

**SISTEM KLASIFIKASI CITRA KANKER ADENOCARCINOMA BODY
FLUID BERBASIS NEURAL NETWORK DENGAN FITUR TEKSTUR**

TUGAS AKHIR

**Diajukan Guna Memenuhi Persyaratan Untuk Mencapai Derajat
Strata-1 Pada Prodi Teknik Elektro Fakultas Teknik
Universitas Muhammadiyah Yogyakarta**



Disusun Oleh:

'ABQARIY AFGHANINA AL-FATH

20200120077

**PROGRAM STUDI TEKNIK ELEKTRO FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH YOGYAKARTA
YOGYAKARTA**

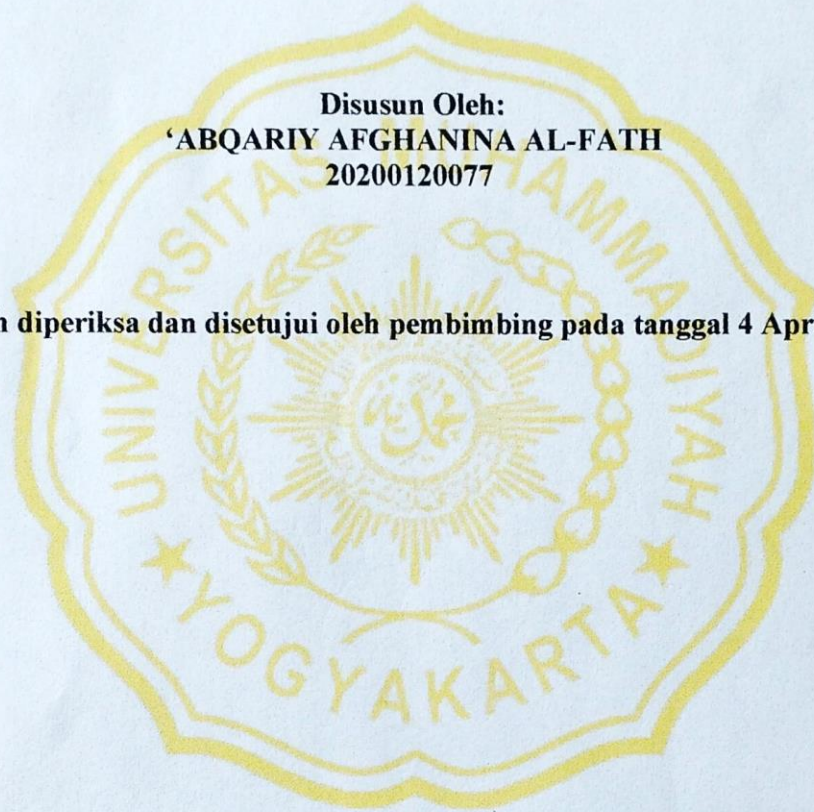
2024

**LEMBAR PENGESAHAN I
TUGAS AKHIR**

**SISTEM KLASIFIKASI CITRA KANKER *ADENOCARCINOMA BODY*
FLUID BERBASIS *NEURAL NETWORK* DENGAN FITUR TEKSTUR**

**Disusun Oleh:
'ABQARIY AFGHANINA AL-FATH
20200120077**

Telah diperiksa dan disetujui oleh pembimbing pada tanggal 4 April 2024



**Mengetahui,
Desen Pembimbing**

**Dr. Yessi Jusman, S.T., M.Sc.
NIK: 19840507201810123106**


LEMBAR PENGESAHAN II

SISTEM KLASIFIKASI CITRA KANKER *ADENOCARCINOMA BODY FLUID* BERBASIS *NEURAL NETWORK* DENGAN FITUR TEKSTUR

Disusun Oleh:
'ABQARIY AFGHANINA AL-FATH
20200120077

Telah dipertahankan pada tim penguji pada tanggal 4 April 2024

Menyetujui,
Desen Pembimbing,



Dr. Yessi Jusman, S.T., M.Sc.
NIK. 19840507201810123106


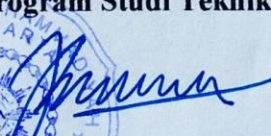
Dosen Penguji,



Ir. Anna Nur Nazilah Chamim, S.T., M.Eng.
NIK. 197608062005012001

Mengesahkan,

Ketua Program Studi Teknik Elektro



Ir. Karisma Trinanda Putra, S.ST., M.T., Ph.D.
NIK. 19900619201604123092

HALAMAN PERNYATAAN

Nama : 'Abqariy Afghana Al-fath
NIM : 20200120077
Program Studi : Teknik Elektro
Fakultas : Teknik
Universitas : Universitas Muhammadiyah Yogyakarta

Saya menyatakan bahwa naskah tugas akhir yang berjudul "**SISTEM KLASIFIKASI CITRA KANKER ADENOCARCINOMA BODY FLUID BERBASIS NEURAL NETWORK DENGAN FITUR TEKSTUR**" merupakan hasil karya tulis sendiri dan tidak terdapat karya yang pernah diajukan untuk memperoleh gelar sarjana pada Perguruan Tinggi serta tidak terdapat karya atau pendapat yang pernah ditulis atau dipublikasikan oleh orang lain, kecuali secara tertulis disebutkan sumbernya dalam naskah dan daftar pustaka.

Yogyakarta, 4 April 2024

Penulis



'Abqariy Afghana Al-fath
20200120077

MOTTO

رَبِّي زِدْنِي عِلْمًا نَافِعًا وَزُرْ قَلْبِي فَهَمًّا

“Ya Allah, tambahkanlah kepadaku ilmu pengetahuan yang bermanfaat dan berikanlah aku pemahaman yang baik.”

رَضِيْتُ بِاللَّهِ رَبًّا وَبِالْإِسْلَامِ دِينًا وَبِمُحَمَّدٍ نَبِيًّا وَرَسُولًا

“Aku ridha Allah SWT sebagai Tuhanku, dan Islam sebagai agamaku, dan Muhammad saw sebagai Nabi dan Rasulku.”

رَبِّ اشْرَحْ لِي صَدْرِي وَيَسِّرْ لِي أَمْرِي وَاخْلُلْ عُنُقَةَ مَنْ لِسَانِي يَفْقَهُوا قَوْلِي

“Ya Rabbku, lapangkanlah untukku dadaku, dan mudahkanlah untukku urusanku, dan lepaskanlah kekakuan dari lidahku, supaya mereka mengerti perkataanku.”

فَإِنَّ مَعَ الْعُسْرِ يُسْرًا ۝ إِنَّ مَعَ الْعُسْرِ يُسْرًا ۝ ٦

“Maka sesungguhnya bersama kesulitan ada kemudahan, sesungguhnya bersama kesulitan ada kemudahan.”

~ Al- Insyirah (94): 5-6 ~

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Alhamdulillah segala puji bagi Allah SWT yang telah melimpahkan rahmat, kemudahan, dan kasih sayang sehingga penulis dapat menyelesaikan tugas akhir sebagai syarat menyelesaikan pendidikan S-1 di Prodi Teknik Elektro Universitas Muhammadiyah Yogyakarta.

Ucapan terima kasih dan rasa hormat penulis sampaikan kepada pihak-pihak yang telah selalu mendoakan, mendukung, dan memberi motivasi sehingga penulis dapat menyelesaikan tugas akhir ini hingga tuntas. Terima kasih yang tulus, penulis sampaikan terkhusus kepada:

1. Bapak Ir. Aris Widyo Nugroho, S.T., M.T., Ph.D. selaku Dekan Fakultas Teknik Universitas Muhammadiyah Yogyakarta.
2. Bapak Ir. Karisma Trinanda Putra, S.ST., M.T., Ph.D. selaku Ketua Program Studi Teknik Elektro Universitas Muhammadiyah Yogyakarta.
3. Ibu Dr. Yessi Jusman, S.T., M.Sc. selaku Dosen Pembimbing yang telah memberikan ilmu, dukungan, serta bimbingan dengan sabar dan tulus.
4. Ibu Ir. Anna Nur Nazilah Chamim, S.T., M.Eng. selaku Dosen Penguji yang telah memberikan masukan selama pendadaran untuk skripsi yang baik.
5. Seluruh Dosen Program Studi Teknik Elektro Universitas Muhammadiyah Yogyakarta yang telah memberikan ilmu dan mendidik selama penulis menjadi mahasiswa.
6. Seluruh tenaga kependidikan Tata Usaha Fakultas Teknik yang telah membantu memberikan pelayanan serta kemudahan mahasiswa.
7. Seluruh Laboran Teknik Elektro yang telah mendampingi dalam praktikum selama penulis menempuh pendidikan di Universitas Muhammadiyah Yogyakarta.
8. Abi, Umi, Mas Yusfi, dan Mas Uqi, my beloved family yang selalu memberikan doa tulus dan dukungan kepada saya untuk bisa segera menyelesaikan tugas akhir.

9. Teman-teman S1 Teknik Elektro angkatan 2020 yang telah berjuang bersama selama menjadi mahasiswa di Universitas Muhammadiyah Yogyakarta.
10. Seluruh teman-teman Ikatan Mahasiswa Muhammadiyah Fakultas Teknik Universitas Muhammadiyah Yogyakarta yang telah menemani selama penulis menjadi mahasiswa di Universitas Muhammadiyah Yogyakarta.

Penulis menyadari bahwa dalam menyusun tugas akhir ini masih terdapat kekurangan dan bisa dikembangkan menjadi lebih baik. Semoga tugas akhir yang telah disusun dan disetujui oleh dosen pembimbing dapat memberikan manfaat dan dapat digunakan sebagai rujukan untuk penelitian selanjutnya.

Yogyakarta, 4 April 2024



'Abqariy Afghanina Al-fath

DAFTAR ISI

LEMBAR PENGESAHAN I.....	ii
LEMBAR PENGESAHAN II.....	iii
HALAMAN PERNYATAAN.....	iv
MOTTO.....	v
KATA PENGANTAR.....	vi
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	x
DAFTAR TABEL.....	xiii
INTISARI.....	xv
ABSTRACT.....	xvi
BAB I.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	4
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian.....	5
1.5 Manfaat Penelitian.....	5
1.6 Sistematika Penelitian.....	5
BAB II.....	7
2.1 Tinjauan Pustaka.....	7
2.2 Landasan Teori.....	11
2.2.1 Kanker <i>Adenocarcinoma</i>	11
2.2.2 <i>Body Fluids</i>	12
2.2.3 Pengolahan Citra Digital.....	13
2.2.4 <i>Gray Level Co-occurrence Matrix</i>	18

2.2.5	<i>Haar Wavelet</i>	23
2.2.6	<i>Multilayer Perceptron</i>	26
2.2.7	Matlab	31
2.2.8	<i>APPS Desaigner</i>	32
BAB III	33
3.1	Metode Penelitian	33
3.2	Perancangan Sistem	35
3.3	Perancangan APPS	48
BAB IV	61
4.1	<i>Preprocessing</i>	61
4.2	Hasil Ekstraksi	63
4.3	Hasil Klasifikasi	65
4.4	Hasil Perbandingan Performa Terbaik	278
4.5	Implementasi APPS	285
BAB V	287
5.1	Kesimpulan	287
5.2	Saran	287
DAFTAR PUSTAKA	288
LAMPIRAN	293

DAFTAR GAMBAR

Gambar 2. 1 Akuisisi Citra Digital.....	13
Gambar 2. 2 Citra Digital Masing-Masing RGB.....	17
Gambar 2. 3 Citra Digital Hasil Konversi RGB Menjadi Biner	18
Gambar 2. 4 Citra Digital Hasil Konversi RGB Menjadi <i>Grayscale</i>	18
Gambar 2. 6 Struktur <i>Multilayer Perceptron</i>	27
Gambar 2. 7 Diagram Alir Proses Klasifikasi, Pelatihan, dan Pengujian.....	28
Gambar 2. 5 Logo Matlab	32
Gambar 3. 1 Diagram Alir Penelitian.....	33
Gambar 3. 2 Flowchart Perancangan Sistem.....	36
Gambar 3. 3 <i>Flowchart Preprocessing</i>	37
Gambar 3. 4 Ekstraksi Fitur	40
Gambar 3. 5 <i>Labelling Citra Body Fluids Kelas Malignant (Adenocarcinoma)</i>	43
Gambar 3. 6 <i>Labelling Citra Body Fluids Kelas Benign (Normal)</i>	43
Gambar 3. 7 Model Klasifikasi	44
Gambar 3. 8 Tampilan Matlab R2020a	45
Gambar 3. 9 Tampilan <i>Editor</i> Untuk <i>Coding</i> Klasifikasi.....	45
Gambar 3. 10 (a) <i>Neural Network Training</i> , (b) <i>Pattern Recognition Neural Network</i>	48
Gambar 3. 11 Tampilan Awal Matlab.....	48
Gambar 3. 12 Tampilan Desain App.....	49
Gambar 3. 13 Tampilan Blank App	49
Gambar 3. 14 Tampilan Desain Awal APPS.....	49
Gambar 3. 15 Tampilan Desain APPS	49
Gambar 3. 16 Tampilan Code View.....	50
Gambar 3. 17 Tampilan <i>Button Function</i>	50
Gambar 3. 18 Tampilan Code APPS.....	50
Gambar 3. 19 Tampilan Akhir APPS	51
Gambar 4. 1 (a) Citra <i>Body Fluids</i> RGB, (b) Citra <i>Body Fluids</i> Grayscale	61
Gambar 4. 2 (a) Pattern Recognition Neural Network Hidden Neuron 1, (b) Pattern Recognition Neural Network Hidden Neuron 5, (c) Pattern Recognition Neural Network Hidden Neuron 10, (d) Pattern Recognition Neural Network Hidden Neuron 15, (e) Pattern Recognition Neural Network Hidden Neuron 20.....	66
Gambar 4. 3 (a1) (b1) Performance <i>Run 1 dan 2 Training</i> SCG HN 1, (a2) (b2) Confusion Matrix <i>Run 1 dan 2 Training</i> SCG HN 1, (a1) (b2) ROC <i>Run 1 dan 2 Training</i> SCG HN 1.....	68
Gambar 4. 4 (a1) (b1) Performance <i>Run 1 dan 2 Training</i> SCG HN 5, (a2) (b2) Confusion Matrix <i>Run 1 dan 2 Training</i> SCG HN 5, (a3) (b3) ROC <i>Run 1 dan 2 Training</i> SCG HN 5.....	75

Gambar 4. 5 (a1) (b1) Performance <i>Run 1</i> dan 2 <i>Training</i> SCG HN 10, (a2) (b2) Confusion Matrix <i>Run 1</i> dan 2 <i>Training</i> SCG HN 10, (a3) (b3) ROC <i>Run 1</i> dan 2 <i>Training</i> SCG HN 10.....	82
Gambar 4. 6 (a1) (b1) Performance <i>Run 1</i> dan 2 <i>Training</i> SCG HN 15, (a2) (b2) Confusion Matrix <i>Run 1</i> dan 2 <i>Training</i> SCG HN 15, (a3) (b3) ROC <i>Run 1</i> dan 2 <i>Training</i> SCG HN 15.....	89
Gambar 4. 7 (a1) (b1) Performance <i>Run 1</i> dan 2 <i>Training</i> SCG HN 20, (a2) (b2) Confusion Matrix <i>Run 1</i> dan 2 <i>Training</i> SCG HN 20, (a3) (b3) ROC <i>Run 1</i> dan 2 <i>Training</i> SCG HN 20.....	96
Gambar 4. 8 (a1) (b1) Performance <i>Run 1</i> dan 2 <i>Training</i> OSS HN 1, (a2) (b2) Confusion Matrix <i>Run 1</i> dan 2 <i>Training</i> OSS HN 1, (a3) (b3) ROC <i>Run 1</i> dan 2 <i>Training</i> OSS HN 1.....	103
Gambar 4. 9 (a1) (b1) Performance <i>Run 1</i> dan 2 <i>Training</i> OSS HN 5, (a2) (b2) Confusion Matrix <i>Run 1</i> dan 2 <i>Training</i> OSS HN 5, (a3) (b3) ROC <i>Run 1</i> dan 2 <i>Training</i> OSS HN 5.....	110
Gambar 4. 10 (a1) (b1) Performance <i>Run 1</i> dan 2 <i>Training</i> OSS HN 10, (a2) (b2) Confusion Matrix <i>Run 1</i> dan 2 <i>Training</i> OSS HN 10, (a3) (b3) ROC <i>Run 1</i> dan 2 <i>Training</i> OSS HN 10.....	117
Gambar 4. 11 (a1) (b1) Performance <i>Run 1</i> dan 2 <i>Training</i> OSS HN 15, (a2) (b2) Confusion Matrix <i>Run 1</i> dan 2 <i>Training</i> OSS HN 15, (a3) (b3) ROC <i>Run 1</i> dan 2 <i>Training</i> OSS HN 15	124
Gambar 4. 12 (a1) (b1) Performance <i>Run 1</i> dan 2 <i>Training</i> OSS HN 20, (a2) (b2) Confusion Matrix <i>Run 1</i> dan 2 <i>Training</i> OSS HN 20, (a3) (b3) ROC <i>Run 1</i> dan 2 <i>Training</i> OSS HN 20.....	131
Gambar 4. 13 (a1) (b1) Performance <i>Run 1</i> dan 2 <i>Training</i> GDM HN 1, (a2) (b2) Confusion Matrix <i>Run 1</i> dan 2 <i>Training</i> GDM HN 1, (a3) (b3) ROC <i>Run 1</i> dan 2 <i>Training</i> GDM HN 1	138
Gambar 4. 14 (a1) (b1) Performance <i>Run 1</i> dan 2 <i>Training</i> GDM HN 5, (a2) (b2) Confusion Matrix <i>Run 1</i> dan 2 <i>Training</i> GDM HN 5, (a3) (b3) ROC <i>Run 1</i> dan 2 <i>Training</i> GDM HN 5	145
Gambar 4. 15 (a1) (b1) Performance <i>Run 1</i> dan 2 <i>Training</i> GDM HN 10, (a2) (b2) Confusion Matrix <i>Run 1</i> dan 2 <i>Training</i> GDM HN 10, (a3) (b3) ROC <i>Run 1</i> dan 2 <i>Training</i> GDM HN 10 .	152
Gambar 4. 16 (a1) (b1) Performance <i>Run 1</i> dan 2 <i>Training</i> GDM HN 15, (a2) (b2) Confusion Matrix <i>Run 1</i> dan 2 <i>Training</i> GDM HN 15, (a3) (b3) ROC <i>Run 1</i> dan 2 <i>Training</i> GDM HN 15 .	159
Gambar 4. 17 (a1) (b1) Performance <i>Run 1</i> dan 2 <i>Training</i> GDM HN 20, (a2) (b2) Confusion Matrix <i>Run 1</i> dan 2 <i>Training</i> GDM HN 20, (a3) (b3) ROC <i>Run 1</i> dan 2 <i>Training</i> GDM HN 20 .	166
Gambar 4. 18 (a) Pattern Recognition Neural Network Hidden Neuron 1, (b) Pattern Recognition Neural Network Hidden Neuron 5, (c) Pattern Recognition Neural Network Hidden Neuron 10, (d) Pattern Recognition Neural Network Hidden Neuron 15, (e) Pattern Recognition Neural Network Hidden Neuron 20.....	172
Gambar 4. 19 (a1) (b1) Performance <i>Run 1</i> dan 2 <i>Training</i> SCG HN 1, (a2) (b2) Confusion Matrix <i>Run 1</i> dan 2 <i>Training</i> SCG HN 1, (a3) (b3) ROC <i>Run 1</i> dan 2 <i>Training</i> SCG HN 1	174
Gambar 4. 20 (a1) (b1) Performance <i>Run 1</i> dan 2 <i>Training</i> SCG HN 5, (a2) (b2) Confusion Matrix <i>Run 1</i> dan 2 <i>Training</i> SCG HN 5, (a3) (b3) ROC <i>Run 1</i> dan 2 <i>Training</i> SCG HN 5.....	181
Gambar 4. 21 (a1) (b1) Performance <i>Run 1</i> dan 2 <i>Training</i> SCG HN 10, (a2) (b2) Confusion Matrix <i>Run 1</i> dan 2 <i>Training</i> SCG HN 10, (i3) (j3) ROC <i>Run 1</i> dan 2 <i>Training</i> SCG HN 10	188

Gambar 4. 22 (a1) (b1) Performance Run 1 dan 2 Training SCG HN 15, (a2) (b2) Confusion Matrix Run 1 dan 2 Training SCG HN 15, (a3) (b3) ROC Run 1 dan 2 Training SCG HN 15	195
Gambar 4. 23 (a1) (b1) Performance Run 1 dan 2 Training SCG HN 20, (a2) (b2) Confusion Matrix Run 1 dan 2 Training SCG HN 20, (a3) (b3) ROC Run 1 dan 2 Training SCG HN 20	202
Gambar 4. 24 (a1) (b1) Performance Run 1 dan 2 Training OSS HN 1, (a2) (b2) Confusion Matrix Run 1 dan 2 Training OSS HN 1, (a3) (b3) ROC Run 1 dan 2 Training OSS HN 1.....	209
Gambar 4. 25 (a1) (b1) Performance Run 1 dan 2 Training OSS HN 5, (a2) (b2) Confusion Matrix Run 1 dan 2 Training OSS HN 5, (a3) (b3) ROC Run 1 dan 2 Training OSS HN 5.....	216
Gambar 4. 26 (a1) (b1) Performance Run 1 dan 2 Training OSS HN 10, (a2) (b2) Confusion Matrix Run 1 dan 2 Training OSS HN 10, (a3) (b3) ROC Run 1 dan 2 Training OSS HN 10.....	223
Gambar 4. 27 (a1) (b1) Performance Run 1 dan 2 Training OSS HN 15, (a2) (b2) Confusion Matrix Run 1 dan 2 Training OSS HN 15, (a3) (b3) ROC Run 1 dan 2 Training OSS HN 15.....	230
Gambar 4. 28 (a1) (b1) Performance Run 1 dan 2 Training OSS HN 20, (a2) (b2) Confusion Matrix Run 1 dan 2 Training OSS HN 20, (a3) (b3) ROC Run 1 dan 2 Training OSS HN 20.....	237
Gambar 4. 29 (a1) (b1) Performance Run 1 dan 2 Training GDM HN 1, (a2) (b2) Confusion Matrix Run 1 dan 2 Training GDM HN 1, (a3) (b3) ROC Run 1 dan 2 Training GDM HN 1.....	244
Gambar 4. 30 (a1) (b1) Performance Run 1 dan 2 Training GDM HN 5, (a2) (b2) Confusion Matrix Run 1 dan 2 Training GDM HN 5, (a3) (b3) ROC Run 1 dan 2 Training GDM HN 5.....	251
Gambar 4. 31 (a1) (b1) Performance Run 1 dan 2 Training GDM HN 10, (a2) (b2) Confusion Matrix Run 1 dan 2 Training GDM HN 10, (a3) (b3) ROC Run 1 dan 2 Training GDM HN 10.....	258
Gambar 4. 32 (a1) (b1) Performance Run 1 dan 2 Training GDM HN 15, (a2) (b2) Confusion Matrix Run 1 dan 2 Training GDM HN 15, (a3) (b3) ROC Run 1 dan 2 Training GDM HN 15.....	265
Gambar 4. 33 (a1) (b1) Performance Run 1 dan 2 Training GDM HN 20, (a2) (b2) Confusion Matrix Run 1 dan 2 Training GDM HN 20, (a3) (b3) ROC Run 1 dan 2 Training GDM HN 20.....	272
Gambar 4. 34 Perbandingan Akurasi <i>Training</i> Tertinggi GLCM	278
Gambar 4. 35 Perbandingan Akurasi Validasi Tertinggi GLCM.....	278
Gambar 4. 36 Perbandingan Akurasi <i>Testing</i> Tertinggi GLCM.....	279
Gambar 4. 37 Perbandingan Akurasi <i>Training</i> Tertinggi GLCM	279
Gambar 4. 38 Perbandingan Akurasi Validasi Tertinggi GLCM.....	280
Gambar 4. 39 Perbandingan Akurasi <i>Testing</i> Tertinggi GLCM.....	280
Gambar 4. 40 Perbandingan Akurasi <i>Training</i> Tertinggi <i>Haar Wavelet</i>	281
Gambar 4. 41 Perbandingan Akurasi Validasi Tertinggi <i>Haar Wavelet</i>	282
Gambar 4. 42 Perbandingan Akurasi <i>Testing</i> Tertinggi <i>Haar Wavelet</i>	282
Gambar 4. 43 Perbandingan Akurasi <i>Training</i> Tertinggi <i>Haar Wavelet</i>	283
Gambar 4. 44 Perbandingan Akurasi Validasi Tertinggi <i>Haar Wavelet</i>	283
Gambar 4. 45 Perbandingan Akurasi <i>Testing</i> Tertinggi <i>Haar Wavelet</i>	284

DAFTAR TABEL

Tabel 2. 1 Rangkuman Tinjauan Pustaka	8
Tabel 2. 2 Penelitian Terkait <i>Gray Level Co-occurrence Matrix</i> (GLCM).....	22
Tabel 2. 3 Penelitian Terkait <i>Haar Wavelet</i>	25
Tabel 2. 4 Penelitian Terkait <i>Multi Layer Perceptron</i> (MLP) atau <i>Artificial Neural Network</i> (ANN)	30
Tabel 4. 1 Perbandingan Jumlah Citra Sebelum dan Sesudah <i>Cropping</i>	62
Tabel 4. 2 Perbandingan Ukuran Citra Sebelum <i>Cropping</i> dan Sesudah <i>Cropping</i>	62
Tabel 4. 3 Perbandingan Empat Jenis <i>Enhancement</i> pada Citra <i>Body Fluids</i>	63
Tabel 4. 4 Hasil Ekstraksi <i>Gray Level Co-occurrence Matrix</i> (GCLM)	64
Tabel 4. 5 Hasil Ekstraksi <i>Haar Wavelet</i>	65
Tabel 4. 6 Hasil Akurasi <i>Training</i> Menggunakan Model SCG <i>Hidden Neuron</i> 1	67
Tabel 4. 7 Hasil Akurasi <i>Training</i> Menggunakan Model SCG <i>Hidden Neuron</i> 5	74
Tabel 4. 8 Hasil Akurasi <i>Training</i> Menggunakan Model SCG <i>Hidden Neuron</i> 10	81
Tabel 4. 9 Hasil Akurasi <i>Training</i> Menggunakan Model SCG <i>Hidden Neuron</i> 15.....	88
Tabel 4. 10 Hasil Akurasi <i>Training</i> Menggunakan Model SCG <i>Hidden Neuron</i> 20	95
Tabel 4. 11 Hasil Akurasi <i>Training</i> Menggunakan Model OSS <i>Hidden Neuron</i> 1.....	102
Tabel 4. 12 Hasil Akurasi <i>Training</i> Menggunakan Model OSS <i>Hidden Neuron</i> 5.....	109
Tabel 4. 13 Hasil Akurasi <i>Training</i> Menggunakan Model OSS <i>Hidden Neuron</i> 10.....	116
Tabel 4. 14 Hasil Akurasi <i>Training</i> Menggunakan Model OSS <i>Hidden Neuron</i> 15.....	123
Tabel 4. 15 Hasil Akurasi <i>Training</i> Menggunakan Model OSS <i>Hidden Neuron</i> 20.....	130
Tabel 4. 16 Hasil Akurasi <i>Training</i> Menggunakan Model GDM <i>Hidden Neuron</i> 1.....	137
Tabel 4. 17 Hasil Akurasi <i>Training</i> Menggunakan Model GDM <i>Hidden Neuron</i> 5.....	144
Tabel 4. 18 Hasil Akurasi <i>Training</i> Menggunakan Model GDM <i>Hidden Neuron</i> 10.....	151
Tabel 4. 19 Hasil Akurasi <i>Training</i> Menggunakan Model GDM <i>Hidden Neuron</i> 15.....	158
Tabel 4. 20 Hasil Akurasi <i>Training</i> Menggunakan Model GDM <i>Hidden Neuron</i> 20.....	165
Tabel 4. 21 Hasil Akurasi <i>Training</i> Menggunakan Model SCG <i>Hidden Neuron</i> 1	173
Tabel 4. 22 Hasil Akurasi <i>Training</i> Menggunakan Model SCG <i>Hidden Neuron</i> 5	180
Tabel 4. 23 Hasil Akurasi <i>Training</i> Menggunakan Model SCG <i>Hidden Neuron</i> 10	187
Tabel 4. 24 Hasil Akurasi <i>Training</i> Menggunakan Model SCG <i>Hidden Neuron</i> 15	194
Tabel 4. 25 Hasil Akurasi <i>Training</i> Menggunakan Model SCG <i>Hidden Neuron</i> 20	201
Tabel 4. 26 Hasil Akurasi <i>Training</i> Menggunakan Model OSS <i>Hidden Neuron</i> 1.....	208
Tabel 4. 27 Hasil Akurasi <i>Training</i> Menggunakan Model OSS <i>Hidden Neuron</i> 5.....	215
Tabel 4. 28 Hasil Akurasi <i>Training</i> Menggunakan Model OSS <i>Hidden Neuron</i> 10.....	222
Tabel 4. 29 Hasil Akurasi <i>Training</i> Menggunakan Model OSS <i>Hidden Neuron</i> 15.....	229
Tabel 4. 30 Hasil Akurasi <i>Training</i> Menggunakan Model OSS <i>Hidden Neuron</i> 20.....	236

Tabel 4. 31 Hasil Akurasi <i>Training</i> Menggunakan Model GDM <i>Hidden Neuron</i> 1	243
Tabel 4. 32 Hasil Akurasi <i>Training</i> Menggunakan Model GDM <i>Hidden Neuron</i> 5	250
Tabel 4. 33 Hasil Akurasi <i>Training</i> Menggunakan Model GDM <i>Hidden Neuron</i> 10	257
Tabel 4. 34 Hasil Akurasi <i>Training</i> Menggunakan Model GDM <i>Hidden Neuron</i> 15	264
Tabel 4. 35 Hasil Akurasi <i>Training</i> Menggunakan Model GDM <i>Hidden Neuron</i> 20	271
Tabel 4. 36 Contoh Pengimplementasian APPS Matlab	285

INTISARI

Adenocarcinoma adalah salah satu jenis kanker yang biasanya menyerang tubuh bagian sel pelapis epitel (sel terluar organ). Kanker *Adenocarcinoma* dapat dideteksi salah satunya melalui *body fluids* yang berada di antara paru-paru dan dinding dada. Penelitian ini bertujuan untuk membantu ahli *cytology* dalam menentukan diagnosis penyakit kanker, terutama kanker yang dapat dideteksi melalui *body fluids*. Melalui teknologi *machine learning*, diagnosis dapat dilakukan dengan efisien dan akurat. Metode yang digunakan dalam penelitian ini adalah *Gray Level Co-occurrence Matrix* (GLCM) dan *Haar Wavelet* sebagai metode ekstraksi dan *Multilayer Perceptron* (MLP) sebagai metode klasifikasi. Jumlah citra yang digunakan adalah 1.273 citra yang terdiri dari 642 citra *benign* dan 631 citra *malignant*. Hasil penelitian menunjukkan bahwa *Haar Wavelet* menghasilkan akurasi lebih tinggi dibandingkan GLCM dengan akurasi sebesar 94,1% oleh model SCG *hidden neuron* 20 dalam waktu *run* 12 detik sedangkan akurasi GLCM adalah 92,4% oleh model SCG *hidden neuron* 20 dalam waktu *run* 3 detik.

Kata Kunci: *Body Fluids*, *Gray Level Co-occurrence Matrix* (GLCM), *Haar Wavelet*, *Multilayer Perceptron* (MLP)

ABSTRACT

Adenocarcinoma is a type of cancer that usually attacks the body's epithelial lining cells (the outermost cells of the organ). Adenocarcinoma cancer can be detected through body fluids between the lungs and chest wall. This research aims to help cytology experts in determining the diagnosis of cancer, especially cancer that can be detected through body fluids. Through machine learning technology, diagnosis can be done efficiently and accurately. The methods used in this research are Gray Level Co-occurrence Matrix (GLCM) and Haar Wavelet as extraction methods and Multilayer Perceptron (MLP) as a classification method. The number of images used is 1,273 images consisting of 642 benign images and 631 malignant images. The results show that Haar Wavelet produces higher accuracy than GLCM with an accuracy of 94.1% by the SCG hidden neuron 20 model in 12 seconds run time while the accuracy of GLCM is 92.4% by the SCG hidden neuron 20 model in 3 seconds run time.

Key Words: Body Fluids, Gray Level Co-occurrence Matrix (GLCM), Haar Wavelet, Multilayer Perceptron (MLP)

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kanker merupakan kondisi dimana sel-sel tubuh mengalami pembelahan yang tidak terkendali. Penyebab dari munculnya penyakit kanker adalah beragam, salah satunya disebabkan karena terjadinya perubahan (mutasi) gen dalam sel. Penyakit kanker masih menjadi salah satu penyakit yang menyebabkan kematian dengan jumlah angka yang tinggi. Data dari Badan Kesehatan Dunia (WHO), di Indonesia, pada tahun 2020 terdapat 396.914 kasus baru penderita penyakit kanker dan terdapat 234.511 kematian akibat kanker (indonesiabaik.id, 2023).

Terdapat banyak sekali jenis kanker yang menyerang organ tubuh manusia terkhusus organ dalam, seperti kanker rahim, kanker serviks, kanker paru-paru, kanker lambung, dan sebagainya. Berdasarkan data *World Health Organization* (WHO) pada tahun 2020, terdapat sepuluh jenis kanker dengan jumlah kasus terbanyak di Indonesia. Kanker payudara memiliki kasus tertinggi di Indonesia, yaitu 65.858 kasus atau 16,6% dari total 396.914 kasus kanker. Kanker serviks (leher rahim) menempati urutan kedua dengan jumlah 36.633 kasus atau 9,2% dari total kasus kanker. Kasus kanker paru-paru berada di urutan ketiga dengan jumlah 34.783 kasus atau 8,8% dari total kasus. Kemudian di urutan keempat terdapat kanker hati dengan jumlah 21.392 kasus atau 5,4% dari total kasus kanker (Rizaty, M.A. 2021)

Adenocarcinoma adalah salah satu jenis kanker yang biasanya menyerang tubuh bagian sel pelapis epitel (sel terluar bagian organ) dengan tipe sel kelenjar. Tipe sel kelenjar adalah sel yang menghasilkan zat tertentu dan ditemukan pada hampir seluruh organ tubuh manusia. Sel kanker *Adenocarcinoma* dapat dideteksi atau diketahui keberadaannya salah satunya melalui *body fluids* atau cairan tubuh. *Body fluids* yang dimaksud disini adalah cairan yang berada di rongga dada dan rongga perut. Di dalam rongga

dada dan rongga perut, terdapat banyak organ dalam yang mana selalu terdapat peluang organ dalam tersebut terserang penyakit kanker.

Di dalam tubuh manusia, paru-paru dilapisi oleh selaput yang disebut dengan pleura. Antara paru-paru dan dinding dada terdapat rongga yang berisi cairan atau dalam penelitian ini disebut dengan *body fluids*. Cairan tersebut berfungsi sebagai pelumas paru-paru saat bekerja. Jumlah cairan tersebut dapat meningkat yang disebabkan oleh beberapa kondisi kesehatan. Peningkatan jumlah cairan tersebut disebut dengan efusi pleura. Terjadinya efusi pleura menjadi salah satu ciri terjadinya kanker.

Penyakit kanker menjadi penyakit ganas bagi manusia karena dapat merenggut nyawa. Seseorang yang diuji oleh Allah SWT dengan penyakit, tentu merasa berat. Namun, seseorang itu tidak boleh menyerah karena Allah telah menyiapkan obat baginya, seperti dalam firman Allah SWT, Q.S. Yunus ayat 57.

يَا أَيُّهَا النَّاسُ قَدْ جَاءَكُمْ مَوْعِظَةٌ مِنْ رَبِّكُمْ وَشِفَاءٌ لِمَا فِي الصُّدُورِ وَهُدًى وَرَحْمَةٌ لِّلْمُؤْمِنِينَ

“Hai manusia, sesungguhnya telah datang kepadamu pelajaran dari Tuhanmu dan penyembuh bagi penyakit-penyakit (yang berada) dalam dada dan petunjuk serta rahmat bagi orang-orang yang beriman”. Oleh karena itu, bagi hamba Allah SWT yang beriman, berdoa meminta kesembuhan itu kewajiban dan ikhtiar pemeriksaan kesehatan itu keharusan.

Pemeriksaan *cytology* sering kali digunakan dalam mendiagnosis penyakit kanker pada efusi pleura karena murah, sederhana, dan alatnya efektif. Jenis pemeriksaan tersebut tergolong manual karena ahli *cytology* akan memeriksa setiap sel menggunakan kamera mikroskopik untuk kemudian diamati apakah terdapat *cytomorphology* yang tidak normal. Selanjutnya ahli *cytology* akan menuliskan kesimpulan tentang pemeriksaan tersebut. Pemeriksaan *cytology* manual tersebut membutuhkan tenaga ahli khusus bidang *cytology*, membutuhkan waktu yang lebih lama, dan rentan terhadap subjektifitas pengamat. Hasil akurasi yang didapatkan pun tergantung pada tingkat fokus dan keahlian dari ahli *cytology* yang

melakukan pemeriksaan. Selain itu, pada beberapa negara berkembang, jumlah ahli *cytology* pun terbatas.

Berdasarkan penjelasan di atas, pengembangan teknologi dalam hal pemeriksaan *cytology*, secara khusus dan dalam bidang kesehatan, secara umum perlu dilakukan. Pengembangan teknologi tersebut terarah pada *artificial intelligence* atau dalam penelitian ini adalah *machine learning*. Dengan pengembangan *machine learning*, tenaga medis akan terbantu dalam proses diagnosis hasil pemeriksaan *cytology*. Kelebihan lain dari penggunaan *machine learning* dalam proses diagnosis kanker adalah mempercepat prosedur diagnosis, mengurangi subjektivitas dari ahli *cytology* dan dapat mendeskripsikan hasil pemeriksaan secara kuantitatif. Hasil dari *machine learning* tersebut dapat melengkapi pendapat para ahli *cytology*.

Metode *machine learning* yang banyak digunakan dalam penelitian tentang *cytology* salah satunya adalah *Artificial Neural Network* (ANN). Beberapa penelitian sebelumnya, telah banyak yang menggunakan ANN dalam proses diagnosis citra sel kanker. Proses tersebut dikenal dengan *image processing*. Salah satu penerapan ANN adalah untuk mendiagnosis kanker dari citra *body fluids*. Contoh penelitian tersebut dilakukan oleh (Baykal et al., 2017) yang menggunakan ANN untuk proses segmentasi citra *cytology pleural effusion*, (Margareth, 2017) menggunakan ANN untuk memprediksi adanya kanker (*mesothelioma*), dan (Mir & Sarwar, 2021) mengumpulkan penelitian yang menggunakan ANN sebagai metode dalam diagnosis *body fluids*.

Teknologi pengolahan citra telah berkembang banyak di bidang medis dan telah membantu banyak tenaga medis menentukan diagnosis awal suatu penyakit, khususnya dalam bidang *cytology*. Penelitian tentang diagnosis kanker melalui citra *body fluids* perlu terus dikembangkan agar dapat membantu pasien yang bergejala tertangani dengan cepat. Berdasarkan latar belakang tersebut, penulis melakukan penelitian tentang klasifikasi kanker *Adenocarcinoma* melalui citra *body fluids* dengan menggunakan *machine learning*. Metode yang digunakan adalah *Gray Level Co-occurrence Matrix*

(GLCM) dan *Haar Wavelet* sebagai metode ekstraksi serta *Multilayer Perceptron* (MLP) sebagai metode klasifikasi. Hasil dari proses klasifikasi tersebut adalah citra akan terklasifikasikan dalam dua kelas, yaitu kelas *Malignant* atau *Benign*.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan di atas, maka dapat diidentifikasi beberapa permasalahan yaitu sebagai berikut.

1. Bagaimana cara mengekstraksi fitur citra *body fluids* yang terdiri dari kelas *malignant* dan *benign*?
2. Bagaimana cara mengklasifikasikan hasil ekstraksi citra *body fluids* ke dalam kelas *malignant* dan *benign*?

1.3 Batasan Masalah

Batasan masalah diperlukan dalam suatu penelitian untuk menghindari perluasan pembahasan. Adapun batasan masalah dalam penelitian ini adalah sebagai berikut.

1. Data sampel yang digunakan adalah citra *body fluids* atau cairan tubuh.
2. Penelitian deteksi kanker *Adenocarcinoma* ini dibatasi dengan dua kelas klasifikasi, yaitu *Malignant* dan *Benign*.
3. Citra *body fluids* yang digunakan adalah citra abu (*grayscale*).
4. Jumlah citra yang digunakan dalam penelitian ini adalah 1.273 citra.
5. Resolusi citra awal sebelum mengalami *cropping* adalah 3.840 x 2.160 *pixel* sedangkan ukuran setelah mengalami *cropping* adalah 1.280 x 720 *pixel*.
6. Metode ekstraksi yang digunakan dalam penelitian ini adalah *Gray Level Co-occurrence Matrix* (GLCM) dan *Haar Wavelet*.
7. Metode klasifikasi yang digunakan adalah *Multilayer Perceptron* (MLP) dengan tiga jenis model, yaitu *Scaled Conjugate Gradient Backpropagation* (SCG), *One Step Secant Backpropagation* (OSS), dan *Gradient Descent with Momentum* (GDM) dengan *hidden neuron* 1, 5, 10, 15, dan 20.

8. Arah sudut *Gray Level Co-occurrence Matrix* (GLCM) adalah 0° , 45° , 90° , 135° .
9. Sistem pemrograman yang digunakan dalam penelitian ini adalah Bahasa pemrograman Matlab R2020a.

1.4 Tujuan Penelitian

Dari uraian permasalahan di atas, maka dapat dirumuskan tujuan diadakannya penelitian tentang klasifikasi sel kanker *Adenocarcinoma* adalah sebagai berikut.

1. Mengekstraksi fitur pembeda antara sel *malignant* dan sel *benign* melalui ekstraksi citra *body fluids* dengan menggunakan metode *Gray Level Co-occurrence Matrix* (GLCM) dan *Haar Wavelet*.
2. Mengklasifikasikan hasil ekstraksi citra *body fluids* menggunakan metode *Multilayer Perceptron* (MLP) dengan model *Scaled Conjugate Gradient Backpropagation* (SCG), *One Step Secant Backpropagation* (OSS), dan *Gradient Descent with Momentum* (GDM) ke dalam kelas *malignant* dan *benign*.

1.5 Manfaat Penelitian

Dari penelitian ini, penulis berharap agar klasifikasi yang memanfaatkan *image processing* sebagai sistem otomasi dalam mengklasifikasikan sel *malignant* dan sel *benign* dapat membantu tenaga medis dalam menentukan diagnosis sementara secara lebih efektif dan efisien. Dengan sistem klasifikasi berbasis *image processing* ini, tenaga medis dapat segera menentukan tindakan penanganan lebih lanjut kepada pasien.

1.6 Sistematika Penelitian

Sistematika yang digunakan dalam penyusunan laporan hasil penelitian tentang sistem klasifikasi citra kanker *Adenocarcinoma body fluids* berbasis *neural network* dengan fitur tekstur adalah sebagai berikut.

1. BAB I: PENDAHULUAN

Pada bab I, yaitu pendahuluan berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penelitian.

2. BAB II: TINJAUAN PUSTAKA

Pada bab II, tinjauan pustaka berisi tentang landasan-landasan teori serta beberapa penelitian sebelumnya yang berkaitan dengan topik penelitian untuk dijadikan rujukan dalam penelitian ini.

3. BAB III: METODOLOGI PENELITIAN

Pada bab III, metodologi penelitian berisi tentang langkah-langkah yang dilakukan dari awal hingga akhir selama penelitian yang tertulis dalam diagram alir.

4. BAB IV: HASIL DAN PEMBAHASAN

Pada bab IV, hasil dan pembahasan berisi tentang tabel, grafik, atau gambar dari hasil pengujian, perancangan sistem serta analisis yang kemudian dijadikan rujukan dalam membuat kesimpulan.

5. BAB V: PENUTUP

Pada bab V, yaitu penutup berisi kesimpulan dari keseluruhan rangkaian penelitian dan saran yang digunakan dalam pengembangan penelitian selanjutnya.

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian terkait penyakit yang dapat dideteksi melalui citra *body fluids* telah dilakukan oleh beberapa peneliti sebelumnya. Metode yang digunakan sangat beragam hingga berkembang dengan metode *machine learning* yang dipadukan dengan berbagai metode segmentasi dan ekstraksi. Beberapa metode segmentasi yang dipadukan dengan *machine learning* dalam penelitian terkait *body fluids* pun beragam, diantaranya *Otsu Thresholding*, *K-Means Clustering*, *mean shift*, *Chan–Vese*, dan *graph cut*. Contoh metode ekstraksi yang dipadukan dengan *machine learning*, diantaranya *Gray Level Co-occurrence Matrix (GLCM)*, *Haar Wavelet*, *Gabor Filter*, dan sebagainya sedangkan metode klasifikasi yang sering digunakan dalam *machine learning*, diantaranya *Random Forest*, *Support Vector Machine (SVM)*, *K-Nearest Neighbor (KNN)*, *Multilayer Perceptron (MLP)*, dan lain sebagainya.

Beberapa penelitian yang menggunakan metode segmentasi untuk mengenali atau membaca citra *body fluids* dilakukan oleh (Zhang et al., 2006), (Win et al., 2017a), dan (Win, Choomchuay, Hamamoto, & Raveesunthornkiat, 2018). Ketiga penelitian tersebut menggunakan metode *Otsu Thresholding* dan *K-means Clustering* dalam segmentasi citra *pleural effusion*. Dari penelitian tersebut, diketahui bahwa segmentasi otomatis dapat membantu tenaga kesehatan dalam mendiagnosa awal keberadaan sel kanker dalam citra *pleural effusion*. Hasil dari penelitian tersebut adalah nilai akurasi *Otsu Thresholding* sebesar 92% dan *K-means Clustering* 94%.

Penelitian lain yang memadukan metode ekstraksi dan *machine learning* sebagai metode klasifikasi adalah penelitian tentang sistem diagnosis kanker dari citra *pleural effusion*. Metode yang digunakan adalah *Gray Level Co-occurrence Matrix (GLCM)* dan *Gray Level Run-Length Matrix (GLRM)*. Hasil dari penelitian tersebut adalah sistem dengan akurasi 98.70%, sensitifitas 87.97%, spesifitas 99.40%, dan F-score 87.79% (Win, Choomchuay,

Hamamoto, Raveesunthornkiat, et al., 2018). Beberapa penelitian lain yang menghasilkan akurasi lebih dari 80% juga dilakukan oleh (Wright et al., 2008) yang menggunakan *Classification and Regression Trees (CART)* sebagai metode klasifikasi dan (Faria et al., 2016) menggunakan *Multilayer Perceptron (MLP)* dalam sistem klasifikasinya.

Selain itu, terdapat juga penelitian yang mengumpulkan publikasi dari peneliti lain untuk dibandingkan metode *machine learning* yang sering digunakan dalam penelitian tentang *body fluids*. Penulis membatasi dalam enam metode *machine learning*, yaitu *Decision Tree*, *ANN*, *k-means clustering*, *CNN*, *SVM*, dan *Fuzzy*. Dari 42 publikasi yang dikumpulkan, penulis menyebutkan bahwa *ANN* berada di urutan pertama metode yang sering digunakan dalam penelitian tentang *body fluids*. Urutan selanjutnya diikuti oleh metode *Fuzzy*, *CNN*, *Decision Tree*, *K-means Clustering*, dan *SVM*. (Mir & Sarwar, 2021). Tabel 2.1 menunjukkan beberapa referensi terkait penelitian tersebut.

Tabel 2. 1 Rangkuman Tinjauan Pustaka

No	Author (Tahun)	Judul	Objek	Metode	Hasil
1	(Zhang et al., 2006)	Research Based on Fuzzy Algorithm of Cancer Cells in Pleural Fluid Microscopic Images Recognition	<i>Body Fluids (Pleural Fluid)</i>	<i>Machine Learning (OTSU thresholding, Fuzzy pattern recognition)</i>	Sistem dapat digunakan sebagai diagnosis awal adanya kanker
2	(Wright et al., 2008)	Using Classification and Regression Trees, Liquid-Based Cytology and Nuclear Morphometry for the Discrimination of Endometrial Lesions	<i>Body Fluids (Endometrial Cytology)</i>	<i>Machine Learning (Classification and Regression Trees (CART))</i>	Akurasi 85%, spesifitas 90,68%, dan sensitifitas 72,05%.
3	(Porcel et al., 2008)	A Decision Tree for Differentiating Tuberculous from Malignant Pleural Effusions	<i>Body Fluids (Pleural Effusions)</i>	<i>Machine Learning (Decision Tree)</i>	Sensitifitas 92,2% spesifitas 98,3% dan ROC 0.976.

4	(Wright et al., 2008)	Machine Learning Techniques to Diagnose Breast Cancer from Image-Processed Nuclear Features of Fine Needle Aspirates	<i>Breast Cancer</i>	<i>Machine Learning</i>	Akurasi dari <i>tenrun cross validation</i> adalah 97%
5	(Faria et al., 2016)	Pleural Tuberculosis Diagnosis Based on Artificial Neural Networks Models	<i>Body Fluids (Pleural Tuberculosis)</i>	Two-Layered Feedforward Multilayer Perceptrons (MLP)	Akurasi 84.7%
6	(Win et al., 2017a)	Automated Segmentation and Isolation of Touching Cell Nuclei in Cytopathology Smear Images of Pleural Effusion Using Distance Transform Watershed Method	<i>Body fluid (pleural fluid)</i>	<i>Machine Learning (OTSU thresholding, Distance Transform Watershed)</i>	Akurasi 92%
7	(Baykal et al., 2017)	Automated Cell Nuclei Segmentation in Pleural Effusion Cytology Using Active Appearance Model	<i>Body Fluids (Pleural Effusion)</i>	<i>Machine Learning (Otsu k-means, Grabcut, dan Active Appearance Model (AMM))</i>	Akurasi Otsu 43,84%, K-means 85,09%, Grabcut 92,31%, dan AMM 98,77%
8	(Win et al., 2017b)	K Mean Clustering Based Automated Segmentation of Overlapping Cell Nuclei in Pleural Effusion Cytology Images	<i>Body Fluids (Pleural Effusion)</i>	<i>Machine Learning (K-Means Clustering)</i>	Presisi 90%, Recall 89%, F-measure 89%
9	(Margareth, 2017)	Predicting Cancer Using Supervised Machine Learning: Mesothelioma	<i>Body Fluids (Pleural Mesothelioma)</i>	<i>Machine Learning (MLP, voted perceptron (VP), Clojure classifier (CC), kernel logistic regression (KLR), stochastic gradient decent</i>	Akurasi SGD 69.23%, AdaBoost.M1 71.29, KLR 69.51%, MLP 64.11%, VFDT 70.38%, VP 70.38%, CC 70.38%

				(SGD), adaptive boosting (AdaBoost), Hoeffding tree (VFDT)	
10	(Baykal et al., 2017)	Artificial Neural Network Based Nuclei Segmentation on Cytology Pleural Effusion Images	<i>Body Fluids (Cytology Pleural Effusion)</i>	<i>Machine Learning (ANN)</i>	Presisi 95%, Recall 86%, dan F-measure 90%
11	(Win, Choomhuay, Hamamoto, Raveesunthornkiat, et al., 2018)	Computer Aided Diagnosis System for Detection of Cancer Cells on Cytological Pleural Effusion Images	<i>Body Fluids (Pleural Effusion)</i>	<i>Gray Level Co-occurrence Matrix (GLCM), Gray Level Run-Length Matrix (GLRM)</i>	Sensitifitas 87.97%, spesifitas 99.40%, akurasi 98.70%, dan F-score 87.79%.
12	(Win, Choomhuay, Hamamoto, & Raveesunthornkiat, 2018)	Comparative Study on Automated Cell Nuclei Segmentation Methods for Cytology Pleural Effusion Images	<i>Body Fluids (Pleural Effusion)</i>	<i>Machine Learning (Otsu, k-means, mean shift, Chan-Vese, dan graph cut)</i>	Akurasi Otsu 94%, K-means 94%, mean shift 95%, Chan-Vese 94%, dan graph cut 93%.
13	(Vargason et al., 2020)	A Clinical Decision Support System for Malignant Pleural Effusion Analysis	<i>Body Fluids (Pleural Effusion)</i>	<i>Machine Learning (Random Forest dan SVM)</i>	Akurasi Random Forest 90%, SVM (Linear Kernel) 91%
14	(Sanyal et al., 2021)	A Machine Learning Model for Screening of Body Fluid Cytology Smears	<i>Body Fluids</i>	<i>Convolutional Neural Network (CNN)</i>	Sensitifitas 97,87%, spesifitas 85,26%, PPV 95,18%, NPV 93,10%.
15	(Mir & Sarwar, 2021)	Artificial Intelligence-Based Techniques for Analysis of Body Cavity Fluids: a Review	<i>Body Cavity Fluids</i>	Membandingkan ANN, Fuzzy, CNN, Decision Tree, SVM, dan k-means clustering	Dari 42 publikasi, ANN menjadi metode paling banyak digunakan, diikuti fuzzy, CNN, Decision tree, k-means

					clustering, dan SVM.
16	(Wei et al., 2023)	Development and Validation of a Machine Learning Model for Differential Diagnosis of Malignant Pleural Effusion Using Routine Laboratory Data	<i>Body Fluids (Pleural Effusion)</i>	<i>Machine Learning (Extreme gradient boosting (XGBoost), logistic regression (LR), random forest (RF), SVM, MLP)</i>	Area Under Curve (AUC) XGBoost 0,903, LR 0,889, RF 0,690, SVM 0,820, MLP 0,801. Akurasi XGBoost 0,827, LR 0,857, RF 0,859, SVM 0,803, MLP 0,827

2.2 Landasan Teori

2.2.1 Kanker *Adenocarcinoma*

Kanker merupakan kondisi dimana sel-sel tubuh mengalami pembelahan yang tidak terkendali. Penyebab dari munculnya penyakit kanker adalah beragam, salah satunya disebabkan karena terjadinya perubahan (mutasi) gen dalam sel. Kanker dapat diklasifikasikan dalam dua cara, yaitu berdasarkan jenis jaringan dimana kanker berasal (tipe histologis) dan berdasarkan tempat utama dimana kanker berkembang. Berdasarkan tipe histologis, kanker dibedakan menjadi enam jenis, yaitu *Carcinoma*, *Sarcoma*, *Myeloma*, *Leukimia*, *Lymphoma*, dan *Mixed Types*.

Kanker *Adenocarcinoma* merupakan salah satu jenis kanker *carcinoma*. *Carcinoma* sendiri dapat didefinisikan sebagai kanker yang muncul dari sel epitel. Sel-sel epitel adalah sel-sel yang melapisi permukaan luar dan dalam organ tubuh, baik yang berbentuk *tubulus* (saluran) maupun *cavum* (rongga). Beberapa jenis kanker *Adenocarcinoma* yang sering terjadi adalah kanker payudara (*breast cancer*), kanker paru-paru (*lung cancer*), kanker pankreas (*pancreas cancer*), dan kanker usus besar (*colon cancer*).

2.2.2 *Body Fluids*

Body fluids atau cairan tubuh adalah semua jenis cairan yang diproduksi di dalam tubuh dan kemudian diedarkan ke seluruh tubuh atau dikeluarkan dari tubuh. Tubuh manusia terdiri dari banyak jenis cairan tubuh, bahkan sebagian besar penyusun tubuh manusia adalah cairan. Manusia terdiri 75% massa air saat masih bayi dan 50% hingga 60% massa air saat dewasa.

Cairan tubuh yang menjadi penyusun tubuh manusia, diantaranya adalah darah (plasma darah, sel darah merah, dan sel darah putih), getah bening (*lymph*), urin, cairan interstisial, cairan *cerebrospinal*, *saliva*, cairan intestinal, *mucus*, *breast milk*, dan sebagainya. Sebagian besar cairan yang ada di dalam tubuh manusia tersebut dihasilkan oleh sel kelenjar. Apabila sel kelenjar ini rusak, maka sistem organ tubuh manusia akan terganggu.

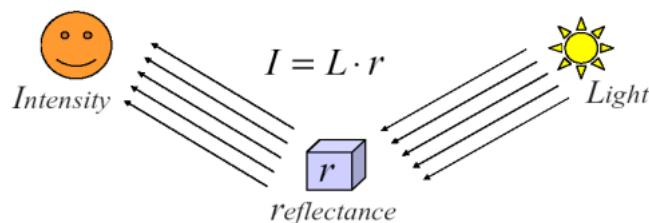
Salah satu contoh cairan yang berperan penting untuk melindungi organ dalam yaitu *mucus* atau lender. *Mucus* berperan sebagai zat pelindung bagi mulut, sinus, tenggorokan, paru-paru, lambung, dan usus. Hal tersebut berlaku juga untuk cairan tubuh lainnya, dimana masing-masing berperan penting untuk melindungi organ tubuh manusia. Dengan teknologi *CT Scan*, kanker *Adenocarcinoma* dapat dideteksi dan diklasifikasikan melalui citra cairan tubuh.

Cairan tubuh atau *body fluids* juga terdapat pada rongga antara paru-paru dan dinding dada. Paru-paru diselimuti oleh pleura dimana pada selaput tersebut terdapat cairan yang berfungsi sebagai pelumas ketika paru-paru bekerja. Jumlah cairan tersebut dapat meningkat yang disebabkan oleh keadaan klinis tertentu. Penumpukan cairan pada bagian tersebut disebut sebagai efusi pleura yang mana merupakan salah satu ciri adanya sel kanker pada organ tersebut (Win et al., 2017b).

2.2.3 Pengolahan Citra Digital

Citra adalah suatu fungsi kontinyu dari intensitas cahaya dalam bidang dua dimensi, dengan (x,y) menyatakan koordinat citra dan nilai f pada koordinat (x,y) menyatakan tingkat kecerahan atau derajat keabuan (Sumijan & Pradani, 2021). Citra dapat dideskripsikan sebagai data 2 dimensi dalam bentuk matrix $M \times N$ yang terdiri atas baris dan kolom untuk menyatakan sebuah titik pada citra. Secara matematis citra disimbolkan dengan $f(x,y)$, dimana : (x,y) : koordinat pada bidang dwi warna $F(x,y)$: intensitas cahaya pada titik (x,y) Nilai $f(x,y)$ adalah hasil kali dari : $i(x,y)$ = jumlah cahaya yang berasal dari sumber, nilainya antara 0 sampai tak terhingga. $r(x,y)$ = derajat kemampuan objek memantulkan cahaya , nilainya antara 0 dan 1 seperti diperlihatkan pada Gambar 2.1.

Jadi $f(x,y) = i(x,y) \cdot r(x,y)$



Gambar 2. 1 Akuisisi Citra Digital

Citra digital adalah bentuk representasi visual dari dunia nyata dalam bentuk digital yang terdiri dari piksel-piksel dengan nilai numerik. Elemen yang disebut dengan piksel tersebut yang kemudian diolah sehingga dapat dipahami oleh komputer. Setiap piksel dalam citra memiliki nilai numerik yang menggambarkan tingkat kecerahan atau warna pada posisi tertentu dalam citra (Rohman et al., 2021).

Teknologi pengolahan citra digital, saat ini banyak sekali digunakan dalam bidang kesehatan karena hasil pengolahan citra membantu tenaga medis mendapatkan kualitas citra yang lebih baik. Adapun tujuan lain dari adanya pengolahan citra digital adalah sebagai berikut.

- a. Peningkatan Kualitas Citra

Penajaman Kontras: Memperjelas perbedaan antara area gelap dan terang dalam citra. Peningkatan Kejelasan: Memperbaiki ketajaman dan rincian dalam citra.

Reduksi Noise: Mengurangi gangguan acak yang dapat mengaburkan citra.

b. Ekstraksi Fitur

Tepi Deteksi: Menemukan lokasi tepi atau perubahan tajam dalam citra.

Ekstraksi Tekstur: Mengidentifikasi pola tekstur dalam citra.

Ekstraksi Fitur Bentuk: Mendeteksi objek dan bentuk khusus dalam citra.

c. Segmentasi Citra

Segmentasi Warna: Membagi citra menjadi kelompok berdasarkan warna.

Segmentasi Objek: Memisahkan objek dari latar belakang dalam citra.

Pemisahan Wilayah: Membagi citra menjadi wilayah-wilayah homogen.

d. Kompresi Citra

Reduksi Ukuran: Mengurangi ukuran citra untuk penyimpanan atau transmisi yang efisien.

Kompresi Tanpa Kehilangan: Mengurangi ukuran citra tanpa kehilangan banyak informasi.

e. Pengenalan Pola

Pengenalan Objek: Mengidentifikasi objek dalam citra.

Pengenalan Karakter: Mentranskripsikan teks dari citra tulisan tangan atau cetakan.

Pengenalan Wajah: Mengidentifikasi wajah dalam citra

Teknologi pengolahan citra banyak sekali digunakan dalam dunia medis.

Proses pengolahan citra digital dilakukan dengan memberi masukan (input) berupa citra (image) dengan hasil keluaran (output) juga

berupa citra (image). Proses tersebut digunakan untuk mengekstrak ciri suatu citra. Beberapa teknik yang digunakan dalam pengolahan citra digital yaitu sebagai berikut.

1.2.4.1 Segmentasi

Segmentasi citra dilakukan dengan tujuan memisahkan antara objek yang diinginkan atau yang biasa disebut ROI (*Region of Interest*) dengan *background* atau bagian lain yang tidak diinginkan. Hasil dari proses segmentasi berupa citra biner dimana nilai 1 untuk nilai objek (ROI) dan nilai 0 untuk *background*.

1.2.4.2 Operasi Morfologi

Teknis operasi morfologi bertujuan untuk memperbaiki nilai dari proses segmentasi serta mengurangi noise dari hasil segmentasi. Operasi-operasi morfologi antara lain:

a. Erosi

Erosi (*erotion*) yaitu pengkikisan area objek yang bertetangga dengan *background*. Teknik ini dapat menyebabkan turunnya jumlah piksel pada tepi objek citra.

b. Dilasi

Dilasi (*dilation*) merupakan teknik untuk memperbesar citra objek dengan menambah lapisan-lapisan di sekeliling titik objek, atau menjadikan titik *background* (nilai biner 0) yang bertetangga dengan objek (nilai biner 1) menjadi titik objek (1). Teknik ini dapat menyebabkan meningkatnya jumlah piksel pada tepi objek citra.

c. *Opening*

Opening merupakan gabungan dari operasi erosi dan operasi dilasi. Dimulai dengan operasi erosi kemudian diikuti dengan operasi dilasi. *Opening* bertujuan untuk membuat tepi citra objek lebih halus.

d. *Closing*

Tahapan pada operasi closing dimulai dari operasi dilasi kemudian diikuti dengan operasi erosi. *Closing* bertujuan untuk menghilangkan dan menutup titik-titik lubang kecil yang terdapat dalam objek serta menggabungkan objek yang berdekatan.

e. *Median filter*

Median filter merupakan operasi untuk menghilangkan *noise* yang biasanya terdapat dalam citra dengan cara mengubah nilai setiap piksel dengan nilai median piksel terdekat.

f. *Filling*

Filling bertujuan untuk mengisi lubang (*hole*) yang mempunyai nilai biner 0 menjadi segmen objek yang solid. Operasi ini mengacu pada nilai piksel tetangganya.

1.2.4.3 Ekstraksi Fitur

Ekstraksi fitur digunakan untuk mengambil informasi berupa nilai pada citra yang dapat merepresentasikan objek pada citra (Fattah et al., 2021). Ekstraksi fitur dapat dilakukan dengan mendapatkan nilai parameter dari ciri bentuk, dan ukuran objek citra (Ashshidiqi, Suprayogi, & Bethaningtyas, 2017).

a. Luas dan keliling

Luas merupakan jumlah piksel yang ada dalam suatu citra, Keliling merupakan jumlah piksel dalam tepian citra objek. Luas citra dapat dihitung dengan Persamaan 1.

$$\text{Luas} = \sum_{i=1}^n \sum_{j=1}^m f(i, j) \quad (1)$$

Di mana m adalah jumlah baris pada citra, n adalah jumlah kolom pada citra, dan $f(i, j)$ merupakan 1 jika (i, j) adalah piksel objek dalam citra biner.

b. *Metric*

Metric merupakan nilai perbandingan antara luas dan keliling pada citra yang mempunyai nilai *range* antara 0 hingga 1.

Rumus perhitungan *metric* dapat dilihat pada Persamaan (2):

$$M = \frac{4\pi A}{C^2} \quad (2)$$

Di mana M merupakan metric, A merupakan area (luas), dan C merupakan keliling.

c. *Eccentricity*

Eccentricity ialah perbandingan nilai jari-jari objek dengan nilai jari-jari objek terpanjang yang tegak lurus. Rumus perhitungan *eccentricity* dapat dilihat pada persamaan (3):

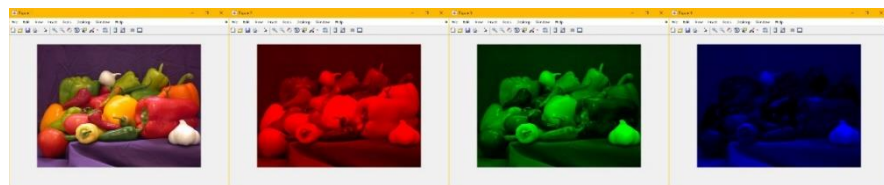
$$e = \sqrt{1 - \frac{b^2}{a^2}} \quad (3)$$

Di mana e adalah nilai *eccentricity*, b merupakan *minor axis*.

Ada tiga jenis citra berdasarkan nilai piksel, yaitu citra warna RGB, citra warna biner, dan citra warna abu (*grayscale*).

a. Citra Berwarna (RGB)

Citra berwarna atau yang sering disebut dengan RGB adalah citra yang mengandung informasi kecerahan dan warna. Piksel dalam citra berwarna direpresentasikan oleh tiga warna, yaitu warna merah (*red*), warna hijau (*green*), dan warna biru (*blue*). Ketiga warna tersebut akan membentuk kombinasi warna baru yang kemudian akan disimpan pada setiap lokasi piksel. Setiap warna memiliki rentang nilai 0-255, dimana masing-masing bernilai 8 bit sehingga format file grafis mampu menyimpan citra warna sebagai citra 24 bit. Hal tersebut menyebabkan citra warna mempunyai 24 juta kemungkinan warna. Citra masing-masing RGB dapat dilihat pada Gambar 2.2.

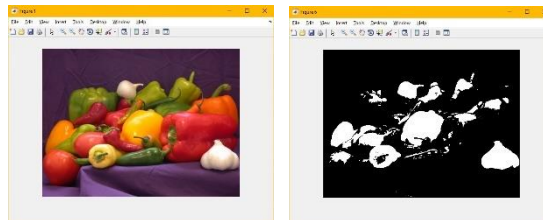


Gambar 2. 2 Citra Digital Masing-Masing RGB

b. Citra Biner

Citra biner merupakan citra hasil segmentasi yang mana piksel-piksel tersebut dikelompokkan menjadi dua kelas berdasarkan ambang

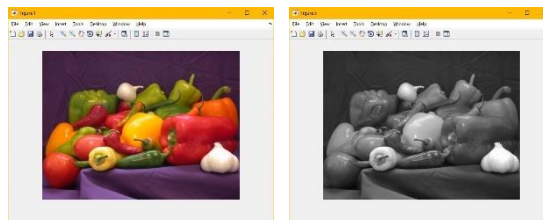
tertentu. Di dalam citra biner, setiap piksel hanya membutuhkan 1 bit memori. Sesuai dengan namanya, citra biner direpresentasikan oleh dua nilai intensitas, yaitu 1 atau 0 dimana sering kali 0 untuk hitam dan 1 untuk putih. Citra biner dapat dilihat pada Gambar 2.3.



Gambar 2. 3 Citra Digital Hasil Konversi RGB Menjadi Biner

c. Citra Abu (*Grayscale*)

Citra *grayscale* adalah citra yang hanya memiliki tingkat kecerahan sebagai informasi. Setiap piksel dalam citra grayscale membutuhkan 8 bit memori yang mana nilai intensitasnya terletak dalam rentang 0 hingga 255. Nilai 0 mewakili warna hitam atau gelap, sementara nilai 255 mewakili warna putih atau terang. Citra *grayscale* dapat dilihat pada Gambar 2.4.



Gambar 2. 4 Citra Digital Hasil Konversi RGB Menjadi *Grayscale*

2.2.4 *Gray Level Co-occurrence Matrix*

Gray Level Co-occurrence Matrix (GLCM) adalah suatu metode yang digunakan untuk analisis tekstur/ekstraksi ciri. GLCM merupakan suatu matriks yang menggambarkan frekuensi munculnya pasangan dua piksel dengan intensitas tertentu dalam jarak dan arah tertentu dalam citra (Maulida et al., 2013). Metode GLCM menentukan hubungan tekstur antar piksel dengan melakukan operasi berdasarkan statistik orde kedua pada gambar.

Jarak dalam koordinat pasangan piksel disimbolkan dengan suatu d dan orientasi sudut disimbolkan dengan Θ . Jarak direpresentasikan dalam

piksel dan sudut direpresentasikan dalam derajat. GLCM merepresentasikan hubungan antara dua piksel yang berdekatan dengan intensitas abu-abu, jarak dan sudut. Terdapat delapan sudut yang dapat digunakan dalam metode GLCM, yaitu 0° , 45° , 90° , 135° , 180° , 225° , 270° atau 315° (Rohman et al., 2021). Langkah-langkah dalam melakukan GLCM yaitu (Yunianto et al., 2021):

1. Pembacaan input citra yang merupakan hasil dari proses *otsu thresholding*.
2. Nilai gray-level ditentukan dengan nilai tertinggi yakni 256 yang digunakan untuk membangun matrik *framework*.
3. Menentukan arah dan jarak piksel referensi dengan piksel tetangga. Arah yang digunakan yakni 0° , 45° , 90° , dan 135° dengan jarak piksel yang digunakan yakni 1.
4. Menghitung jumlah nilai kookurensi berdasarkan arah dan jarak yang telah ditentukan.
5. Membagi nilai kookurensi matrik simetris dengan jumlah keseluruhan nilai kookurensi untuk diperoleh matrik normal.
6. Menghitung 11 fitur statistik, yaitu *contrast*, *correlation*, *energy*, *homogeneity*, *mean*, *standart deviation*, *entropy*, *variance*, *smoothness*, *kurtosis*, *skewness*.
7. Fitur statistik dihitung untuk masing-masing matrik kookurensi pada arah yang sudah ditentukan.

Bagian citra yang digunakan dalam perhitungan matriks kookurensi adalah bagian kanker yang digambarkan pada citra input yang telah melalui proses *pre-processing*. Adapun teknik atau fitur statistik yang dapat digunakan dalam GLCM yaitu:

1. Kontras (*contrast*), merupakan jumlah ragam intensitas keabuan dalam citra.

Persamaan matematis kontras dapat dilihat pada persamaan 4.

$$\sum_{i_1} \sum_{i_2} (i_1 - i_2)^2 p(i_1, i_2) \quad (4)$$

2. Korelasi (*correlation*), merupakan keterkaitan linier pada derajat keabuan citra.

Persamaan matematis korelasi dapat dilihat pada persamaan 5.

$$\sum_i \sum_j \frac{(i - \mu_i)(j - \mu_j) p_{(ij)}}{\sigma_i \sigma_j} \quad (5)$$

3. Energi (*energy*), merupakan nilai konsentrasi pada pasangan intensitas matriks kookurensi. Persamaan matematisnya dapat dilihat pada persamaan 6.

$$\sum_{i1} \sum_{i2} p^2(i1, i2) \quad (6)$$

4. Homogenitas (*homogeneity*), merupakan ukuran nilai kesamaan variasi dari intensitas citra. Persamaan matematisnya dapat dilihat pada persamaan 7.

$$\sum_{i1} \sum_{i2} \frac{p(i1, i2)}{1 + |i1 - i2|} \quad (7)$$

5. Rata-rata (*mean*), merupakan nilai rata-rata yang ada pada citra. Rumus matematis mencari rata-rata dapat dilihat pada persamaan 8.

$$\sum_i \sum_j (i, j) p(i, j) \quad (8)$$

6. Standar deviasi (*standart deviation*), merupakan ragam histogram dari suatu citra. Rumus dalam menghitung standar deviasi dapat dilihat pada persamaan 9.

$$\sqrt{\sum_{i=1}^L \sum_{j=1}^L (i - \mu)^2 p(i, j)} \quad (9)$$

7. Entropi (*entropy*), merupakan ketidakteraturan atas keabuan yang terdapat dalam citra. Persamaan matematis entropi dapat dilihat pada persamaan 10.

$$\sum_{i1} \sum_{i2} p(i1, i2) \log p(i1, i2) \quad (10)$$

8. *Variance*, merupakan variasi dari elemen matriks kookurensi dimana jika citra memiliki transisi derajat keabuan kecil akan memiliki variasi yang kecil. Persamaan matematisnya dapat dilihat pada persamaan 11

$$\sum_i \sum_j (i - \mu)^2 p(i, j) \quad (11)$$

9. Kehalusan (*smoothness*), merupakan tingkat kehalusan dari suatu intensitas citra. Persamaan matematis kehalusan dapat dilihat pada persamaan 12.

$$R = 1 - \frac{1}{1+\sigma^2} \quad (12)$$

10. Kurtosis (kurtosis), merupakan tingkat keruncingan relatif kurva histogram yang ada pada suatu citra. Persamaan matematisnya dapat dilihat pada persamaan 13.

$$\frac{1}{\sigma^4} \sum_i \sum_j (i - \mu)^4 p(i, j) - 3 \quad (13)$$

11. *Skewness*, merupakan tingkat kemiringan relatif kurva histogram suatu citra. Persamaan matematis untuk mencari skewness dapat dilihat pada persamaan 14.

$$\frac{1}{\sigma^3} \sum_i \sum_j (i - \mu)^3 p(i, j) \quad (14)$$

Dimana: i/j = nilai derajat keabuan; $p(i,j)$ = nilai histogram; L = nilai tertinggi dari intensitas; σ = standar deviasi; dan μ = mean.

12. *Inverse Difference Moment (IDM)*, biasanya disebut homogenitas yang mengukur homogenitas lokal suatu gambar. Fitur IDM memperoleh ukuran kedekatan distribusi elemen GLCM ke diagonal GLCM. Nilai bobot IDM adalah kebalikan dari bobot Kontras, dengan bobot menurun secara eksponensial menjauh dari diagonal (Raut et al., 2016). Persamaan matematis IDM terdapat pada persamaan 15.

$$IDM = \sum_i^M \sum_j^N \frac{P(i,j)}{1+(i,j)^2} \quad (15)$$

Dimana:

M = maksimum nilai level keabuan baris ke- i

N = maksimum nilai level keabuan kolom ke- j

i = nilai level keabuan baris ke- i

j = nilai level keabuan kolom ke- j

$P(i,j)$ = peluang nilai level keabuan pada baris ke- i dan kolom ke- j

Metode ekstraksi *Gray Level Co-occurrence Matrix (GLCM)* telah banyak digunakan peneliti sebelumnya. Metode ini telah diuji dengan menggunakan objek yang berbeda-beda sehingga menghasilkan akurasi

yang berbeda. Beberapa penelitian sebelumnya telah menggunakan metode ini dapat dilihat pada Tabel 2.2.

Tabel 2. 2 Penelitian Terkait *Gray Level Co-occurrence Matrix* (GLCM)

No	Author (Tahun)	Judul	Objek	Metode	Hasil
1	(Potghan et al., 2018)	Multi Layer Perceptron Based Lung Tumor Classification	<i>Lung Tumor</i>	<i>Gray Level Co-occurrence Matrix</i> (GLCM)	Dari pengujian GLCM dengan K-NN menghasilkan akurasi 98,30% sedangkan GLCM dengan MLP menghasilkan akurasi 98,31%.
2	(Hussain & Khunteta, 2020)	Semantic Segmentation of Brain Tumor from MRI Images and SVM Classification using GLCM Features	<i>Brain Tumor</i>	<i>Gray Level Co-occurrence Matrix</i> (GLCM)	Dari hasil pengujian diperoleh hasil rata-rata akurasi adalah 93,05%.
3	(Yunianto et al., 2021)	Klasifikasi Kanker Paru Paru menggunakan Naïve Bayes dengan Variasi Filter dan Ekstraksi Ciri GLCM	Kanker paru-paru	<i>Gray Level Co-occurrence Matrix</i> (GLCM)	Hasil akurasi paling tinggi sebesar 88.33%

Berdasarkan Tabel 2.2 Metode ekstraksi *Gray Level Co-occurrence Matrix* (GCLM) dapat digunakan pada beberapa objek yang berbeda. Seperti penelitian yang dilakukan oleh Potghan, S., Rajamenakshi, R., & Dr. Bhise, A (2018), objek yang digunakan adalah tumor paru-paru. Metode klasifikasi yang digunakan merupakan perbandingan antara K-NN dengan MLP. Hasil penelitian menunjukkan bahwa metode GLCM dan MLP menghasilkan akurasi lebih tinggi yaitu 98,31%. Penelitian yang dilakukan oleh Hussain, A. & Khunteta, A (2020) menggunakan tumor otak sebagai objek penelitian. Hasil penelitian menunjukkan bahwa

metode GLCM yang dikombinasikan dengan metode *Support Vector Machine* (SVM) memiliki rata-rata akurasi sebesar 93,05%. Penelitian yang dilakukan oleh Yuniarto, M., Soeparmi., dkk (2021) menggunakan kanker paru-paru sebagai objek penelitian. Pada penelitian ini, metode GLCM dikombinasikan dengan Naïve Bayes sehingga menghasilkan akurasi sebesar 88,33%.

2.2.5 Haar Wavelet

Wavelet adalah fungsi matematika yang dikembangkan untuk menyortir data berdasarkan frekuensi. Transformasi *Wavelet* mengubah data dari spasial menjadi domain frekuensi dan kemudian menyimpan setiap komponen dengan skala resolusi yang sesuai (Kaur & Malhotra, 2015). Kata *Wavelet* berasal dari *wave* yang artinya gelombang. *Wavelet* merupakan gelombang yang dibatasi atau terlokalisasi. Penggunaan *wavelet* pada saat ini sudah semakin berkembang dengan munculnya area sains terpisah yang berhubungan dengan analisis *wavelet* dan teori transformasi *wavelet* atau dapat dikatakan sebagai gelombang pendek (Satria et al., 2018).

Haar adalah *wavelet* yang paling tua dan sederhana, diperkenalkan oleh Alfred Haar pada tahun 1909. Metode ini bertujuan untuk melakukan ekstraksi ciri tekstur pada citra digital. *Haar Wavelet* ini memakai fungsi *Haar* sebagai fungsi basisnya. *Haar Wavelet* pada suatu citra digital dilakukan dengan menggunakan dua penapis yaitu penapis lolos rendah (*Low Pass Filter/ LPF*) dan penapis lolos tinggi (*High Pass Filter/ HPS*) oleh karenanya diperoleh koefisien *wavelet*. Penjelasan sederhananya ialah *Low Pass Filter/ LPF* adalah proses *averaging* untuk mendapatkan bagian dari citra yang berfrekuensi rendah dan penggunaan *High Pass Filter/ HPF* berarti melakukan proses *differencing* untuk mendapatkan bagian dari citra yang berfrekuensi tinggi (Faiqoh et al., 2020).

Beberapa matriks yang digunakan untuk transformasi *Haar Wavelet* adalah seperti di bawah ini (Satria et al., 2018)..

Matriks *lowpass* Haar $H_0(n)$ dan *highpass* Haar $H_1(n)$ adalah sebagai berikut.

$$H_0^{(n)} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & 1 & \dots & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & \dots & 1 & 1 \end{bmatrix} \in R^{n \times n}$$

$$H_1^{(n)} = \frac{1}{2} \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 1 & -1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & 1 & -1 \end{bmatrix} \in R^{n \times n}$$

Matrik untuk melakukan *down sampling* 2 $D(n)$ adalah sebagai berikut.

$$D^{(n)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & 1 & 0 \end{bmatrix} \in R^{\frac{n}{2} \times n}$$

Hasil proses *down sampling* dengan ekspresi $D(n)$ $H_0(n)$ (*filter lowpass*) dan $D(n)$ $H_1(n)$ (*filter highpass*) adalah sebagai berikut.

$$T^{(n)} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & \vdots & & \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \in R^{\frac{n}{2} \times n}$$

$$B^{(n)} = \frac{1}{2} \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & \vdots & & \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \in R^{\frac{n}{2} \times n}$$

Berikut adalah langkah-langkah metode Transformasi *Haar Wavelet*.

1. *Transpose* matriks input.
2. Mengkalikan filter *low* dan *high* yang sudah digabung dengan matriks input.
3. Melakukan proses *transpose* matriks hasil dari langkah 2.
4. Mengkalikan filter *low* dan *high* yang sudah digabung dengan hasil *transpose* langkah 3.
5. Menampilkan output yang terdiri dari 4 subbidang yaitu LL, HL, LH, LL.
6. Jika lebih dari 1 level, ulangi lagi langkah 1-5 dengan input matriks subbidang LL dari level sebelumnya, begitu seterusnya.

Matriks input dari metode Transformasi *Haar Wavelet* tersebut adalah matriks citra yang sudah berupa *grayscale*. Setelah proses 1 sampai dengan 5, maka akan didapat matriks fitur (karakteristik) dari citra

tersebut. Jadi, dengan kata lain transformasi *Haar Wavelet* mengambil fitur dari matriks citra yang akan diolah.

Tabel 2. 3 Penelitian Terkait *Haar Wavelet*

No	Author (Tahun)	Judul	Objek	Metode	Hasil
1	(Karthiga & Narasimhan, 2018)	Automated Diagnosis of Breast Cancer Using Wavelet Based Entropy Features	<i>Breast Cancer</i>	<i>Wavelet dan Support Vector Machine (SVM)</i>	Wavelet - Linear SVM 93,3%, Wavelet - Quadratic SVM 92,7%, Wavelet - Fine Gaussian SVM 91,3%.
2	(Bahaalddin & Ahmed, 2020)	Breast Mass Classification Based on Hybrid Discrete Cosine Transformation–Haar Wavelet Transformation	<i>Breast Cancer</i>	<i>Haar Wavelet dan KNN</i>	Rata-rata akurasi adalah 95%
3	(Melekoo dappattu et al., 2021)	Early Detection of Breast Malignancy Using Wavelet Features and Optimized Classifier	<i>Breast Cancer</i>	Wavelet Transform, (ANN), (SVM), <i>Extreme Learning Machine (ELM)</i> , dan ELM-Grasshopper Optimization Algorithm (GOA)	ELM-GOA menghasilkan akurasi 99.33%, sensitifitas 98%, presisi 100%, dan spesifitas 100%.

Dari Tabel 2.3 dapat diketahui bahwa metode *Haar Wavelet* dapat digunakan sebagai metode ekstraksi pada *breast cancer*. Pada penelitian yang dilakukan oleh Karthiga, R. & Narasimhan, K. (2018), metode *Wavelet Transform* dapat digunakan sebagai fitur ekstraksi citra *breast cancer*. Pada penelitiannya, penulis menggabungkan *Wavelet Transform* dengan tiga macam SVM, yaitu Linear SVM, Quadratic SVM, dan Fine Gaussian SVM. Jumlah data yang digunakan adalah 300 citra yang terbagi menjadi training set dan testing set. Hasil dari penelitian tersebut ialah

metode Wavelet - Linear SVM mendapat hasil akurasi tertinggi, yaitu 93,3% sedangkan Wavelet - Quadratic SVM 92,7% dan Wavelet – Fine Gaussian SVM 91,3%. Penelitian serupa juga dilakukan oleh Bahaalddin, B. M. & Ahmed, H. O (2020). Dalam penelitian tersebut, metode *Haar Wavelet* dan *K-Nearest Number* digunakan sebagai metode ekstraksi dan klasifikasi. Jumlah data yang digunakan adalah 322 citra yang terbagi dalam 3 kelas, yaitu normal, *benign*, dan *malignant*. Hasil dari penelitian tersebut adalah model dengan akurasi mencapai 95%. Penelitian dengan objek sama pun dilakukan oleh Melekoodappattu et al., (2021). Metode yang digunakan untuk ekstraksi adalah *Wavelet Transform* sedangkan metode klasifikasinya adalah ANN, SVM, ELM, dan ELM-GOA. Peneliti membandingkan empat metode klasifikasi tersebut, kemudian diperoleh hasil bahwa ELM-GOA menghasilkan akurasi 99.33%, sensitifitas 98%, presisi 100%, dan spesifitas 100%.

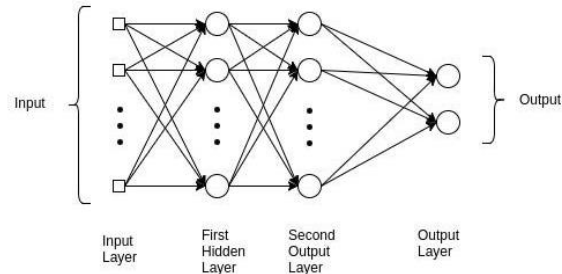
2.2.6 *Multilayer Perceptron*

Metode klasifikasi *Multilayer Perceptron* (MLP) merupakan bagian dari *Artificial Neural Network* (ANN). Algoritma yang diterapkan dalam *Multilayer Perceptron* (MLP) ini meniru cara kerja syaraf manusia yang memiliki kemampuan belajar dan menyesuaikan model dengan data-data terbaru meskipun data yang dimiliki terbatas. Keuntungan dari pemodelan dengan MLP ini adalah:

- a. Model ini dapat dibuat hanya berdasar data historis proses masukan-keluaran.
- b. Dapat memberikan simulasi hubungan antara banyak masukan-banyak keluaran.

Multilayer Perceptron (MLP) terdiri dari beberapa lapisan dimana setiap lapisannya terdiri dari beberapa *neuron*. *Neuron* yang ada di lapisan-lapisan tersebut kemudian akan dihubungkan oleh bobot dan unit output yang merupakan fungsi penjumlahan dari *input* ke *neuron* dimodifikasi

oleh transfer *non-linear* sederhana atau aktivasi. Ilustrasi jaringan antar *neuron* dapat dilihat pada Gambar 2.6.



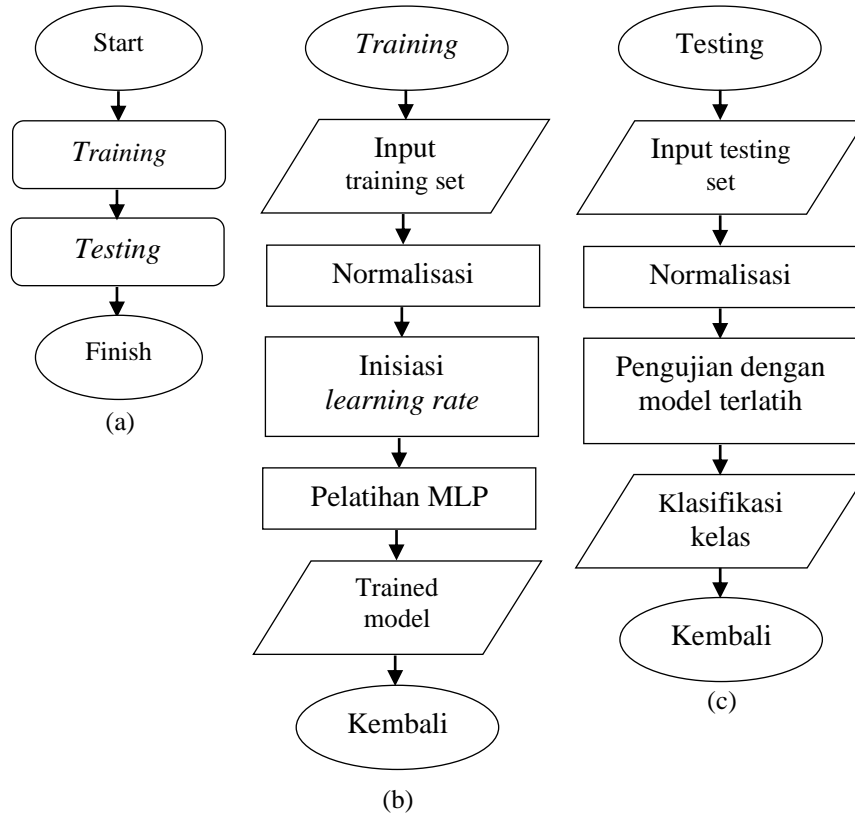
Gambar 2. 5 Struktur *Multilayer Perceptron*

Komponen MLP secara umum terdiri dari tiga lapisan utama, yaitu *input*, *hidden*, dan *output*. Berikut penjelasan mengenai struktur komponen penyusun MLP.

- a. *Neuron* atau *Node*; sel syaraf tiruan yang merupakan elemen pengolahan dalam MLP.
- b. *Input layer* atau lapisan masukan; terdiri dari n jumlah *neuron* yang mewakili n jumlah variabel dari dataset. Pada lapisan ini, nilai masukan akan disimpan sebagai awal proses.
- c. *Hidden layer*; terdiri dari beberapa lapisan tersembunyi dimana jumlahnya tergantung dari arsitektur yang dirancang. Semua proses pelatihan dan pengenalan dijalankan di lapisan ini.
- d. *Output layer* atau lapisan keluaran; terdiri dari m *neuron output* yang jumlahnya disesuaikan dengan jumlah kelas dataset. Pada lapisan ini, hasil perhitungan sistem berdasarkan fungsi aktivasi pada *hidden layer* akan dimunculkan.

Terdapat dua parameter penting yang ada dalam MLP, yaitu fungsi aktivasi dan fungsi optimisasi. Fungsi aktivasi menentukan *output* pada sebuah *neuron* dari beberapa elemen *input*. Sedangkan fungsi optimisasi berfungsi untuk menentukan bobot paling tepat sesuai dengan *input* dan *output*. MLP termasuk algoritma *supervised* yang membutuhkan proses pembelajaran untuk menentukan bobot optimal yang digunakan dalam proses pengujian data. Selama proses pelatihan, MLP secara berkala terjadi perubahan bobot dengan data latih pada jaringan sampai map *input-*

output yang diinginkan terjadi. Diagram alir proses klasifikasi, pelatihan, dan pengujian dapat dilihat pada Gambar 2.7.



Gambar 2. 6 Diagram Alir Proses Klasifikasi, Pelatihan, dan Pengujian

Terdapat beberapa model *Multilayer Perceptron* (MLP) yang dapat digunakan dalam *image processing*. Perbedaan model yang digunakan akan mempengaruhi hasil akurasi. Beberapa model *Multilayer Perceptron* (MLP) tersebut adalah sebagai berikut.

a. *Scaled Conjugate Gradient Backpropagation* (*trainscg*)

Scaled Conjugate Gradient (SCG) adalah algoritma *supervised learning* yang digunakan untuk feedforward jaringan saraf tiruan (NN) dan bagian dari kelas metode *Conjugate Gradient*. SCG menggunakan konsep serupa dengan strategi optimasi umum, tetapi memilih arah dan ukuran langkah secara lebih efektif dengan menggunakan informasi dari pendekatan orde kedua seperti terlihat pada persamaan 16.

$$E(w+y) = E(w) + E'(w)^T + \frac{1}{2} y^T E''(w)y \quad (16)$$

Di SCG, setiap iterasi menghitung jarak optimal. *Line search* kemudian digunakan untuk menentukan jarak optimal untuk bergerak di sepanjang arah pencarian saat ini sebagai terlihat pada persamaan 17.

$$W_{k+1} = W_k + a_k * P_k \quad (17)$$

Kemudian arah pencarian berikutnya dilakukan sehingga, terkonjugasi ke instruksi pencarian sebelumnya. P_k adalah fungsi dari a_k , matriks Hessian dari fungsi error dan juga matriks turunan kedua. SCG menggunakan skalar a_k yang seharusnya mengatur ketidakterbatasan matriks Hessian (Nayak, 2017).

Penulisan *syntax trainscg* dalam bahasa Matlab adalah sebagai berikut.

```
net.trainFcn = 'trainscg'  
[trainedNet,tr] = train(net,...)
```

b. *One Step Secant Backpropagation (trainoss)*

Algoritma *One Step Secant* (OSS) adalah algoritma yang dapat melatih jaringan apa pun selama memiliki *weight*, *net input* dan *activation functions* yang memiliki fungsi turunan. Algoritma ini akan memaksimalkan waktu dan meningkatkan akurasi yang dihasilkan oleh *standard backpropagation* (Solikhun et al., 2020).

Penulisan *syntax trainoss* dalam bahasa Matlab adalah sebagai berikut.

```
net.trainFcn = 'trainoss'  
[net,tr] = train(net,...)
```

c. *Gradient Descent with Momentum (traingdm)*

Dalam pendekatan *standard gradient descent*, tujuannya adalah untuk meminimalkan nilai *error* dari *error function* yang memiliki parameter *weight* dalam jaringan. *Gradient error* fungsional yang berkaitan dengan *weight* akan menjadi bahan evaluasi. Dengan nilai ini, *weights* jaringan disesuaikan sepanjang arah negatif gradien dengan tujuan untuk meminimalkan *error* (Masood et al., 2016).

Penulisan *syntax traingdm* dalam bahasa Matlab adalah sebagai berikut.

```
net.trainFcn = 'traingdm'
[net,tr] = train(net,...)
```

Metode klasifikasi *Multi Layer Perceptron* (MLP) telah banyak digunakan dalam penelitian. Peneliti sering menggabungkan metode ini dengan berbagai metode ekstraksi, seperti *Gray Level Co-occurrence Matrix* (GLCM), *Haar Wavelet*, dan sebagainya. Metode kombinasi dan objek yang berbeda akan menghasilkan akurasi yang berbeda. Tabel 2.4 menunjukkan beberapa penelitian yang menggunakan metode MLP dapat dilihat pada

Tabel 2. 4 Penelitian Terkait *Multi Layer Perceptron* (MLP) atau *Artificial Neural Network* (ANN)

No	Author (Tahun)	Judul	Objek	Metode	Hasil
1	(Technology et al., 2019)	Lung Cancer Detection Using Artificial Neural Network	Kanker paru-paru	Artificial Neural Network (ANN)	Model ANN dapat mendeteksi kanker dengan tingkat akurasi 96.67%.
2	(Pradana et al., 2022)	Klasifikasi Penyakit Jantung Menggunakan Metode Artificial Neural Network	Penyakit jantung	Artificial Neural Network (ANN)	Hasil pengukuran performa, diperoleh akurasi 73.77%, presisi 80.43%, recall 84.09% dan f1-score sebesar 82.22%
3	(Max, 2023)	Penerapan Multilayer Perceptron Untuk Identifikasi Kanker Payudara	Kanker payudara	Multilayer Perceptron (MLP)	Model dapat menghasilkan prediksi dengan nilai akurasi sebesar 93.59%

Dari Tabel 2.4 dapat diketahui bahwa metode *Multi Layer Perceptron* (MLP) atau *Artificial Neural Network* (ANN) dapat digunakan untuk mengklasifikasikan beberapa objek yang berbeda. Seperti penelitian yang dilakukan oleh Nasser, I. M. & Abu-Naser, S. S. (2019) yang menggunakan kanker paru-paru sebagai objek penelitian. Hasil penelitian

menunjukkan bahwa model dapat mendeteksi adanya kanker dengan tingkat akurasi sebesar 96,67%. Penelitian yang dilakukan Pradana, D. G., Alghifari, M. L., dkk (2022) menggunakan penyakit jantung sebagai objek dalam penelitian. Metode ANN yang dengan 304 data ini menghasilkan akurasi sebesar 73.77%, presisi 80.43%, recall 84.09% dan f1-score sebesar 82.22%. Penelitian yang dilakukan Sari, N. R. & Mar'atullatifah, Y. (2023) menggunakan kanker payudara sebagai objek penelitian. Dari beberapa pengujian, model dapat memprediksi adanya kanker payudara dengan nilai akurasi sebesar 93,59%.

2.2.7 Matlab

Matrix Laboratory atau yang biasa disebut dengan Matlab merupakan sebuah program yang digunakan untuk analisis dan komputasi numerik. Saat ini, matlab telah dikembangkan menggunakan bahasa C++ dan Assembler dari yang awalnya menggunakan bahasa Fortran. Matlab bersifat *extensible*, dalam arti bahwa seorang pengguna dapat menulis fungsi baru untuk ditambahkan di library jika fungsi – fungsi *built-in* yang ada dalam *environment* tidak dapat melakukan tugas tertentu.

Matlab merupakan bahasa pemrograman tingkat tinggi yang berbasis pada matriks sehingga sering digunakan untuk teknik komputasi dan digunakan untuk menyelesaikan masalah-masalah yang melibatkan operasi matematika, elemen matriks, optimasi, aproksimasi, dan lain-lain. Selain itu Matlab banyak digunakan untuk

- Matematika dan komputasi.
- Pengembangan dan algoritma.
- Pemrograman pemodelan, simulasi dan pembuatan prototipe.
- Analisa data, eksplorasi dan visualisasi.
- Analisa numerik dan statistik.
- Pengembangan aplikasi teknik



Gambar 2. 7 Logo Matlab

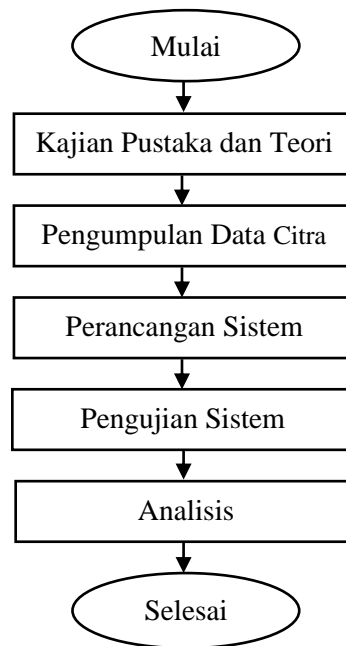
2.2.8 APPS Designer

App Designer adalah pengembangan *environment* interaktif untuk mendesain tata letak aplikasi atau *layout* dan memprogramnya. *App Designer* menyediakan Editor yang terintegrasi penuh pada MATLAB® dan terdiri dari banyak komponen interaktif untuk UI. *App Designer* juga menawarkan bantuan tata letak *grid* untuk mengatur *interface* dan juga memiliki kemampuan *reflow* otomatis untuk membuat aplikasi tersebut dapat mendeteksi dan merespons perubahan ukuran layar. *App Designer* mengintegrasikan dua tugas utama pembuatan aplikasi, yaitu menata komponen visual *user interface* (GUI) dan membuat pemrogramannya. *App Designer* direkomendasikan untuk digunakan dalam membuat aplikasi di MATLAB.

BAB III METODOLOGI PENELITIAN

3.1 Metode Penelitian

Metode penelitian merupakan langkah-langkah atau tahapan yang dilakukan untuk mencapai tujuan dalam penelitian. Adapun metode penelitian yang digunakan dalam penelitian ini disajikan dalam diagram alir pada Gambar 3.1.



Gambar 3. 1 Diagram Alir Penelitian

Diagram alir pada Gambar 3.1 digunakan sebagai landasan dalam penelitian yang menggunakan metode *Gray Level Co-occurrence Matrix* (GLCM) dan *Haar Wavelet* sebagai metode ekstraksi serta *Multilayer Perceptron* sebagai metode klasifikasi. Adapun penjelasan setiap proses dalam diagram alir adalah sebagai berikut.

3.1.1 Mulai

Tahap mulai ini adalah tahap dimana penelitian diawali. Penelitian diawali dengan diskusi bersama dosen pembimbing terkait tema atau topik yang akan dibahas dalam tugas akhir. Setelah membahas topik maka dilanjutkan dengan membahas judul dari tugas akhir tersebut.

3.1.2 Kajian Pustakan dan Teori

Tahap kajian pustaka dan teori merupakan tahap pencarian referensi yang berkaitan dengan topik penelitian. Dalam hal ini, penulis mencari beberapa penelitian terdahulu yang membahas tentang klasifikasi atau identifikasi kanker melalui *body fluids*, metode ekstraksi berupa *Gray Level Co-occurrence Matrix* (GLCM) dan *Haar Wavelet*, serta metode klasifikasi berupa *Multilayer Perceptron* (MLP). Beberapa sumber rujukan tersebut kemudian digunakan sebagai bahan observasi dan tambahan informasi dalam penelitian.

3.1.3 Pengumpulan Data Citra

Tahap pengumpulan data ini meliputi pengumpulan citra kanker *Adenocarcinoma body fluids* yang diambil dari Hospital University of Science Malaysia (HUSM), Kelantan, Malaysia, bagian *pathology*. Data tersebut telah divalidasi oleh ahli *pathology* yaitu Assoc. Prof. Dr. Anani Aila Mat Zin (Department Pathology, University of Science Malaysia). Jumlah citra yang akan digunakan dalam penelitian adalah 1.273 citra yang terdiri dari 303 citra *Adenocarcinoma Peritoneal*, 263 citra *Normal Peritoneal*, 339 citra *Adenocarcinoma Pleura*, dan 368 citra *Normal Pleura*. Data citra tersebut akan melalui beberapa proses tahapan pengolahan citra sebelum masuk dalam klasifikasi.

3.1.4 Perancangan Sistem

Perancangan sistem meliputi pembuatan desain APPS, pembuatan *coding* atau program yang mengimplementasikan citra kanker *Adenocarcinoma* sebagai inputan sesuai dengan desain APPS yang telah dirancang, serta pengolahan citra. Citra kanker *Adenocarcinoma* akan melalui tahap *preprocessing*, seperti *cropping* untuk menyesuaikan ukuran masing-masing citra yang akan digunakan. Tahap *processing*, yaitu ekstraksi fitur dengan metode *Gray Level Co-occurrence Matrix* (GLCM) dan *Haar Wavelet* serta klasifikasi dengan menggunakan metode *Multilayer Perceptron* (MLP).

3.1.5 Pengujian Sistem

Pengujian sistem dilakukan dengan menggunakan dataset yang telah ada. Dataset tersebut akan dibagi menjadi data *training*, data *testing*, dan data validasi. Jumlah citra yang digunakan sebagai data *training* adalah 90% dari keseluruhan dataset, 5% untuk data *testing*, dan 5% digunakan sebagai data validasi. Tahap pengujian dilakukan dengan melakukan *training* terlebih dahulu kemudian dilakukan *testing* untuk mengetahui akurasi dari sistem yang dibuat. Dalam penelitian ini, metode evaluasi yang digunakan dalam pengujian sistem adalah *10 run cross validation*. Data akan dibagi menjadi 10 bagian dimana 9 diantaranya digunakan sebagai *training* dan 1 bagian untuk *testing*. Prosedur tersebut akan diulang hingga 10 kali.

3.1.6 Analisis

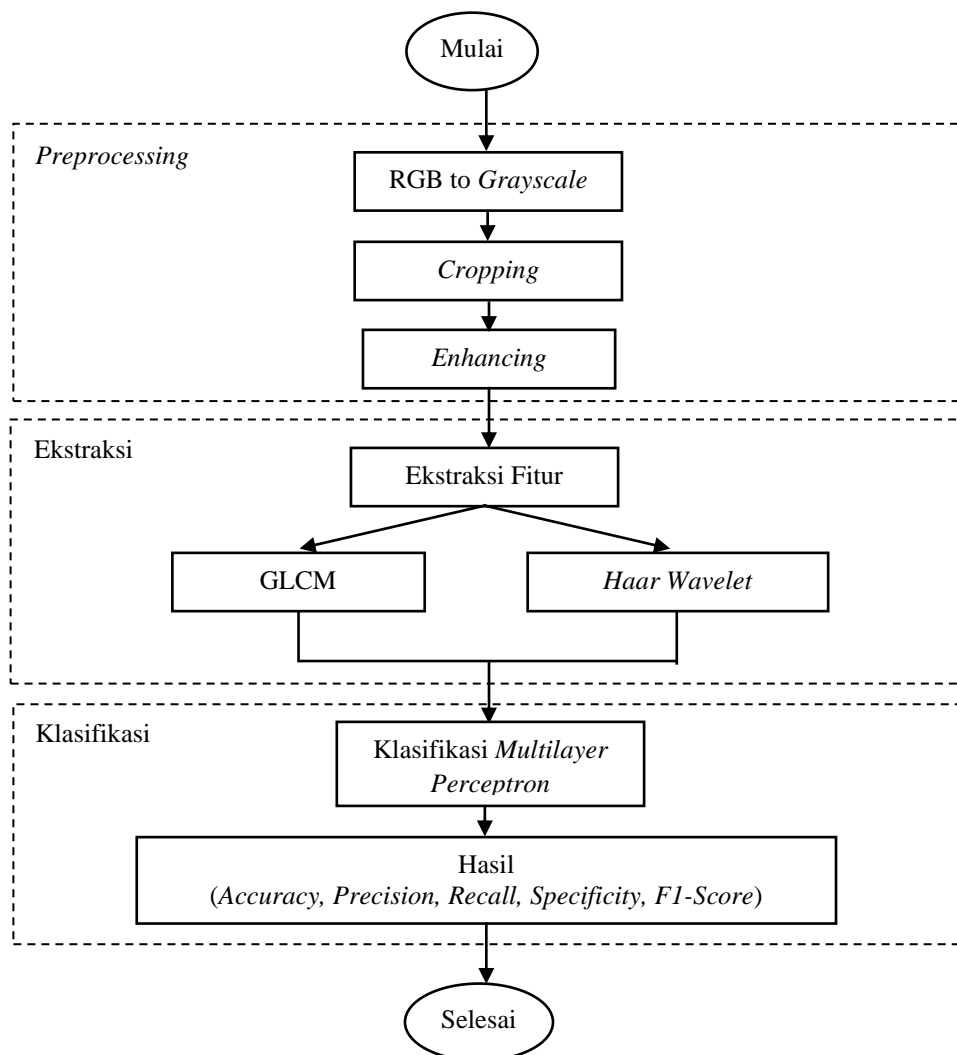
Analisis dilakukan setelah citra melalui tahap klasifikasi. Output dari klasifikasi tersebut yang digunakan sebagai bahan analisis. Proses analisis meliputi tingkat akurasi sistem serta besar *error* dari sistem tersebut. Dalam penelitian ini, analisis yang digunakan adalah analisis kualitatif berupa pengamatan secara visual hasil dari perbandingan empat jenis *enhancement* dan analisis kuantitatif berupa penghitungan *mean* dan standar deviasi dari hasil ekstraksi, analisis grafik *performance*, *confusion matrix*, serta kurva ROC. Tahap analisis merupakan tahap akhir dalam penelitian, yang mana nantinya hasil dari analisis ini juga dapat digunakan sebagai bahan evaluasi untuk penelitian selanjutnya.

3.1.7 Selesai

Penutup dari semua proses penelitian adalah penyusunan laporan tugas akhir. Semua proses yang telah dilaksanakan akan dicantumkan dalam laporan, termasuk penarikan kesimpulan dan pemberian saran.

3.2 Perancangan Sistem

Sistem dirancang dengan menggunakan *software* MATLAB. Perancangan ini dibuat agar sistem dapat bekerja sesuai dengan yang diharapkan, yaitu melakukan pengolahan citra sehingga diperoleh hasil klasifikasi kanker *Adenocarcinoma* secara lebih akurat dan efisien. Adapun perancangan sistem secara keseluruhan dapat dilihat pada Gambar 3.2.



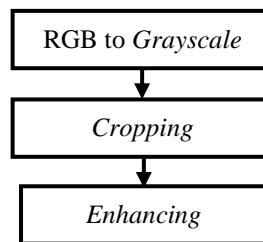
Gambar 3. 2 Flowchart Perancangan Sistem

Gambar 3.2 merupakan tahapan yang dilalui masing-masing citra mulai dari input hingga output. Input dari proses ini adalah citra *body fluids* yang mana dalam prosesnya akan melalui tahap *preprocessing*, ekstraksi, dan klasifikasi. *Output* atau hasil proses klasifikasi akan menunjukkan level kelas sel *body fluids* dan tingkat akurasi yang dihasilkan sistem. Adapun detail dari masing-masing proses adalah sebagai berikut.

3.2.1 Preprocessing

Proses *preprocessing* dilakukan dengan tujuan mempersiapkan citra yang nantinya akan digunakan pada tahap ekstraksi. Tahap *preprocessing* ini dilakukan dengan menggunakan MATLAB R2020a dengan citra yang

digunakan berjumlah 1.273 citra. Pada tahap *preprocessing* ini, citra akan diberi tiga perlakuan, yaitu mengubah citra RGB menjadi *Grayscale*, *Cropping*, dan *Enhancing*. Ketiga perlakuan tersebut dapat dilihat pada Gambar 3.3.



Gambar 3. 3 Flowchart Preprocessing

Proses yang terjadi di setiap perlakuan tersebut berbeda-beda. Tujuan akhir dari masing-masing perlakuan adalah untuk menghasilkan sebuah citra dengan kualitas terbaik sehingga dapat menghasilkan akurasi yang terbaik dari model yang diuji.

a. *RGB to Grayscale*

Tahap awal pada proses *preprocessing*, yaitu mengubah citra RGB menjadi citra abu (*grayscale*). Tujuan dari perubahan format image ini adalah untuk menyederhanakan citra yang awalnya terdiri dari tiga layer matriks, yaitu *R-layer* (merah), *G-layer* (hijau), dan *B-layer* (biru) menjadi satu layer matriks sehingga durasi pemrosesan menjadi lebih singkat. Berikut adalah *code* untuk mengubah citra RGB menjadi *grayscale*.

```
clc;
clear all;
%Load Folder yang Berisi Images (Data)
%Proses RGB to Grayscale dilakukan setiap folder
(NPeritoneal, NPleural, Aperitoneal, Apleural)
folder = cd('D:\College\Semester
7\Data\Peritoneal\NPeritoneal');
images = dir([folder, '*.jpg']);
nimages = length(images);
for i = 1:nimages
    currentimagename=images(i).name;
    imagenames(i,:)={currentimagename};

currentimage=imread(fullfile(folder,currentimagename), 'jpg');

%RGB2GRAYSCALE
I=rgb2gray(currentimage);
End
```

b. Cropping

Cropping adalah tahap untuk mengubah ukuran (*pixel*) citra agar citra memiliki ukuran yang sama sehingga memudahkan dalam pengolahan. Ukuran awal *image body fluid* yang digunakan adalah 3.840 x 2.160 pixel kemudian dipotong menjadi 1.280 x 720 pixel. Berikut adalah *code* untuk *crop* dan *save images*.

```
clc;
clear all;
%Load Folder yang Berisi Images (Data)
%Proses RGB to Grayscale dilakukan setiap folder (NPeritoneal,
Npleural, Aperitoneal, Apleural)
folder = cd('D:\College\Semester
7\Data\Peritoneal\NPeritoneal');
images = dir([folder, '\*.jpg']);
nimages = length(images);
for i = 1:nimages
    currentimagename=images(i).name;
    imagenames(i,:)={currentimagename};
    currentimage=imread(fullfile(folder,currentimagename), 'jpg');

%RGB2GRAYSCALE
I=rgb2gray(currentimage);

%CROPPING IMAGE (3840 X 2160 PIXEL MENJADI 1280 X 720 PIXEL)
%Proses cropping dilakukan di masing-masing folder
(NPeritoneal, Npleural, Aperitoneal, Apleural)
C1=I(1:720,1:1280);
C2=I(1:720,1281:2560);
C3=I(1:720,2561:3840);
C4=I(721:1440,1:1280);
C5=I(721:1440,1281:2560);
C6=I(721:1440,2561:3840);
C7=I(1441:2160,1:1280);
C8=I(1441:2160,1281:2560);
C9=I(1441:2160,2561:3840);

%SAVE IMAGE YANG DICROP KE FOLDER
%'NP1_' adalah penamaan file, disesuaikan dg citra
imwrite(C1, strcat(['NP1_', num2str(i), '.jpg']));
imwrite(C2, strcat(['NP2_', num2str(i), '.jpg']));
imwrite(C3, strcat(['NP3_', num2str(i), '.jpg']));
imwrite(C4, strcat(['NP4_', num2str(i), '.jpg']));
imwrite(C5, strcat(['NP5_', num2str(i), '.jpg']));
imwrite(C6, strcat(['NP6_', num2str(i), '.jpg']));
imwrite(C7, strcat(['NP7_', num2str(i), '.jpg']));
imwrite(C8, strcat(['NP8_', num2str(i), '.jpg']));
imwrite(C9, strcat(['NP9_', num2str(i), '.jpg']));

end
```


c. *Enhancing*

Enhancement adalah proses membuat gambar memiliki kualitas yang lebih baik dan lebih berguna. Tujuan *enhancement* ini antara lain untuk menghilangkan noise dari gambar, menonjolkan suatu ciri dari citra tersebut, menonjolkan fitur spesifik, menyoroti detail gambar, dan sebagainya. Terdapat beberapa jenis *enhancement* yang dapat digunakan, diantaranya *Adjust*, *Sharpen*, *Histogram Equalization* (Histeq), dan *Contrast-Limited Adaptive Histogram Equalization* (CLAHE). Pada penelitian ini, jenis *enhancement* yang digunakan adalah *Adapthisteq* (CLAHE). Sebelum proses *enhancement*, seluruh images yang telah mengalami *cropping* akan melalui proses seleksi *images*, yaitu penghapusan *images* yang tidak menunjukkan bagian sel. Selanjutnya, semua images yang telah lolos seleksi dijadikan satu dalam folder baru untuk proses *enhancement*. Berikut adalah *code* untuk *enhancement Adapthisteq* (CLAHE).

```
%Nama folder awal adalah APeritoneal, NPeritoneal, APleural,
Npleural
%Nama folder yang baru adalah Body Fluids/All

clc;
clear all;
%Load Folder yang Berisi Images (Data)
folder = cd('D:\College\Semester 7\Body Fluids\All');
images = dir([folder, '\*.jpg']);
nimages = length(images);
for i = 1:nimages
    currentimagename=images(i).name;
    imagenames(i,:)={currentimagename};
    currentimage=imread(fullfile(folder,currentimagename), 'jpg');

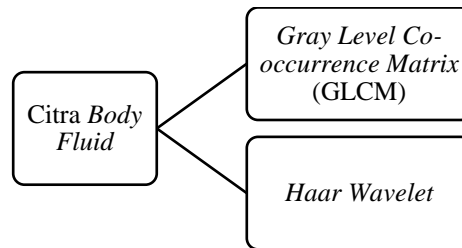
    %Enhancement
    enhance = adapthisteq(currentimage);

    .
    .
    .
```

3.2.2 Ekstraksi Fitur

Ekstraksi fitur pada penelitian ini menggunakan dua metode, yaitu *Gray Level Co-occurrence Matrix* (GLCM) dan *Haar Wavelet*. Tujuan dari proses ekstraksi adalah untuk mendapatkan nilai fitur yang kemudian akan digunakan

untuk menghasilkan fitur ciri. Fitur-fitur ciri dari masing-masing metode akan mempengaruhi tingkat akurasi karena setiap metode akan menghasilkan nilai fitur ciri yang berbeda. Ekstraksi fitur yang digunakan dalam penelitian ini dapat dilihat pada Gambar 3.4.



Gambar 3. 4 Ekstraksi Fitur

Proses ekstraksi yang digunakan dalam penelitian ini adalah sebagai berikut.

- a. Pertama, citra yang telah diubah menjadi *Grayscale* dan telah melalui tahap *cropping* akan diekstraksi dengan dua metode, yaitu *Gray Level Co-occurrence Matrix (GLCM)* dan *Haar Wavelet*. Code untuk proses ekstraksi GLCM dan *Haar Wavelet* dapat dilihat sebagai berikut.

Code Ekstraksi Metode *Gray Level Co-occurrence Matrix (GLCM)*

```

.
.
.

%Gray Level Co-occurrence Matrix
offsets = [0 30; -30 30; -30 0; -30 -30]; %d = distance
[glcm, SI] = graycomatrix (enhance, 'Offset', offsets);
stats =
graycoprops (glcm, {'Contrast', 'Correlation', 'Energy', 'Homogeneity'});

%Menghitung Nilai Contrast, Correlation, Energy, Homogeneity
Contrast = stats.Contrast;
Correlation = stats.Correlation;
Energy = stats.Energy;
Homogeneity = stats.Homogeneity;

%Menghitung Entropy dan IDM Menggunakan file .M
entropy_result = zeros(1,4);
idm_result = zeros(1,4);

for j = 1:4
    glcm_j = glcm(:, :, j);
    entropy_result(j) = entropysearch (glcm_j);
    idm_result(j) = IDMsearch (glcm_j);
end
  
```

```

%Menghitung Rata-Rata Semua Fitur GLCM
mean_Contrast = mean(Contrast);
mean_Correlation = mean(Correlation);
mean_Energy = mean(Energy);
mean_Homogeneity = mean(Homogeneity);
mean_Entropy = mean(entropy_result);
mean_IDM = mean(idm_result);

Z = [Contrast(1) Contrast(2) Contrast(3) Contrast(4)
Correlation(1) Correlation(2) Correlation(3) Correlation(4)
Energy(1) Energy(2) Energy(3) Energy(4) Homogeneity(1)
Homogeneity(2) Homogeneity(3) Homogeneity(4)
entropy_result(1) entropy_result(2) entropy_result(3)
entropy_result(4) idm_result(1) idm_result(2) idm_result(3)
idm_result(4) mean(Contrast) mean(Correlation) mean(Energy)
mean(Homogeneity) mean(entropy_result) mean(idm_result)];
Z1(i,:) = Z;

end

%Export ke Excel
sheet1 = 1;
filename = 'GLCM.xlsx'; %Nama file disesuaikan
xlswrite(filename,Z1,sheet1,'A1:AD9545'); %9545 adlh jumlah
citra
xlswrite(filename,imagenames,sheet1,'AE1:AE9545');

```

Dalam penggunaan *code* di atas, untuk fitur *Entropy* dan *Inverse Difference Moment (IDM)* harus menggunakan *function* dengan format file *.M* yang terletak pada file terpisah dari *code* utama. *Code Function Entropy* dan *IDM* dapat dilihat sebagai berikut.

Code Function Entropy

```

function [entropy] = entropysearch(image)

image = image ./sum(image(:));
[baris, kolom] = size(image);
tampung = zeros (baris,kolom);

for i = 1:baris
    for j = 1:kolom
        temp2 = image(i,j);
        if(temp2==0)
            tampung(i,j) = 0;
        else
            a = -(temp2);
            b = log10(temp2);
            tampung(i,j)=a.*b;
        end
    end
end

entropy = tampung;

```

```
entropy = sum(sum(entropy));
end
```

Code Function IDM

```
function IDM = IDMsearch(image)
%Mendapatkan ukuran citra (jumlah baris dan kolom)
[baris,kolom] = size(image);
%Normalisasi GLCM (membagi dengan jumlah total elemen GLCM)
norm_glcm = image/sum(image(:));
%Inisialisasi IDM
IDM = 0;
%Menghitung IDM
for i = 1:baris
    for j = 1:kolom
        IDM = IDM + (norm_glcm(i,j)/(1+(i-j)^2));
    end
end
```

Code Ekstraksi Metode Haar Wavelet (Level 1)

```
clc;
clear all;
%Load Folder yang Berisi Images (Data)
folder = cd('D:\College\Semester 7\Body Fluids\All');
images = dir([folder,'*.jpg']);
nimages = length(images);
for i = 1:nimages
    currentimagename=images(i).name;
    imagenames(i,:)={currentimagename};

    currentimage=imread(fullfile(folder,currentimagename),'jpg');
end;
%Resize
A=imresize(currentimage,[1280 720]);

%Haar Wavelet (Level 1)
[C,S] = wavedec2(A,2,'haar');
[H1,V1,D1] = detcoef2('all',C,S,1);%ubah angka untuk level
%%Z=[H1,V1,D1]
A1=appcoef2(C,S,'haar',1); %ubah angka untuk level
V1img=wcodemat(V1,255,'mat',1);
H1img=wcodemat(D1,255,'mat',1);
A1img=wcodemat(H1,255,'mat',1);
D1img=wcodemat(A1,255,'mat',1);
%%Mean A1 = mean(A1)
meanH1 = mean(H1);
meanV1 = mean(V1);
meanD1 = mean(D1);
stdH1 = std(H1);
stdV1 = std(V1);
stdD1 = std(D1);
[Ea,Eh,Ev,Ed] = wenergy2(C,S);
Z = [Ea Eh Ev Ed meanH1 meanV1 meanD1 stdH1 stdV1 stdD1];
FeatureExtracted(i,:) = Z;
```

```

end

%Export ke Excel
sheet1 = 1;
filename = 'HAAR_LEVEL1.xlsx';
%Tukar sesuai jumlah data image
xlswrite(filename, FeatureExtracted, sheet1, 'A1:CEV9545');
xlswrite(filename, imagenames, sheet, 'CEW1:CEW9545');

```

b. Selanjutnya, citra yang telah diekstraksi akan diberi label, yaitu *Malignant* dan *Benign*. Label untuk citra kelas *Malignant* adalah [1 0] dan citra kelas *Benign* adalah [0 1]. Gambar 3.5 dan Gambar 3.6 menunjukkan hasil *labelling*.

	Z	AA	AB	AC	AD	AE	AF	AG
1732	0,81506	0,05895	0,65354	1,48732	0,61671	AdePeritoneal (992).jpg	1	0
1733	0,63198	0,0276	0,56201	1,65697	0,51143	AdePeritoneal (993).jpg	1	0
1734	0,73074	0,06242	0,63204	1,4994	0,59021	AdePeritoneal (994).jpg	1	0
1735	0,6676	0,21313	0,72695	1,18537	0,69626	AdePeritoneal (995).jpg	1	0
1736	0,82323	0,09354	0,68187	1,37498	0,64966	AdePeritoneal (996).jpg	1	0
1737	0,81795	0,07273	0,65984	1,44622	0,62439	AdePeritoneal (997).jpg	1	0
1738	0,74671	0,05959	0,62769	1,53077	0,58353	AdePeritoneal (998).jpg	1	0
1739	0,77193	0,05587	0,63171	1,50237	0,5946	AdePeritoneal (999).jpg	1	0
1740	0,64422	0,57561	0,87066	0,64736	0,85542	AdenoPleura (1).jpg	1	0
1741	0,78751	0,89756	0,97183	0,1626	0,96888	AdenoPleura (10).jpg	1	0
1742	0,65114	0,47482	0,82869	0,75487	0,80814	AdenoPleura (100).jpg	1	0
1743	0,55032	0,03338	0,54134	1,65941	0,48308	AdenoPleura (1000).jpg	1	0
1744	0,52317	0,03497	0,53317	1,67475	0,47173	AdenoPleura (1001).jpg	1	0
1745	0,579	0,04717	0,57055	1,60195	0,51799	AdenoPleura (1002).jpg	1	0
1746	0,64623	0,04829	0,59684	1,55425	0,55148	AdenoPleura (1003).jpg	1	0
1747	0,62503	0,03791	0,58489	1,57755	0,53954	AdenoPleura (1004).jpg	1	0
1748	0,68359	0,06266	0,62262	1,48031	0,58288	AdenoPleura (1005).jpg	1	0
1749	0,70389	0,14973	0,69662	1,2301	0,66676	AdenoPleura (1006).jpg	1	0

Gambar 3. 5 Labelling Citra Body Fluids Kelas Malignant

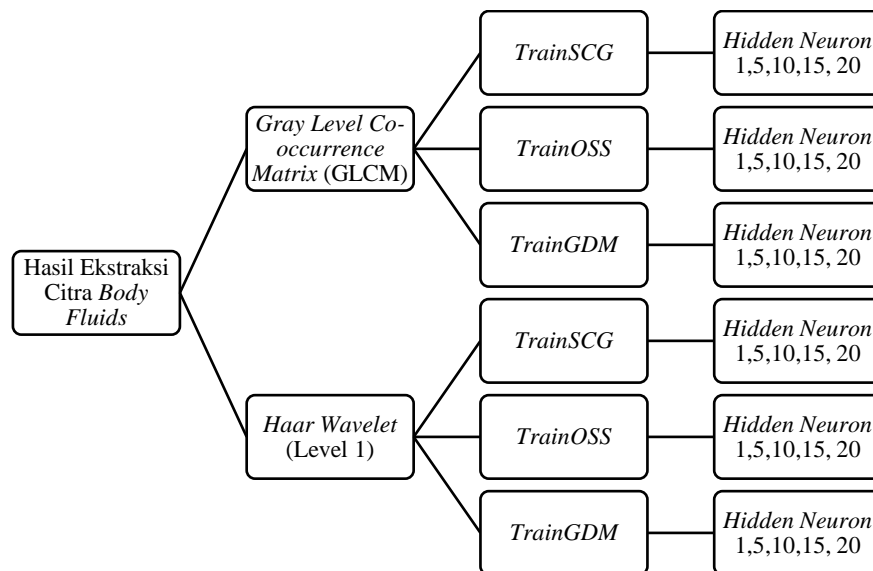
	Z	AA	AB	AC	AD	AE	AF	AG
2092	0,324575	0,244159	0,708111	1,118223	0,670962	NPeritoneal (842).jpg	0	1
2093	0,304046	0,300694	0,749267	1,002787	0,720705	NPeritoneal (850).jpg	0	1
2094	0,359788	0,288181	0,752688	1,012281	0,725302	NPeritoneal (856).jpg	0	1
2095	0,371126	0,081199	0,604406	1,474026	0,555204	NPeritoneal (858).jpg	0	1
2096	0,445189	0,175028	0,681807	1,274593	0,642825	NPeritoneal (86).jpg	0	1
2097	0,303281	0,058962	0,56322	1,507852	0,506388	NPeritoneal (863).jpg	0	1
2098	0,264575	0,033418	0,497664	1,657093	0,429804	NPeritoneal (867).jpg	0	1
2099	0,220956	0,062175	0,554502	1,522621	0,495089	NPeritoneal (999).jpg	0	1
2100	0,234432	0,066304	0,530375	1,496757	0,465069	NPleural_ (1).jpg	0	1
2101	0,512143	0,452231	0,792495	0,806408	0,762385	NPleural_ (10).jpg	0	1
2102	0,589155	0,110007	0,607701	1,453981	0,554455	NPleural_ (100).jpg	0	1
2103	0,511316	0,090027	0,5795	1,502425	0,522629	NPleural_ (1000).jpg	0	1
2104	0,526512	0,155149	0,64124	1,360336	0,592136	NPleural_ (1001).jpg	0	1
2105	0,464017	0,046911	0,534576	1,632285	0,472556	NPleural_ (1002).jpg	0	1
2106	0,459169	0,067444	0,552972	1,571965	0,492134	NPleural_ (1003).jpg	0	1
2107	0,659449	0,250666	0,757371	1,045471	0,732651	NPleural_ (1004).jpg	0	1
2108	0,586861	0,141482	0,701836	1,248679	0,667984	NPleural_ (1005).jpg	0	1
2109	0,727597	0,455982	0,823339	0,792943	0,80393	NPleural_ (1006).jpg	0	1
2110	0,57173	0,138712	0,629913	1,398808	0,580237	NPleural_ (1007).jpg	0	1

Gambar 3. 6 Labelling Citra Body Fluids Kelas Benign

3.2.3 Klasifikasi

Proses klasifikasi adalah proses yang dilakukan setelah citra diekstraksi menjadi nilai-nilai kuantitatif yang mengandung informasi. Pada penelitian

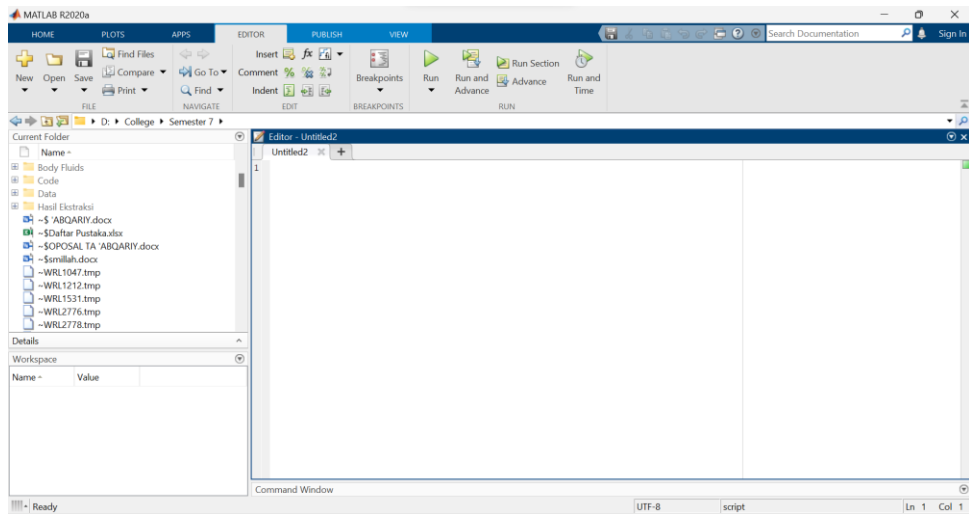
ini, metode klasifikasi yang digunakan adalah *Multilayer Perceptron* (MLP) dengan model *Scaled Conjugate Gradient Backpropagation* (SCG), *One Step Secant Backpropagation* (OSS), dan *Gradient Descent with Momentum* (GDM). Masing-masing model akan diuji dengan jumlah *hidden neuron* 1, 5, 10, 15, dan 20. Kemudian masing-masing *hidden neuron* akan diuji sebanyak 10 kali atau disebut dengan *10 run cross validation*. Perbandingan jumlah citra yang digunakan untuk *training*, *testing*, dan *validation* adalah 90%, 5%, dan 5% secara berurutan. Model yang digunakan pada proses klasifikasi dapat dilihat pada Gambar 3.7.



Gambar 3. 7 Model Klasifikasi

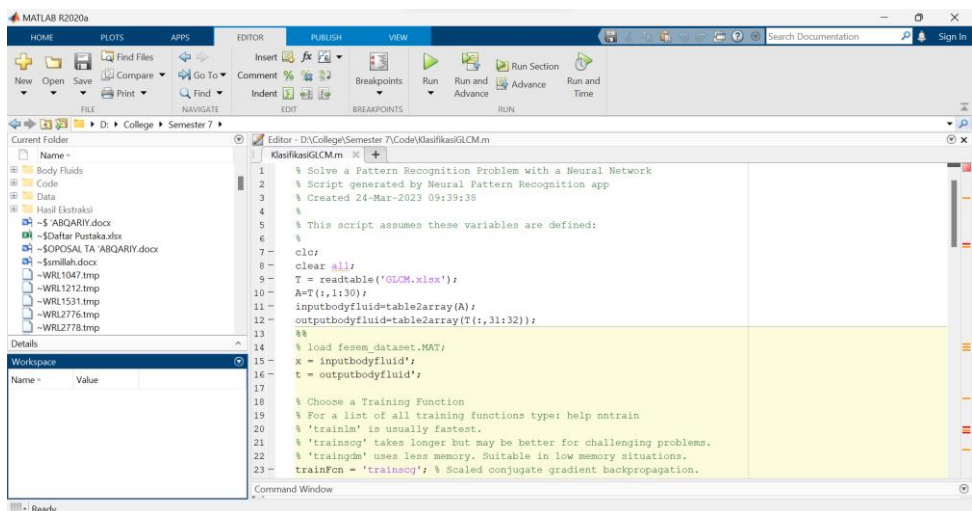
Hasil dari proses klasifikasi tersebut merupakan hasil akhir untuk mengetahui tingkat akurasi dari model yang digunakan. *Output* dari proses klasifikasi akan menunjukkan apakah citra *body fluids* tersebut *Malignant* atau *Benign*. Tahapan dalam proses klasifikasi tersebut adalah sebagai berikut.

1. Buka Matlab R2020a.



Gambar 3. 8 Tampilan Matlab R2020a

2. Pada menu editor, tulis *code* yang akan digunakan untuk klasifikasi.



Gambar 3. 9 Tampilan Editor Untuk Coding Klasifikasi

Code untuk klasifikasi GLCM dan Haar Wavelet

```
% Solve a Pattern Recognition Problem with a Neural Network
% Script generated by Neural Pattern Recognition app
% Created 24-Mar-2023 09:39:38
% This script assumes these variables are defined:
%
clc;
clear all;
T = readtable('GLCM.xlsx'); % disesuaikan nama file
A=T(:,1:30); % disesuaikan jumlah fitur
inputbodyfluid=table2array(A);
outputbodyfluid=table2array(T(:,31:32));
%%
```

```

% load fesem_dataset.MAT;
x = inputbodyfluid';
t = outputbodyfluid';

% Choose a Training Function
% For a list of all training functions type: help ntrain
% 'trainlm' is usually fastest.
% 'trainscg' takes longer but may be better for challenging
problems.
% 'traingdm' uses less memory. Suitable in low memory
situations.
trainFcn = 'trainscg'; % Scaled conjugate gradient
backpropagation.
% trainbfg | traincgb | traincgf | traincgp | traingda |
traingdm | % traingdx | trainlm | trainoss | trainrp
% Create a Pattern Recognition Network
hiddenNeuronSize = 1; % 1 5 10 15 20
net = patternnet(hiddenNeuronSize, trainFcn);
% Choose Input and Output Pre/Post-Processing Functions
% For a list of all processing functions type: help
nnprocess
net.input.processFcns = {'removeconstantrows', 'mapminmax'};
% Setup Division of Data for Training, Validation, Testing
% For a list of all data division functions type: help
nndivision
net.divideFcn = 'dividerand'; % Divide data randomly
net.divideMode = 'sample'; % Divide up every sample
net.divideParam.trainRatio = 90/100;
net.divideParam.valRatio = 5/100;
net.divideParam.testRatio = 5/100;
% Choose a Performance Function
% For a list of all performance functions type: help
nnperformance
net.performFcn = 'crossentropy'; % Cross-Entropy
% Choose Plot Functions
% For a list of all plot functions type: help nnplot
net.plotFcns =
{'plotperform', 'plottrainstate', 'ploterrhist', ...
'plotconfusion', 'plotroc'};
% Train the Network
[net,tr] = train(net,x,t);
% Test the Network
y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y);
tind = vec2ind(t);
yind = vec2ind(y);
percentErrors = sum(tind ~= yind)/numel(tind);
% Recalculate Training, Validation and Test Performance

```

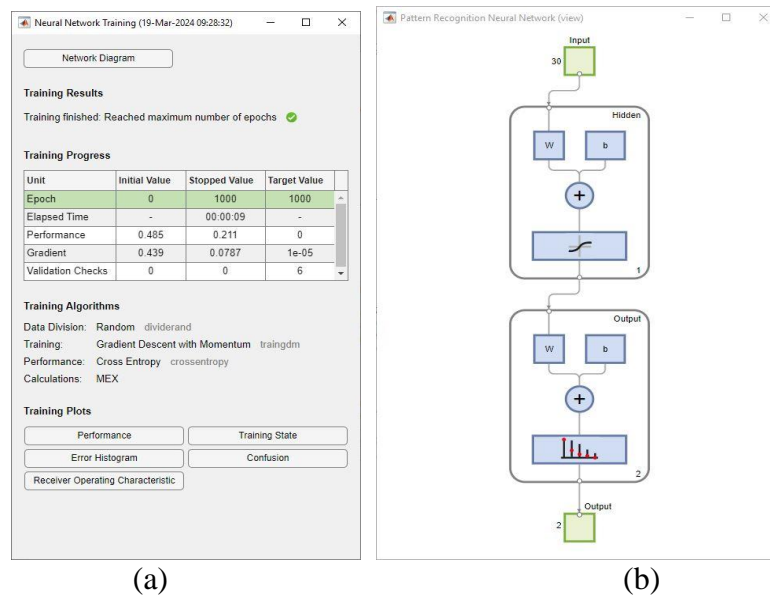


```

trainTargets = t .* tr.trainMask{1};
valTargets = t .* tr.valMask{1};
testTargets = t .* tr.testMask{1};
trainPerformance = perform(net,trainTargets,y)
valPerformance = perform(net,valTargets,y)
testPerformance = perform(net,testTargets,y)
% View the Network
view(net)
% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, ploterrhist(e)
%figure, plotconfusion(t,y)
%figure, plotroc(t,y)
% Deployment
% Change the (false) values to (true) to enable the
following code blocks.
% See the help for each generation function for more
information.
if (false)
    % Generate MATLAB function for neural network for
application
    % deployment in MATLAB scripts or with MATLAB Compiler and
Builder
    % tools, or simply to examine the calculations your
trained neural
    % network performs.
    genFunction(net,'myNeuralNetworkFunction');
    y = myNeuralNetworkFunction(x);
end
if (false)
    % Generate a matrix-only MATLAB function for neural network
code
    % generation with MATLAB Coder tools.
    genFunction(net,'myNeuralNetworkFunction','Matrix
Only',
'yes');
    y = myNeuralNetworkFunction(x);
end
if (false)
    % Generate a Simulink diagram for simulation or deployment
with.
    % Simulink CCoder tools.
    gensim(net);
end

```

3. Tekan tombol *run* untuk proses training.



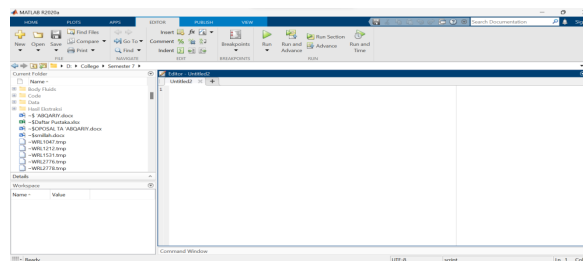
Gambar 3. 10 (a) *Neural Network Training*, (b) *Pattern Recognition Neural Network*

3.3 Perancangan APPS

APPS (*App Designer*) merupakan aplikasi *display* dari Matlab yang mengandung perintah serta bertujuan untuk memudahkan *user* dalam menjalankan program di Matlab. Pada APPS terdapat beberapa *control* yang dapat digunakan untuk memberikan perintah, seperti tombol atau *button*, *textbox*, menu, slider, dan sebagainya. Pada perancangan APPS ini, terdapat dua rujukan (Anggraini, M., 2024) dan (Kurniawati, R., 2024). Tahapan dalam pembuatan APPS Matlab ini adalah sebagai berikut.

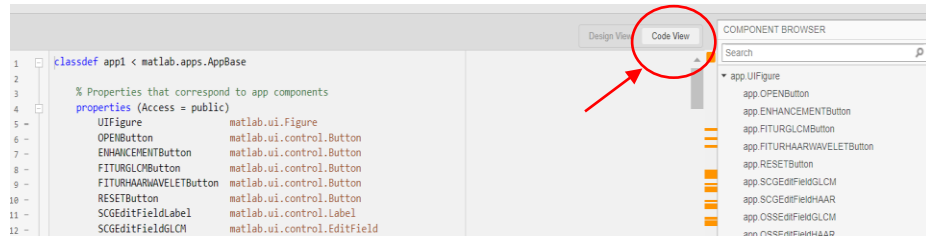
3.3.1 Desain APPS

1. Membuka aplikasi Matlab (Matlab R2020a)



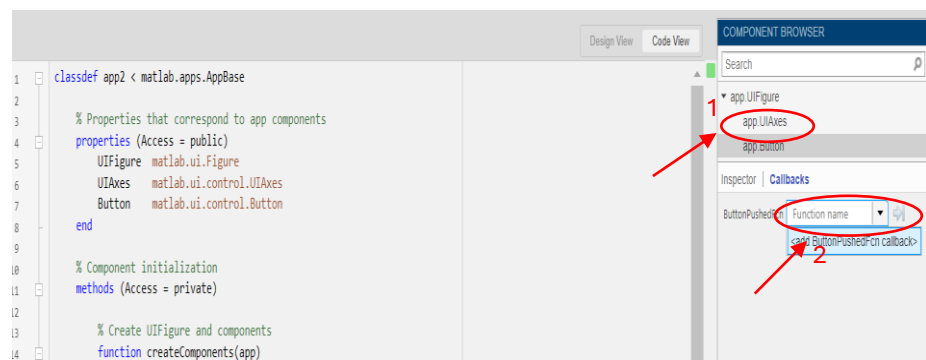
Gambar 3. 11 Tampilan Awal Matlab

- Klik pada *code view* maka akan muncul *coding* dari komponen yang dipilih



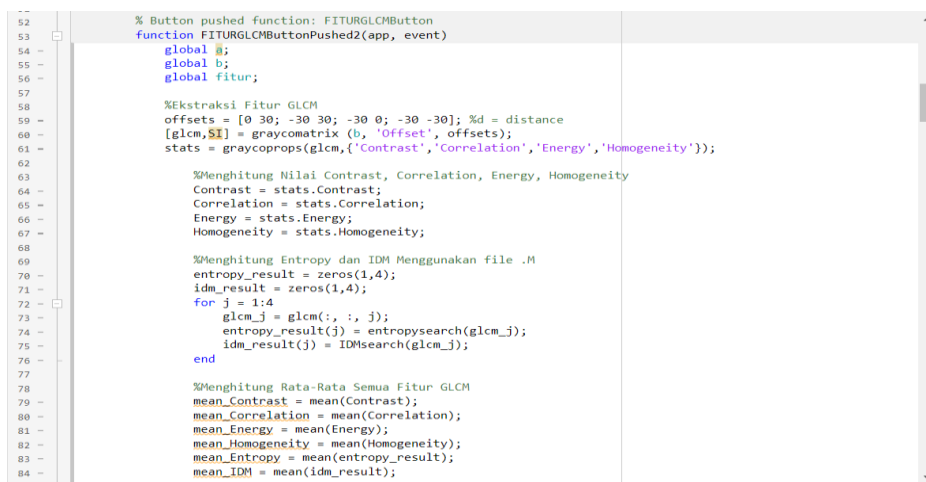
Gambar 3. 16 Tampilan Code View

- Klik *app.button* dan pilih *callbacks* untuk memunculkan *button function*



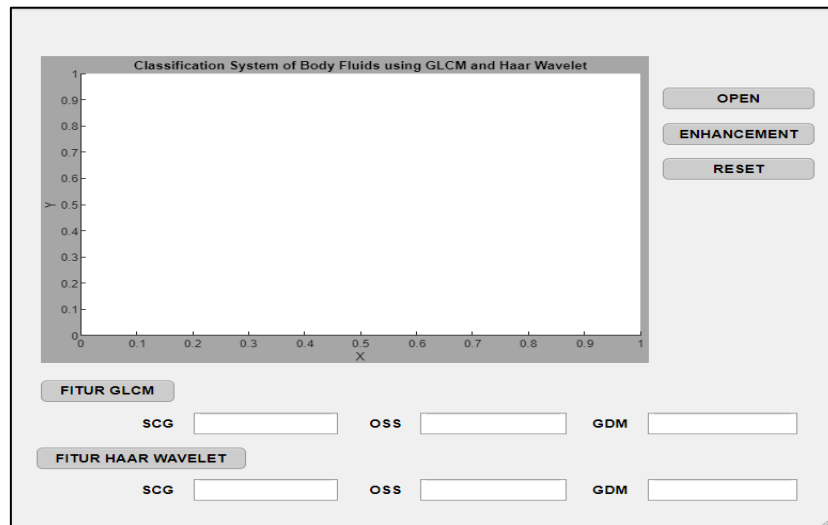
Gambar 3. 17 Tampilan Button Function

- Memasukkan code ekstraksi fitur GLCM dan Haar Wavelet serta otak hasil ekstraksi fitur GLCM dan Haar Wavelet yang telah dilatih dengan tiga model, yaitu SCG, OSS, dan GDM



Gambar 3. 18 Tampilan Code APPS

9. APPS siap digunakan



Gambar 3. 19 Tampilan Akhir APPS

3.3.2 Code APPS

Code dalam Pembuatan APPS

```
classdef app1 < matlab.apps.AppBase

% Properties that correspond to app components
properties (Access = public)
    UIFigure matlab.ui.Figure
    OPENButton matlab.ui.control.Button
    ENHANCEMENTButton matlab.ui.control.Button
    FITURGLCMButton matlab.ui.control.Button
    FITURHAARWAVELETButton matlab.ui.control.Button
    RESETButton matlab.ui.control.Button
    SCGEditFieldLabel matlab.ui.control.Label
    SCGEditFieldGLCM matlab.ui.control.EditField
    SCGEditField_2Label matlab.ui.control.Label
    SCGEditFieldHAAR matlab.ui.control.EditField
    OSSEditFieldLabel matlab.ui.control.Label
    OSSEditFieldGLCM matlab.ui.control.EditField
    OSSEditField_2Label matlab.ui.control.Label
    OSSEditFieldHAAR matlab.ui.control.EditField
    GDMEditFieldLabel matlab.ui.control.Label
    GDMEditFieldGLCM matlab.ui.control.EditField
    GDMEditField_2Label matlab.ui.control.Label
    GDMEditFieldHAAR matlab.ui.control.EditField
    UIAxes matlab.ui.control.UIAxes
end

% Callbacks that handle component events
```

```

methods (Access = private)

% Callback function
function FITURGLCMButtonPushed(app, event)
end

% Button pushed function: OPENButton
function OPENButtonPushed(app, event)
global a;
[filename, pathname] = uigetfile('*..*', 'Pick a MATLAB code file');
filename=strcat(pathname, filename);
a=imread(filename);
imshow(a, 'Parent', app.UIAxes);
end

% Button pushed function: ENHANCEMENTButton
function ENHANCEMENTButtonPushed(app, event)
global a;
global b;
A=a;
b = adapthisteq(a);
imshow(b, 'Parent', app.UIAxes);
end

% Button pushed function: FITURGLCMButton
function FITURGLCMButtonPushed2(app, event)
global a;
global b;
global fitur;
%Ekstraksi Fitur GLCM
offsets = [0 30; -30 30; -30 0; -30 -30]; %d = distance
[glcm,SI] = graycomatrix (b, 'Offset', offsets);
stats =
graycoprops(glcm,{'Contrast', 'Correlation', 'Energy', 'Homogeneity'});

%Menghitung Nilai Contrast, Correlation, Energy, Homogeneity
Contrast = stats.Contrast;
Correlation = stats.Correlation;
Energy = stats.Energy;
Homogeneity = stats.Homogeneity;
%Menghitung Entropy dan IDM Menggunakan file .M
entropy_result = zeros(1,4);
idm_result = zeros(1,4);
for j = 1:4

```

```

glcm_j = glcm(:, :, j);
entropy_result(j) = entropysearch(glcm_j);
idm_result(j) = IDMsearch(glcm_j);
end
%Menghitung Rata-Rata Semua Fitur GLCM
mean_Contrast = mean(Contrast);
mean_Correlation = mean(Correlation);
mean_Energy = mean(Energy);
mean_Homogeneity = mean(Homogeneity);
mean_Entropy = mean(entropy_result);
mean_IDM = mean(idm_result);
fitur = [Contrast(1) Contrast(2) Contrast(3) Contrast(4) Correlation(1)
Correlation(2) Correlation(3) Correlation(4) Energy(1) Energy(2)
Energy(3) Energy(4) Homogeneity(1) Homogeneity(2) Homogeneity(3)
Homogeneity(4) entropy_result(1) entropy_result(2) entropy_result(3)
entropy_result(4) idm_result(1) idm_result(2) idm_result(3) idm_result(4)
mean(Contrast) mean(Correlation) mean(Energy) mean(Homogeneity)
mean(entropy_result) mean(idm_result)];
%Memanggil Otak SCG
load OTAKGLCMSCG20.mat %GANTI OTAK
%Klasifikasi Citra
output1=round (sim(net,fitur'));
if output1==[1 0]';
output1 = 'Adenocarcinoma';
else output1==[0 1]';
output1 = 'Benign';
end
I1 = string(output1);
str1 = I1;
app.SCGEditFieldGLCM.Value = str1;
%Memanggil Otak OSS
load OTAKGLCMOSS20.mat %GANTI OTAK
%Klasifikasi Citra
output2=round (sim(net,fitur'));
if output2==[1 0]';
output2 = 'Adenocarcinoma';
else output2==[0 1]';
output2 = 'Benign';
end
I2 = string(output2);
str2 = I2;
app.OSSEditFieldGLCM.Value = str2;
%Memanggil Otak GDM
load OTAKGLCMGDM20.mat %GANTI OTAK
%Klasifikasi Citra
output3=round (sim(net,fitur'));
if output3==[1 0]';
output3 = 'Adenocarcinoma';
else output3==[0 1]';

```

```

output3 = 'Benign';
end
I3 = string(output3);
str3 = I3;
app.GMEditFieldGLCM.Value = str3;
end

% Button pushed function: FITURHAARWAVELETButton
function FITURHAARWAVELETButtonPushed(app, event)
global a;
global b;
global haar;
% Cropping or Resize
crop=imresize(b,[1280 720]);
% Ekstraksi Fitur Haar Wavelet
[C,S] = wavedec2(crop,2,'haar');
[H1,V1,D1] = detcoef2('all',C,S,1);%ubah angka untuk level
%%Z=[H1,V1,D1]
A1=appcoef2(C,S,'haar',1); %ubah angka untuk level
V1img=wcodemat(V1,255,'mat',1);
H1img=wcodemat(D1,255,'mat',1);
A1img=wcodemat(H1,255,'mat',1);
D1img=wcodemat(A1,255,'mat',1);
%%Mean A1 = mean(A1)
meanH1 = mean(H1);
meanV1 = mean(V1);
meanD1 = mean(D1);
stdH1 = std(H1);
stdV1 = std(V1);
stdD1 = std(D1);
[Ea,Eh,Ev,Ed] = wenergy2(C,S);
haar = [Ea Eh Ev Ed meanH1 meanV1 meanD1 stdH1 stdV1 stdD1];
% Memanggil Otak SCG
load OTAKHAARSCG20.mat %GANTI OTAK
% Klasifikasi Citra
output4=round (sim(net,haar'));
if output4==[1 0]';
output4 = 'Adenocarcinoma';
else output4==[0 1]';
output4 = 'Benign';
end
I4 = string(output4);
str4 = I4;
app.SCGEditFieldHAAR.Value = str4;
% Memanggil Otak OSS
load OTAKHAAROSS15.mat %GANTI OTAK
% Klasifikasi Citra
output5=round (sim(net,haar'));

```



```

if output5==[1 0]';
output5 = 'Adenocarcinoma';
else output5==[0 1]';
output5 = 'Benign';
end
I5 = string(output5);
str5 = I5;
app.OSSEditFieldHAAR.Value = str5;
%Memanggil Otak GDM
load OTAKHAARGDM1.mat %GANTI OTAK
%Klasifikasi Citra
output6=round (sim(net,haar'));
if output6==[1 0]';
output6 = 'Adenocarcinoma';
else output6==[0 1]';
output6 = 'Benign';
end
I6 = string(output6);
str6 = I6;
app.GMEditFieldHAAR.Value = str6;
end

% Button pushed function: RESETButton
function RESETButtonPushed(app, event)
app.OPENButton;
app.ENHANCEMENTButton;
app.RESETButton;
app.FITURGLCMButton;
app.FITURHAARWAVELETButton;
app.SCGEditFieldGLCM.Value=" ";
app.OSSEditFieldGLCM.Value=" ";
app.GMEditFieldGLCM.Value=" ";
app.SCGEditFieldHAAR.Value=" ";
app.OSSEditFieldHAAR.Value=" ";
app.GMEditFieldHAAR.Value=" ";
app.UIAxes;
app.UIFigure;
end

% Value changed function: SCGEditFieldGLCM
function SCGEditFieldGLCMValueChanged(app, event)
value = app.SCGEditFieldGLCM.Value;
end

% Value changed function: SCGEditFieldHAAR
function SCGEditFieldHAARValueChanged(app, event)

```

```

value = app.SCGEditFieldHAAR.Value;
end
end

% Component initialization
methods (Access = private)

% Create UIFigure and components
function createComponents(app)

% Create UIFigure and hide until all components are created
app.UIFigure = uifigure('Visible', 'off');
app.UIFigure.Position = [100 100 650 523];
app.UIFigure.Name = 'MATLAB App';

% Create OPENButton
app.OPENButton = uibutton(app.UIFigure, 'push');
app.OPENButton.ButtonPushedFcn = createCallbackFcn(app,
@OPENButtonPushed, true);
app.OPENButton.BackgroundColor = [0.8 0.8 0.8];
app.OPENButton.FontWeight = 'bold';
app.OPENButton.Position = [511.5 424 119 22];
app.OPENButton.Text = 'OPEN';

% Create ENHANCEMENTButton
app.ENHANCEMENTButton = uibutton(app.UIFigure, 'push');
app.ENHANCEMENTButton.ButtonPushedFcn = createCallbackFcn(app,
@ENHANCEMENTButtonPushed, true);
app.ENHANCEMENTButton.BackgroundColor = [0.8 0.8 0.8];
app.ENHANCEMENTButton.FontWeight = 'bold';
app.ENHANCEMENTButton.Position = [511.5 388 119 22];
app.ENHANCEMENTButton.Text = 'ENHANCEMENT';

% Create FITURGLCMBButton
app.FITURGLCMBButton = uibutton(app.UIFigure, 'push');
app.FITURGLCMBButton.ButtonPushedFcn = createCallbackFcn(app,
@FITURGLCMBButtonPushed2, true);
app.FITURGLCMBButton.BackgroundColor = [0.8 0.8 0.8];
app.FITURGLCMBButton.FontWeight = 'bold';
app.FITURGLCMBButton.Position = [24 129 105 22];
app.FITURGLCMBButton.Text = 'FITUR GLCM';

% Create FITURHAARWAVELETButton

```

```

app.FITURHAARWAVELEETButton = uibutton(app.UIFigure, 'push');
app.FITURHAARWAVELEETButton.ButtonPushedFcn = createCallbackFcn(app,
@FITURHAARWAVELEETButtonPushed, true);
app.FITURHAARWAVELEETButton.BackgroundColor = [0.8 0.8 0.8];
app.FITURHAARWAVELEETButton.FontWeight = 'bold';
app.FITURHAARWAVELEETButton.Position = [20.5 61 164 22];
app.FITURHAARWAVELEETButton.Text = 'FITUR HAAR WAVELET';

% Create RESETButton
app.RESETButton = uibutton(app.UIFigure, 'push');
app.RESETButton.ButtonPushedFcn = createCallbackFcn(app,
@RESETButtonPushed, true);
app.RESETButton.BackgroundColor = [0.8 0.8 0.8];
app.RESETButton.FontWeight = 'bold';
app.RESETButton.Position = [511.5 353 119 22];
app.RESETButton.Text = 'RESET';

% Create SCGEditFieldLabel
app.SCGEditFieldLabel = uilabel(app.UIFigure);
app.SCGEditFieldLabel.HorizontalAlignment = 'right';
app.SCGEditFieldLabel.FontWeight = 'bold';
app.SCGEditFieldLabel.Position = [97 96 32 22];
app.SCGEditFieldLabel.Text = 'SCG';

% Create SCGEditFieldGLCM
app.SCGEditFieldGLCM = uieditfield(app.UIFigure, 'text');
app.SCGEditFieldGLCM.ValueChangedFcn = createCallbackFcn(app,
@SCGEditFieldGLCMValueChanged, true);
app.SCGEditFieldGLCM.FontWeight = 'bold';
app.SCGEditFieldGLCM.Position = [144 96 113 22];

% Create SCGEditField_2Label
app.SCGEditField_2Label = uilabel(app.UIFigure);
app.SCGEditField_2Label.HorizontalAlignment = 'right';
app.SCGEditField_2Label.FontWeight = 'bold';
app.SCGEditField_2Label.Position = [97 29 32 22];
app.SCGEditField_2Label.Text = 'SCG';

% Create SCGEditFieldHAAR
app.SCGEditFieldHAAR = uieditfield(app.UIFigure, 'text');
app.SCGEditFieldHAAR.ValueChangedFcn = createCallbackFcn(app,
@SCGEditFieldHAARValueChanged, true);
app.SCGEditFieldHAAR.FontWeight = 'bold';
app.SCGEditFieldHAAR.Position = [144 29 113 22];

```

```

% Create OSSEditFieldLabel
app.OSSEditFieldLabel = uilabel(app.UIFigure);
app.OSSEditFieldLabel.HorizontalAlignment = 'right';
app.OSSEditFieldLabel.FontWeight = 'bold';
app.OSSEditFieldLabel.Position = [276 96 31 22];
app.OSSEditFieldLabel.Text = 'OSS';

% Create OSSEditFieldGLCM
app.OSSEditFieldGLCM = uieditfield(app.UIFigure, 'text');
app.OSSEditFieldGLCM.FontWeight = 'bold';
app.OSSEditFieldGLCM.Position = [322 96 114 22];

% Create OSSEditField_2Label
app.OSSEditField_2Label = uilabel(app.UIFigure);
app.OSSEditField_2Label.HorizontalAlignment = 'right';
app.OSSEditField_2Label.FontWeight = 'bold';
app.OSSEditField_2Label.Position = [276 29 31 22];
app.OSSEditField_2Label.Text = 'OSS';

% Create OSSEditFieldHAAR
app.OSSEditFieldHAAR = uieditfield(app.UIFigure, 'text');
app.OSSEditFieldHAAR.FontWeight = 'bold';
app.OSSEditFieldHAAR.Position = [322 29 114 22];

% Create GDMEditFieldLabel
app.GDMEditFieldLabel = uilabel(app.UIFigure);
app.GDMEditFieldLabel.HorizontalAlignment = 'right';
app.GDMEditFieldLabel.FontWeight = 'bold';
app.GDMEditFieldLabel.Position = [451 96 34 22];
app.GDMEditFieldLabel.Text = 'GDM';

% Create GDMEditFieldGLCM
app.GDMEditFieldGLCM = uieditfield(app.UIFigure, 'text');
app.GDMEditFieldGLCM.FontWeight = 'bold';
app.GDMEditFieldGLCM.Position = [500 96 117 22];

% Create GDMEditField_2Label
app.GDMEditField_2Label = uilabel(app.UIFigure);
app.GDMEditField_2Label.HorizontalAlignment = 'right';
app.GDMEditField_2Label.FontWeight = 'bold';
app.GDMEditField_2Label.Position = [451 29 34 22];
app.GDMEditField_2Label.Text = 'GDM';

```

```

% Create GDMeditFieldHAAR
app.GDMeditFieldHAAR = uieditfield(app.UIFigure, 'text');
app.GDMeditFieldHAAR.FontWeight = 'bold';
app.GDMeditFieldHAAR.Position = [500 29 117 22];

% Create UIAxes
app.UIAxes = uiaxes(app.UIFigure);
title(app.UIAxes, 'Classification System of Body Fluids using
GLCM and Haar Wavelet')
xlabel(app.UIAxes, 'X')
ylabel(app.UIAxes, 'Y')
app.UIAxes.PlotBoxAspectRatio = [1.65743073047859 1 1];
app.UIAxes.FontSize = 10;
app.UIAxes.XTick = [0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1];
app.UIAxes.YTick = [0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1];
app.UIAxes.BackgroundColor = [0.651 0.651 0.651];
app.UIAxes.Position = [24 168 477 310];

% Show the figure after all components are created
app.UIFigure.Visible = 'on';
end
end

% App creation and deletion
methods (Access = public)

% Construct app
function app = app1

% Create UIFigure and components
createComponents(app)

% Register the app with App Designer
registerApp(app, app.UIFigure)

if nargin == 0
    clear app
end
end
end

```

```
% Code that executes before app deletion
function delete(app)

    % Delete UIFigure when app is deleted
    delete(app.UIFigure)
end
end
end
```

3.4 Instrumen Penelitian

Dalam mengolah data dan menjalankan program pada sistem klasifikasi kanker *Adenocarcinoma* melalui citra *body fluids* dibutuhkan perangkat keras (*hardware*) dan lunak (*software*) dengan spesifikasi tertentu. Rincian instrumen yang digunakan dalam penelitian ini adalah sebagai berikut.

- 1. Sistem Operasi Komputer : 64-bit operating system
- 2. Processor : Intel® Core™ i5-9400 CPU @2.90GHz
- 3. RAM : 16.0 GB
- 4. Aplikasi *Software* : MATLAB R2020a

BAB IV HASIL DAN ANALISIS

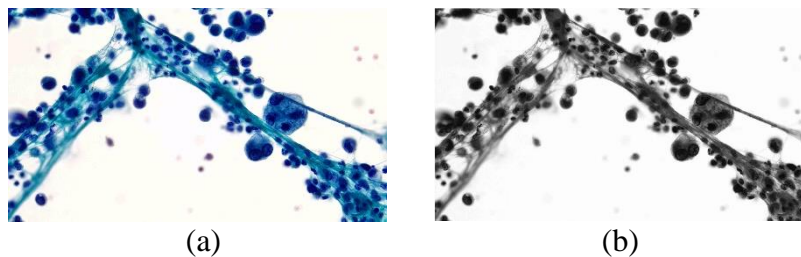
Pada Bab IV membahas tentang hasil *image processing* dari citra *body fluids*. Citra *body fluids* awal yang dibedakan dalam empat kelompok, yaitu *Adenocarcinoma Peritoneal*, *Adenocarcinoma Pleural*, *Normal Peritoneal*, dan *Normal Pleural* digunakan sebagai input dalam proses klasifikasi. Output dari klasifikasi dibagi menjadi dua kelas yaitu *Malignant* dan *Benign*. Seluruh citra akan melalui proses *preprocessing* sebelum masuk tahap ekstraksi dan klasifikasi.

4.1 *Preprocessing*

Image processing dimulai dari tahap *preprocessing*. Pada tahap ini, citra akan melalui beberapa tahapan lagi, seperti perubahan format citra dari RGB menjadi *Grayscale*, *cropping*, pemilihan gambar, dan perbaikan kualitas gambar. Tahapan *preprocessing* tersebut adalah sebagai berikut.

4.1.1 *RGB to Grayscale*

Proses yang pertama adalah mengubah gambar dari citra RGB menjadi *Grayscale* dimana tujuan dari proses adalah untuk menyederhanakan citra sehingga durasi pemrosesan lebih singkat. Salah satu hasil proses *RGB to Grayscale* dapat dilihat pada Gambar 4.1.



Gambar 4. 1 (a) Citra *Body Fluids* RGB, (b) Citra *Body Fluids* Grayscale

4.1.2 *Cropping*

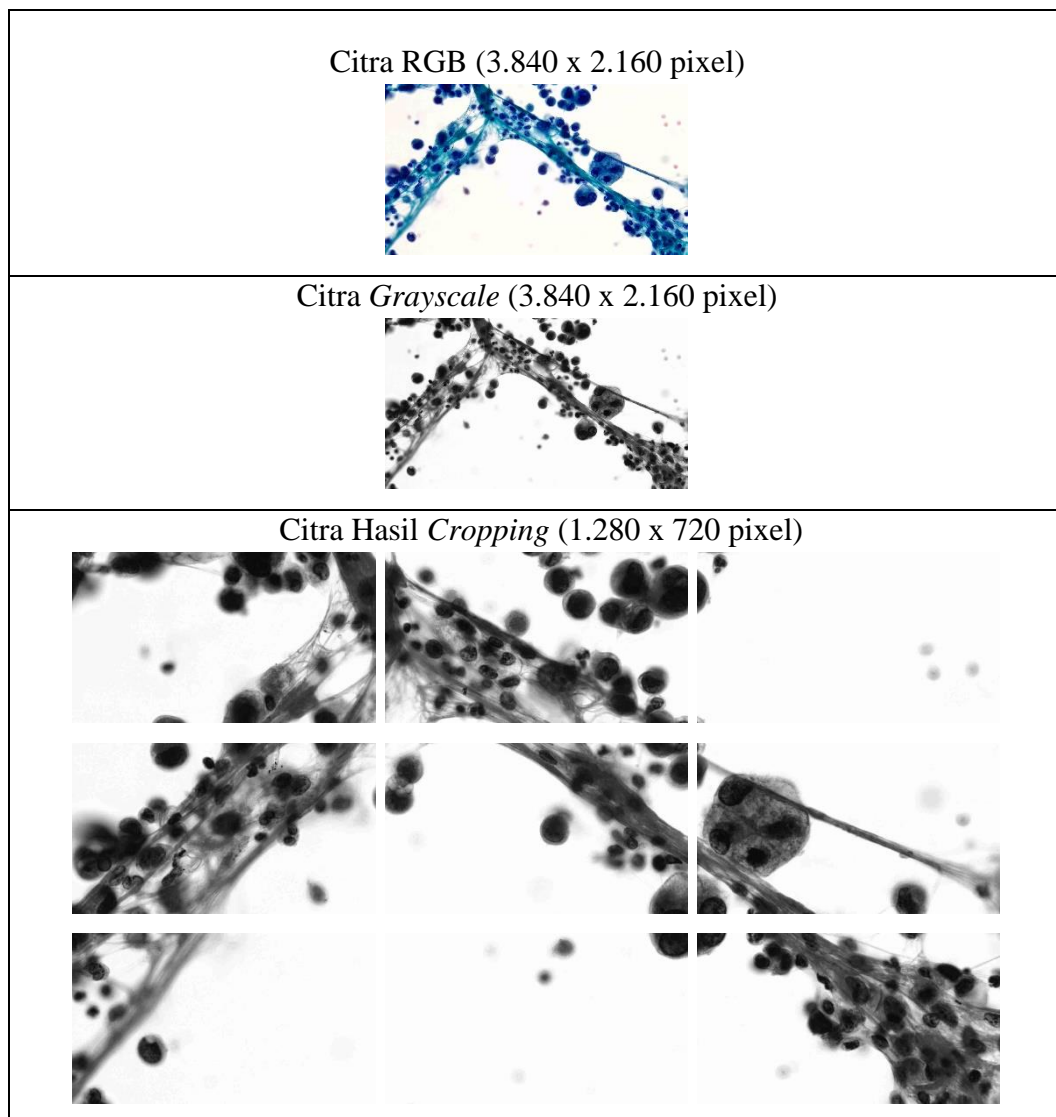
Setelah seluruh citra RGB menjadi citra *Grayscale*, masing-masing citra akan dipotong menjadi sembilan bagian. Citra yang awalnya berukuran 3.840 x 2.160 pixel kemudian dipotong menjadi 1.280 x 720 pixel. Jumlah citra awal *body fluids* adalah 1.273 citra. Setelah melalui *cropping*, jumlah citra menjadi 9.545 citra. Tabel 4.1 merupakan perbandingan jumlah citra sebelum dan sesudah proses

cropping dan Tabel 4.2 menunjukkan perbandingan ukuran citra sebelum dan sesudah *cropping*.

Tabel 4. 1 Perbandingan Jumlah Citra Sebelum dan Sesudah *Cropping*

Nama Citra	Jumlah Sebelum Cropping	Jumlah Sesudah Cropping
Adenocarcinoma Peritoneal	303	1.739
Adenocarcinoma Pleural	339	2.895
Normal Peritoneal	263	2.099
Normal Pleural	368	2.812
Total	1.273	9.545

Tabel 4. 2 Perbandingan Ukuran Citra Sebelum Cropping dan Sesudah Cropping



4.1.3 Enhancing

Tahap *enhancing* adalah tahap setelah citra diubah menjadi *Grayscale* dan melalui proses *cropping*. Jenis *enhancement* yang digunakan adalah *Imadjust*, *Sharpen*, *Histeq*, dan *Adapthisteq*. *Enhancement* tersebut diuji coba pada empat jenis citra *body fluids*. Pada penelitian ini, jenis *enhancement* yang digunakan adalah *Adapthisteq*. Tabel 4.3 menunjukkan hasil perbandingan empat jenis *enhancement* pada citra *body fluids*.

Tabel 4. 3 Perbandingan Empat Jenis *Enhancement* pada Citra *Body Fluids*

Jenis Citra	Jenis <i>Enhancement</i>				
	<i>Grayscale</i>	<i>Adapthisteq</i>	<i>Sharpen</i>	<i>Histeq</i>	<i>Imadjust</i>
<i>Adenocarcinoma Peritoneal (Malignant)</i>					
<i>Adenocarcinoma Pleural (Malignant)</i>					
<i>Normal Peritoneal (Benign)</i>					
<i>Normal Pleural (Benign)</i>					

4.2 Hasil Ekstraksi

Metode ekstraksi yang digunakan dalam penelitian ini adalah *Gray Level Co-occurrence Matrix (GLCM)* dan *Haar Wavelet*. Hasil dari ekstraksi tersebut akan digunakan dalam proses klasifikasi. Berikut adalah hasil proses ekstraksi dengan metode *GLCM* dan *Haar Wavelet*.

4.2.1 Ekstraksi Fitur dengan Metode *Gray Level Co-occurrence Matrix (GLCM)*

Proses ekstraksi fitur yang pertama menggunakan metode *Gray Level Co-occurrence Matrix (GLCM)*. Terdapat enam fitur yang digunakan dalam penelitian ini, yaitu *Contrast*, *Correlation*, *Energy*, *Homogeneity*, *Entropy*, dan *Inverse Difference Moment (IDM)* serta *mean* dari masing-masing fitur sehingga

jumlah fitur adalah 30. Citra yang digunakan dalam proses ekstraksi ini adalah citra *body fluids* dari dua kelas, yaitu *Malignant* dan *Benign*. Tabel 4.4 menunjukkan hasil dari proses ekstraksi fitur dengan menggunakan metode *Gray Level Co-occurrence Matrix* (GLCM).

Tabel 4. 4 Hasil Ekstraksi *Gray Level Co-occurrence Matrix* (GLCM)

Gray Level Co-occurrence Matrix (GLCM)		
Fitur	Kelas 1 (<i>Malignant</i>)	Kelas 2 (<i>Benign</i>)
	Average ± Standar Deviasi	Average ± Standar Deviasi
1	2,021 ± 1,238	3,656 ± 1,939
2	2,913 ± 1,656	4,947 ± 2,511
3	2,000 ± 1,223	3,788 ± 2,019
4	2,925 ± 1,680	4,993 ± 2,539
5	0,752 ± 0,120	0,505 ± 0,118
6	0,639 ± 0,156	0,327 ± 0,140
7	0,754 ± 0,121	0,484 ± 0,143
8	0,638 ± 0,158	0,323 ± 0,141
9	0,284 ± 0,255	0,229 ± 0,230
10	0,271 ± 0,254	0,215 ± 0,227
⋮	⋮	⋮
30	0,689 ± 0,148	0,615 ± 0,158

Tabel 4.4 berisi nilai rata-rata dan standar deviasi dari hasil ekstraksi 30 fitur *Gray Level Co-occurrence Matrix* (GLCM). Nilai rata-rata digunakan untuk membandingkan antara kelas 1 dan kelas 2. Standar deviasi dihitung untuk mengetahui ragam nilai dalam data hasil ekstraksi GLCM karena standar deviasi akan memperlihatkan jarak setiap titik data terhadap nilai rata-rata.

4.2.2 Ekstraksi Fitur dengan Metode *Haar Wavelet*

Metode ekstraksi yang kedua adalah *Haar Wavelet*. Fitur yang digunakan dalam penelitian ini adalah *Energy*, *Mean*, dan *Standar deviasi* pada *Wavelet*. Fitur energi dibagi menjadi empat ciri, yaitu *Ea* (Aproksimasi), *Eh* (arah horizontal), *Ev* (arah vertikal), dan *Ed* (arah diagonal) sehingga jumlah fitur pada saat proses ekstraksi adalah 2.167 fitur. Citra yang digunakan adalah citra *body fluids* dari dua kelas, yaitu *Malignant* dan *Benign*. Hasil proses ekstraksi citra *body fluids* dengan metode *Haar Wavelet* dapat dilihat pada Tabel 4.5.

Tabel 4. 5 Hasil Ekstraksi *Haar Wavelet*

<i>Haar Wavelet</i>		
Fitur	Kelas 1 (<i>Malignant</i>)	Kelas 2 (<i>Benign</i>)
	Average ± Standar Deviasi	Average ± Standar Deviasi
1	99,852 ± 0,178	99,797 ± 0,157
2	0,005 ± 0,006	0,007 ± 0,006
3	0,013 ± 0,015	0,019 ± 0,019
4	0,034 ± 0,042	0,044 ± 0,033
5	0,092 ± 0,111	0,128 ± 0,097
6	0,0006 ± 0,0007	0,0007 ± 0,0005
7	0,003 ± 0,004	0,003 ± 0,003
8	0,013 ± 0,086	0,012 ± 0,074
9	0,014 ± 0,085	0,012 ± 0,074
10	0,015 ± 0,085	0,012 ± 0,074
⋮	⋮	⋮
2.167	0,592 ± 0,399	0,753 ± 0,326

Tabel 4.5 berisi nilai rata-rata dan standar deviasi dari hasil ekstraksi 2.167 fitur *Haar Wavelet*. Nilai rata-rata digunakan untuk membandingkan antara kelas 1 dan kelas 2. Standar deviasi dihitung untuk mengetahui ragam nilai dalam data hasil ekstraksi *Haar Wavelet* karena standar deviasi akan memperlihatkan jarak setiap titik data terhadap nilai rata-rata.

4.3 Hasil Klasifikasi

Jumlah citra *body fluids* yang digunakan adalah 9.545 citra, dimana citra tersebut dibagi menjadi data *training*, data *testing*, dan data validasi. Citra yang digunakan untuk data *training* adalah 90% dari total data, maka secara kuantitatif jumlahnya adalah 8.591 citra. Sebanyak 10% dari keseluruhan data digunakan sebagai data *testing* dan data validasi, maka secara kuantitatif jumlahnya adalah 477 citra untuk masing-masing data *testing* dan data validasi. Output dari klasifikasi ini terdiri dari dua kelas, yaitu kelas *Malignant* dan *Benign*.

4.3.1 Klasifikasi dari Metode *Gray Level Co-occurrence Matrix* (GLCM)

Sebanyak 9.545 citra yang telah diekstraksi menggunakan metode *Gray Level Co-occurrence Matrix* (GLCM) dilatih dan diuji menggunakan tiga model *Multilayer Perceptron* (MLP), yaitu *Scaled Conjugate Gradient Backpropagation* (SCG), *One Step Secant Backpropagation* (OSS), dan *Gradient Descent with*

Momentum (GDM) dengan *hidden neuron* 1, 5, 10, 15, dan 20. Setiap *training* akan menghasilkan *Pattern Recognition Neural Network* yang menggambarkan proses input hingga output. *Pattern Recognition Neural Network* masing-masing *hidden neuron* dapat dilihat pada Gambar 4.3.



Gambar 4. 2 (a) Pattern Recognition Neural Network Hidden Neuron 1, (b) Pattern Recognition Neural Network Hidden Neuron 5, (c) Pattern Recognition Neural Network Hidden Neuron 10, (d) Pattern Recognition Neural Network Hidden Neuron 15, (e) Pattern Recognition Neural Network Hidden Neuron 20

4.3.1.1 Scaled Conjugate Gradient Backpropagation (SCG)

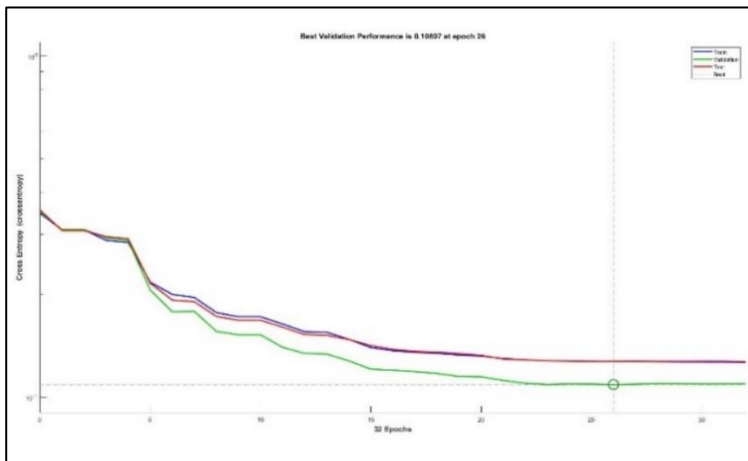
a. Hidden Neuron 1

Hasil akurasi *training* citra *body fluids* menggunakan model *Scaled Conjugate Gradient Backpropagation* (SCG) *hidden neuron* 1 dapat dilihat pada Tabel 4.6.

Tabel 4. 6 Hasil Akurasi *Training* Menggunakan Model SCG *Hidden Neuron* 1

<i>Run</i>	<i>Epoch</i>		MLP-TrainSCG Hidden Neuron 1			<i>Elapsed Time</i>
	<i>All</i>	<i>Best</i>	<i>Acc Training</i>	<i>Acc Validation</i>	<i>Acc Testing</i>	
1	32	26	90,0%	92,2%	89,7%	00:00:02
2	94	88	90,8%	91,4%	91,4%	00:00:02
3	42	36	90,1%	89,5%	89,9%	00:00:02
4	27	21	89,9%	91,6%	92,2%	00:00:02
5	36	30	90,0%	91,0%	89,7%	00:00:02
6	36	30	89,9%	89,5%	89,3%	00:00:02
7	53	47	90,2%	89,9%	90,4%	00:00:02
8	46	40	90,0%	89,7%	89,7%	00:00:02
9	33	27	89,8%	91,6%	91,2%	00:00:02
10	27	21	90,1%	87,8%	92,2%	00:00:02
Average			90,1%	90,4%	90,6%	00:00:02
Standar Deviasi			0,0027809	0,0135794	0,010935	

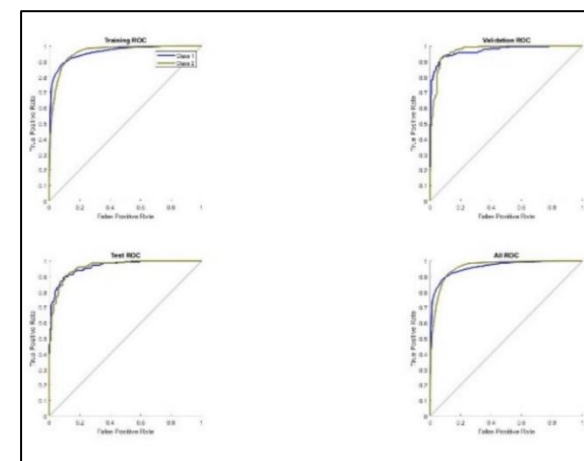
Berdasarkan Tabel 4.6 diketahui bahwa rata-rata akurasi *training* menggunakan model SCG dengan *hidden neuron* 1 adalah 89-90%. Hasil akurasi *training* terbaik terdapat pada *run* 2 dengan akurasi *training* sebesar 90,8%, akurasi validasi 91,4%, dan akurasi *testing* 91,4%. Akurasi *training* terbaik ini diperoleh pada *epoch* ke 88 dari total 94 *epoch* dengan waktu *training* selama 2 detik. Sementara itu, hasil akurasi *training* terendah terdapat pada *run* 9 dengan akurasi *training* 89,8%, akurasi validasi 91,6%, dan akurasi *testing* 91,2%. Dari pengujian diatas, diperoleh hasil bahwa rata-rata akurasi *training* adalah 90,1%, akurasi validasi 90,4%, dan akurasi *testing* 90,6%. Rata-rata waktu *training* menggunakan model SCG *hidden neuron* 1 adalah 2 detik. Visualisasi hasil *training* dapat dilihat pada Gambar 4.3.



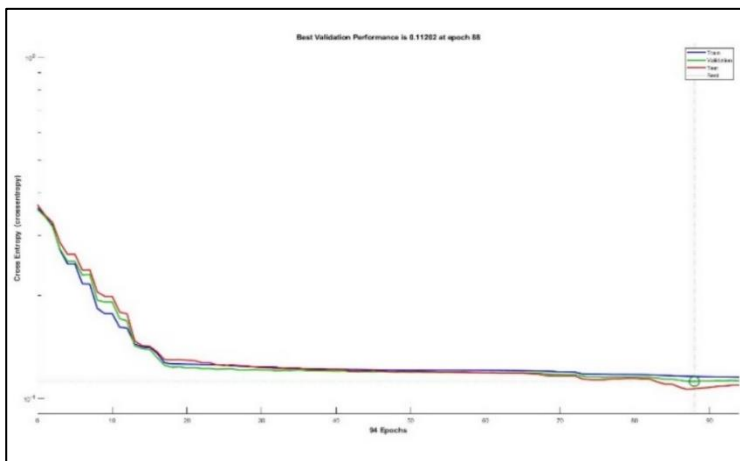
(a1)



(a2)



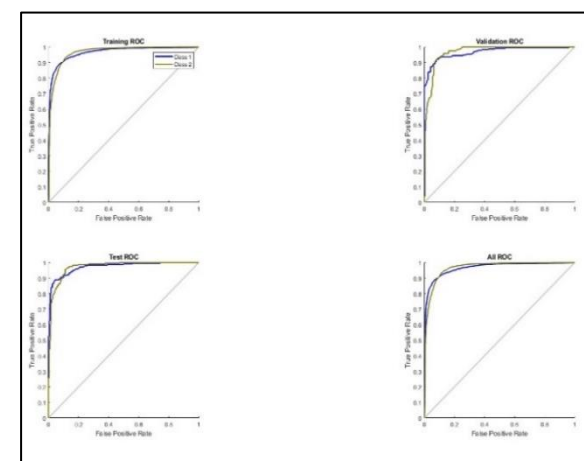
(a3)



(b1)

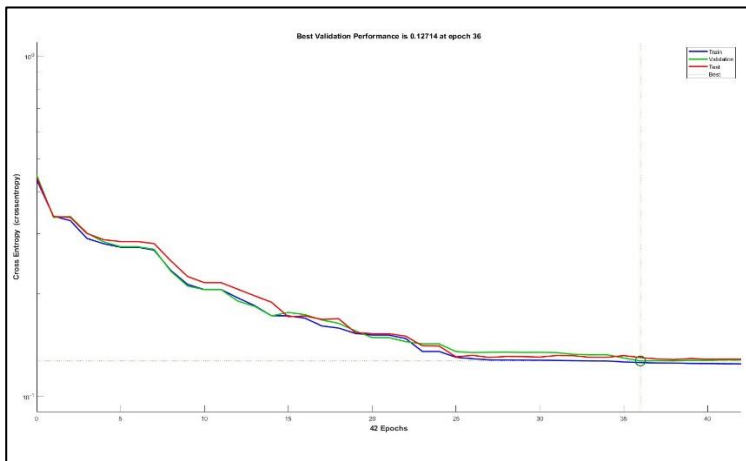


(b2)



(b3)

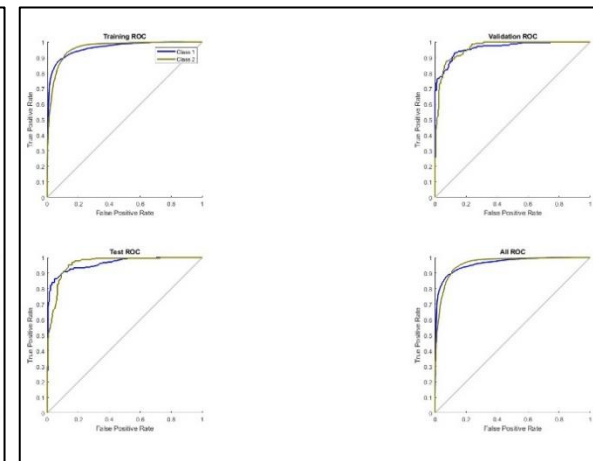
Gambar 4. 3 (a1) (b1) Performance Run 1 dan 2 Training SCG HN 1, (a2) (b2) Confusion Matrix Run 1 dan 2 Training SCG HN 1, (a1) (b2) ROC Run 1 dan 2 Training SCG HN 1



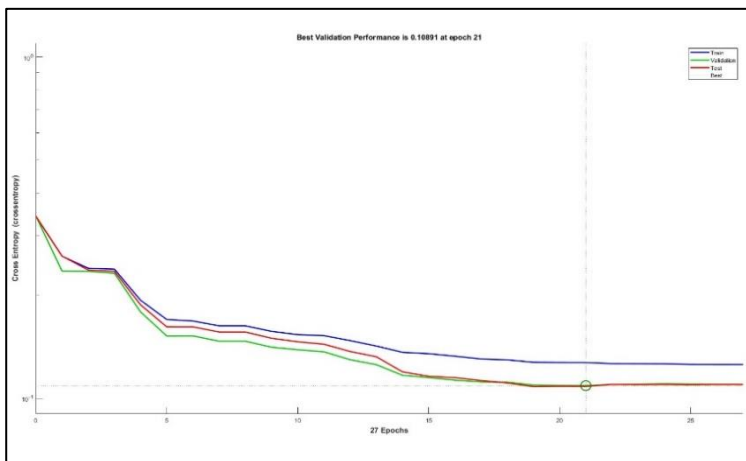
(c1)



(c2)



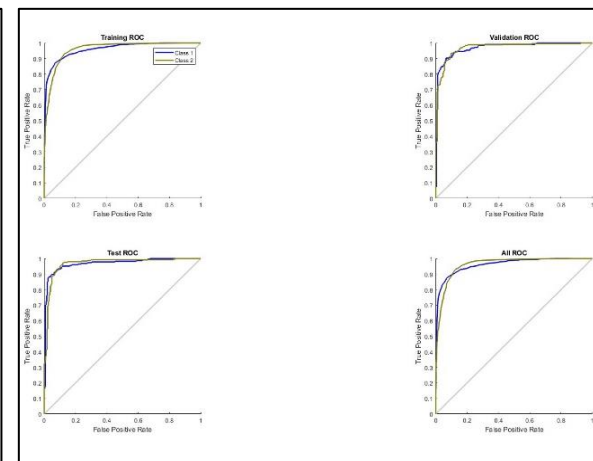
(c3)



(d1)

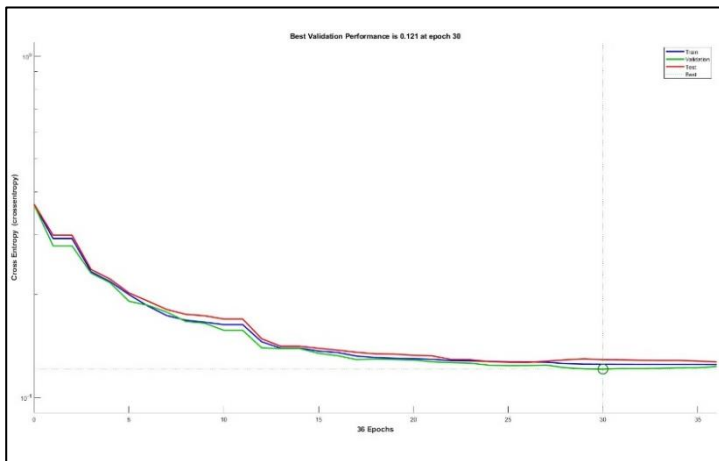


(d2)



(d3)

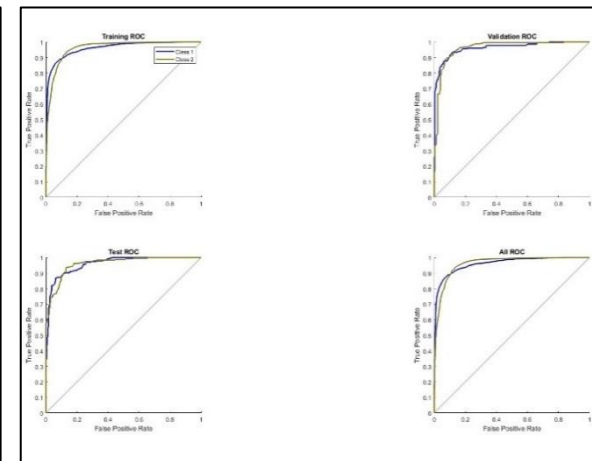
Gambar 4.3 (c1) (d1) Performance Run 3 dan 4 Training SCG HN 1, (c2) (d2) Confusion Matrix Run 3 dan 4 Training SCG HN 1, (c1) (d2) ROC Run 3 dan 4 Training SCG HN 1



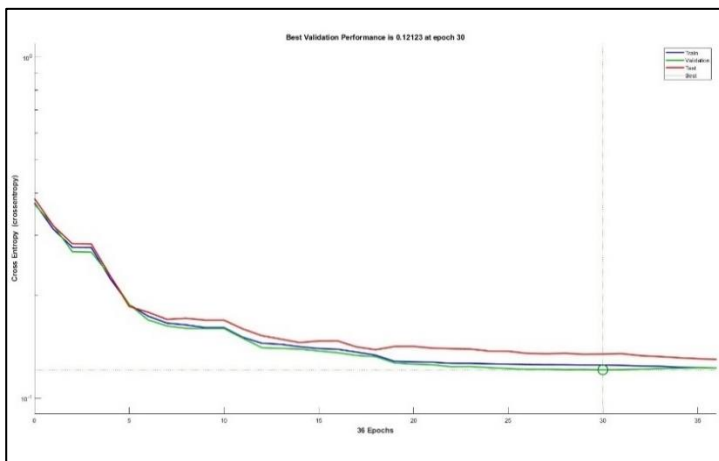
(e1)



(e2)



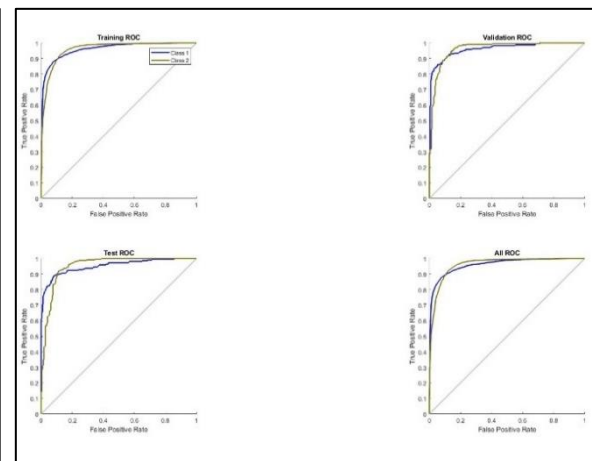
(e3)



(f1)

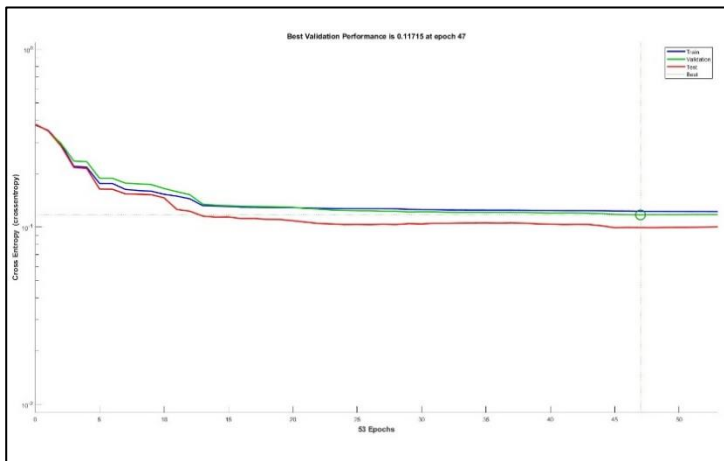


(f2)



(f3)

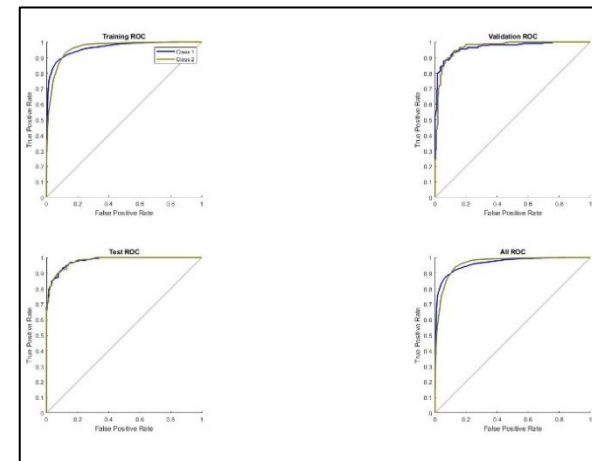
Gambar 4.3 (e1) (f1) Performance Run 5 dan 6 Training SCG HN 1, (e2) (f2) Confusion Matrix Run 5 dan 6 Training SCG HN 1, (e3) (f3) ROC Run 5 dan 6 Training SCG HN 1



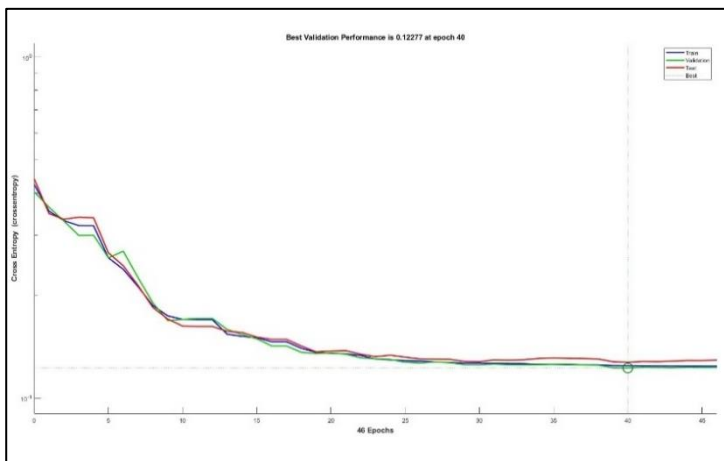
(g1)



(g2)



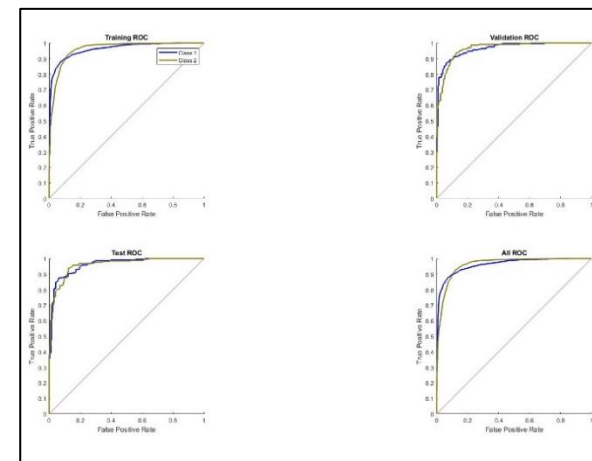
(g3)



(h1)

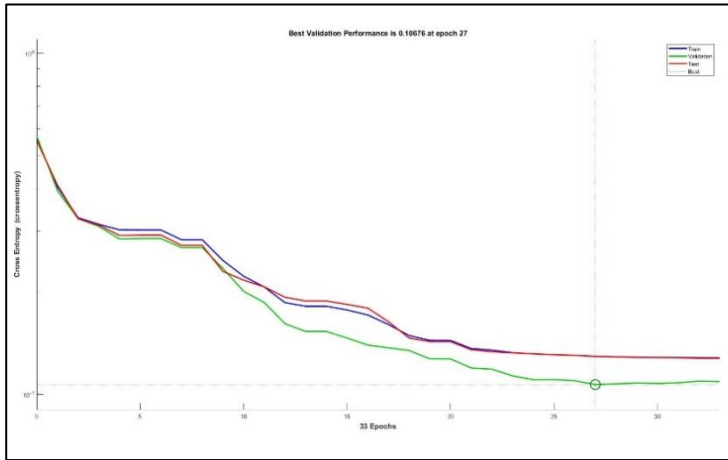


(h2)



(h3)

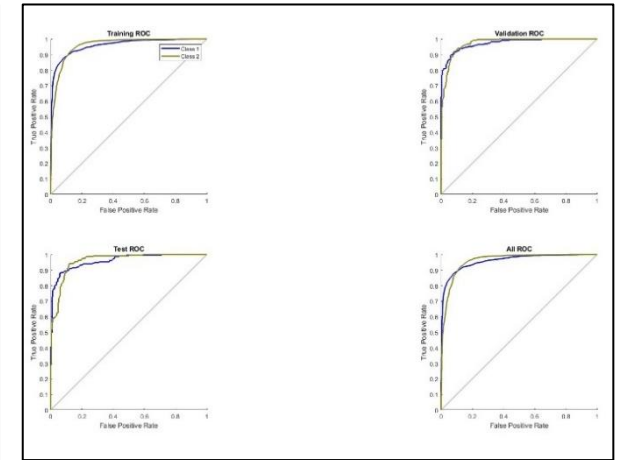
Gambar 4.3 (g1) (gh1) Performance Run 7 dan 8 Training SCG HN 1, (g2) (h2) Confusion Matrix Run 7 dan 8 Training SCG HN 1, (g3) (h3) ROC Run 7 dan 8 Training SCG HN 1



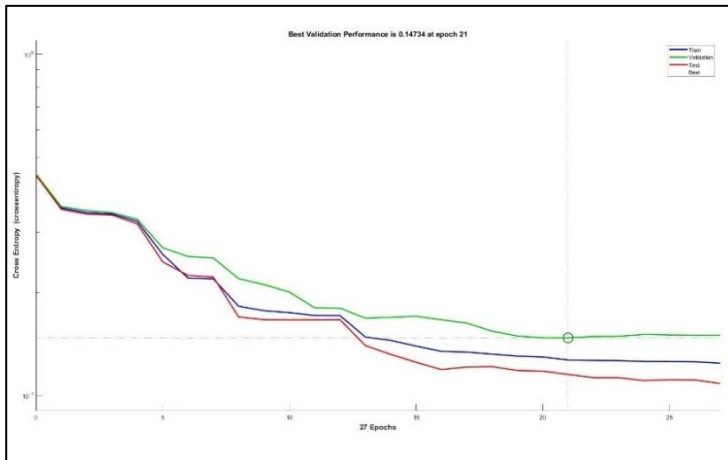
(i1)



(i2)



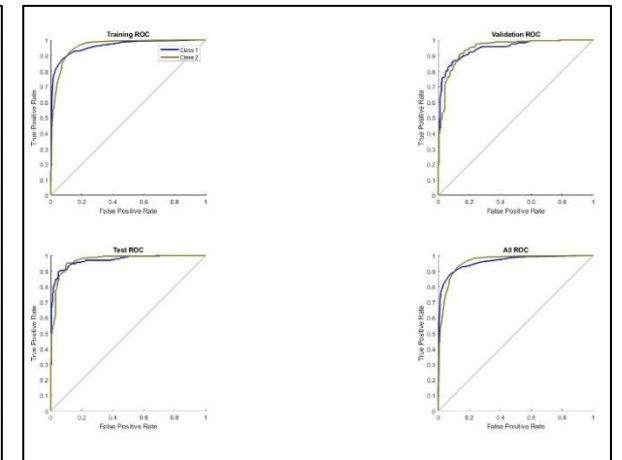
(i3)



(j1)



(j2)



(j3)

Gambar 4.3 (i1) (j1) Performance Run 9 dan 10 Training SCG HN 1, (i2) (j2) Confusion Matrix Run 9 dan 10 Training SCG HN 1, (i3) (j3) ROC Run 9 dan 10 Training SCG HN 1

Berdasarkan visualisasi hasil *training* di atas, dapat diketahui bahwa grafik *performance* terdiri dari 3 grafik, yaitu grafik biru merepresentasikan hasil *training*, grafik hijau merepresentasikan hasil validasi, dan grafik merah merepresentasikan hasil *testing*. Gambar 4.3 menampilkan sepuluh grafik *performane training SCG hidden neuron 1* dimana rata-rata hasil penurunan error signifikan terjadi pada *range epoch* 0-5 kecuali pada *run* 2 dan 8. Rata-rata grafik mengalami penurunan tidak signifikan (landai) dari *epoch* 6 sampai *epoch* maksimal. Sementara untuk *run* 2 dan 8, penurunan tidak signifikan tersebut mulai dari *epoch* 10 sampai *epoch* maksimalnya. Dari uraian di atas, dapat disimpulkan bahwa semakin rendah nilai entropi dari grafik (nilai error) maka semakin baik performa dari model.

Hasil *training* yang menunjukkan akurasi dari sistem ditunjukkan oleh *confusion matrix* dimana warna hijau merepresentasikan citra yang berhasil diklasifikasikan ke dalam kelas secara benar dan warna merah adalah citra yang gagal diklasifikasikan dalam kelasnya. Perbandingan data yang digunakan untuk *10-run cross validation* yaitu 90% dari total data (8.591) digunakan sebagai data *training*, 5% dari total data (477) digunakan sebagai data validasi, dan 5% lainnya (477) digunakan sebagai data *testing*. Dari *confusion matrix* tersebut, diketahui bahwa rata-rata akurasi *training*, validasi, dan *testing* di atas 90%. Nilai rata-rata *error training* adalah $\leq 10\%$ (859) dan rata-rata nilai *error testing* adalah $\leq 10\%$ (47). Hal ini menunjukkan bahwa model memiliki performa yang baik.

Visualisasi grafik lainnya adalah grafik *Receiver Operating Characteristic* (ROC). Fungsi dari grafik ROC ini adalah untuk mengetahui hasil evaluasi performa *machine learning*. Pada grafik ROC, terdapat dua warna grafik yaitu biru yang menunjukkan klasifikasi citra kelas 1 dan grafik hijau menunjukkan klasifikasi citra kelas 2. Dari *run* 1-10, dapat diketahui bahwa nilai grafik *training*, validasi, dan *testing* mendekati angka 1 dan semakin menjauhi garis *threshold*. Hal tersebut menunjukkan bahwa *machine learning* (*Multilayer Perceptron* model *SCG hidden neuron 1*) memiliki performa *perfect*.

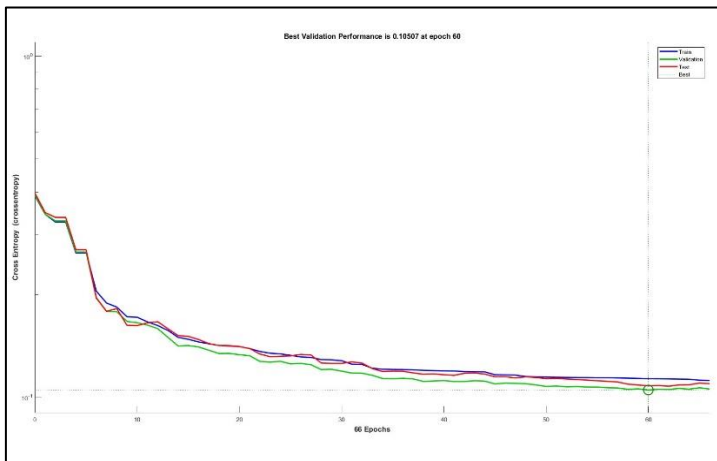
b. *Hidden Neuron 5*

Hasil akurasi *training* citra *body fluids* menggunakan model *Scaled Conjugate Gradient Backpropagation* (SCG) *hidden neuron 5* dapat dilihat pada Tabel 4.7.

Tabel 4. 7 Hasil Akurasi *Training* Menggunakan Model SCG *Hidden Neuron 5*

<i>Run</i>	<i>Epoch</i>		MLP-TrainSCG <i>Hidden Neuron 5</i>			<i>Elapsed Time</i>
	<i>All</i>	<i>Best</i>	<i>Acc Training</i>	<i>Acc Validation</i>	<i>Acc Testing</i>	
1	66	60	90,5%	92,0%	91,0%	00:00:02
2	101	95	91,6%	93,7%	90,1%	00:00:03
3	39	33	90,3%	92,9%	90,4%	00:00:02
4	39	33	90,4%	88,1%	89,5%	00:00:02
5	23	17	89,9%	88,5%	91,8%	00:00:02
6	27	21	90,1%	91,4%	90,1%	00:00:02
7	34	28	89,9%	89,1%	90,8%	00:00:02
8	61	55	90,9%	89,1%	90,4%	00:00:18
9	23	17	89,9%	90,1%	87,0%	00:00:02
10	72	66	91,4%	92,2%	93,9%	00:00:02
Average			90,5%	90,7%	90,5%	00:00:04
Standar Deviasi			0,0062084	0,0198183	0,017378	

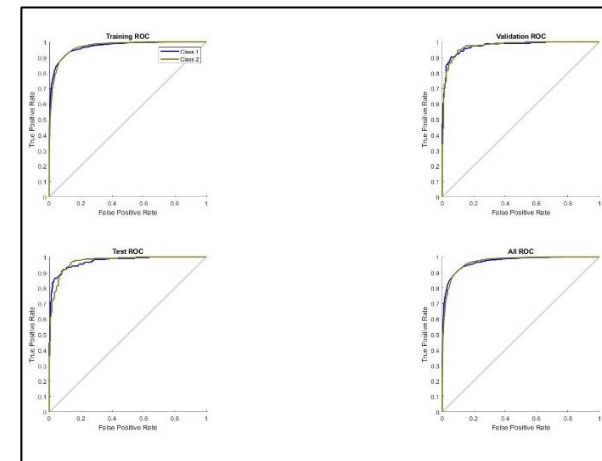
Berdasarkan Tabel 4.7 diketahui bahwa rata-rata akurasi *training* menggunakan model SCG dengan *hidden neuron 5* adalah 89-91%. Hasil akurasi *training* terbaik terdapat pada *run 2* dengan akurasi *training* sebesar 91,6%, akurasi validasi 93,7%, dan akurasi *testing* 90,1%. Akurasi *training* terbaik ini diperoleh pada *epoch* ke 95 dari total 101 *epoch* dengan waktu *training* selama 3 detik. Sementara itu, hasil akurasi *training* terendah terdapat pada *run 9* dengan akurasi *training* 89,9%, akurasi validasi 88,5%, dan akurasi *testing* 91,8%. Akurasi *training* terendah ini diperoleh pada *epoch* ke 17 dari total 23 *epoch* dengan waktu *training* selama 2 detik. Dari pengujian diatas, diperoleh hasil bahwa rata-rata akurasi *training* adalah 90,5%, akurasi validasi 90,7%, dan akurasi *testing* 90,5%. Rata-rata waktu *training* menggunakan model SCG *hidden neuron 5* adalah 4 detik. Visualisasi hasil *training* dapat dilihat pada Gambar 4.4.



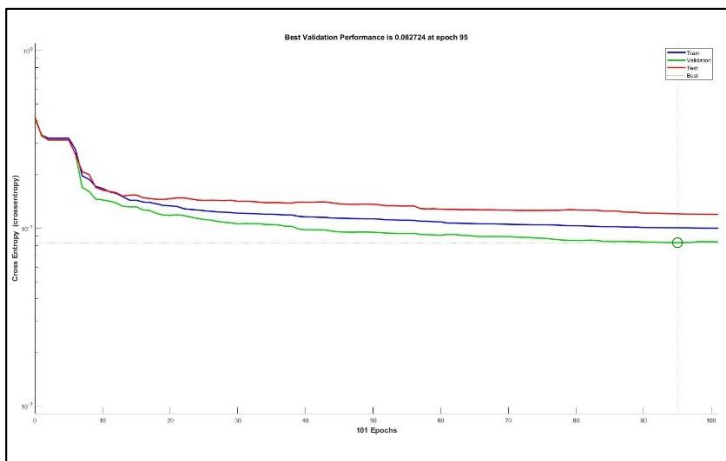
(a1)



(a2)



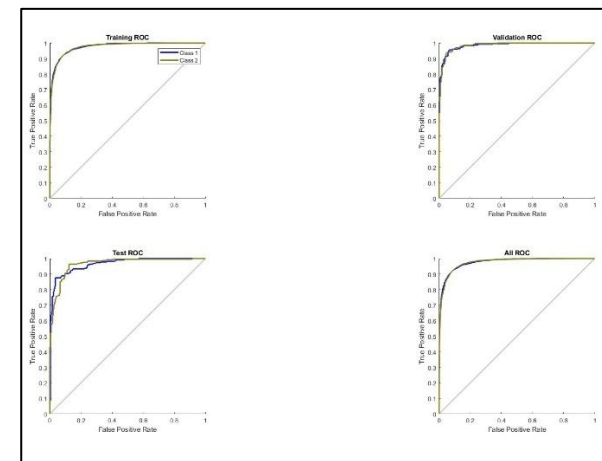
(a3)



(b1)

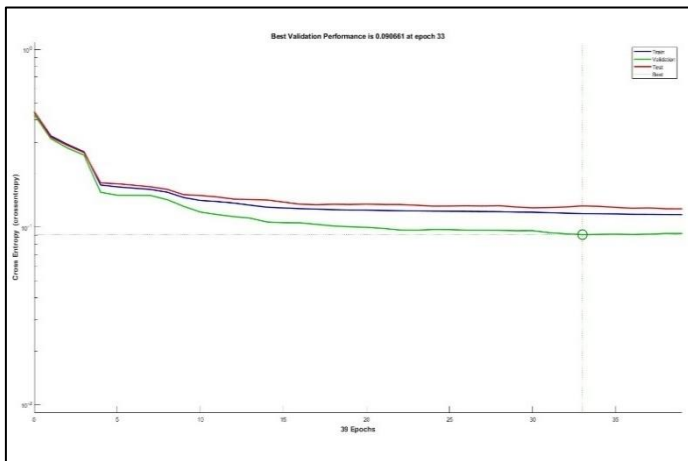


(b2)



(b3)

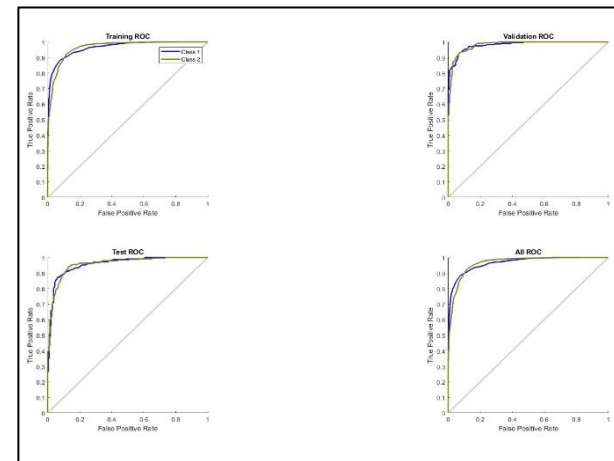
Gambar 4. 4 (a1) (b1) Performance Run 1 dan 2 Training SCG HN 5, (a2) (b2) Confusion Matrix Run 1 dan 2 Training SCG HN 5, (a3) (b3) ROC Run 1 dan 2 Training SCG HN 5



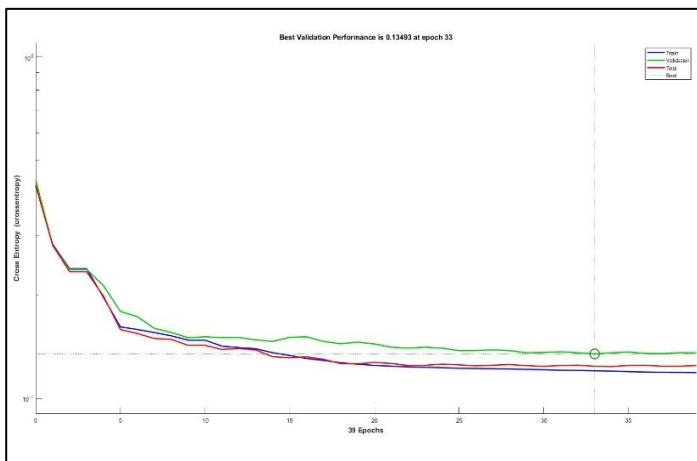
(c1)



(c2)



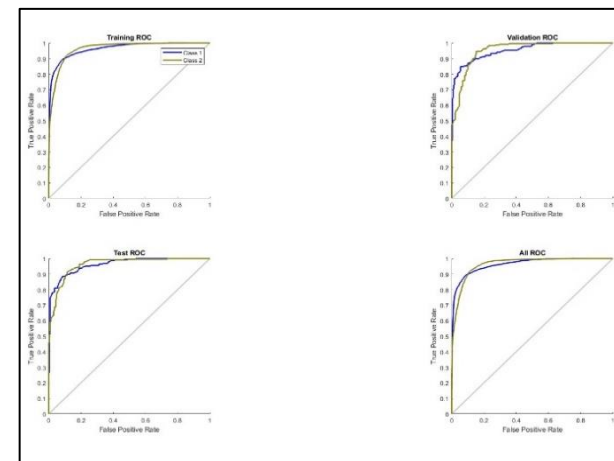
(c3)



(d1)

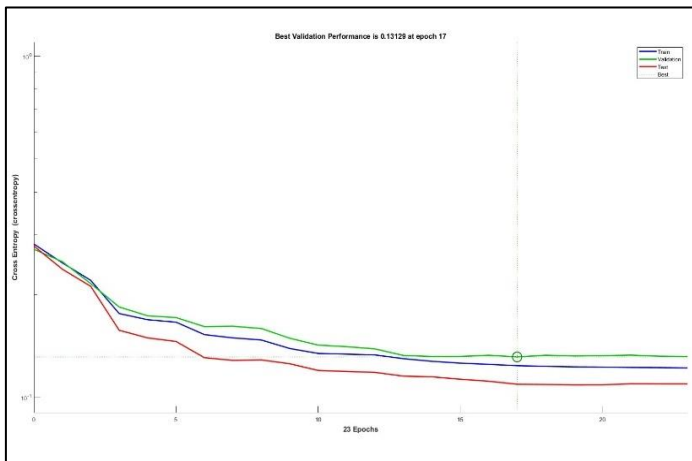


(d2)



(d3)

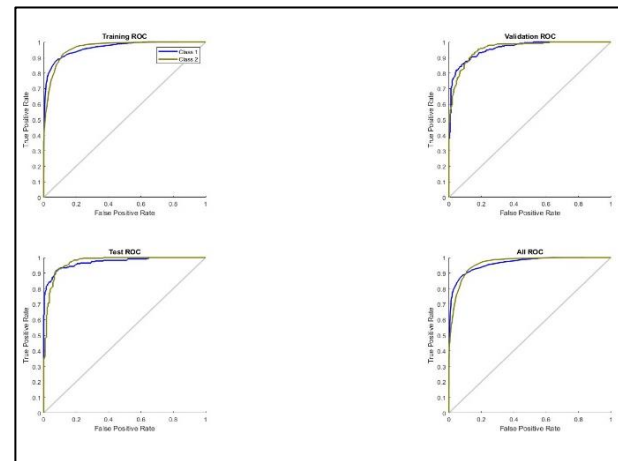
Gambar 4.4 (c1) (d1) Performance Run 3 dan 4 Training SCG HN 5, (c2) (d2) Confusion Matrix Run 3 dan 4 Training SCG HN 5, (c3) (d3) ROC Run 3 dan 4 Training SCG HN 5



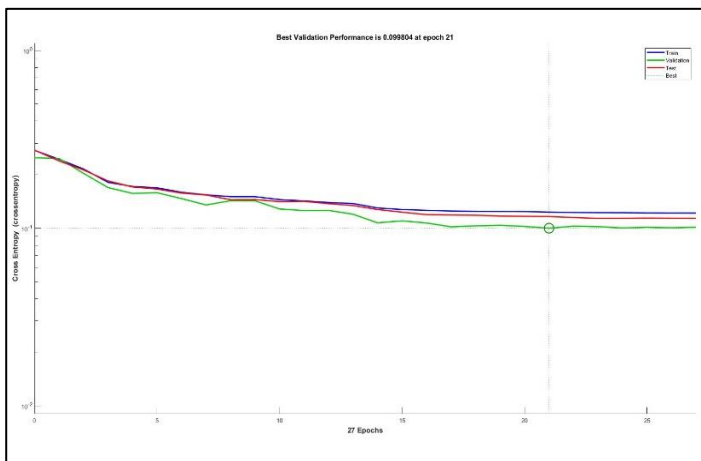
(e1)



(e2)



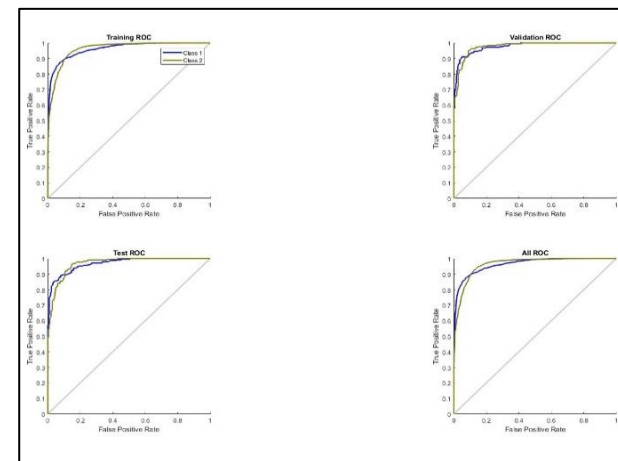
(e3)



(f1)

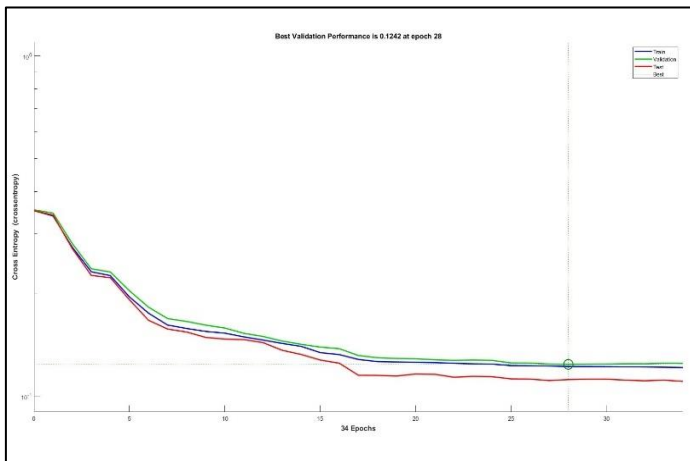


(f2)



(f3)

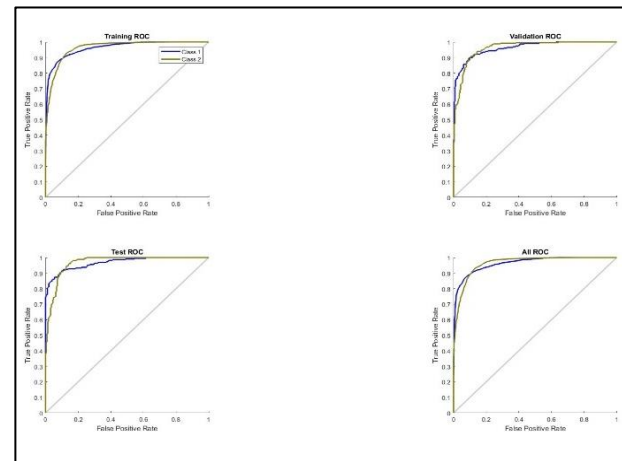
Gambar 4.4 (e1) (f1) Performance Run 5 dan 6 Training SCG HN 5, (e2) (f2) Confusion Matrix Run 5 dan 6 Training SCG HN 5, (e3) (f3) ROC Run 5 dan 6 Training SCG HN 5



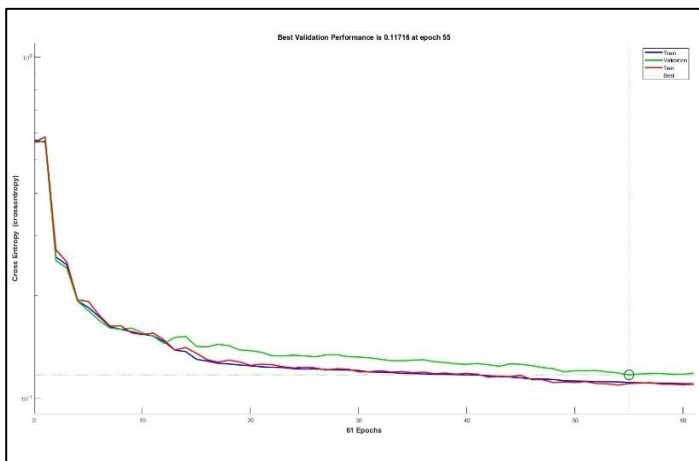
(g1)



(g2)



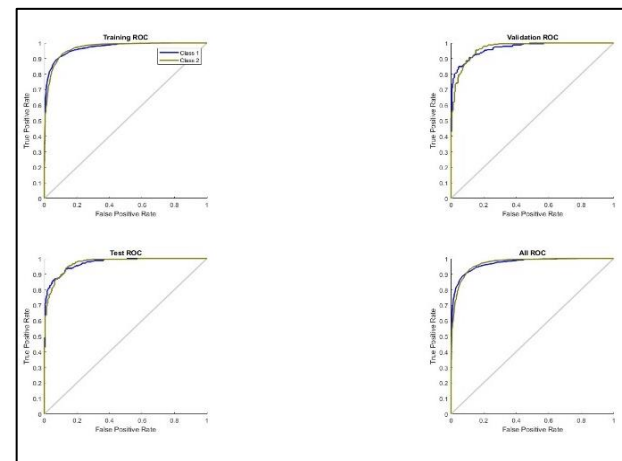
(g3)



(h1)

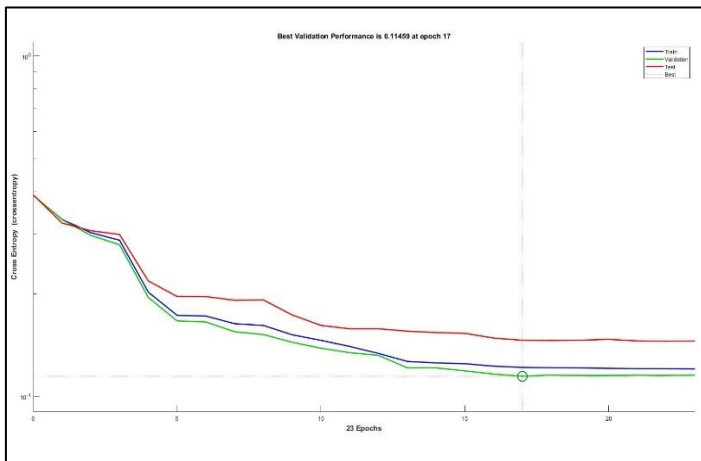


(h2)



(h3)

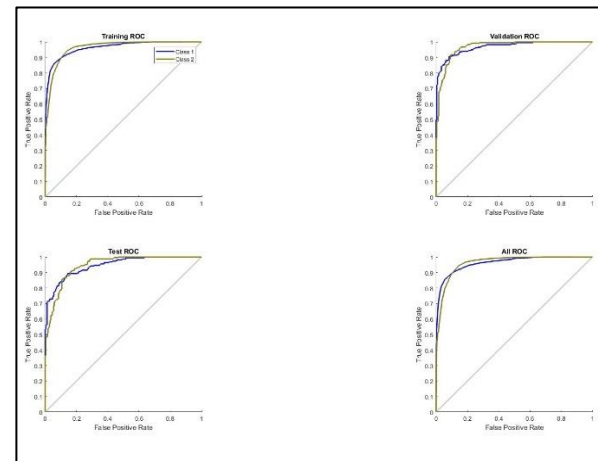
Gambar 4.4 (g1) (h1) Performance Run 7 dan 8 Training SCG HN 5, (g2) (h2) Confusion Matrix Run 7 dan 8 Training SCG HN 5, (g3) (h3) ROC Run 7 dan 8 Training SCG HN 5



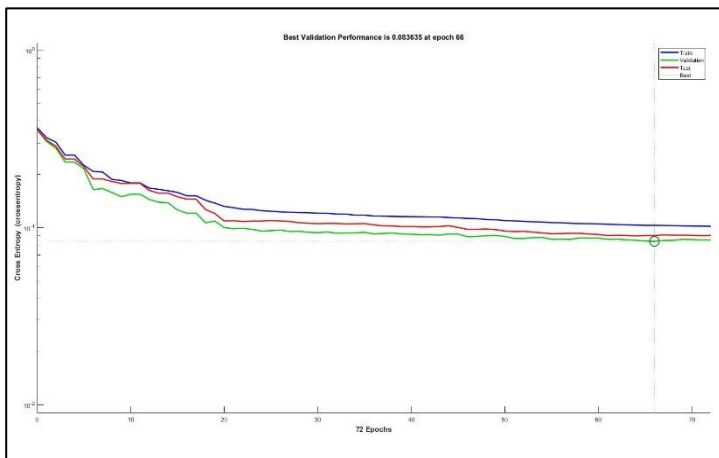
(i1)



(i2)



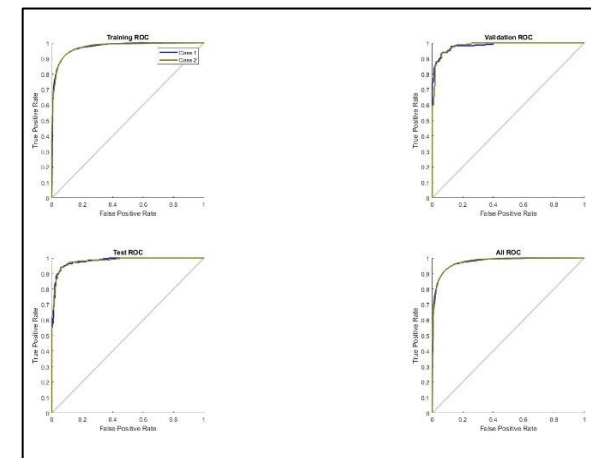
(i3)



(j1)



(j2)



(j3)

Gambar 4.4 (i1) (j1) Performance Run 9 dan 10 Training SCG HN 5, (i2) (j2) Confusion Matrix Run 9 dan 10 Training SCG HN 5, (i3) (j3) ROC Run 9 dan 10 Training SCG HN 5

Berdasarkan visualisasi hasil *training* di atas, diketahui bahwa grafik *performance* terdiri dari 3 grafik, yaitu grafik biru adalah hasil *training*, grafik hijau adalah hasil validasi, dan grafik merah adalah hasil *testing*. Gambar 4.4 menampilkan sepuluh grafik *performance training SCG hidden neuron 5* dimana enam grafik diantaranya mengalami penurunan signifikan pada *range epoch* 0-5, yaitu pada *run* 3,4,5,6,7, dan 9. Pada enam grafik tersebut, penurunan tidak signifikan dimulai dari *epoch* 6 sampai maksimalnya. Empat grafik lainnya, yaitu grafik *run* 1,2,8, dan 10 mengalami penurunan signifikan pada *range epoch* 0-10 kemudian penurunan tidak signifikan mulai dari *epoch* 11 sampai maksimalnya. Dari uraian di atas, dapat disimpulkan bahwa semakin rendah nilai entropi dari grafik (nilai error) maka semakin baik performa dari model.

Hasil *training* yang menunjukkan akurasi dari sistem ditunjukkan oleh *confusion matrix* dimana warna hijau merepresentasikan citra yang berhasil diklasifikasikan ke dalam kelas secara benar dan warna merah adalah citra yang gagal diklasifikasikan dalam kelasnya. Perbandingan data yang digunakan untuk *10-run cross validation* yaitu 90% dari total data (8.591) digunakan sebagai data *training*, 5% dari total data (477) digunakan sebagai data validasi, dan 5% lainnya (477) digunakan sebagai data *testing*. Dari *confusion matrix* tersebut, diketahui bahwa rata-rata akurasi *training* dan *testing* $\geq 90\%$ sedangkan validasi $\geq 88\%$. Nilai rata-rata *error training* adalah $\leq 10\%$ (859) dan rata-rata nilai *error testing* adalah $\leq 10\%$ (47) kecuali pada *run* 9. Meskipun demikian model tetap memiliki performa yang baik.

Visualisasi grafik lainnya adalah grafik *Receiver Operating Characteristic* (ROC). Fungsi dari grafik ROC ini adalah untuk mengetahui hasil evaluasi performa *machine learning*. Pada grafik ROC, terdapat dua warna grafik yaitu biru yang menunjukkan klasifikasi citra kelas 1 dan grafik hijau menunjukkan klasifikasi citra kelas 2. Dari *run* 1-10, dapat diketahui bahwa nilai grafik *training*, validasi, dan *testing* mendekati angka 1 dan semakin menjauhi garis *threshold* kecuali pada *run* 4, kurva *validation* pada kelas 1 menunjukkan hasil pada angka 0,8. Meskipun demikian, performa *machine learning* (*Multilayer Perceptron* model *SCG hidden neuron 5*) menunjukkan performa perfect.

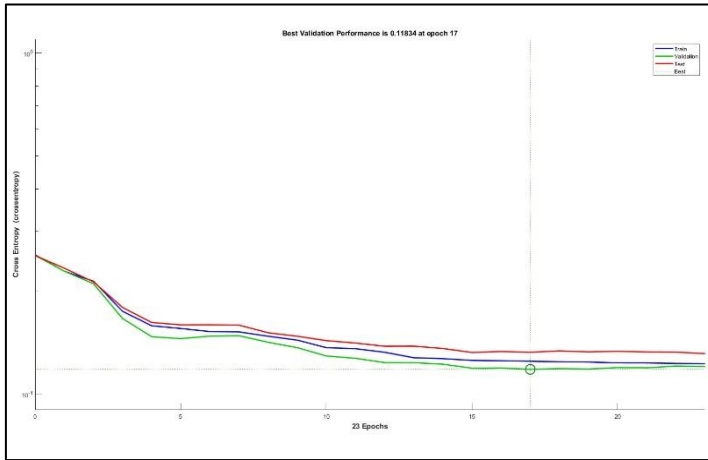
c. *Hidden Neuron 10*

Hasil akurasi *training* citra *body fluids* menggunakan model *Scaled Conjugate Gradient Backpropagation* (SCG) *hidden neuron 10* dapat dilihat pada Tabel 4.8.

Tabel 4. 8 Hasil Akurasi *Training* Menggunakan Model SCG *Hidden Neuron 10*

<i>Run</i>	<i>Epoch</i>		MLP-TrainSCG <i>Hidden Neuron 10</i>			<i>Elapsed Time</i>
	<i>All</i>	<i>Best</i>	<i>Acc Training</i>	<i>Acc Validation</i>	<i>Acc Testing</i>	
1	23	17	89,9%	89,5%	89,1%	00:00:02
2	62	56	91,1%	92,9%	93,1%	00:00:02
3	28	22	90,1%	89,9%	91,0%	00:00:02
4	173	167	94,7%	94,5%	94,8%	00:00:04
5	57	51	90,9%	89,9%	90,1%	00:00:02
6	30	24	90,2%	89,5%	91,0%	00:00:02
7	49	43	90,5%	91,4%	88,7%	00:00:02
8	60	54	91,0%	93,1%	92,5%	00:00:02
9	35	29	90,3%	87,4%	90,4%	00:00:02
10	36	30	90,0%	89,3%	91,2%	00:00:02
Average			90,9%	90,7%	91,2%	00:00:02
Standar Deviasi			0,0141189	0,0217368	0,018514	

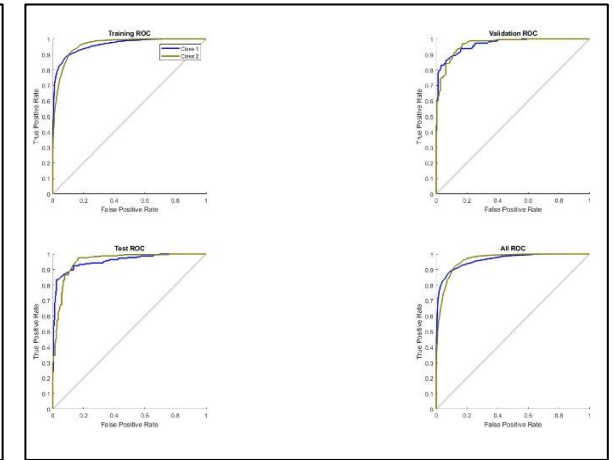
Berdasarkan Tabel 4.8 diketahui bahwa rata-rata akurasi *training* menggunakan model SCG dengan *hidden neuron 10* adalah 89-94%. Hasil akurasi *training* terbaik terdapat pada *run 4* dengan akurasi *training* sebesar 94,7%, akurasi validasi 94,5%, dan akurasi *testing* 94,8%. Akurasi *training* terbaik ini diperoleh pada *epoch* ke 167 dari total 173 *epoch* dengan waktu *training* selama 4 detik. Sementara itu, hasil akurasi *training* terendah terdapat pada *run 1* dengan akurasi *training* 89,9%, akurasi validasi 89,5%, dan akurasi *testing* 89,1%. Akurasi *training* terendah ini diperoleh pada *epoch* ke 17 dari total 23 *epoch* dengan waktu *training* selama 2 detik. Dari pengujian diatas, diperoleh hasil bahwa rata-rata akurasi *training* adalah 90,9%, akurasi validasi 90,7%, dan akurasi *testing* 91,2%. Rata-rata waktu *training* menggunakan model SCG *hidden neuron 10* adalah 2 detik. Visualisasi hasil *training* dapat dilihat pada Gambar 4.5.



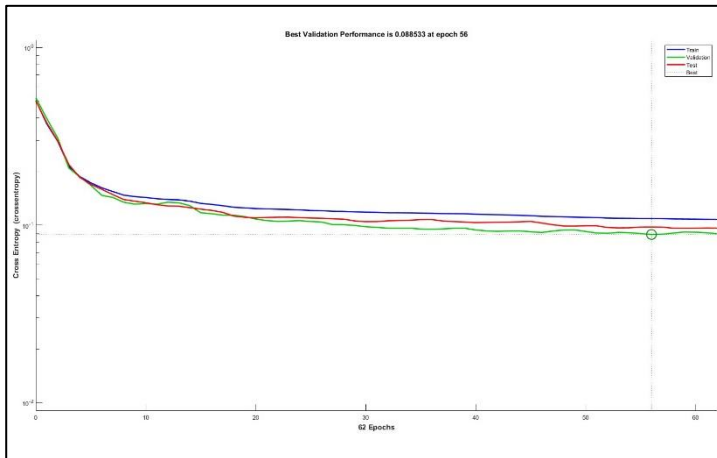
(a1)



(a2)



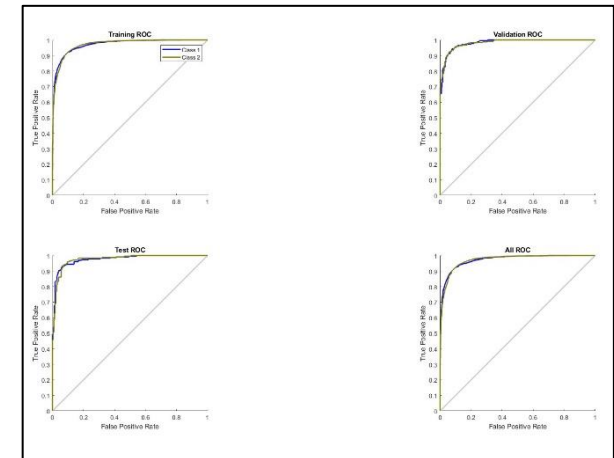
(a3)



(b1)

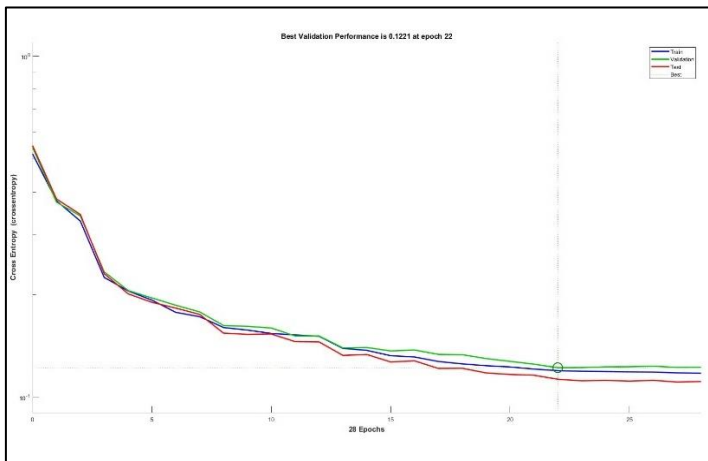


(b2)



(b3)

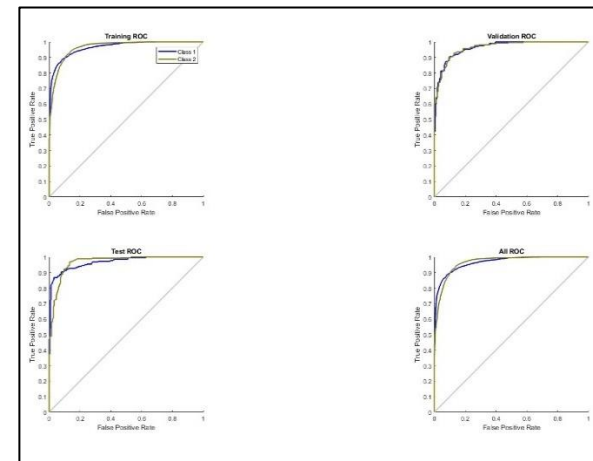
Gambar 4. 5 (a1) (b1) Performance Run 1 dan 2 Training SCG HN 10, (a2) (b2) Confusion Matrix Run 1 dan 2 Training SCG HN 10, (a3) (b3) ROC Run 1 dan 2 Training SCG HN 10



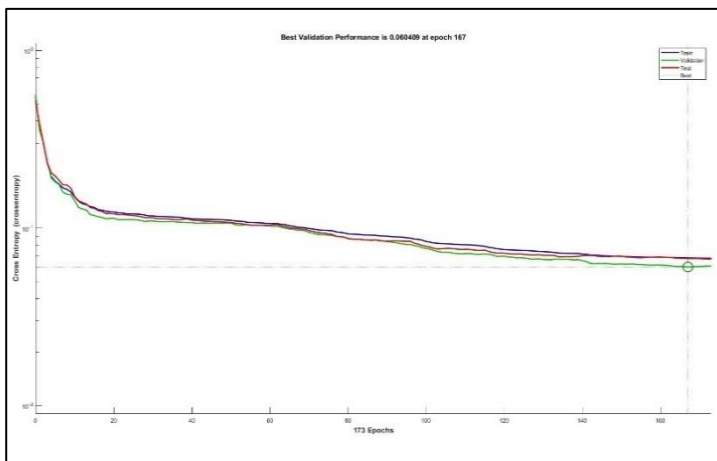
(c1)



(c2)



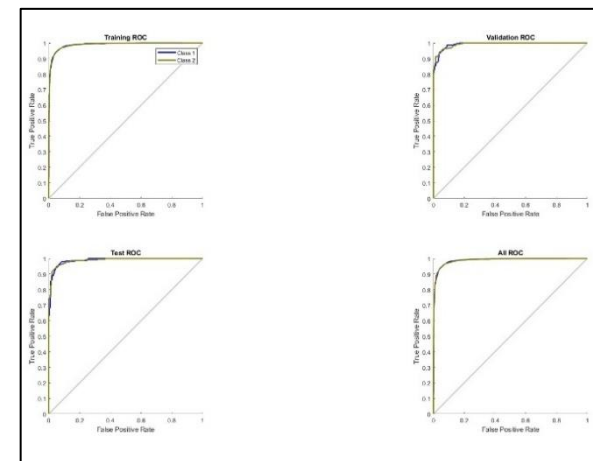
(c3)



(d1)

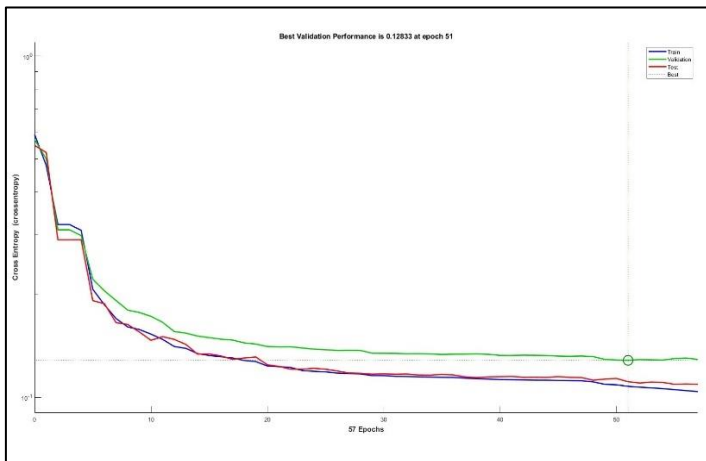


(d2)



(d3)

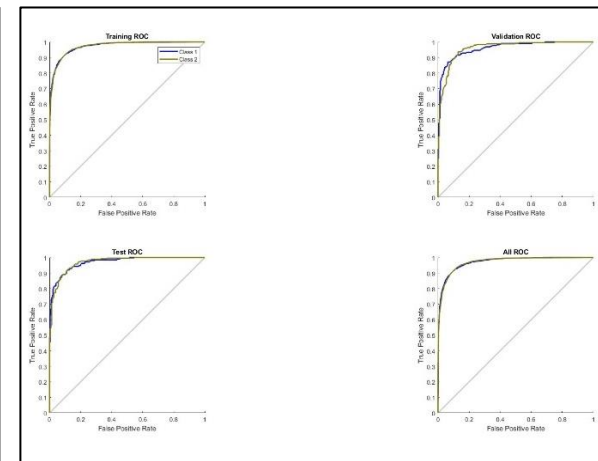
Gambar 4.5 (c1) (d1) Performance Run 3 dan 4 Training SCG HN 10, (c2) (d2) Confusion Matrix Run 3 dan 4 Training SCG HN 10, (c3) (d3) ROC Run 3 dan 4 Training SCG HN 10



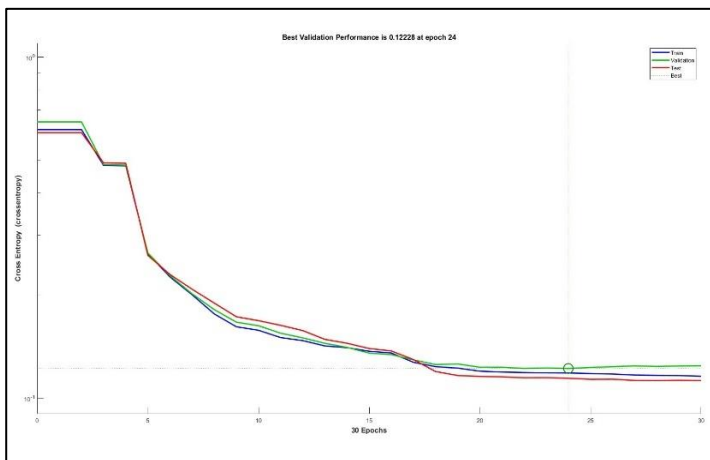
(e1)



(e2)



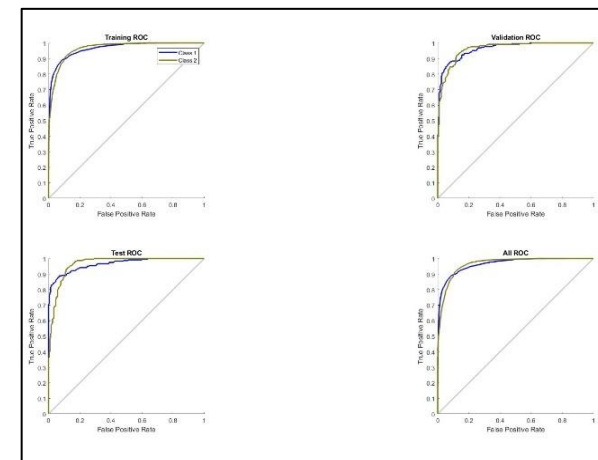
(e3)



(f1)

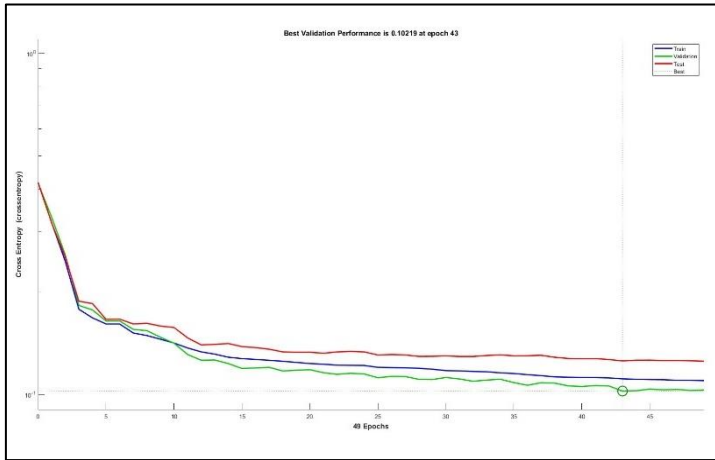


(f2)



(f3)

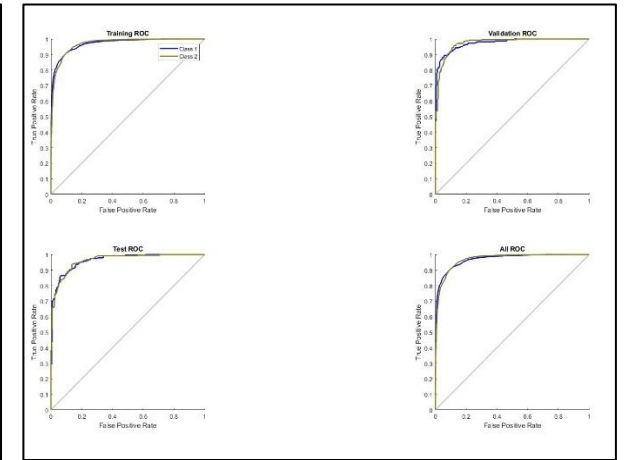
Gambar 4.5 (e1) (f1) Performance Run 5 dan 6 Training SCG HN 10, (e2) (f2) Confusion Matrix Run 5 dan 6 Training SCG HN 10, (e3) (f3) ROC Run 5 dan 6 Training SCG HN 10



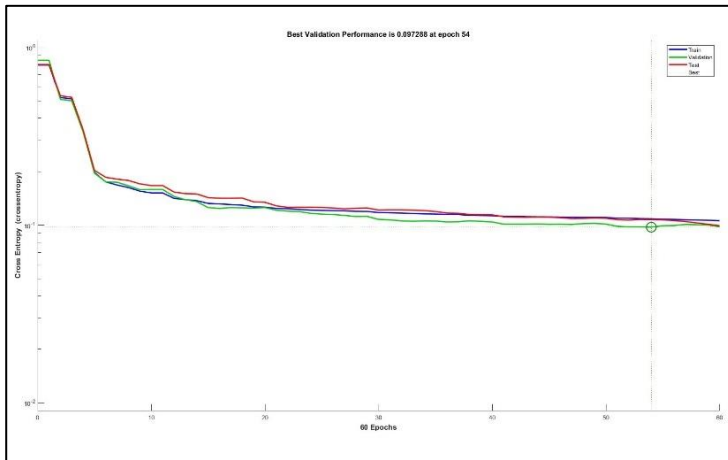
(g1)



(g2)



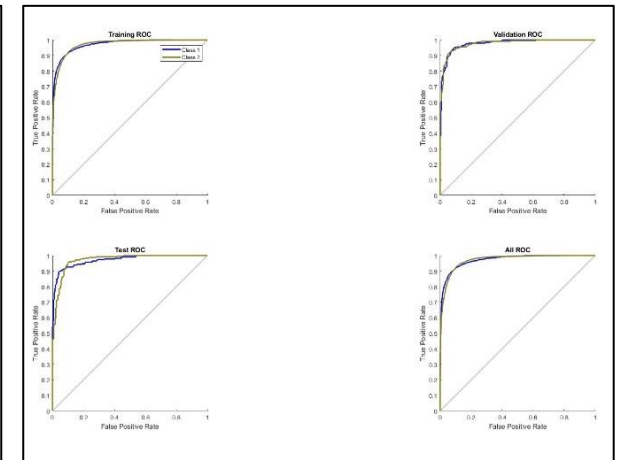
(g3)



(h1)

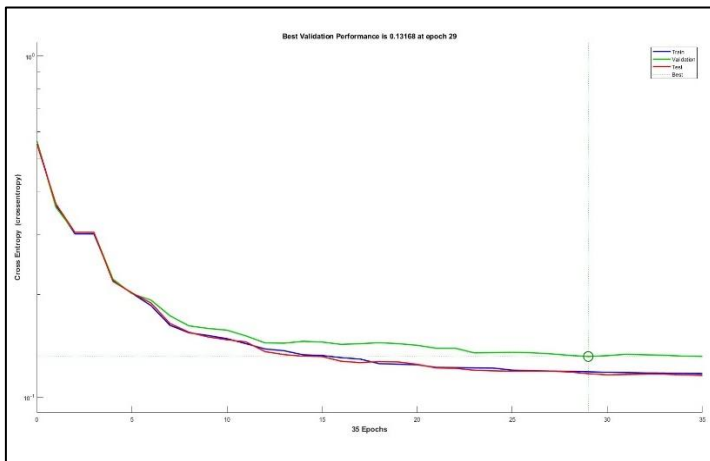


(h2)



(h3)

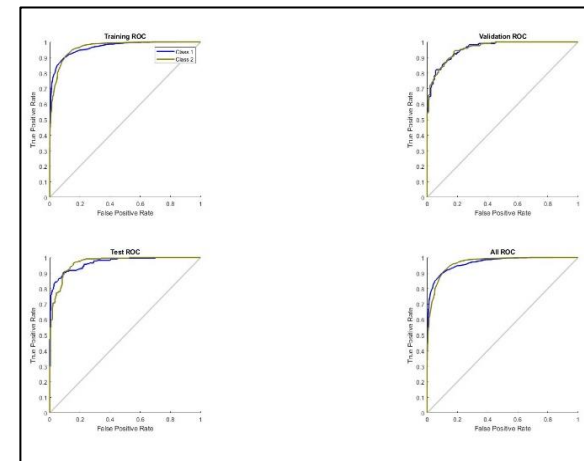
Gambar 4.5 (g1) (h1) Performance Run 7 dan 8 Training SCG HN 10, (g2) (h2) Confusion Matrix Run 7 dan 8 Training SCG HN 10, (g3) (h3) ROC Run 7 dan 8 Training SCG HN 10



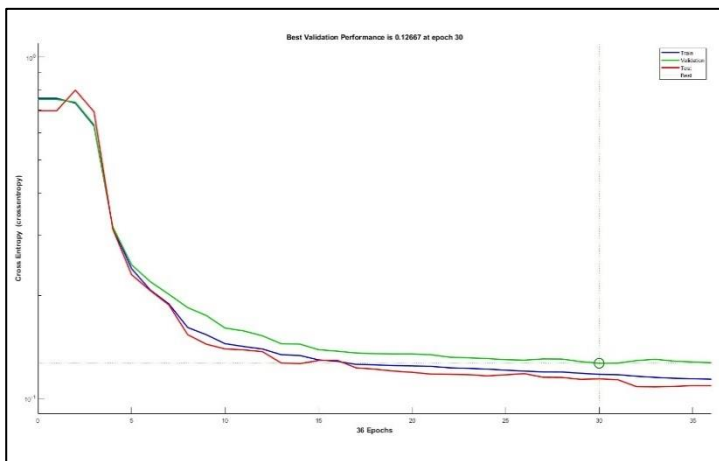
(i1)



(i2)



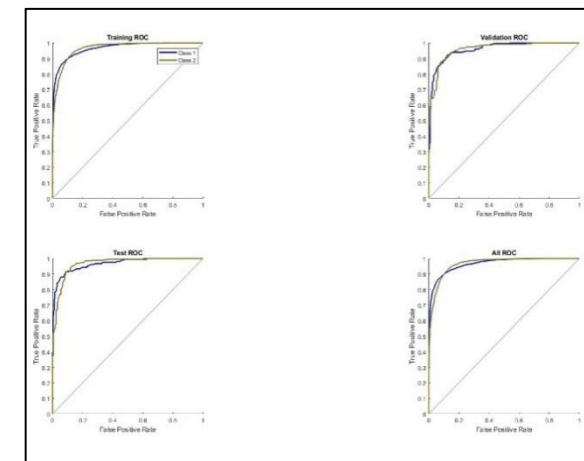
(i3)



(j1)



(j2)



(j3)

Gambar 4.5 (i1) (j1) Performance Run 9 dan 10 Training SCG HN 10, (i2) (j2) Confusion Matrix Run 9 dan 10 Training SCG HN 10, (i3) (j3) ROC Run 9 dan 10 Training SCG HN 10

Berdasarkan visualisasi hasil *training* di atas, dapat diketahui bahwa grafik *performance* terdiri dari 3 grafik, yaitu grafik biru merepresentasikan hasil *training*, grafik hijau merepresentasikan hasil validasi, dan grafik merah merepresentasikan hasil *testing*. Gambar 4.5 menampilkan sepuluh grafik *performane training SCG hidden neuron 10* dimana rata-rata grafik mengalami penurunan signifikan pada *range epoch 0-5*, kecuali pada *run 5*. Grafik mulai mengalami penurunan tidak signifikan pada *epoch 3* hingga maksimal dapat dilihat pada *run 3* sampai *5*, *run 7*, dan *run 9*. Sementara itu, hasil *run 1* dan *2*, grafik tidak mengalami penurunan signifikan dari awal hingga maksimalnya *epoch*. Dari uraian di atas, dapat disimpulkan bahwa semakin rendah nilai entropi dari grafik (nilai error) maka semakin baik performa dari model.

Hasil *training* yang menunjukkan akurasi dari sistem ditunjukkan oleh *confusion matrix* dimana warna hijau merepresentasikan citra yang berhasil diklasifikasikan ke dalam kelas secara benar dan warna merah adalah citra yang gagal diklasifikasikan dalam kelasnya. Perbandingan data yang digunakan untuk *10-run cross validation* yaitu 90% dari total data (8.591) digunakan sebagai data *training*, 5% dari total data (477) digunakan sebagai data validasi, dan 5% lainnya (477) digunakan sebagai data *testing*. Dari *confusion matrix* tersebut, diketahui bahwa rata-rata akurasi *training*, validasi, dan *testing* lebih dari 90%. Nilai rata-rata *error training* adalah $\leq 10\%$ (859) dan rata-rata nilai *error testing* adalah $\leq 10\%$ (47). Hal ini menunjukkan bahwa model memiliki performa yang baik.

Visualisasi grafik lainnya adalah grafik *Receiver Operating Characteristic* (ROC). Fungsi dari grafik ROC ini adalah untuk mengetahui hasil evaluasi performa *machine learning*. Pada grafik ROC, terdapat dua warna grafik yaitu biru yang menunjukkan klasifikasi citra kelas 1 dan grafik hijau menunjukkan klasifikasi citra kelas 2. Dari *run 1-10*, dapat diketahui bahwa nilai grafik *training*, validasi, dan *testing* mendekati angka 1 dan semakin menjauhi garis *threshold*. Dengan demikian, dapat disimpulkan bahwa performa *machine learning* (*Multilayer Perceptron* model *SCG hidden neuron 10*) menunjukkan performa perfect.

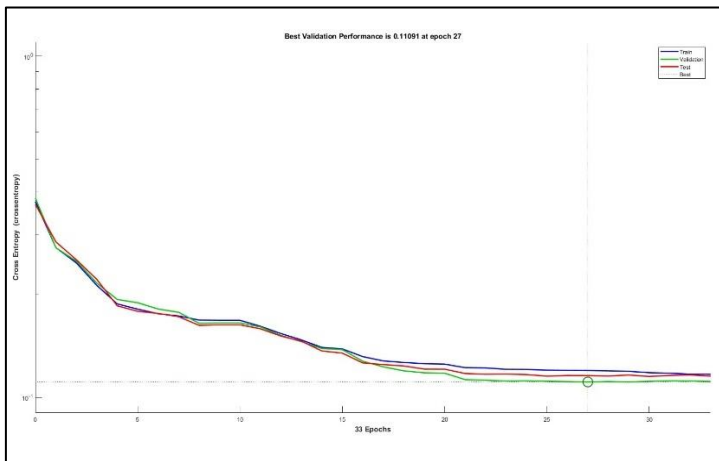
d. *Hidden Neuron 15*

Hasil akurasi *training* citra *body fluids* menggunakan model *Scaled Conjugate Gradient Backpropagation* (SCG) *hidden neuron 15* dapat dilihat pada Tabel 4.9.

Tabel 4. 9 Hasil Akurasi Training Menggunakan Model SCG *Hidden Neuron 15*

<i>Run</i>	<i>Epoch</i>		MLP-TrainSCG <i>Hidden Neuron 15</i>			<i>Elapsed Time</i>
	<i>All</i>	<i>Best</i>	<i>Acc Training</i>	<i>Acc Validation</i>	<i>Acc Testing</i>	
1	33	27	89,9%	91,8%	89,9%	00:00:02
2	25	19	90,0%	89,3%	89,5%	00:00:02
3	44	38	90,6%	89,3%	89,9%	00:00:02
4	90	84	93,2%	92,5%	94,5%	00:00:03
5	55	49	91,0%	89,7%	90,6%	00:00:02
6	91	85	92,6%	95,0%	94,1%	00:00:03
7	84	78	91,9%	93,1%	92,2%	00:00:03
8	31	25	90,3%	89,9%	89,9%	00:00:02
9	40	34	90,5%	91,8%	89,9%	00:00:02
10	104	98	93,5%	93,9%	94,1%	00:00:03
Average			91,4%	91,6%	91,5%	00:00:02
Standar Deviasi			0,0134763	0,0202926	0,02056	

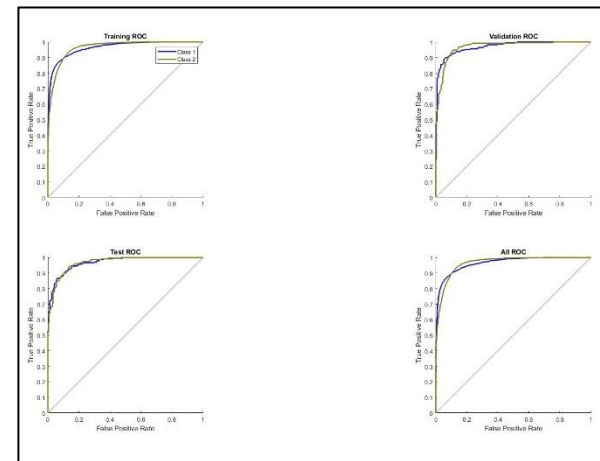
Berdasarkan Tabel 4.9 diketahui bahwa rata-rata akurasi *training* menggunakan model SCG dengan *hidden neuron 15* adalah 89-93%. Hasil akurasi *training* terbaik terdapat pada *run 10* dengan akurasi *training* sebesar 93,5%, akurasi validasi 93,9%, dan akurasi *testing* 94,1%. Akurasi *training* terbaik ini diperoleh pada *epoch* ke 98 dari total 104 *epoch* dengan waktu *training* selama 3 detik. Sementara itu, hasil akurasi *training* terendah terdapat pada *run 1* dengan akurasi *training* 89,9%, akurasi validasi 91,8%, dan akurasi *testing* 89,9%. Akurasi *training* terendah ini diperoleh pada *epoch* ke 27 dari total 33 *epoch* dengan waktu *training* selama 2 detik. Dari pengujian diatas, diperoleh hasil bahwa rata-rata akurasi *training* adalah 91,4%, akurasi validasi 91,6%, dan akurasi *testing* 91,5%. Rata-rata waktu *training* menggunakan model SCG *hidden neuron 15* adalah 2 detik. Visualisasi hasil *training* dapat dilihat pada Gambar 4.6.



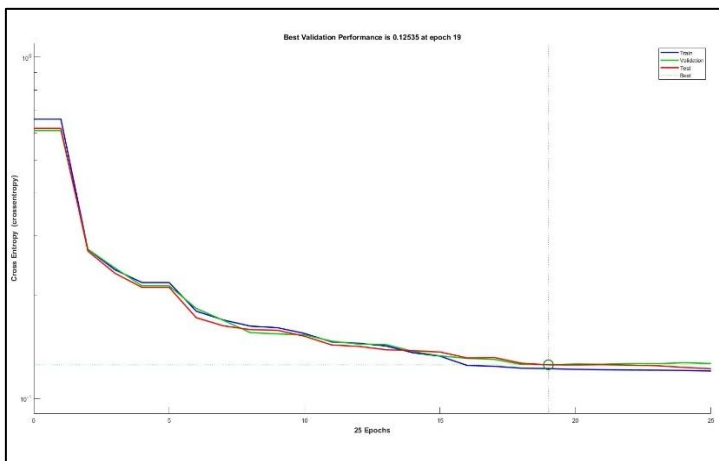
(a1)



(a2)



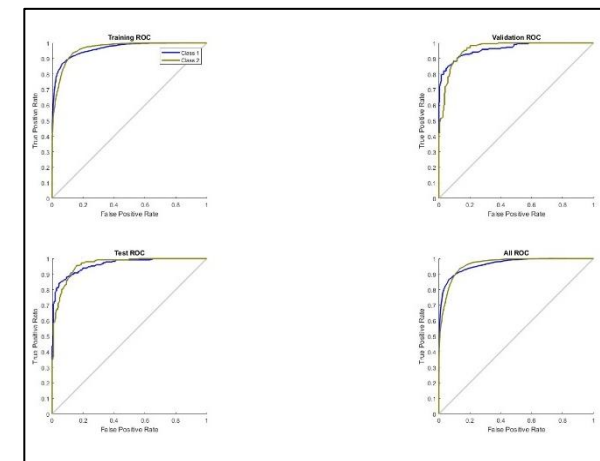
(a3)



(b1)

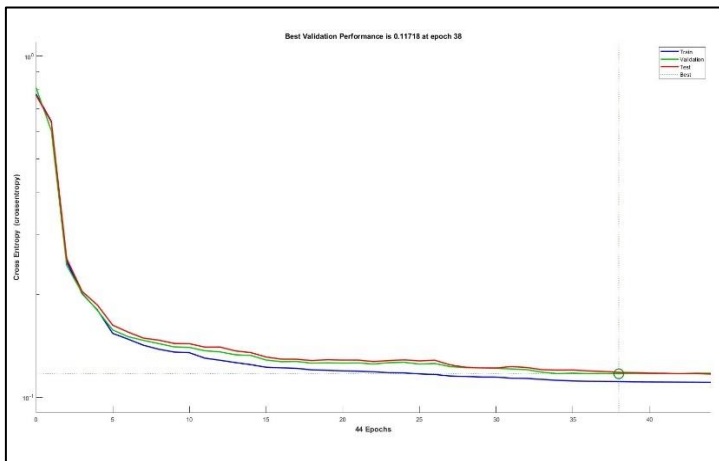


(b2)



(b3)

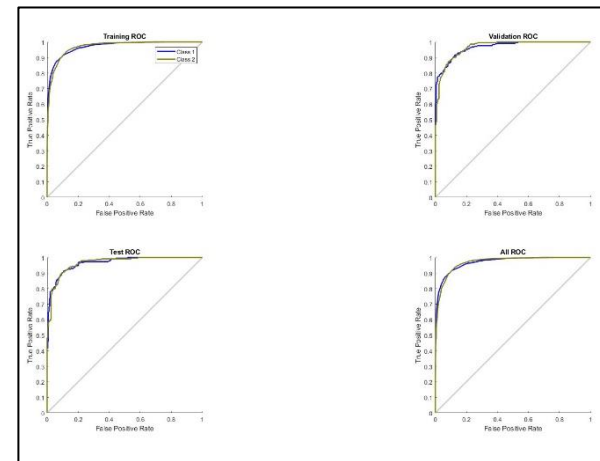
Gambar 4. 6 (a1) (b1) Performance Run 1 dan 2 Training SCG HN 15, (a2) (b2) Confusion Matrix Run 1 dan 2 Training SCG HN 15, (a3) (b3) ROC Run 1 dan 2 Training SCG HN 15



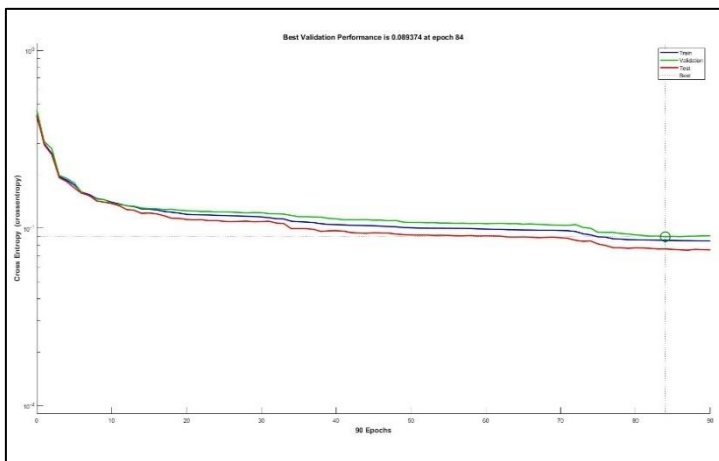
(c1)



(c2)



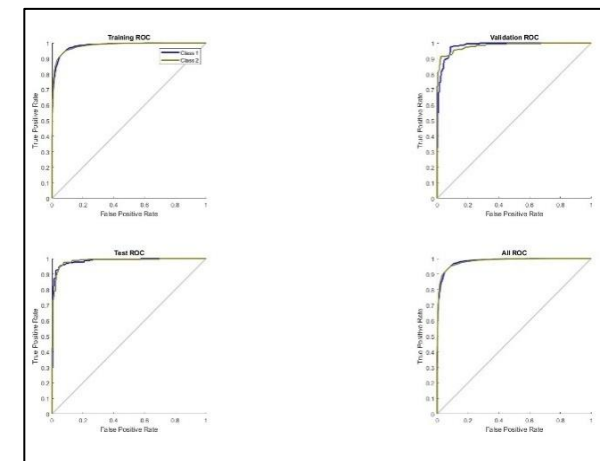
(c3)



(d1)

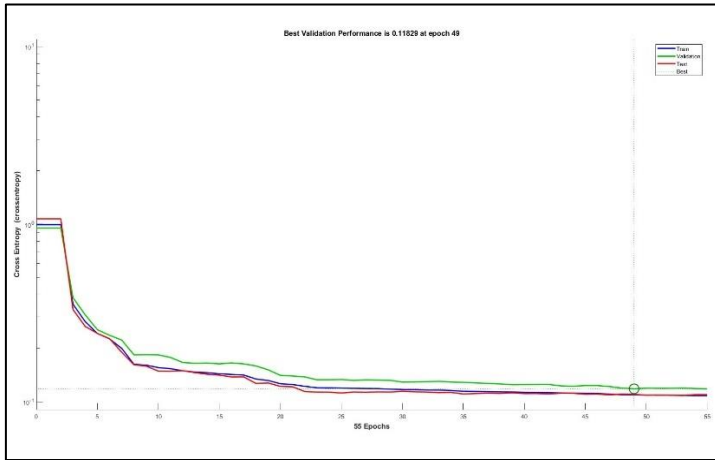


(d2)



(d3)

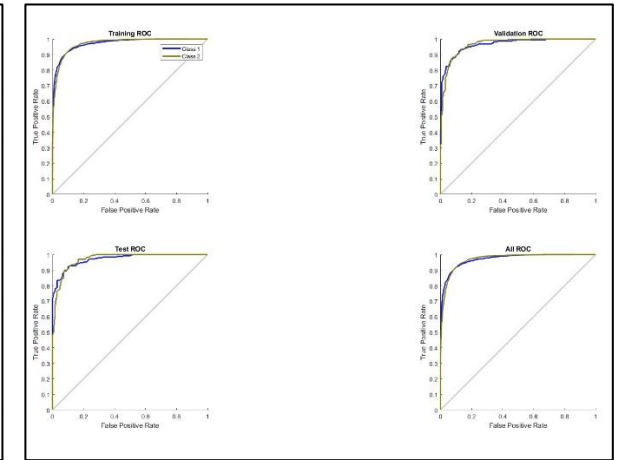
Gambar 4.6 (c1) (d1) Performance Run 3 dan 4 Training SCG HN 15, (c2) (d2) Confusion Matrix Run 3 dan 4 Training SCG HN 15, (c3) (d3) ROC Run 3 dan 4 Training SCG HN 15



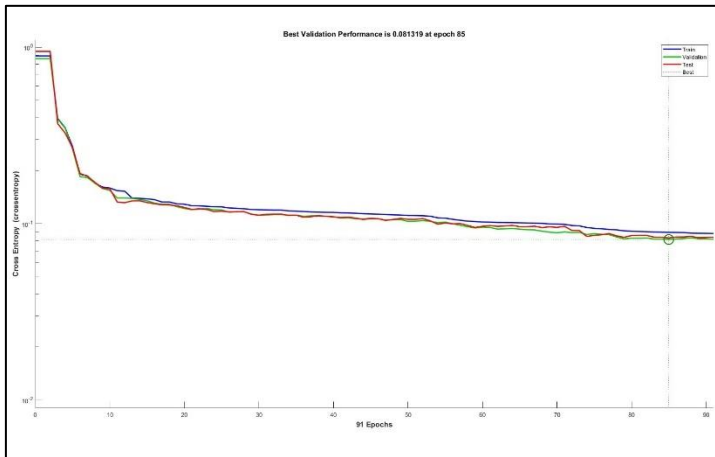
(e1)



(e2)



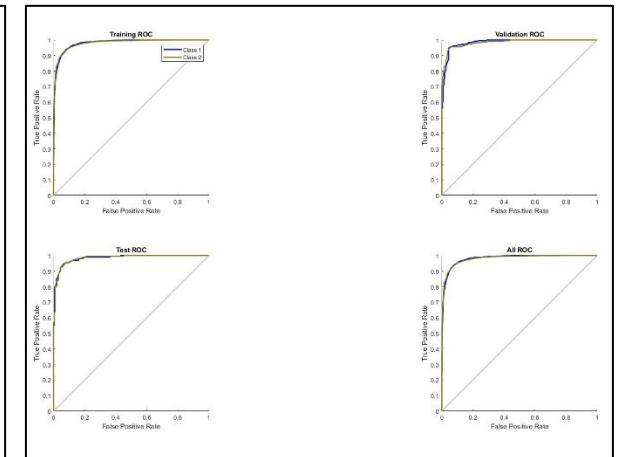
(e3)



(f1)

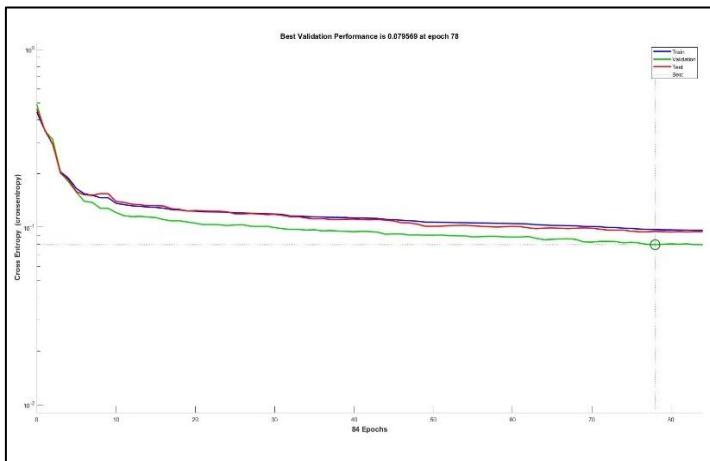


(f2)



(f3)

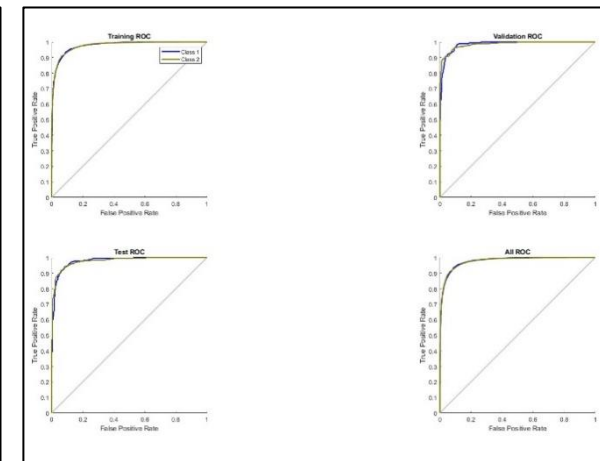
Gambar 4.6 (e1) (f1) Performance Run 5 dan 6 Training SCG HN 15, (e2) (f2) Confusion Matrix Run 5 dan 6 Training SCG HN 15, (e3) (f3) ROC Run 5 dan 6 Training SCG HN 15



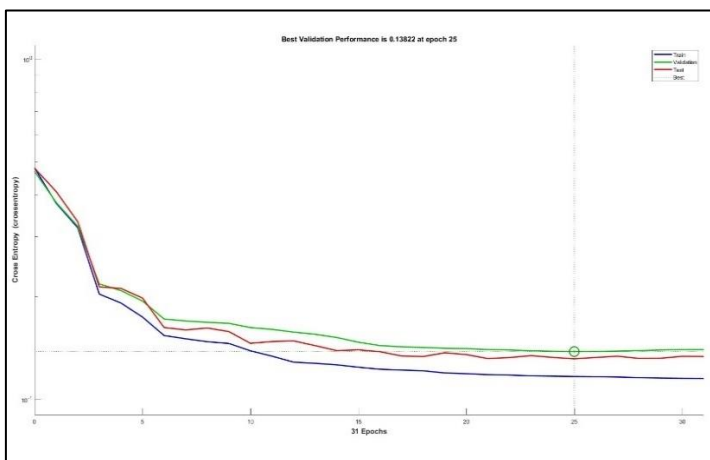
(g1)



(g2)



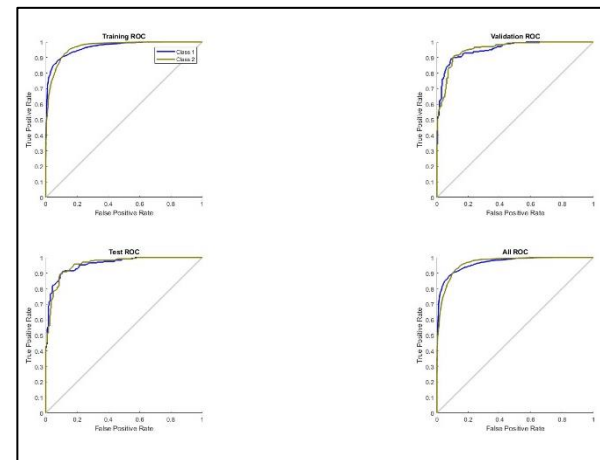
(g3)



(h1)

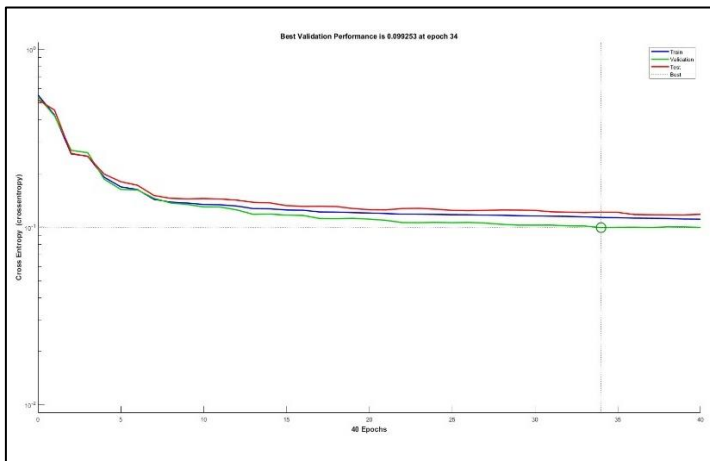


(h2)



(h3)

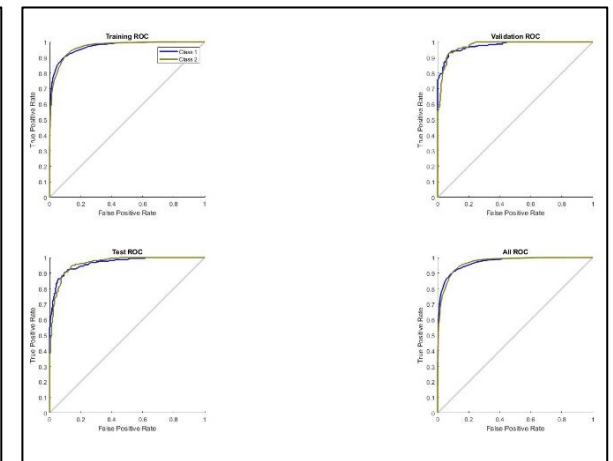
Gambar 4.6 (g1) (h1) Performance Run 7 dan 8 Training SCG HN 15, (g2) (h2) Confusion Matrix Run 7 dan 8 Training SCG HN 15, (g3) (h3) ROC Run 7 dan 8 Training SCG HN 15



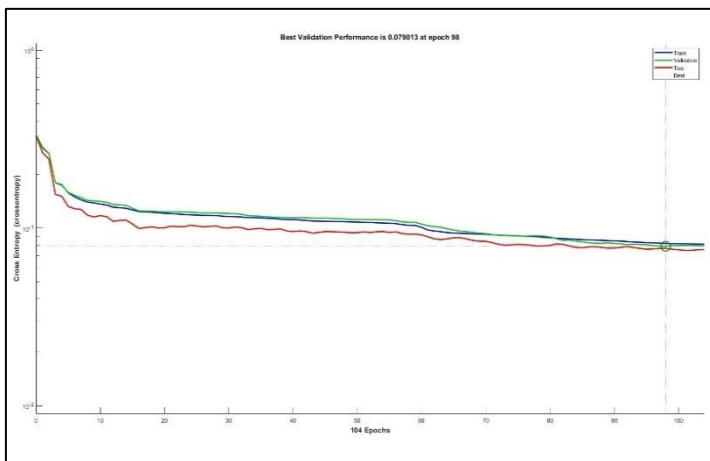
(i1)



(i2)



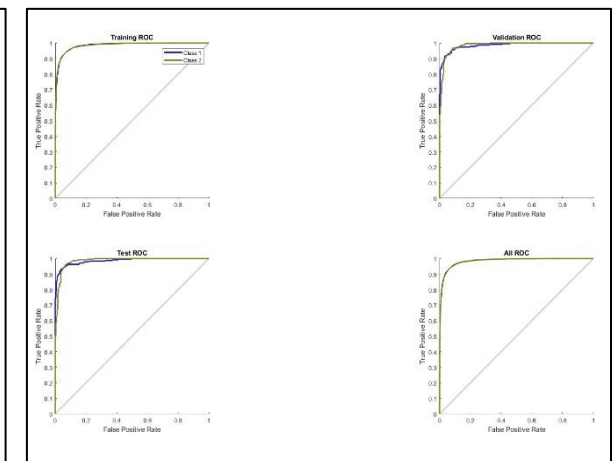
(i3)



(j1)



(j2)



(j3)

Gambar 4.6 (i1) (j1) Performance Run 9 dan 10 Training SCG HN 15, (i2) (j2) Confusion Matrix Run 9 dan 10 Training SCG HN 15, (i3) (j3) ROC Run 9 dan 10 Training SCG HN 15

Berdasarkan visualisasi hasil *training* di atas, dapat diketahui bahwa grafik *performance* terdiri dari 3 grafik, yaitu grafik biru merepresentasikan hasil *training*, grafik hijau merepresentasikan hasil validasi, dan grafik merah merepresentasikan hasil *testing*. Gambar 4.6 menampilkan sepuluh grafik *performane training SCG hidden neuron 15* dimana grafik *run 2, 3, 5, 6, dan run 8* mengalami penurunan sangat signifikan pada *range epoch 0-2* sedangkan hasil *run 1,7, dan 9* penurunannya landai. Pada *run 4 dan run 10*, grafik tidak mengalami penurunan signifikan dari *epoch* awal hingga *epoch* maksimalnya. Dari uraian di atas, dapat disimpulkan bahwa semakin rendah nilai entropi dari grafik (nilai error) maka semakin baik performa dari model.

Hasil *training* yang menunjukkan akurasi dari sistem ditunjukkan oleh *confusion matrix* dimana warna hijau merepresentasikan citra yang berhasil diklasifikasikan ke dalam kelas secara benar dan warna merah adalah citra yang gagal diklasifikasikan dalam kelasnya. Perbandingan data yang digunakan untuk *10-run cross validation* yaitu 90% dari total data (8.591) digunakan sebagai data *training*, 5% dari total data (477) digunakan sebagai data validasi, dan 5% lainnya (477) digunakan sebagai data *testing*. Dari *confusion matrix* tersebut, diketahui bahwa rata-rata akurasi *training*, validasi, dan *testing* lebih dari 90%. Nilai rata-rata *error training* adalah $\leq 10\%$ (859) dan rata-rata nilai *error testing* adalah $\leq 10\%$ (47). Hal ini menunjukkan bahwa model memiliki performa yang baik

Visualisasi grafik lainnya adalah grafik *Receiver Operating Characteristic* (ROC). Fungsi dari grafik ROC ini adalah untuk mengetahui hasil evaluasi performa *machine learning*. Pada grafik ROC, terdapat dua warna grafik yaitu biru yang menunjukkan klasifikasi citra kelas 1 dan grafik hijau menunjukkan klasifikasi citra kelas 2. Dari *run 1-10*, dapat diketahui bahwa kurva *training*, validasi, dan *testing* mendekati angka 1 dan semakin menjauhi garis *threshold*. Dengan demikian, dapat disimpulkan bahwa performa *machine learning* (*Multilayer Perceptron* model *SCG hidden neuron 15*) menunjukkan performa perfect.

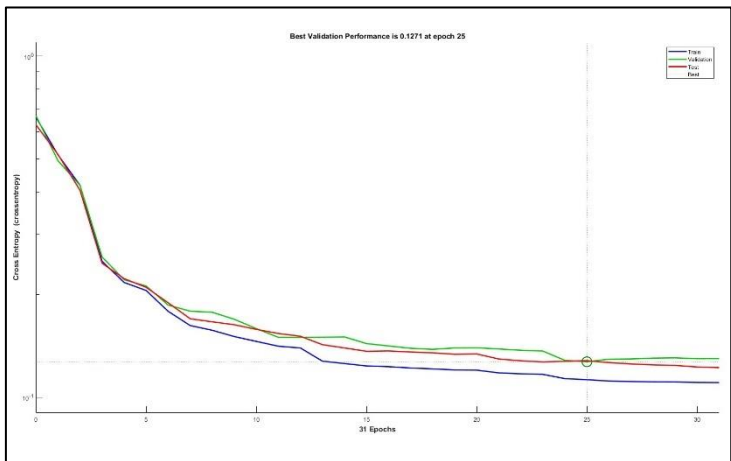
e. *Hidden Neuron 20*

Hasil akurasi *training* citra *body fluids* menggunakan model *Scaled Conjugate Gradient Backpropagation* (SCG) *hidden neuron 20* dapat dilihat pada Tabel 4.10.

Tabel 4. 10 Hasil Akurasi *Training* Menggunakan Model SCG *Hidden Neuron 20*

<i>Run</i>	<i>Epoch</i>		MLP-TrainSCG <i>Hidden Neuron 20</i>			<i>Elapsed Time</i>
	<i>All</i>	<i>Best</i>	<i>Acc Training</i>	<i>Acc Validation</i>	<i>Acc Testing</i>	
1	31	25	90,7%	88,7%	87,4%	00:00:02
2	143	137	94,9%	96,2%	92,5%	00:00:04
3	79	73	91,3%	92,5%	90,8%	00:00:03
4	107	101	93,6%	95,2%	94,8%	00:00:03
5	80	74	93,1%	92,7%	91,6%	00:00:03
6	56	50	91,7%	91,2%	90,4%	00:00:02
7	89	83	91,6%	91,6%	92,2%	00:00:03
8	147	141	94,7%	94,5%	93,7%	00:00:04
9	41	35	90,0%	89,7%	89,7%	00:00:02
10	77	71	92,4%	92,2%	92,5%	00:00:03
Average			92,4%	92,5%	91,6%	00:00:03
Standar Deviasi			0,0164857	0,0235384	0,021067	

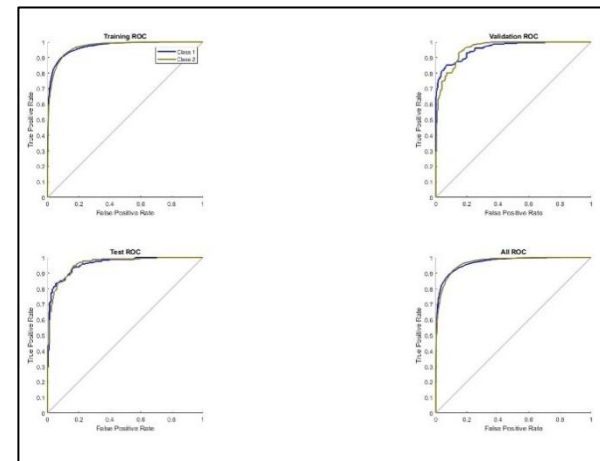
Berdasarkan Tabel 4.10 diketahui bahwa rata-rata akurasi *training* menggunakan model SCG dengan *hidden neuron 20* adalah 90-94%. Hasil akurasi *training* terbaik terdapat pada *run 2* dengan akurasi *training* sebesar 94,9%, akurasi validasi 96,2%, dan akurasi *testing* 92,5%. Akurasi *training* terbaik ini diperoleh pada *epoch* ke 137 dari total 143 *epoch* dengan waktu *training* selama 4 detik. Sementara itu, hasil akurasi *training* terendah terdapat pada *run 9* dengan akurasi *training* 90,0%, akurasi validasi 89,7%, dan akurasi *testing* 89,7%. Akurasi *training* terendah ini diperoleh pada *epoch* ke 35 dari total 41 *epoch* dengan waktu *training* selama 2 detik. Dari pengujian diatas, diperoleh hasil bahwa rata-rata akurasi *training* adalah 92,4%, akurasi validasi 92,5%, dan akurasi *testing* 91,6%. Rata-rata waktu *training* menggunakan model SCG *hidden neuron 20* adalah 3 detik. Visualisasi hasil *training* dapat dilihat pada Gambar 4.7.



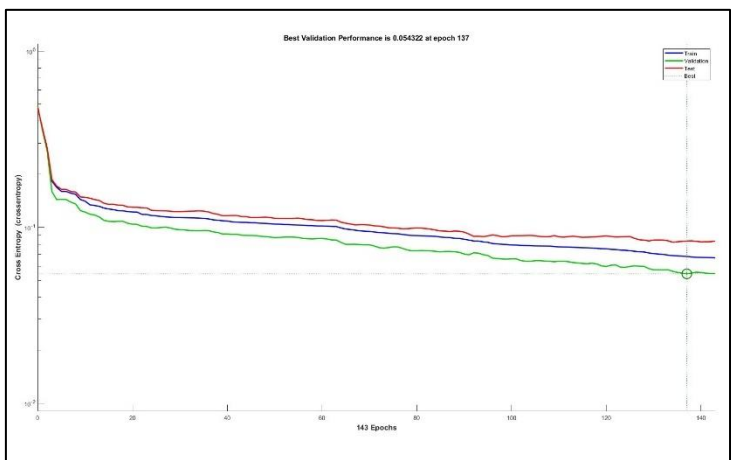
(a1)



(a2)



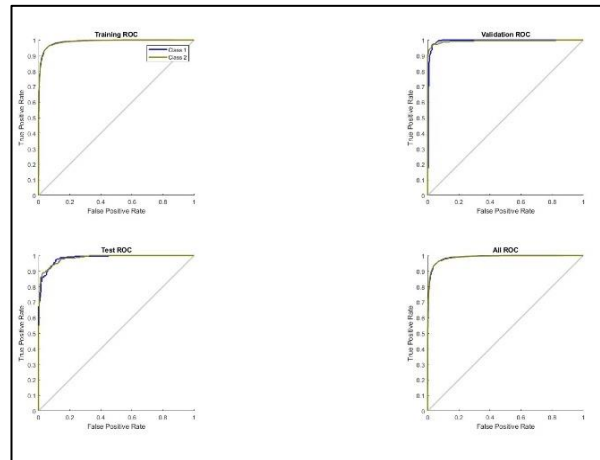
(a3)



(b1)

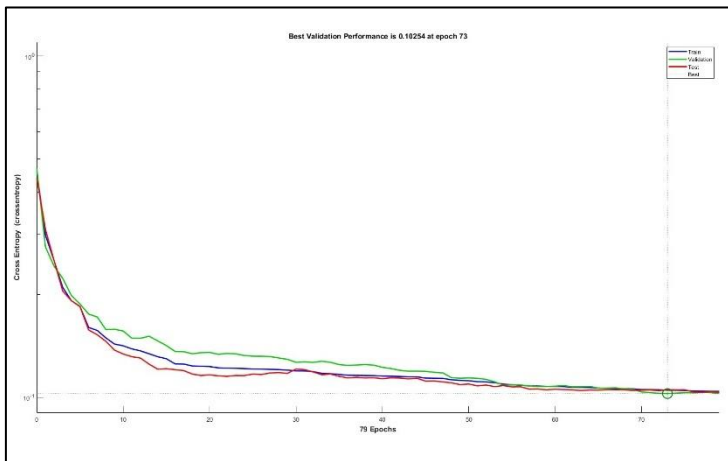


(b2)



(b3)

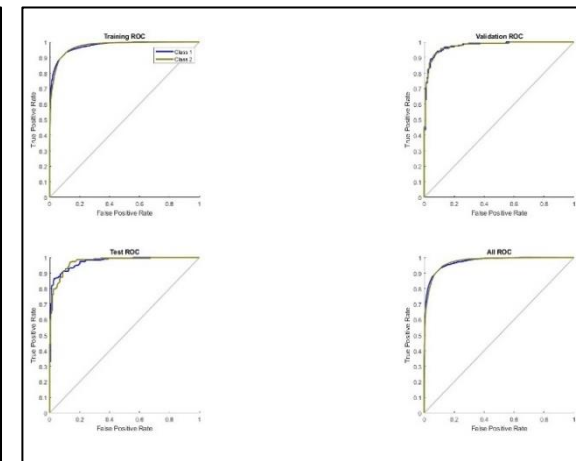
Gambar 4. 7 (a1) (b1) Performance Run 1 dan 2 Training SCG HN 20, (a2) (b2) Confusion Matrix Run 1 dan 2 Training SCG HN 20, (a3) (b3) ROC Run 1 dan 2 Training SCG HN 20



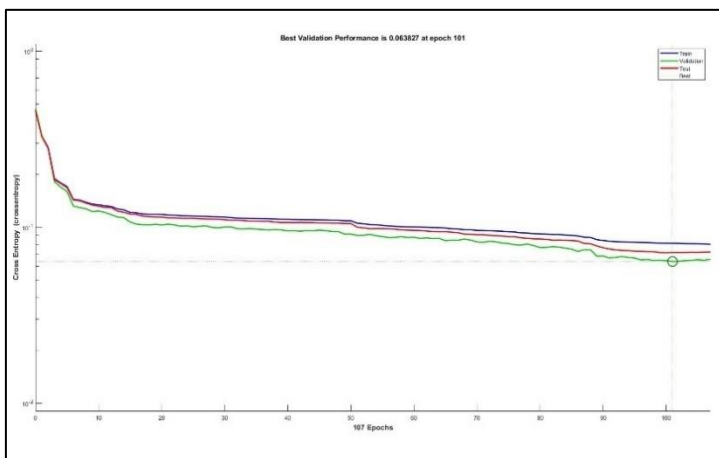
(c1)



(c2)



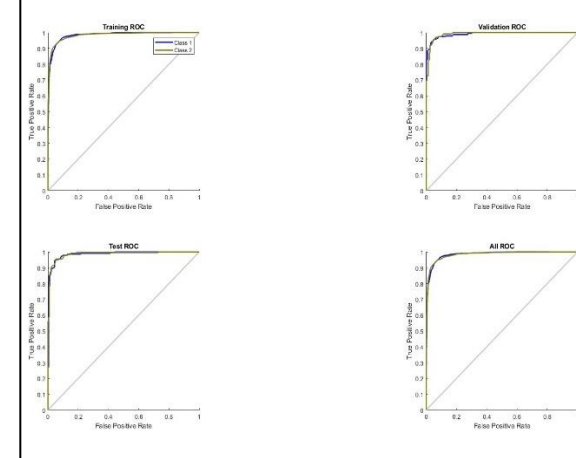
(c3)



(d1)

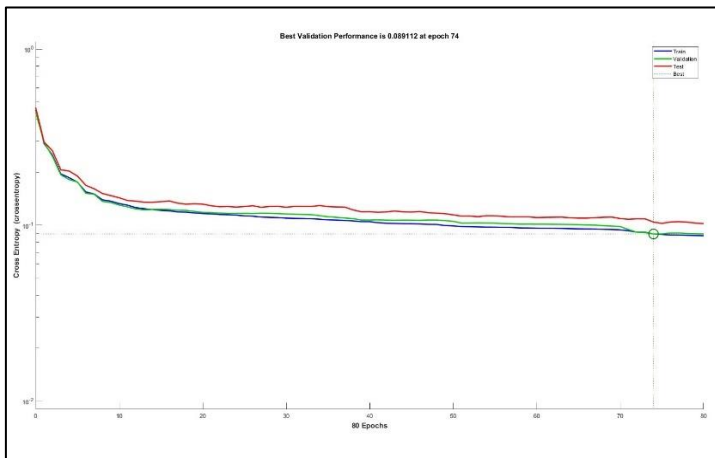


(d2)



(d3)

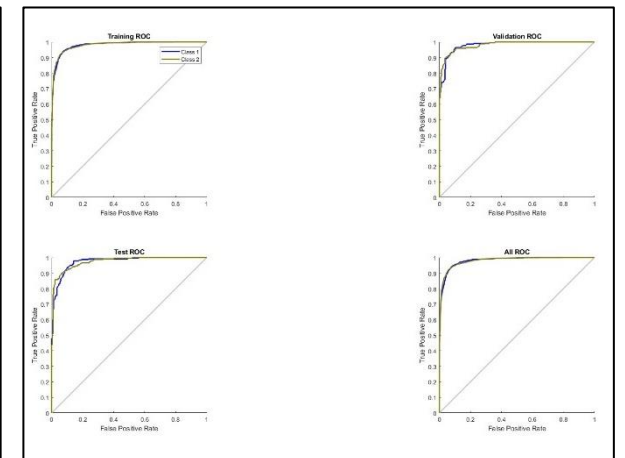
Gambar 4.7 (c1) (d1) Performance Run 3 dan 4 Training SCG HN 20, (c2) (d2) Confusion Matrix Run 3 dan 4 Training SCG HN 20, (c3) (d3) ROC Run 3 dan 4 Training SCG HN 20



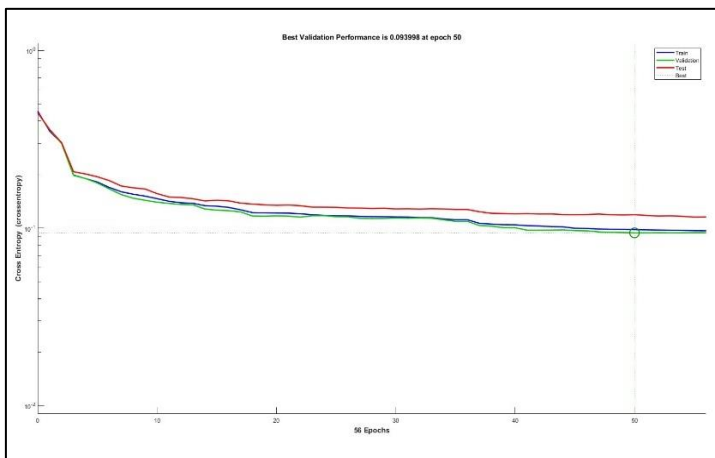
(e1)



(e2)



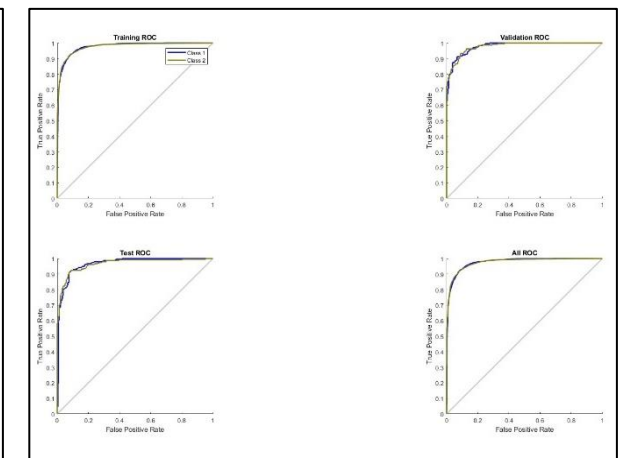
(e3)



(f1)

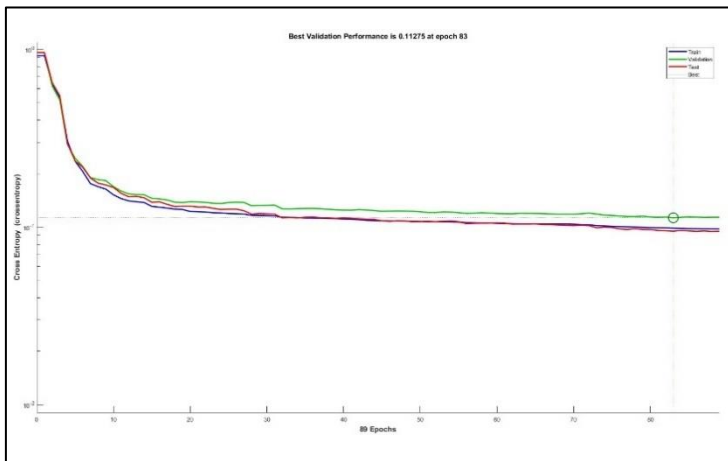


(f2)



(f3)

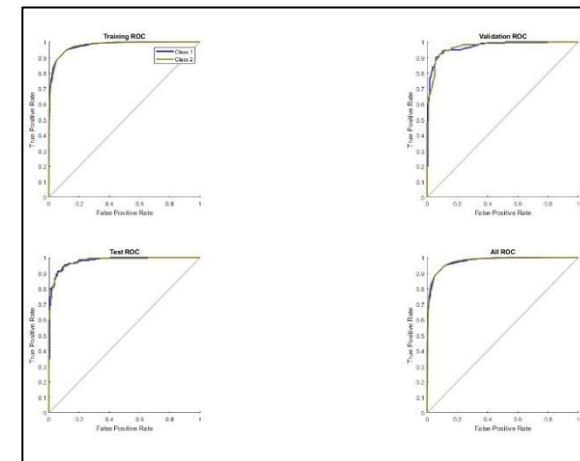
Gambar 4.7 (e1) (f1) Performance Run 5 dan 6 Training SCG HN 20, (e2) (f2) Confusion Matrix Run 5 dan 6 Training SCG HN 20, (e3) (f3) ROC Run 5 dan 6 Training SCG HN 20



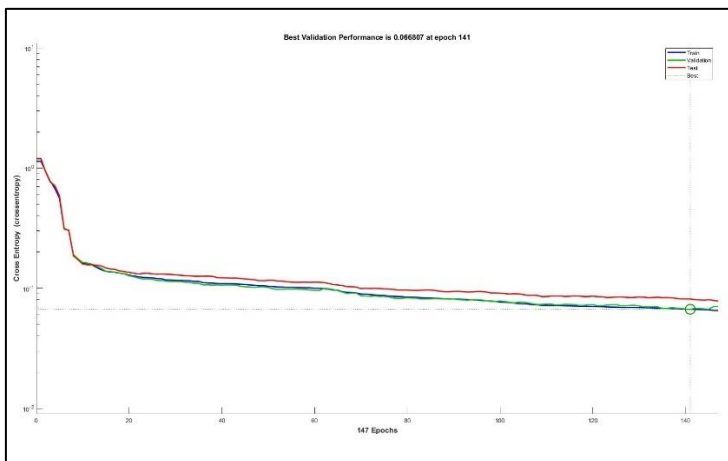
(g1)



(g2)



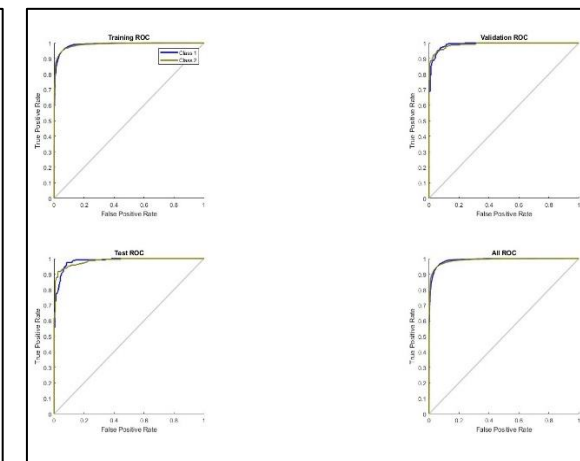
(g3)



(h1)

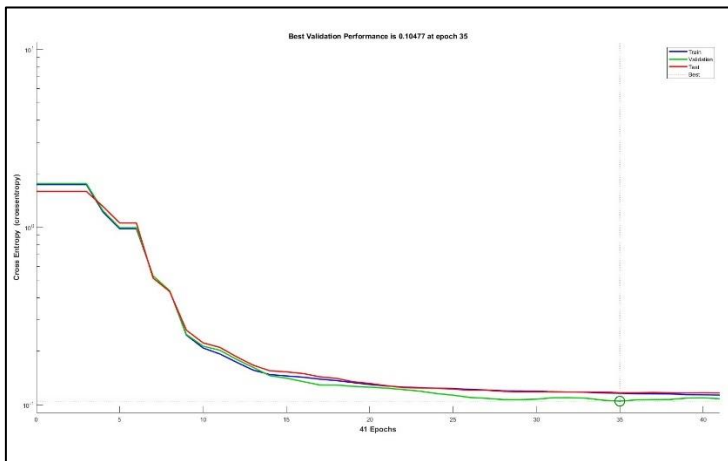


(h2)



(h3)

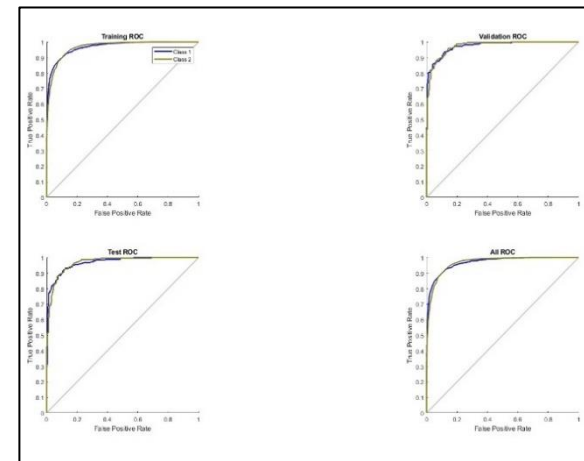
Gambar 4.7 (g1) (h1) Performance Run 7 dan 8 Training SCG HN 20, (g2) (h2) Confusion Matrix Run 7 dan 8 Training SCG HN 20, (g3) (h3) ROC Run 7 dan 8 Training SCG HN 20



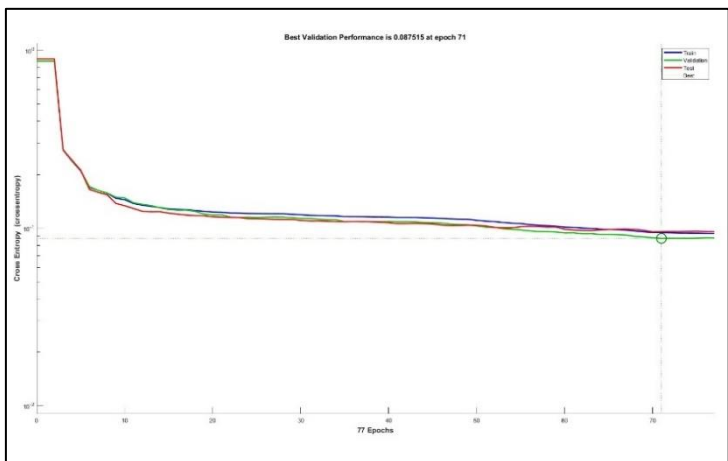
(i1)



(i2)



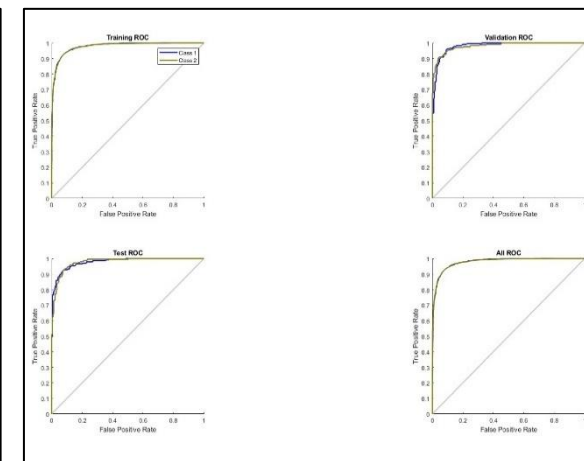
(i3)



(j1)



(j2)



(j3)

Gambar 4.7 (i1) (j1) Performance Run 9 dan 10 Training SCG HN 20, (i2) (j2) Confusion Matrix Run 9 dan 10 Training SCG HN 20, (i3) (j3) ROC Run 9 dan 10 Training SCG HN 20

Berdasarkan visualisasi hasil *training* di atas, dapat diketahui bahwa grafik *performance* terdiri dari 3 grafik, yaitu grafik biru merepresentasikan hasil *training*, grafik hijau merepresentasikan hasil validasi, dan grafik merah merepresentasikan hasil *testing*. Gambar 4.7 menampilkan sepuluh grafik *performane training SCG hidden neuron 20* dimana rata-rata grafik hasil run mengalami penurunan signifikan pada *epoch* 0-5, terutama pada *run* 7 dan 10. Untuk hasil *run* lainnya, grafik mulai mengalami penurunan landai pada *epoch* 5 sampai *epoch* maksimalnya. Dari uraian di atas, dapat disimpulkan bahwa semakin rendah nilai entropi dari grafik (nilai error) maka semakin baik performa dari model.

Hasil *training* yang menunjukkan akurasi dari sistem ditunjukkan oleh *confusion matrix* dimana warna hijau merepresentasikan citra yang berhasil diklasifikasikan ke dalam kelas secara benar dan warna merah adalah citra yang gagal diklasifikasikan dalam kelasnya. Perbandingan data yang digunakan untuk *10-run cross validation* yaitu 90% dari total data (8.591) digunakan sebagai data *training*, 5% dari total data (477) digunakan sebagai data validasi, dan 5% lainnya (477) digunakan sebagai data *testing*. Dari *confusion matrix* tersebut, diketahui bahwa akurasi *training* $\geq 90\%$ dan akurasi validasi dan *testing* $\geq 87\%$. Nilai rata-rata *error training* adalah $\leq 10\%$ (859) dan rata-rata nilai *error testing* adalah $\leq 10\%$ (47). Hal ini menunjukkan bahwa model memiliki performa yang baik.

Visualisasi grafik lainnya adalah grafik *Receiver Operating Characteristic* (ROC). Fungsi dari grafik ROC ini adalah untuk mengetahui hasil evaluasi performa *machine learning*. Pada grafik ROC, terdapat dua warna grafik yaitu biru yang menunjukkan klasifikasi citra kelas 1 dan grafik hijau menunjukkan klasifikasi citra kelas 2. Pada *run* 1, kurva validasi dan *testing* berada pada angka 0,8 sedangkan *run* 2-10, dapat diketahui bahwa kurva *training*, validasi, dan *testing* mendekati angka 1 dan semakin menjauhi garis *threshold*. Dengan demikian, dapat disimpulkan bahwa performa *machine learning* (*Multilayer Perceptron* model *SCG hidden neuron 20*) menunjukkan performa perfect.

4.3.1.2 One Step Secant Backpropagation (OSS)

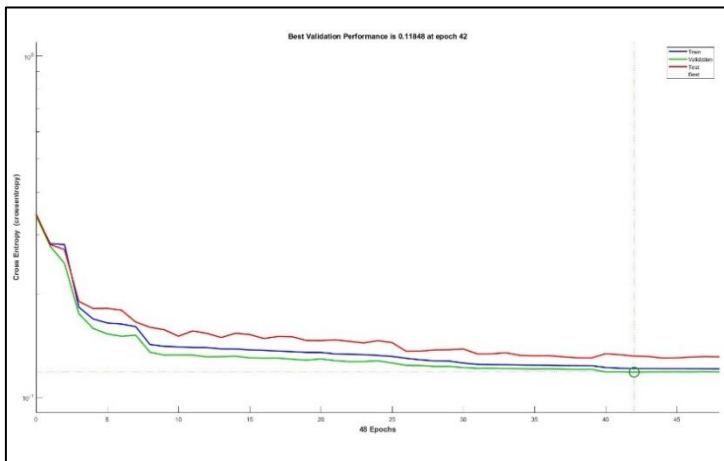
a. Hidden Neuron 1

Tabel 4.11 merupakan hasil akurasi *training* citra *body fluids* menggunakan model *One Step Secant Backpropagation* (OSS) *hidden neuron 1*.

Tabel 4. 11 Hasil Akurasi *Training* Menggunakan Model OSS *Hidden Neuron 1*

<i>Run</i>	<i>Epoch</i>		MLP-TrainOSS Hidden Neuron 1			<i>Elapsed Time</i>
	<i>All</i>	<i>Best</i>	<i>Acc Training</i>	<i>Acc Validation</i>	<i>Acc Testing</i>	
1	48	42	90,4%	91,6%	88,9%	00:00:02
2	23	17	89,7%	91,4%	91,8%	00:00:02
3	33	27	89,3%	89,7%	90,6%	00:00:02
4	41	35	90,2%	88,7%	87,6%	00:00:02
5	10	4	48,5%	52,0%	49,1%	00:00:02
6	58	52	90,3%	89,3%	91,4%	00:00:02
7	32	26	90,1%	87,4%	89,3%	00:00:02
8	38	32	90,2%	89,7%	90,4%	00:00:02
9	59	53	90,1%	91,2%	92,0%	00:00:03
10	49	43	90,3%	88,9%	91,2%	00:00:02
Average			85,9%	86,0%	86,2%	00:00:02
Standar Deviasi			0,1314868	0,1201614	0,1312039	

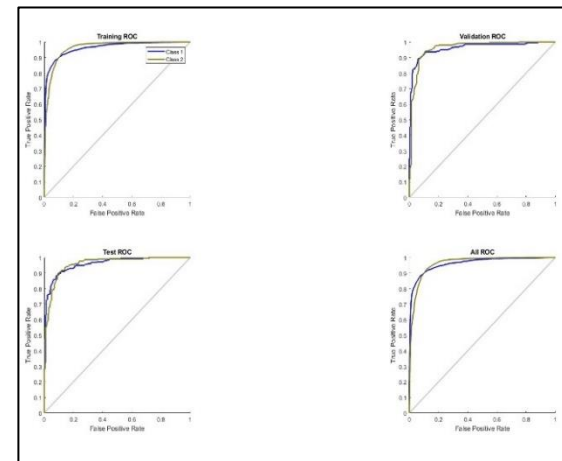
Berdasarkan Tabel 4.11 diketahui bahwa rata-rata akurasi *training* menggunakan model OSS dengan *hidden neuron 1* adalah 48-90%. Hasil akurasi *training* terbaik terdapat pada *run 1* dengan akurasi *training* sebesar 90,4%, akurasi validasi 91,6%, dan akurasi *testing* 88,9%. Akurasi *training* terbaik ini diperoleh pada *epoch* ke 42 dari total 48 *epoch* dengan waktu *training* selama 2 detik. Sementara itu, hasil akurasi *training* terendah terdapat pada *run 5* dengan akurasi *training* 48,5%, akurasi validasi 52,0%, dan akurasi *testing* 49,1%. Dari pengujian diatas, diperoleh hasil bahwa rata-rata akurasi *training* adalah 85,9%, akurasi validasi 86,0%, dan akurasi *testing* 86,2%. Rata-rata waktu *training* menggunakan model OSS *hidden neuron 1* adalah 2 detik. Visualisasi hasil *training* dapat dilihat pada Gambar 4.8.



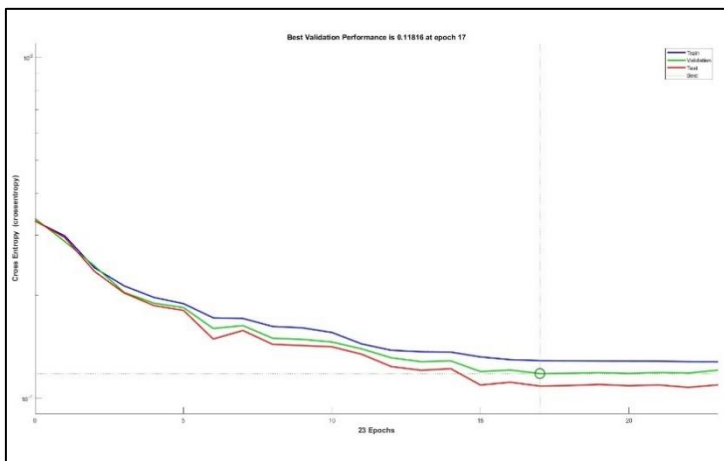
(a1)



(a2)



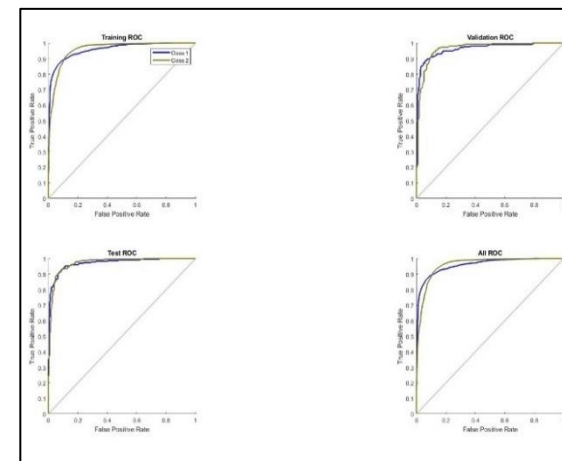
(a3)



(b1)

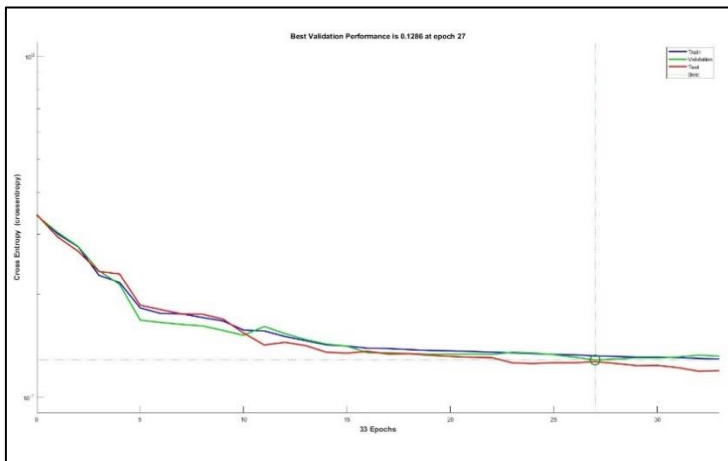


(b2)



(b3)

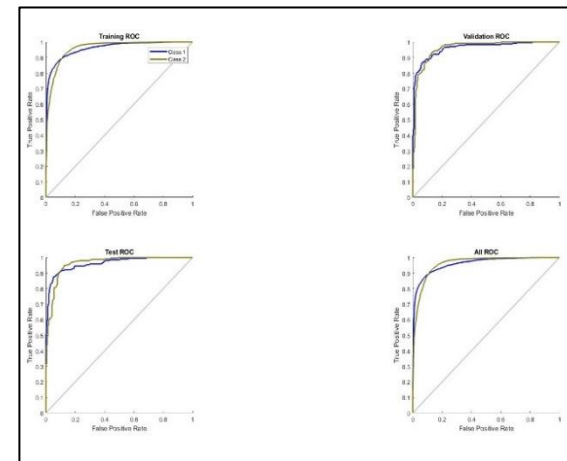
Gambar 4. 8 (a1) (b1) Performance Run 1 dan 2 Training OSS HN 1, (a2) b2) Confusion Matrix Run 1 dan 2 Training OSS HN 1, (a3) (b3) ROC Run 1 dan 2 Training OSS HN 1



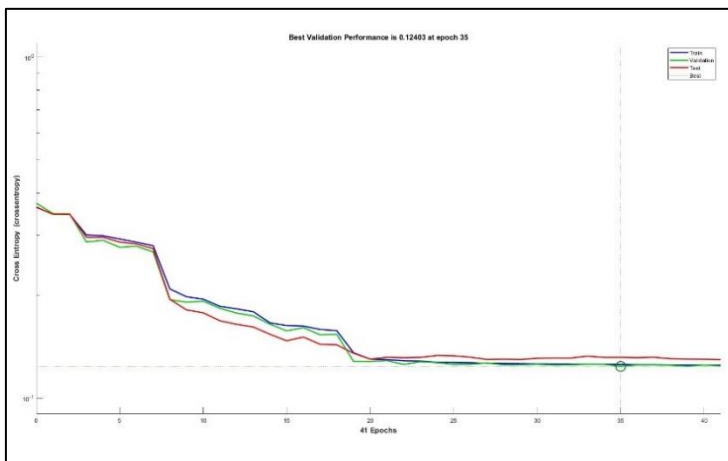
(c1)



(c2)



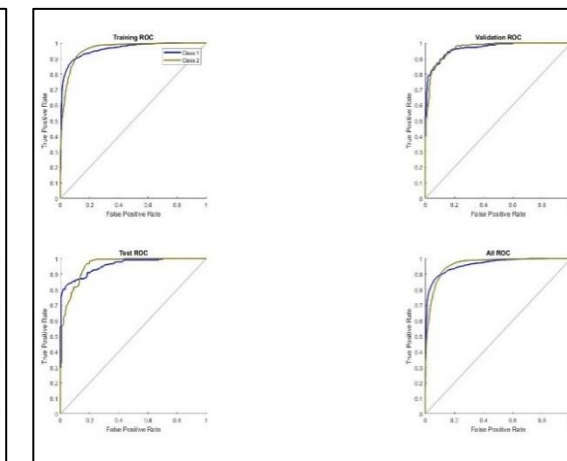
(c3)



(d1)

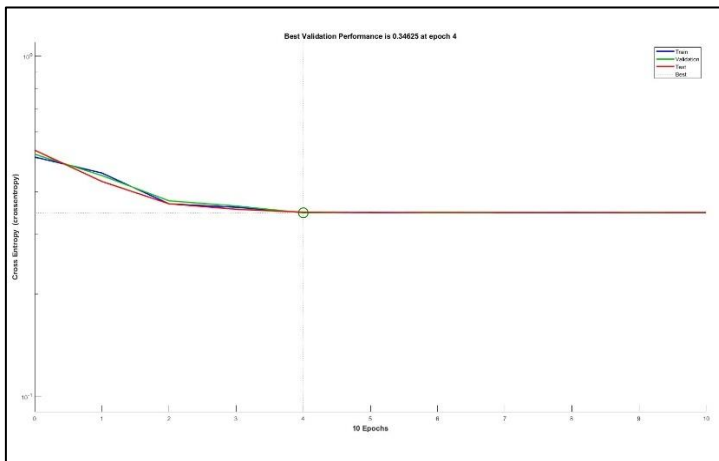


(d2)



(d3)

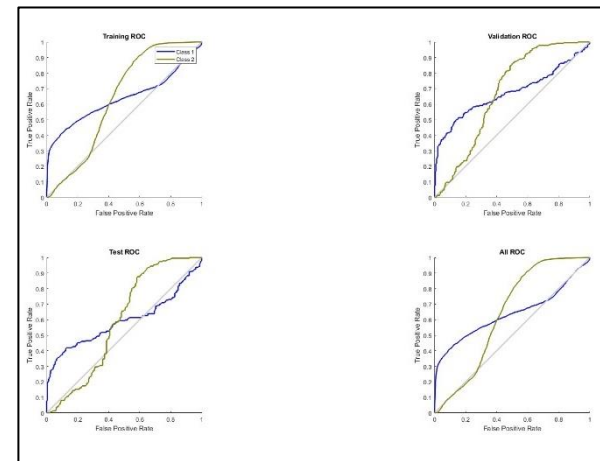
Gambar 4.8 (c1) (d1) Performance Run 3 dan 4 Training OSS HN 1, (c2) (d2) Confusion Matrix Run 3 dan 4 Training OSS HN 1, (c3) (d3) ROC Run 3 dan 4 Training OSS HN 1



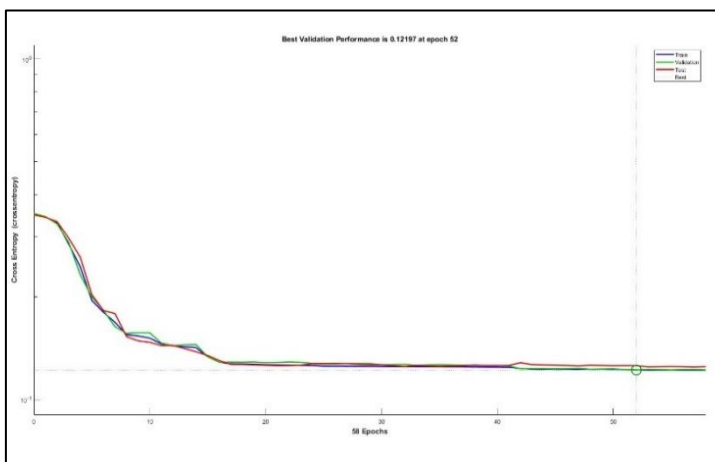
(e1)



(e2)



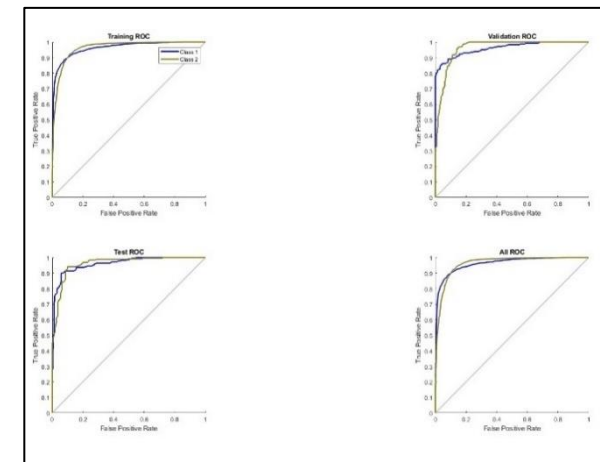
(e3)



(f1)

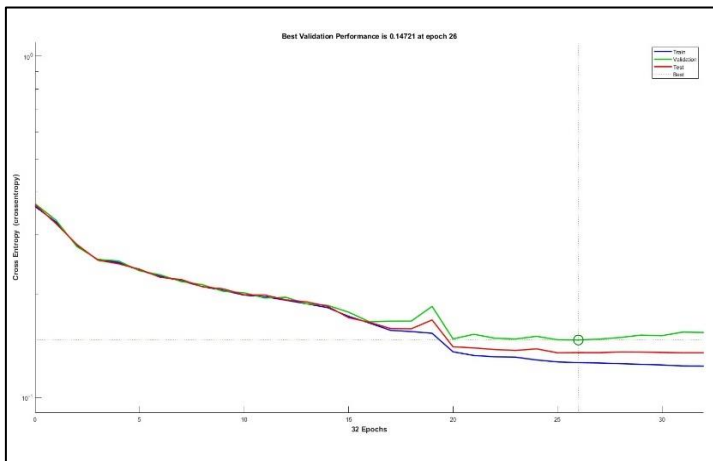


(f2)



(f3)

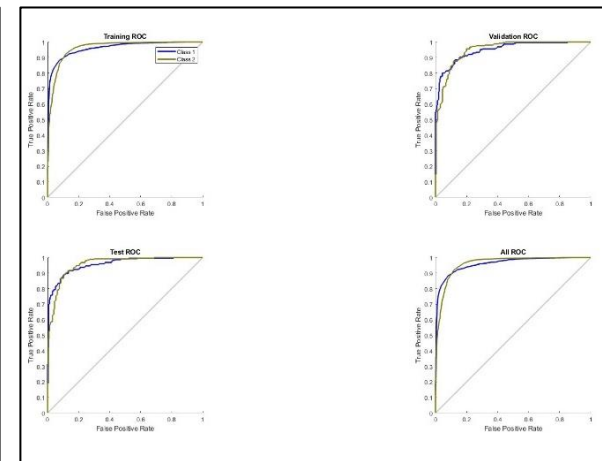
Gambar 4.8 (e1) (f1) Performance Run 5 dan 6 Training OSS HN 1, (e2) (f2) Confusion Matrix Run 5 dan 6 Training OSS HN 1, (e3) (f3) ROC Run 5 dan 6 Training OSS HN 1



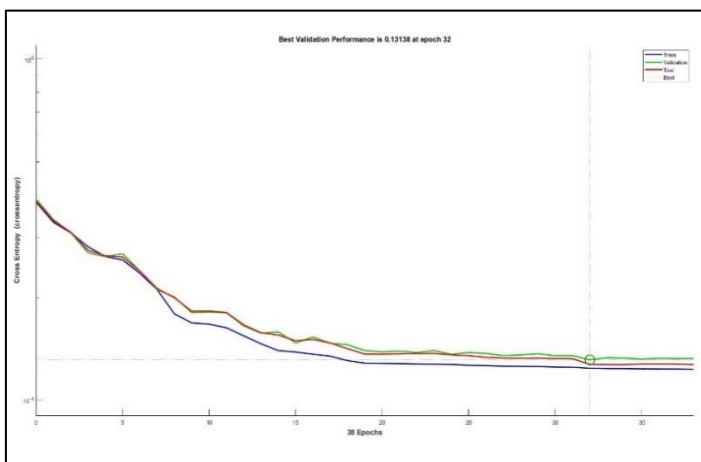
(g1)



(g2)



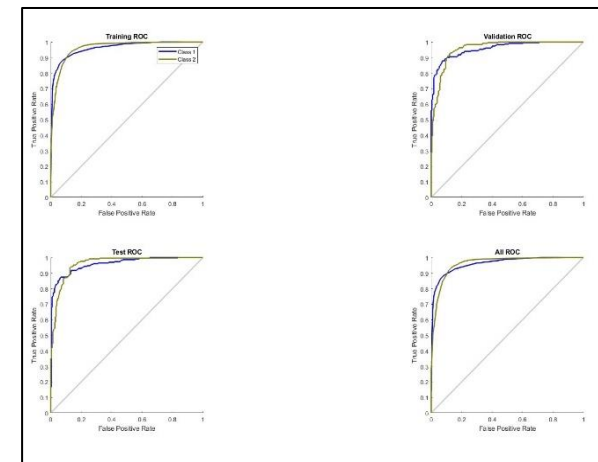
(g3)



(h1)

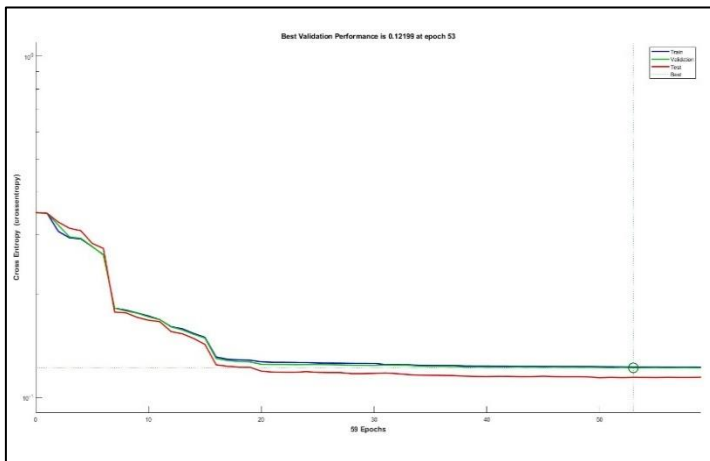


(h2)



(h3)

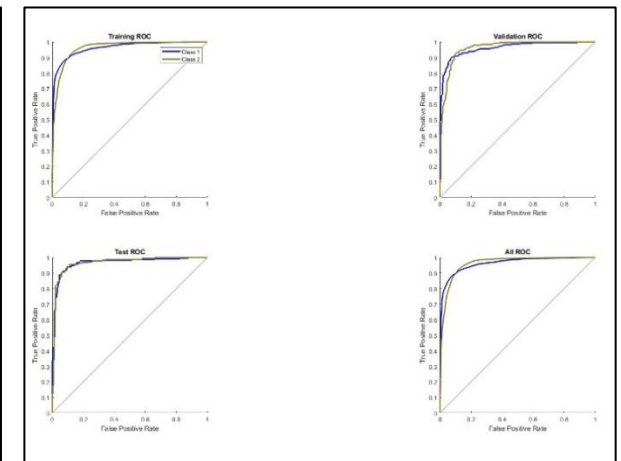
Gambar 4.8 (g1) (h1) Performance Run 7 dan 8 Training OSS HN 1, (g2) (h2) Confusion Matrix Run 7 dan 8 Training OSS HN 1, (g3) (h3) ROC Run 7 dan b Training OSS HN 1



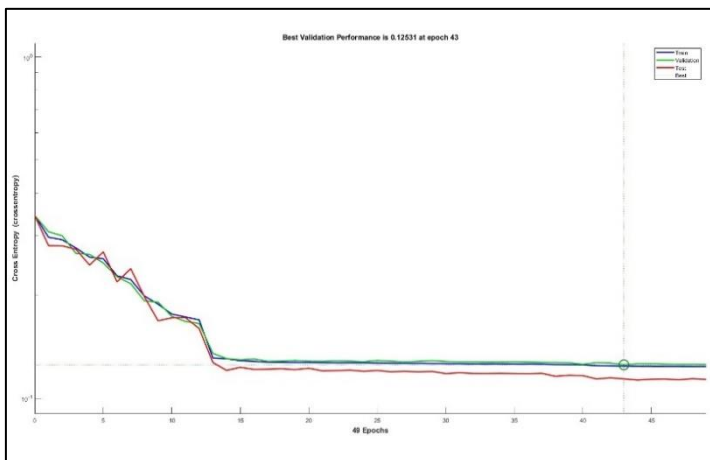
(i1)



(i2)



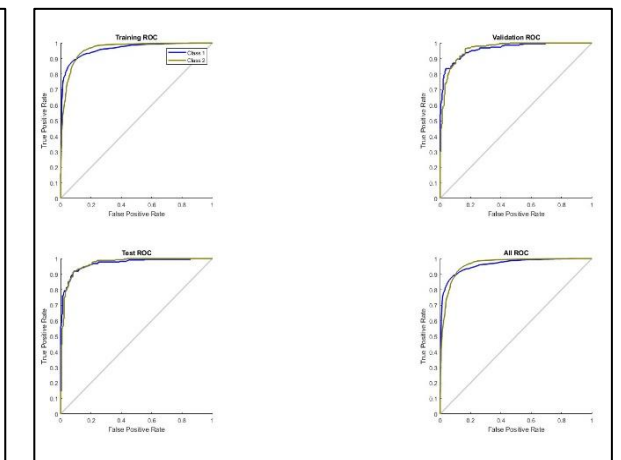
(i3)



(j1)



(j2)



(j3)

Gambar 4.8 (i1) (j1) Performance Run 9 dan 10 Training OSS HN 1, (i2) (j2) Confusion Matrix Run 9 dan 10 Training OSS HN 1, (i3) (j3) ROC Run 9 dan 10 Training OSS HN 1

Berdasarkan visualisasi hasil *training* di atas, dapat diketahui bahwa grafik *performance* terdiri dari 3 grafik, yaitu grafik biru merepresentasikan hasil *training*, grafik hijau merepresentasikan hasil validasi, dan grafik merah merepresentasikan hasil *testing*. Gambar 4.8 menampilkan sepuluh grafik *performane training OSS hidden neuron 1* dimana grafik *run 1* sampai 3 mengalami penurunan signifikan pada *range epoch 0-5* sedangkan *run 4, 6, 8* sampai 10 mengalami penurunan signifikan pada *range epoch 0-10*. Penurunan signifikan lainnya terjadi pada *range epoch 0-2* yaitu *run 2* dan *range epoch 0-20* pada *run 7*. Dari uraian di atas, dapat disimpulkan bahwa semakin rendah nilai entropi dari grafik (nilai error) maka semakin baik performa dari model.

Hasil *training* yang menunjukkan akurasi dari sistem ditunjukkan oleh *confusion matrix* dimana warna hijau merepresentasikan citra yang berhasil diklasifikasikan ke dalam kelas secara benar dan warna merah adalah citra yang gagal diklasifikasikan dalam kelasnya. Perbandingan data yang digunakan untuk *10-run cross validation* yaitu 90% dari total data (8.591) digunakan sebagai data *training*, 5% dari total data (477) digunakan sebagai data validasi, dan 5% lainnya (477) digunakan sebagai data *testing*. Dari *confusion matrix* tersebut, diketahui bahwa akurasi *training*, validasi, dan *testing* $\geq 89\%$ kecuali pada *run 5*. Hasil akurasi *testing run 5* adalah 48,5%. Nilai rata-rata *error training* adalah $\leq 10\%$ (859) dan rata-rata nilai *error testing* adalah $\leq 10\%$ (47) kecuali *run 5*. Meskipun demikian, model tetap memiliki performa yang baik.

Visualisasi grafik lainnya adalah grafik *Receiver Operating Characteristic (ROC)*. Fungsi dari grafik ROC ini adalah untuk mengetahui hasil evaluasi performa *machine learning*. Pada grafik ROC, terdapat dua warna grafik yaitu biru yang menunjukkan klasifikasi citra kelas 1 dan grafik hijau menunjukkan klasifikasi citra kelas 2. Kurva validasi dan *testing* berada pada angka 0,9 mendekati 1 kecuali pada *run 5* yang semakin mendekati garis *threshold*. Meskipun demikian, dapat disimpulkan bahwa performa *machine learning (Multilayer Perceptron model OSS hidden neuron 1)* menunjukkan hasil performa cukup baik.

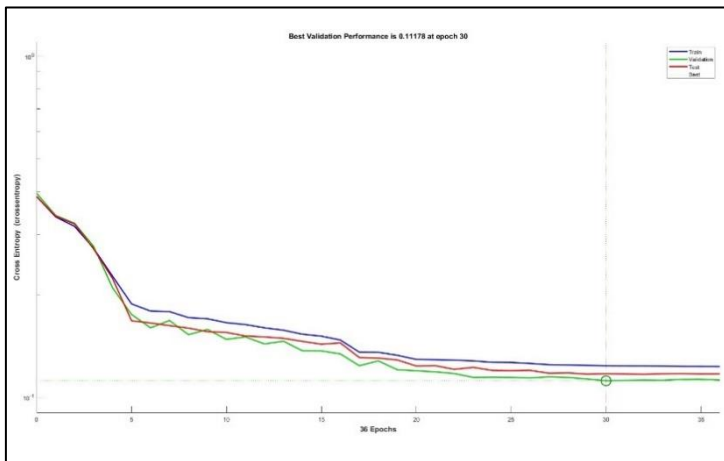
b. *Hidden Neuron 5*

Tabel 4.12 merupakan hasil akurasi *training* citra *body fluids* menggunakan model *One Step Secant Backpropagation* (OSS) *hidden neuron 5*

Tabel 4. 12 Hasil Akurasi *Training* Menggunakan Model OSS *Hidden Neuron 5*

<i>Run</i>	<i>Epoch</i>		MLP-TrainOSS Hidden Neuron 5			<i>Elapsed Time</i>
	<i>All</i>	<i>Best</i>	<i>Acc Training</i>	<i>Acc Validation</i>	<i>Acc Testing</i>	
1	36	30	89,8%	91,6%	89,1%	00:00:02
2	85	79	90,5%	91,6%	88,9%	00:00:03
3	74	68	90,3%	92,7%	89,9%	00:00:03
4	71	65	91,1%	93,3%	91,4%	00:00:03
5	33	27	90,1%	89,1%	89,7%	00:00:03
6	47	41	90,5%	90,6%	89,9%	00:00:03
7	34	28	90,1%	89,9%	89,7%	00:00:03
8	109	103	91,7%	89,9%	92,2%	00:00:04
9	31	25	89,6%	91,8%	90,6%	00:00:02
10	38	32	90,2%	89,9%	88,9%	00:00:03
Average			90,4%	91,0%	90,0%	00:00:03
Standar Deviasi			0,0061725	0,0137129	0,0108428	

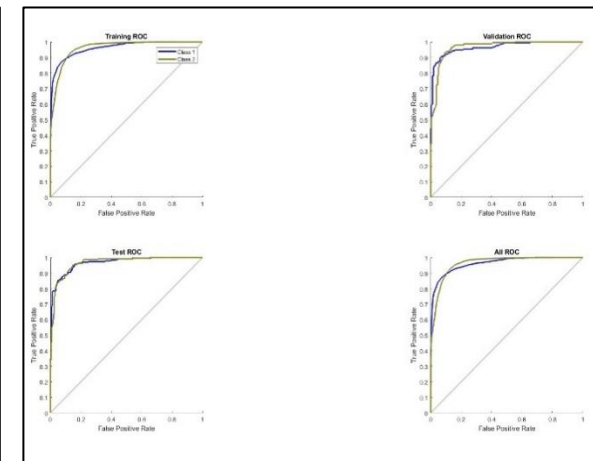
Berdasarkan Tabel 4.12 diketahui bahwa rata-rata akurasi *training* menggunakan model OSS dengan *hidden neuron 5* adalah 89-91%. Hasil akurasi *training* terbaik terdapat pada *run 8* dengan akurasi *training* sebesar 91,7%, akurasi validasi 89,9%, dan akurasi *testing* 92,2%. Akurasi *training* terbaik ini diperoleh pada *epoch* ke 103 dari total 109 *epoch* dengan waktu *training* selama 4 detik. Sementara itu, hasil akurasi *training* terendah terdapat pada *run 9* dengan akurasi *training* 89,6%, akurasi validasi 91,8%, dan akurasi *testing* 90,6%. Dari pengujian diatas, diperoleh hasil bahwa rata-rata akurasi *training* adalah 90,4%, akurasi validasi 91,0%, dan akurasi *testing* 90,0%. Rata-rata waktu *training* menggunakan model OSS *hidden neuron 5* adalah 3 detik. Visualisasi hasil *training* dapat dilihat pada Gambar 4.9.



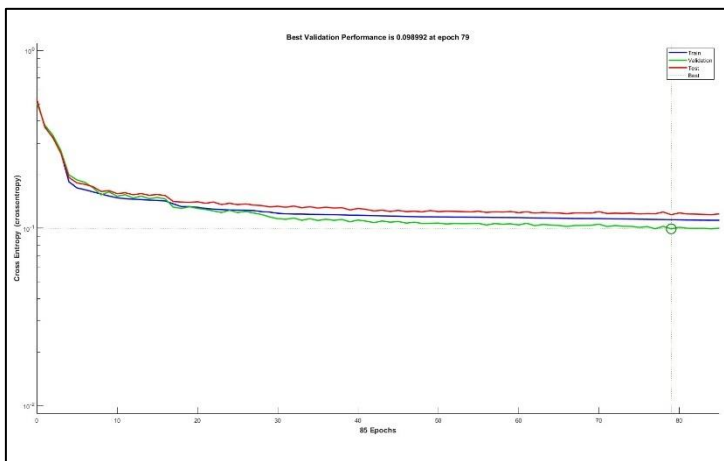
(a1)



(a2)



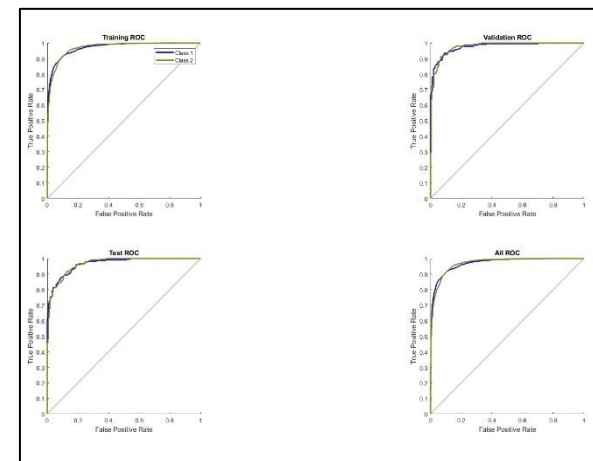
(a3)



(b1)

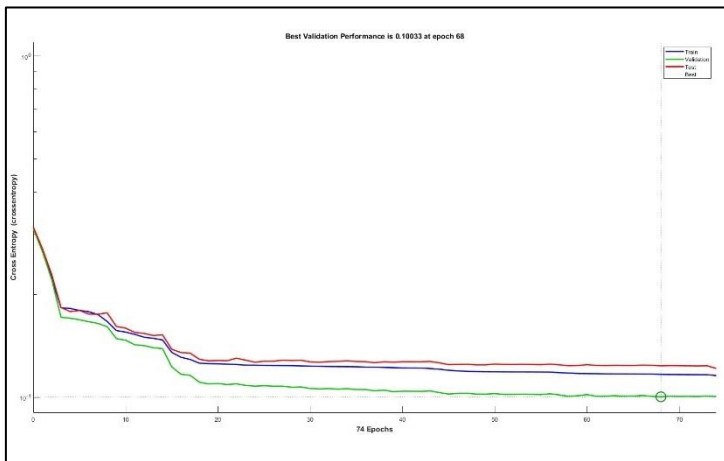


(b2)



(b3)

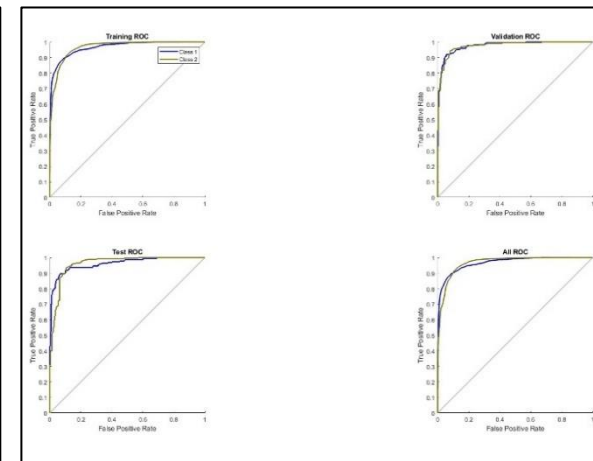
Gambar 4. 9 (a1) (b1) Performance Run 1 dan 2 Training OSS HN 5, (a2) (b2) Confusion Matrix Run 1 dan 2 Training OSS HN 5, (a3) (b3) ROC Run 1 dan 2 Training OSS HN 5



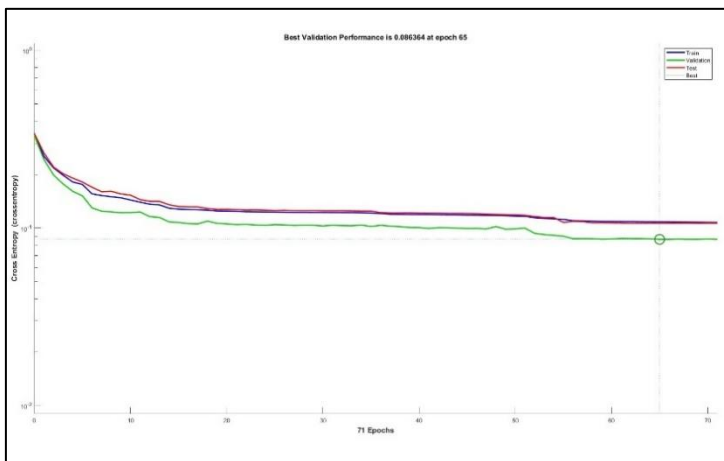
(c1)



(c2)



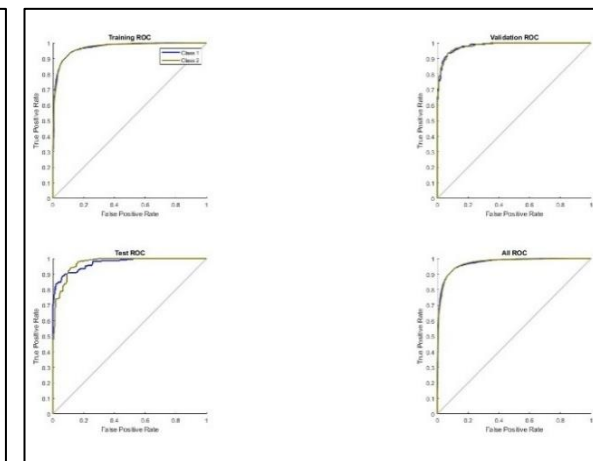
(c3)



(d1)

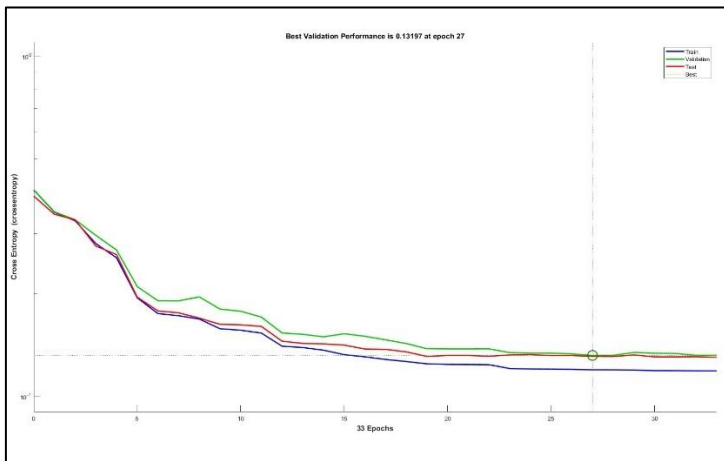


(d2)



(d3)

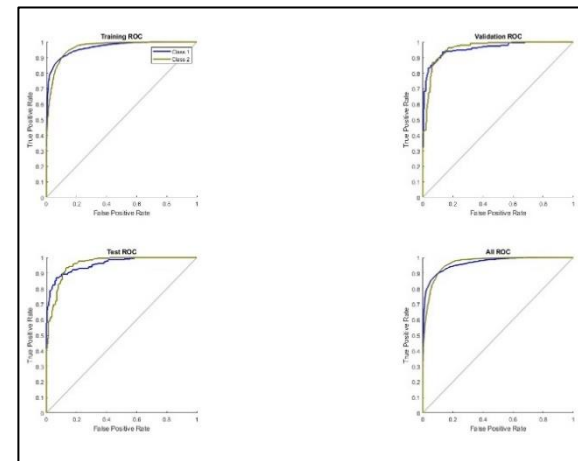
Gambar 4.9 (c1) (d1) Performance Run 3 dan 4 Training OSS HN 5, (c2) (d2) Confusion Matrix Run 3 dan 4 Training OSS HN 5, (c3) (d3) ROC Run 3 dan 4 Training OSS HN 5



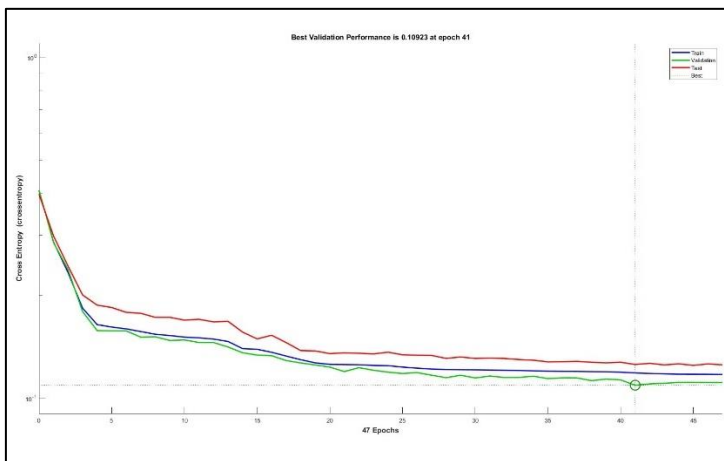
(e1)



(e2)



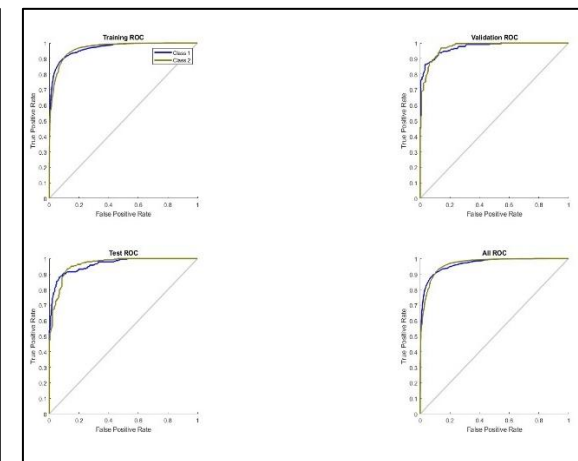
(e3)



(f1)

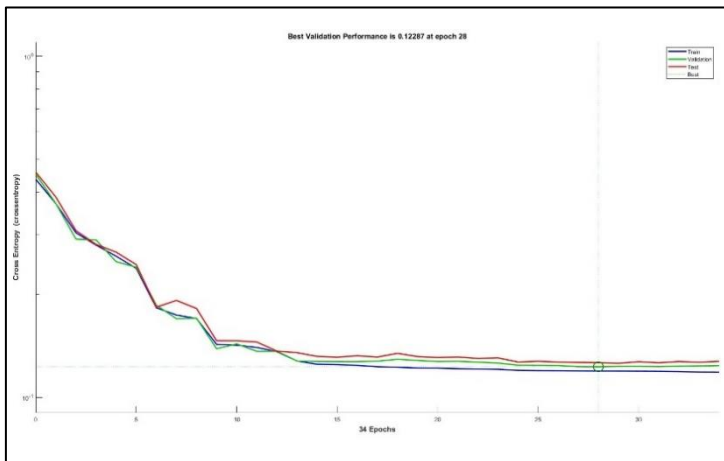


(f2)



(f3)

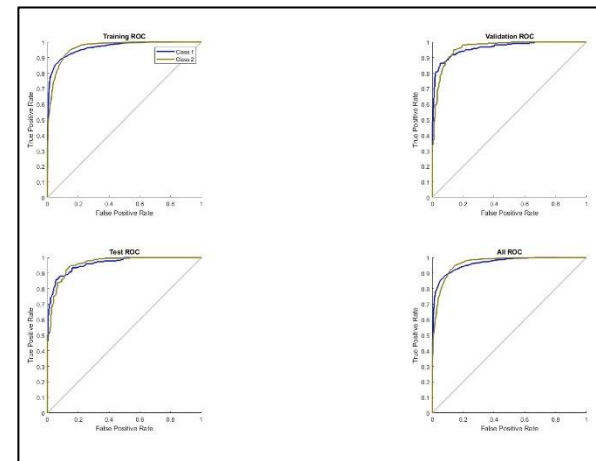
Gambar 4.9 (e1) (f1) Performance Run 5 dan 6 Training OSS HN 5, (e2) (f2) Confusion Matrix Run 5 dan 6 Training OSS HN 5, (e3) (f3) ROC Run 5 dan 6 Training OSS HN 5



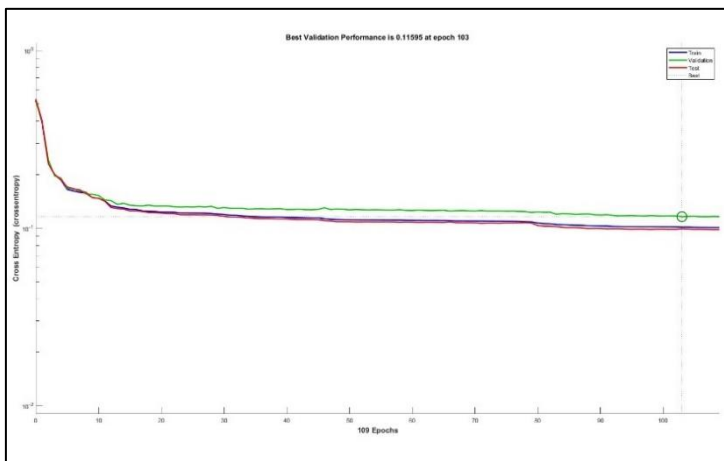
(g1)



(g2)



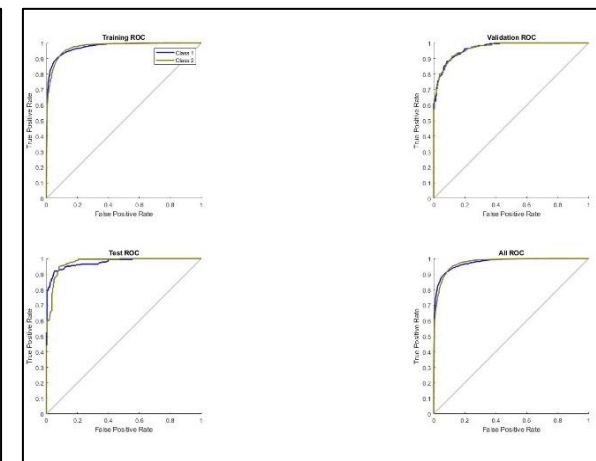
(g3)



(h1)

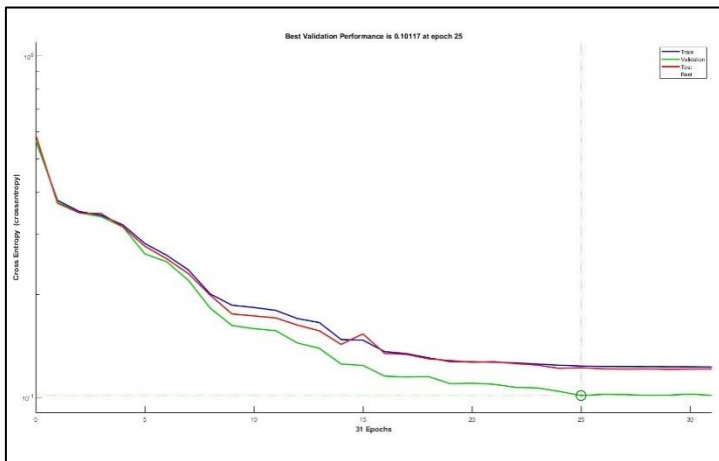


(h2)



(h3)

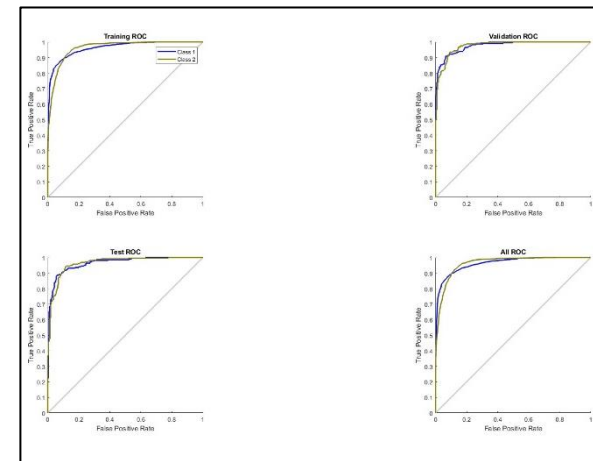
Gambar 4.9 (g1) (h1) Performance Run 7 dan 8 Training OSS HN 5, (g2) (h2) Confusion Matrix Run 7 dan 8 Training OSS HN 5, (g3) (h3) ROC Run 7 dan 8 Training OSS HN 5



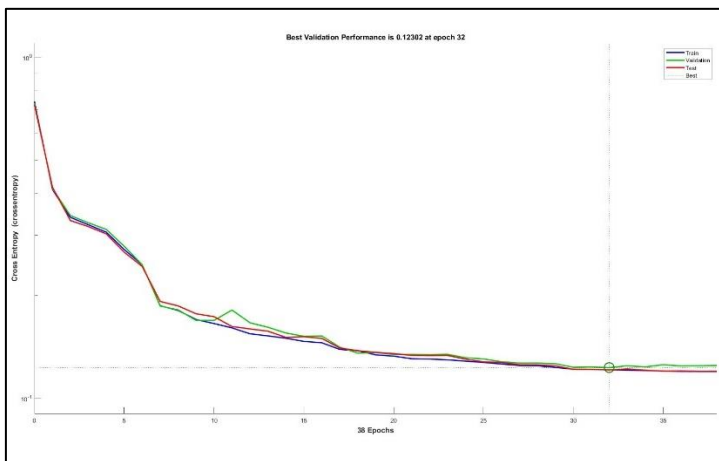
(i1)



(i2)



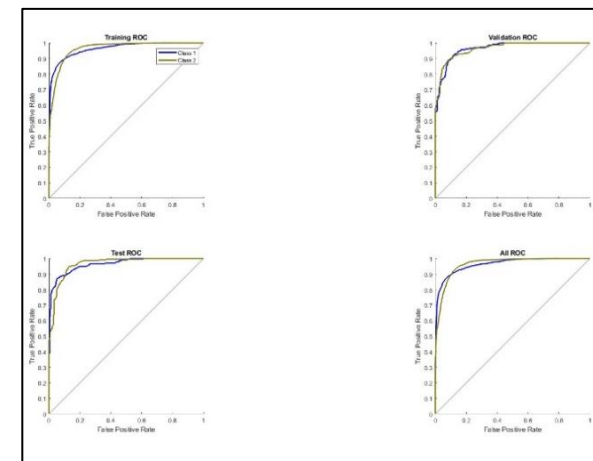
(i3)



(j1)



(j2)



(j3)

Gambar 4.9 (i1) (j1) Performance Run 9 dan 10 Training OSS HN 5, (i2) (j2) Confusion Matrix Run 9 dan 10 Training OSS HN 5, (i3) (j3) ROC Run 9 dan 10 Training OSS HN 5

Berdasarkan visualisasi hasil training di atas, dapat diketahui bahwa grafik performance terdiri dari 3 grafik, yaitu grafik biru merepresentasikan hasil training, grafik hijau merepresentasikan hasil validasi, dan grafik merah merepresentasikan hasil testing. Gambar 4.9 menampilkan sepuluh grafik performane training OSS *hidden neuron 5* dimana grafik hasil *run* mengalami penurunan signifikan pada *range epoch* 0-5, kecuali *run* 4, 7 dan 9. Grafik hasil *run* 7 dan 9 mengalami penurunan signifikan pada *range epoch* 0-10 sedangkan *run* 4 grafik tidak mengalami penurunan signifikan dari awal *epoch* hingga *epoch* maksimalnya. Dari uraian di atas, dapat disimpulkan bahwa semakin rendah nilai entropi dari grafik (nilai error) maka semakin baik performa dari model

Hasil training yang menunjukkan akurasi dari sistem ditunjukkan oleh *confusion matrix* dimana warna hijau merepresentasikan citra yang berhasil diklasifikasikan ke dalam kelas secara benar dan warna merah adalah citra yang gagal diklasifikasikan dalam kelasnya. Perbandingan data yang digunakan untuk *10-run cross validation* yaitu 90% dari total data (8.591) digunakan sebagai data *training*, 5% dari total data (477) digunakan sebagai data validasi, dan 5% lainnya (477) digunakan sebagai data *testing*. Dari *confusion matrix* tersebut, diketahui bahwa akurasi *training* mencapai lebih dari 90% serta akurasi validasi dan testing \geq lebih dari 88%. Nilai rata-rata *error training* adalah $\leq 10\%$ (859) dan rata-rata nilai *error testing* adalah $\leq 10\%$ (47). Hal ini menunjukkan bahwa model memiliki performa yang baik.

Visualisasi grafik lainnya adalah grafik *Receiver Operating Characteristic* (ROC). Fungsi dari grafik ROC ini adalah untuk mengetahui hasil evaluasi performa *machine learning*. Pada grafik ROC, terdapat dua warna grafik yaitu biru yang menunjukkan klasifikasi citra kelas 1 dan grafik hijau menunjukkan klasifikasi citra kelas 2. Kurva *training*, validasi, dan *testing* mendekati angka 1 dan semakin menjauhi garis *threshold*. Dengan demikian, dapat disimpulkan bahwa performa *machine learning* (*Multilayer Perceptron* model OSS *hidden neuron 5*) menunjukkan performa *perfect*.

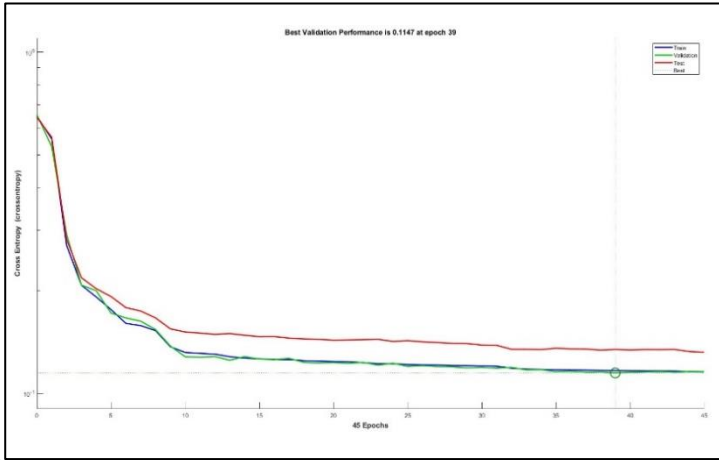
c. *Hidden Neuron 10*

Tabel 4.13 merupakan hasil akurasi *training* citra *body fluids* menggunakan model *One Step Secant Backpropagation* (OSS) *hidden neuron 10*

Tabel 4. 13 Hasil Akurasi *Training* Menggunakan Model OSS *Hidden Neuron 10*

<i>Run</i>	<i>Epoch</i>		MLP-TrainOSS <i>Hidden Neuron 10</i>			<i>Elapsed Time</i>
	<i>All</i>	<i>Best</i>	<i>Acc Training</i>	<i>Acc Validation</i>	<i>Acc Testing</i>	
1	45	39	90,3%	92,5%	88,7%	00:00:03
2	30	24	89,7%	88,9%	88,9%	00:00:03
3	32	26	89,8%	91,6%	87,8%	00:00:03
4	51	45	90,5%	89,3%	89,3%	00:00:03
5	134	128	92,8%	94,1%	91,8%	00:00:05
6	19	13	89,2%	91,8%	89,5%	00:00:02
7	107	101	92,4%	92,5%	91,6%	00:00:04
8	69	63	91,0%	92,0%	90,4%	00:00:03
9	102	96	91,4%	90,6%	90,1%	00:00:04
10	60	54	91,0%	91,6%	90,6%	00:00:03
Average			90,8%	91,5%	89,9%	00:00:03
Standar Deviasi			0,0115993	0,0154665	0,0127545	

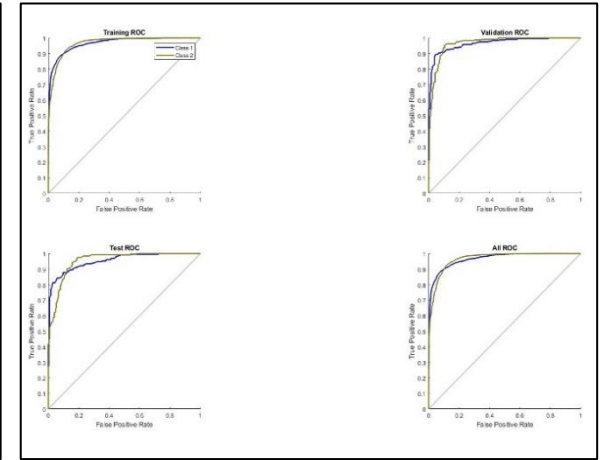
Berdasarkan Tabel 4.13 diketahui bahwa rata-rata akurasi *training* menggunakan model OSS dengan *hidden neuron 10* adalah 89-92%. Hasil akurasi *training* terbaik terdapat pada *run 5* dengan akurasi *training* sebesar 92,8%, akurasi validasi 94,1%, dan akurasi *testing* 91,8%. Akurasi *training* terbaik ini diperoleh pada *epoch* ke 128 dari total 134 *epoch* dengan waktu *training* selama 5 detik. Sementara itu, hasil akurasi *training* terendah terdapat pada *run 6* dengan akurasi *training* 89,2%, akurasi validasi 91,8%, dan akurasi *testing* 89,5%. Dari pengujian diatas, diperoleh hasil bahwa rata-rata akurasi *training* adalah 90,8%, akurasi validasi 91,5%, dan akurasi *testing* 89,9%. Rata-rata waktu *training* menggunakan model OSS *hidden neuron 10* adalah 3 detik. Visualisasi hasil *training* dapat dilihat pada Gambar 4.10.



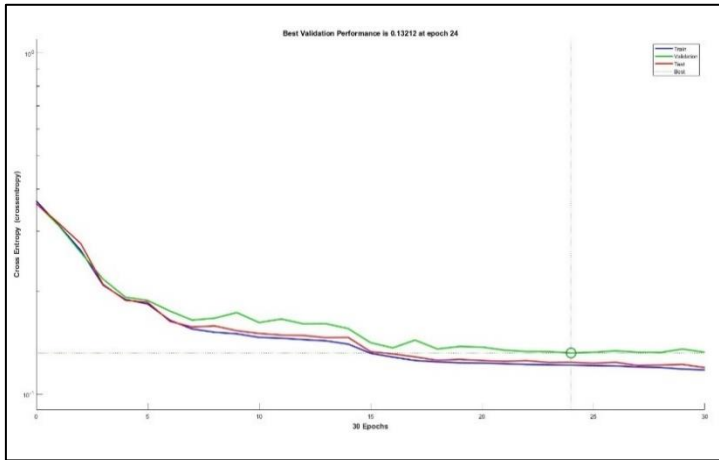
(a1)



(a2)



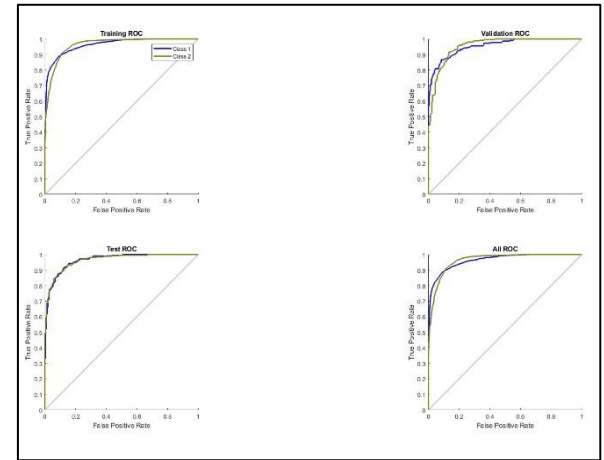
(a3)



(b1)

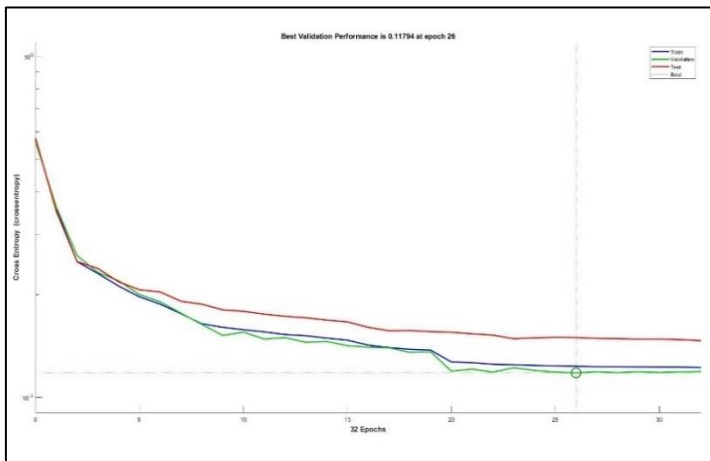


(b2)



(b3)

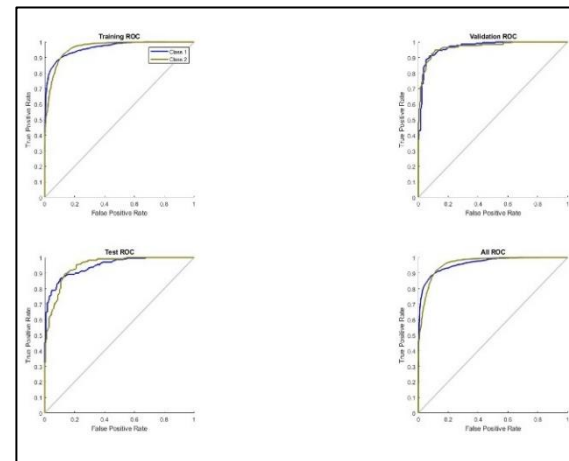
Gambar 4. 10 (a1) (b1) Performance Run 1 dan 2 Training OSS HN 10, (a2) (b2) Confusion Matrix Run 1 dan 2 Training OSS HN 10, (a3) (b3) ROC Run 1 dan 2 Training OSS HN 10



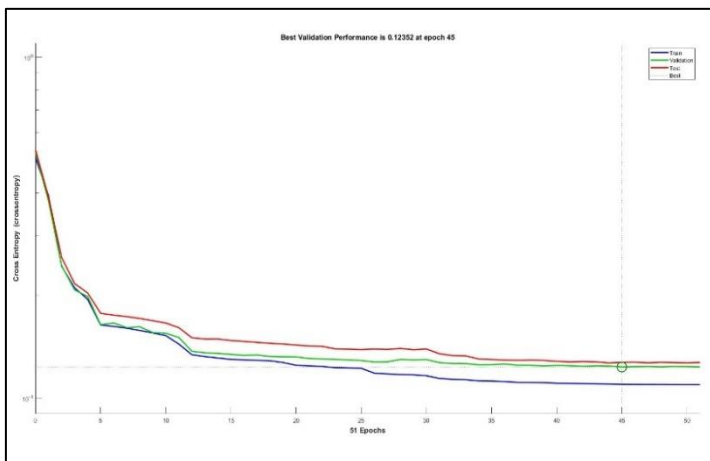
(c1)



(c2)



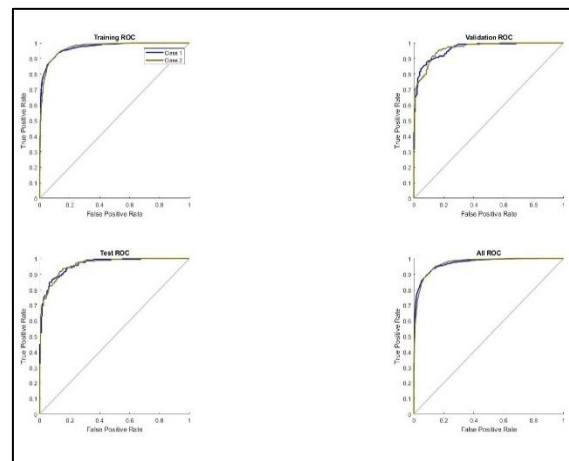
(c3)



(d1)

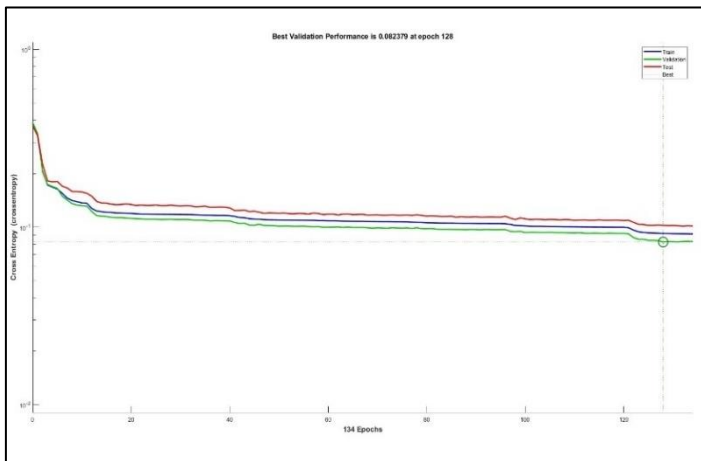


(d2)



(d3)

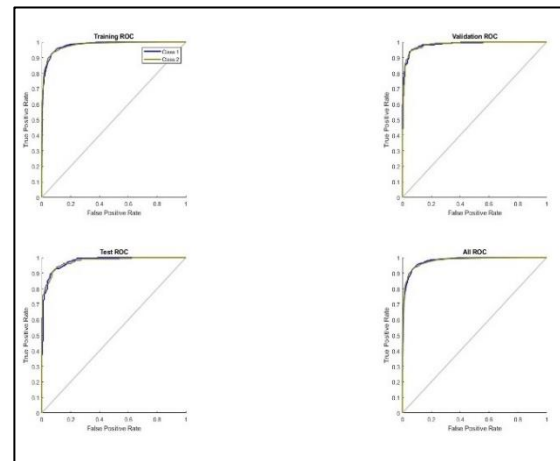
Gambar 4.10 (c1) (d1) Performance Run 3 dan 4 Training OSS HN 10, (c2) (d2) Confusion Matrix Run 3 dan 4 Training OSS HN 10, (c3) (d3) ROC Run 3 dan 4 Training OSS HN 10



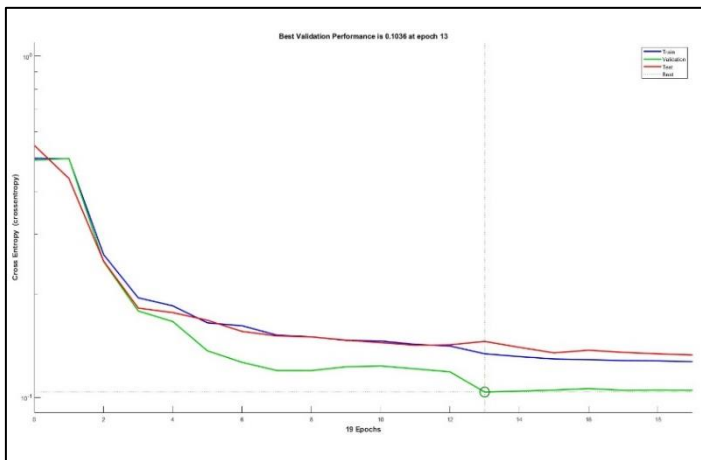
(e1)



(e2)



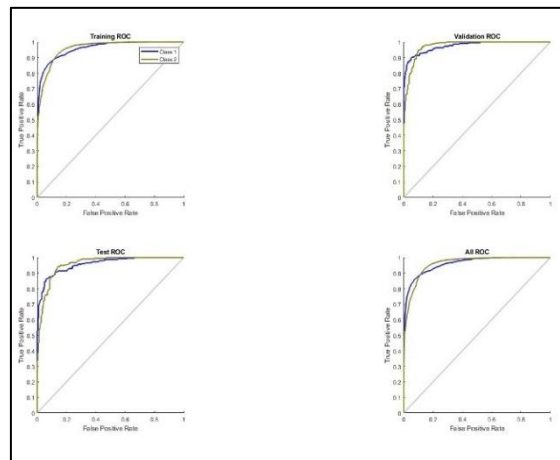
(e3)



(f1)

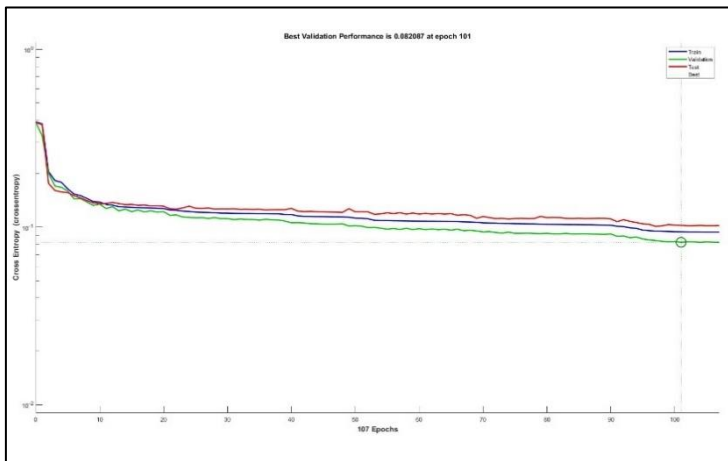


(f2)



(f3)

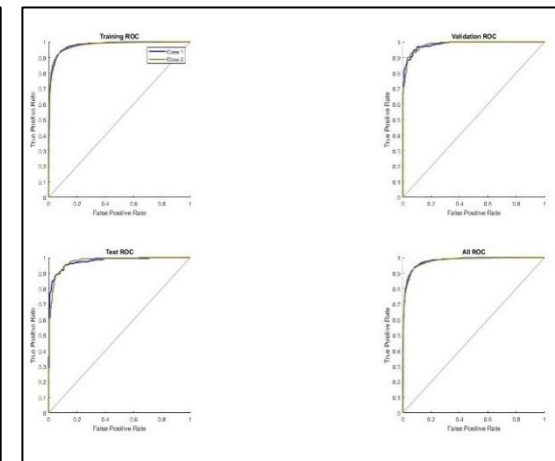
Gambar 4.10 (e1) (f1) Performance Run 5 dan 6 Training OSS HN 10, (e2) (f2) Confusion Matrix Run 5 dan 6 Training OSS HN 10, (e3) (f3) ROC Run 5 dan 6 Training OSS HN 10



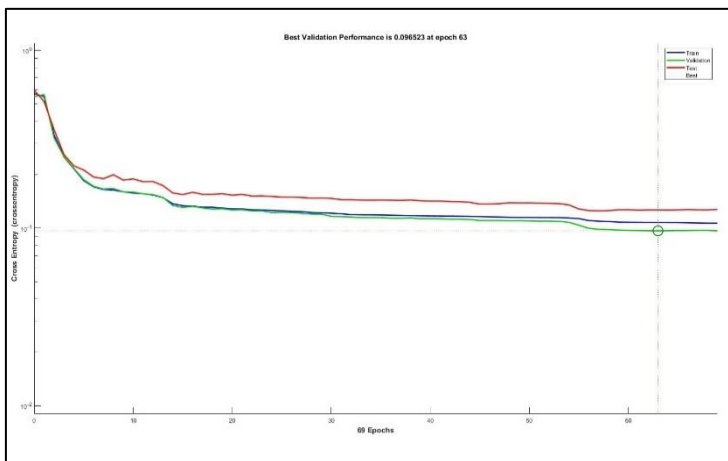
(g1)



(g2)



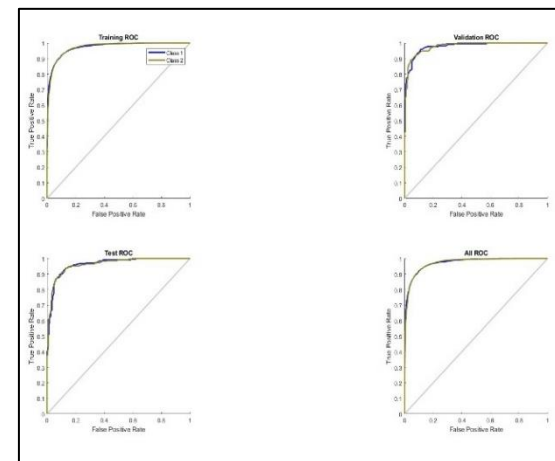
(g3)



(h1)

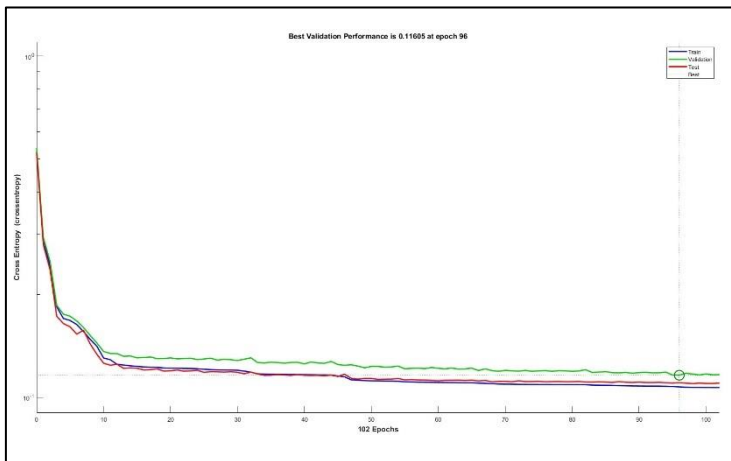


(h2)



(h3)

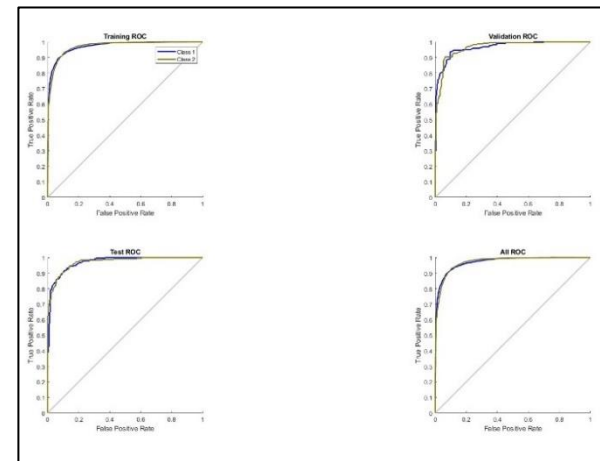
Gambar 4.10 (g1) (h1) Performance Run 7 dan 8 Training OSS HN 10, (g2) (h2) Confusion Matrix Run 7 dan 8 Training OSS HN 10, (g3) (h3) ROC Run 7 dan 8 Training OSS HN 10



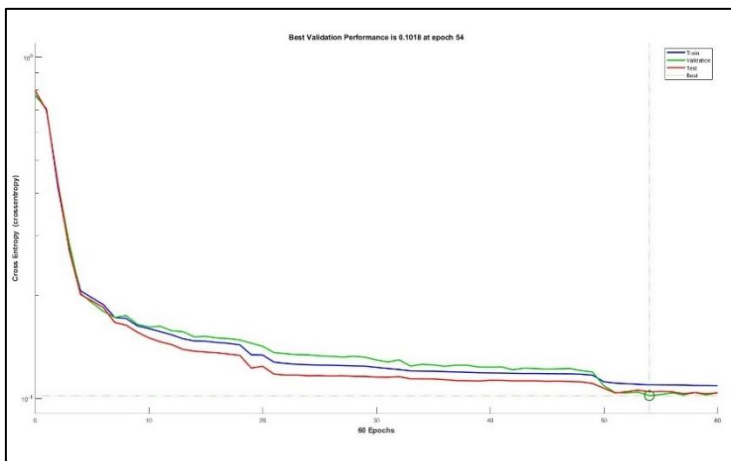
(i1)



(i2)



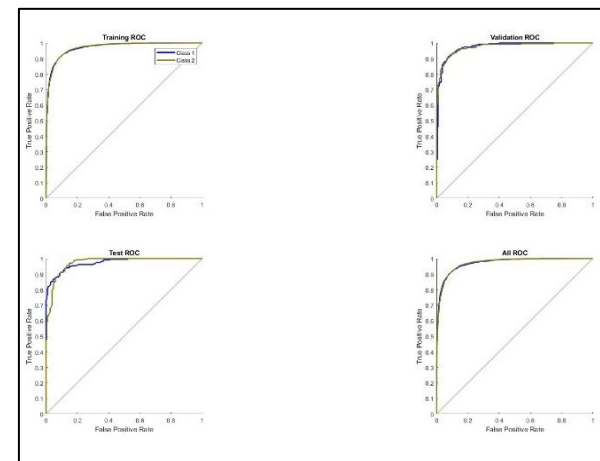
(i3)



(j1)



(j2)



(j3)

Gambar 4.10 (i1) (j1) Performance Run 9 dan 10 Training OSS HN 10, (i2) (j2) Confusion Matrix Run 9 dan 10 Training OSS HN 10, (i3) (j3) ROC Run 9 dan 10 Training OSS HN 10

Berdasarkan visualisasi hasil training di atas, dapat diketahui bahwa grafik *performance* terdiri dari 3 grafik, yaitu grafik biru merepresentasikan hasil *training*, grafik hijau merepresentasikan hasil validasi, dan grafik merah merepresentasikan hasil *testing*. Gambar 4.10 menampilkan sepuluh grafik *performane training* OSS *hidden neuron* 10 dimana penurunan signifikan terjadi pada *range epoch* 0-5 yaitu pada *run* 1, 2, 4, 9, dan 10. Penurunan *error* secara landai terjadi pada *range epoch* 3 hingga maksimal terlihat pada *run* 3, 6, 7, dan 8 sedangkan grafik *run* 5 tidak mengalami penurunan signifikan dari awal *epoch* hingga *epoch* maksimalnya. Dari uraian di atas, dapat disimpulkan bahwa semakin rendah nilai entropi dari grafik (nilai *error*) maka semakin baik performa dari model

Hasil *training* yang menunjukkan akurasi dari sistem ditunjukkan oleh *confusion matrix* dimana warna hijau merepresentasikan citra yang berhasil diklasifikasikan ke dalam kelas secara benar dan warna merah adalah citra yang gagal diklasifikasikan dalam kelasnya. Perbandingan data yang digunakan untuk *10-run cross validation* yaitu 90% dari total data (8.591) digunakan sebagai data *training*, 5% dari total data (477) digunakan sebagai data validasi, dan 5% lainnya (477) digunakan sebagai data *testing*. Dari *confusion matrix* tersebut, diketahui bahwa akurasi *training* $\geq 90\%$ serta akurasi validasi dan *testing* $\geq 87\%$. Nilai rata-rata *error training* adalah $\leq 10\%$ (859) dan rata-rata nilai *error testing* adalah $\leq 10\%$ (47) kecuali *run* 3. Meskipun demikian, model tetap memiliki performa yang cukup baik.

Visualisasi grafik lainnya adalah grafik *Receiver Operating Characteristic* (ROC). Fungsi dari grafik ROC ini adalah untuk mengetahui hasil evaluasi performa *machine learning*. Pada grafik ROC, terdapat dua warna grafik yaitu biru yang menunjukkan klasifikasi citra kelas 1 dan grafik hijau menunjukkan klasifikasi citra kelas 2. Pada *run* 10, kurva validasi dan *testing* mendekati angka 1 dan semakin menjauhi garis *threshold*. Dengan demikian, dapat disimpulkan bahwa performa *machine learning* (*Multilayer Perceptron* model SCG *hidden neuron* 10) menunjukkan performa perfect.

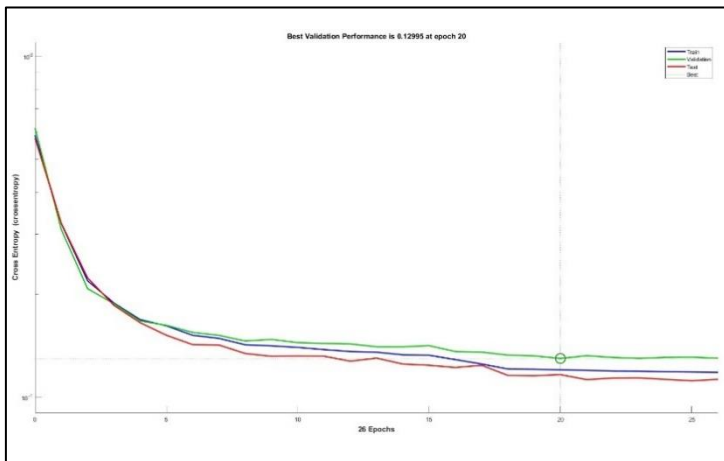
d. *Hidden Neuron 15*

Tabel 4.14 merupakan hasil akurasi *training* citra *body fluids* menggunakan model *One Step Secant Backpropagation* (OSS) *hidden neuron 15*.

Tabel 4. 14 Hasil Akurasi *Training* Menggunakan Model OSS *Hidden Neuron 15*

<i>Run</i>	<i>Epoch</i>		MLP-TrainOSS Hidden Neuron 15			<i>Elapsed Time</i>
	<i>All</i>	<i>Best</i>	<i>Acc Training</i>	<i>Acc Validation</i>	<i>Acc Testing</i>	
1	26	20	90,0%	89,3%	89,3%	00:00:03
2	42	36	90,3%	89,3%	91,0%	00:00:03
3	64	58	91,2%	89,3%	90,1%	00:00:04
4	47	41	90,8%	91,2%	90,4%	00:00:03
5	23	17	89,8%	92,0%	91,0%	00:00:02
6	34	28	90,0%	90,4%	89,9%	00:00:03
7	43	37	90,3%	90,4%	89,1%	00:00:03
8	24	18	89,8%	87,8%	89,3%	00:00:02
9	39	33	89,9%	91,6%	88,7%	00:00:03
10	143	137	93,2%	95,4%	91,0%	00:00:06
Average			90,5%	90,7%	90,0%	00:00:03
Standar Deviasi			0,0104249	0,0208968	0,0085997	

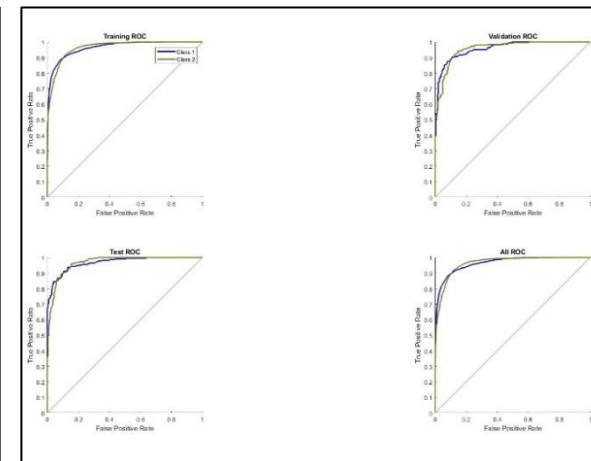
Berdasarkan Tabel 4.14 diketahui bahwa rata-rata akurasi *training* menggunakan model OSS dengan *hidden neuron 15* adalah 89-93%. Hasil akurasi *training* terbaik terdapat pada *run 10* dengan akurasi *training* sebesar 93,2%, akurasi validasi 95,4%, dan akurasi *testing* 91,0%. Akurasi *training* terbaik ini diperoleh pada *epoch* ke 137 dari total 148 *epoch* dengan waktu *training* selama 6 detik. Sementara itu, hasil akurasi *training* terendah terdapat pada *run 8* dengan akurasi *training* 89,8%, akurasi validasi 87,8%, dan akurasi *testing* 89,3%. Dari pengujian diatas, diperoleh hasil bahwa rata-rata akurasi *training* adalah 90,5%, akurasi validasi 90,7%, dan akurasi *testing* 90,0%. Rata-rata waktu *training* menggunakan model OSS *hidden neuron 15* adalah 3 detik. Visualisasi hasil *training* dapat dilihat pada Gambar 4.11.



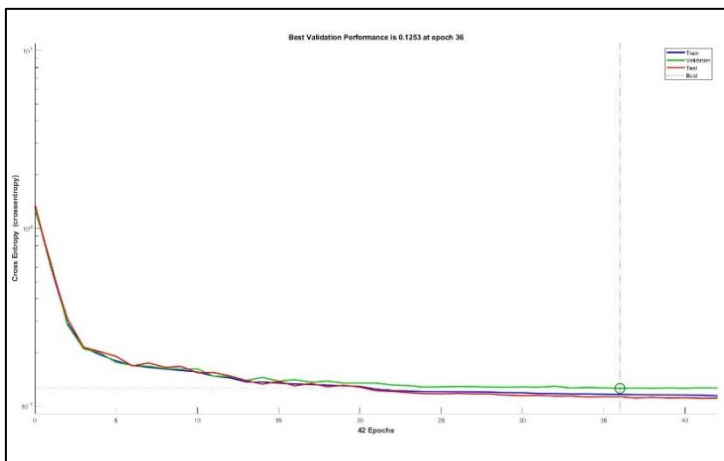
(a1)



(a2)



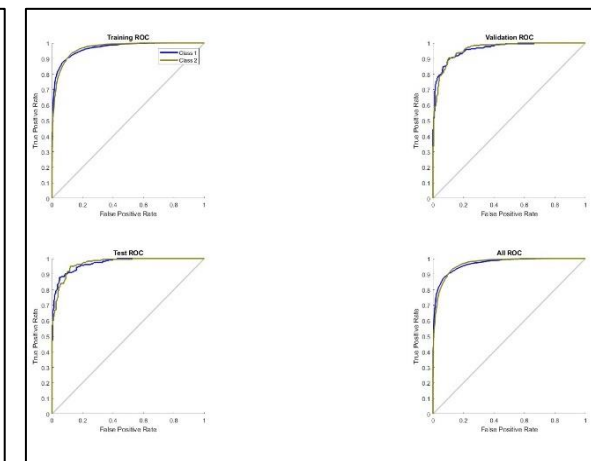
(a3)



(b1)

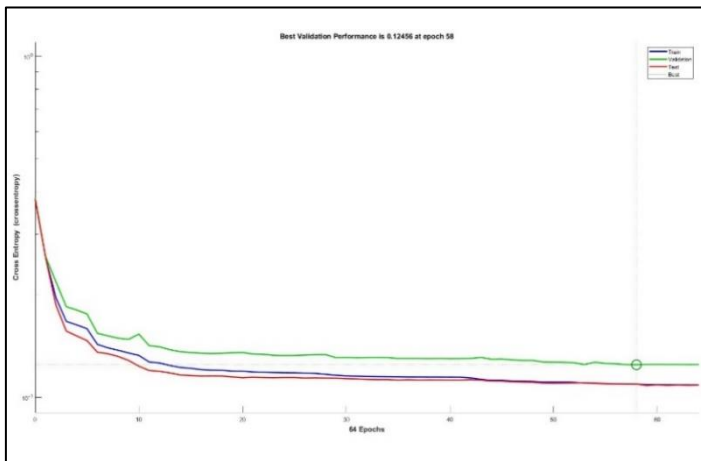


(b2)



(b3)

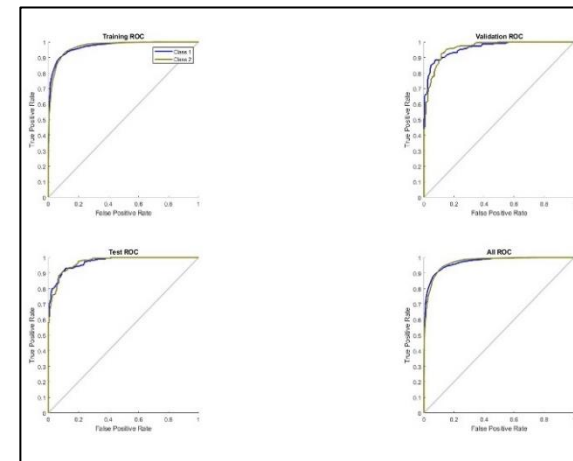
Gambar 4. 11 (a1) (b1) Performance Run 1 dan 2 Training OSS HN 15, (a2) (b2) Confusion Matrix Run 1 dan 2 Training OSS HN 15, (a3) (b3) ROC Run 1 dan 2 Training OSS HN 15



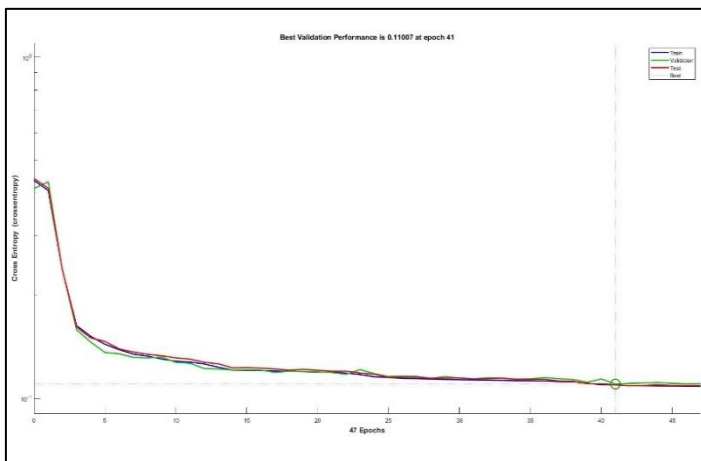
(c1)



(c2)



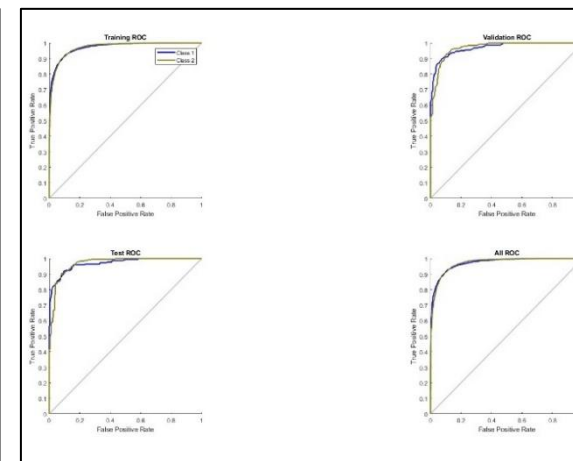
(c3)



(d1)

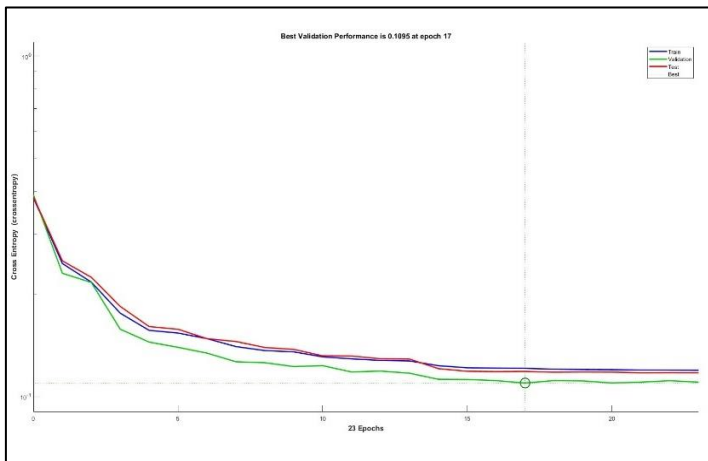


(d2)



(d3)

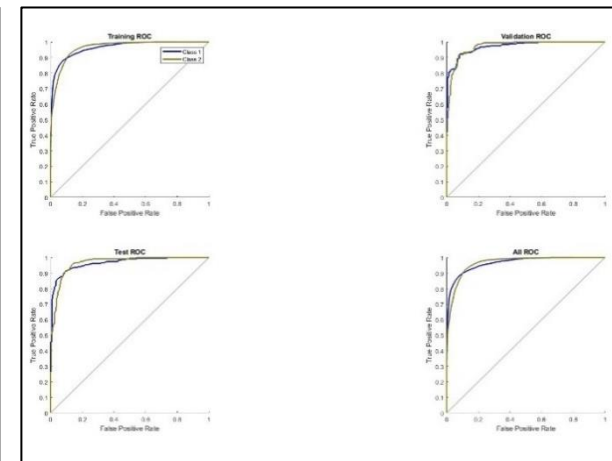
Gambar 4.11 (c1) (d1) Performance Run 3 dan 4 Training OSS HN 15, (c2) (d2) Confusion Matrix Run 3 dan 4 Training OSS HN 15, (c3) (d3) ROC Run 3 dan 4 Training OSS HN 15



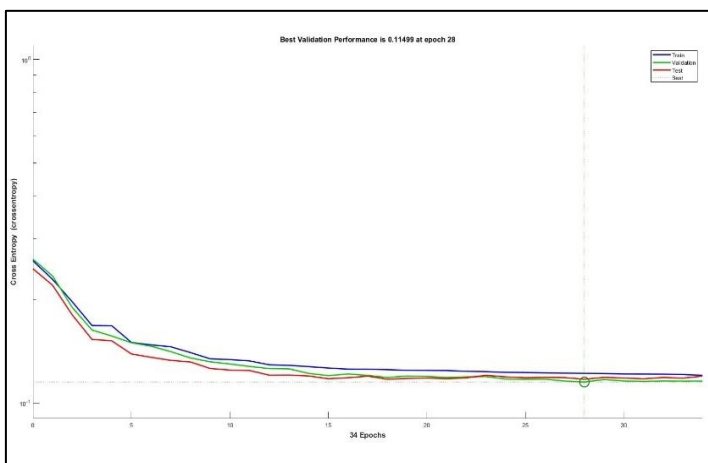
(e1)



(e2)



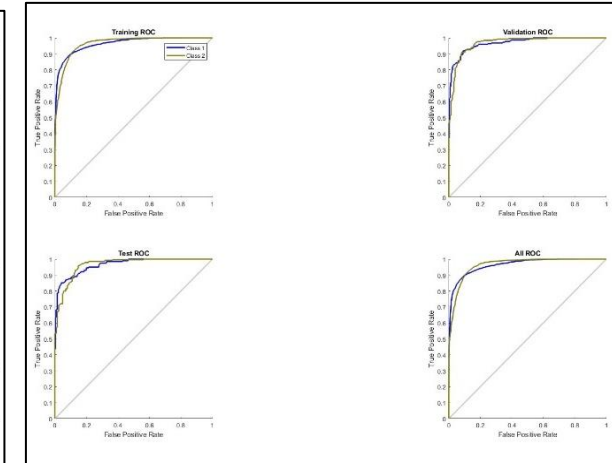
(e3)



(f1)

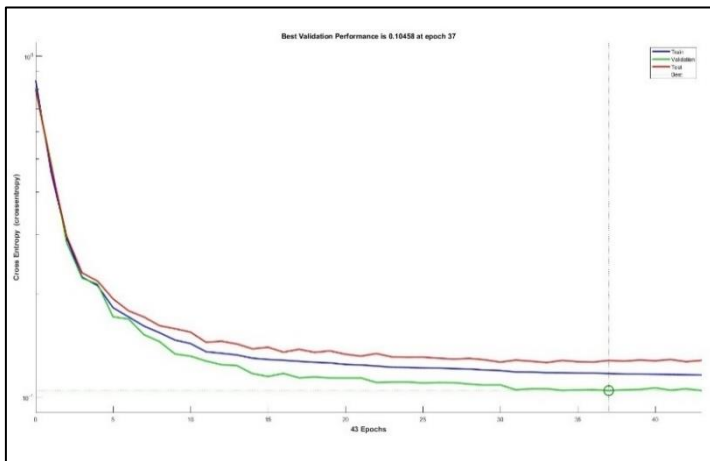


(f2)



(f3)

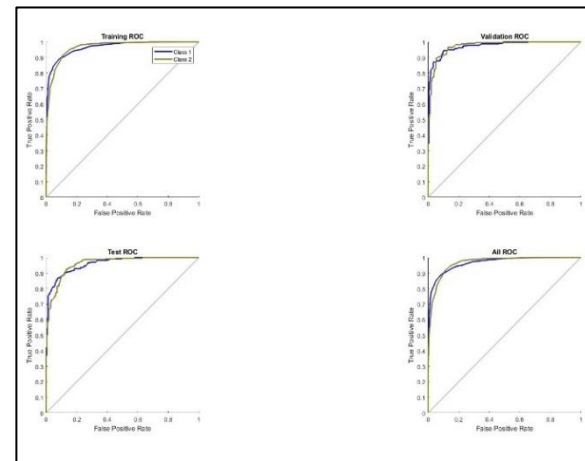
Gambar 4.11 (e1) (f1) Performance Run 5 dan 6 Training OSS HN 15, (e2) (f2) Confusion Matrix Run 5 dan 6 Training OSS HN 15, (e3) (f3) ROC Run 5 dan 6 Training OSS HN 15



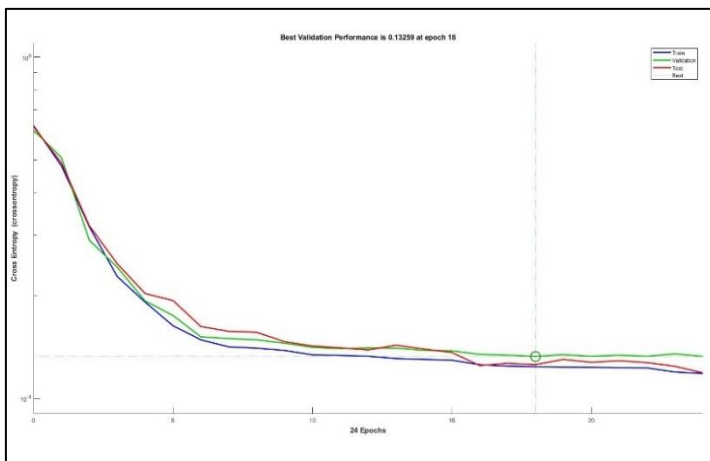
(g1)



(g2)



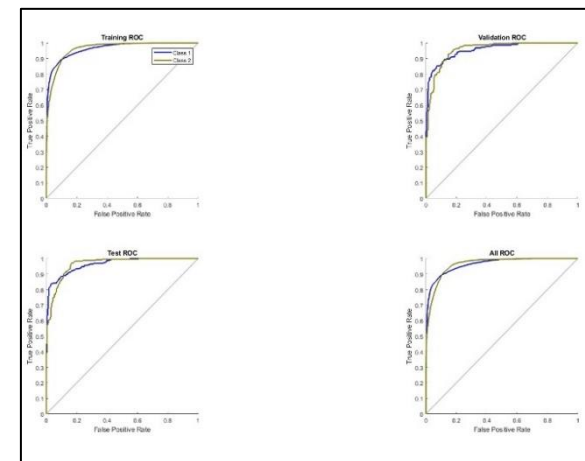
(g3)



(h1)

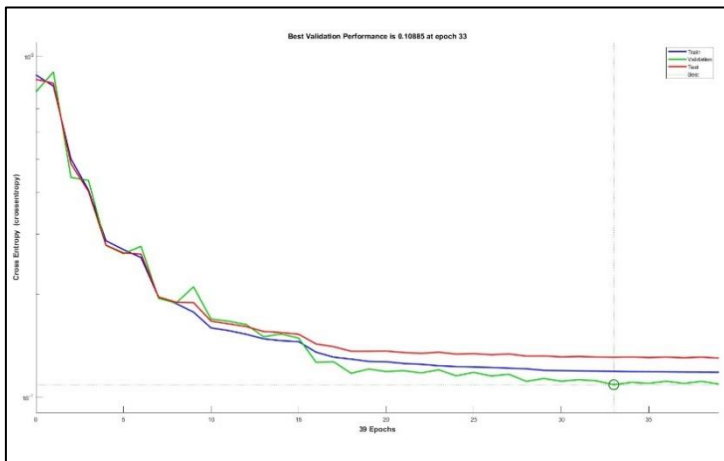


(h2)



(h3)

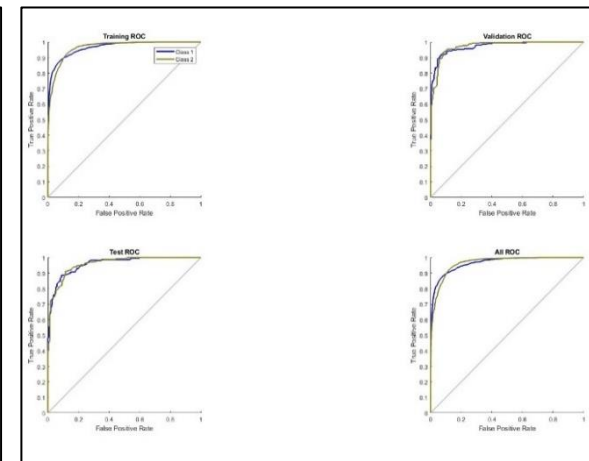
Gambar 4.11 (g1) (h1) Performance Run 7 dan 8 Training OSS HN 15, (g2) (h2) Confusion Matrix Run 7 dan 8 Training OSS HN 15, (g3) (h3) ROC Run 7 dan 8 Training OSS HN 15



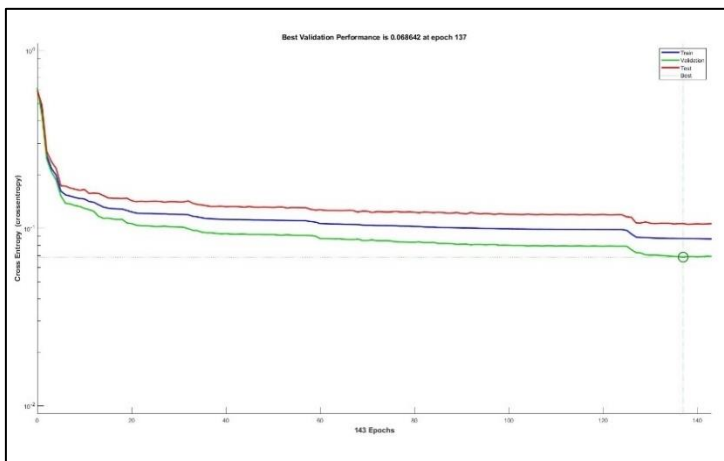
(i1)



(i2)



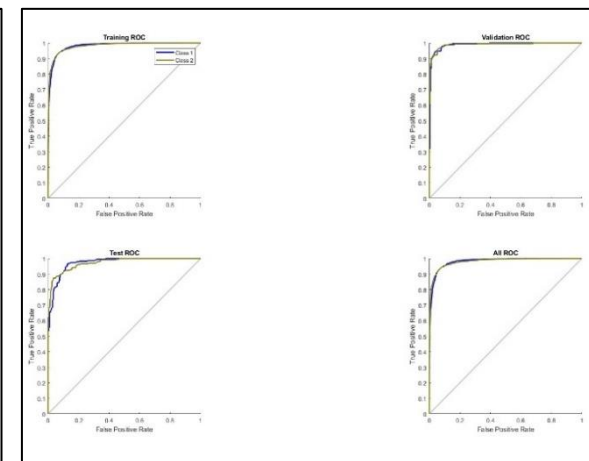
(i3)



(j1)



(j2)



(j3)

Gambar 4.11 (i1) (j1) Performance Run 9 dan 10 Training OSS HN 15, (i2) (j2) Confusion Matrix Run 9 dan 10 Training OSS HN 15, (i3) (j3) ROC Run 9 dan 10 Training OSS HN 15

Berdasarkan visualisasi hasil *training* di atas, dapat diketahui bahwa grafik *performance* terdiri dari 3 grafik, yaitu grafik biru merepresentasikan hasil *training*, grafik hijau merepresentasikan hasil validasi, dan grafik merah merepresentasikan hasil *testing*. Gambar 4.11 menampilkan sepuluh grafik *performane training OSS hidden neuron 15* dimana grafik *error* mengalami penurunan signifikan pada *range epoch* 0-5 kecuali run 6, 9, dan 10. Penurunan performa *error* run 9 terjadi signifikan pada *range epoch* 0-10 sedangkan pada run 6 dan 10, grafik mengalami penurunan *error* secara melandai mulai *epoch* hingga *epoch* maksimalnya. Dari uraian di atas, dapat disimpulkan bahwa semakin rendah nilai entropi dari grafik (nilai error) maka semakin baik performa dari model.

Hasil *training* yang menunjukkan akurasi dari sistem ditunjukkan oleh *confusion matrix* dimana warna hijau merepresentasikan citra yang berhasil diklasifikasikan ke dalam kelas secara benar dan warna merah adalah citra yang gagal diklasifikasikan dalam kelasnya. Perbandingan data yang digunakan untuk *10-run cross validation* yaitu 90% dari total data (8.591) digunakan sebagai data *training*, 5% dari total data (477) digunakan sebagai data validasi, dan 5% lainnya (477) digunakan sebagai data *testing*. Dari *confusion matrix* tersebut, diketahui bahwa akurasi *training*, validasi, dan *testing* mencapai $\geq 90\%$. Nilai rata-rata *error training* adalah $\leq 10\%$ (859) dan rata-rata nilai *error testing* adalah $\leq 10\%$ (47). Hal ini menunjukkan bahwa model memiliki performa yang baik.

Visualisasi grafik lainnya adalah grafik *Receiver Operating Characteristic* (ROC). Fungsi dari grafik ROC ini adalah untuk mengetahui hasil evaluasi performa *machine learning*. Pada grafik ROC, terdapat dua warna grafik yaitu biru yang menunjukkan klasifikasi citra kelas 1 dan grafik hijau menunjukkan klasifikasi citra kelas 2. Kurva validasi dan *testing* berada pada angka 0,9 mendekati angka 1 serta semakin menjauhi garis *threshold*. Dengan demikian, dapat disimpulkan bahwa performa *machine learning* (*Multilayer Perceptron* model *OSS hidden neuron 15*) menunjukkan performa perfect.

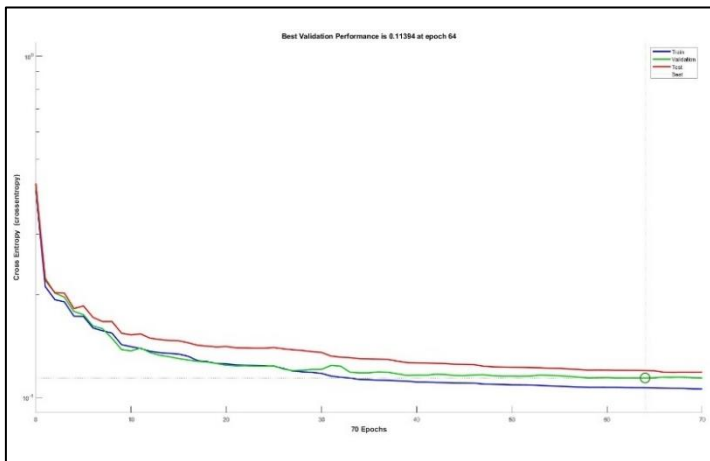
e. *Hidden Neuron 20*

Tabel 4.15 merupakan hasil akurasi *training* citra *body fluids* menggunakan model *One Step Secant Backpropagation* (OSS) *hidden neuron 20*.

Tabel 4. 15 Hasil Akurasi *Training* Menggunakan Model OSS *Hidden Neuron 20*

<i>Run</i>	<i>Epoch</i>		MLP-TrainOSS Hidden Neuron 20			<i>Elapsed Time</i>
	<i>All</i>	<i>Best</i>	<i>Acc Training</i>	<i>Acc Validation</i>	<i>Acc Testing</i>	
1	70	64	91,2%	89,7%	90,6%	00:00:04
2	133	127	92,6%	95,2%	91,2%	00:00:06
3	61	55	90,9%	91,2%	87,8%	00:00:04
4	116	110	92,9%	92,5%	93,3%	00:00:05
5	77	71	91,5%	91,4%	91,2%	00:00:04
6	122	116	92,0%	92,7%	90,1%	00:00:05
7	58	52	90,3%	90,1%	91,6%	00:00:03
8	71	65	90,9%	92,0%	91,8%	00:00:04
9	53	47	90,8%	91,4%	90,4%	00:00:03
10	34	28	90,0%	89,3%	89,9%	00:00:03
Average			91,3%	91,6%	90,8%	00:00:04
Standar Deviasi			0,0094569	0,017161	0,014418	

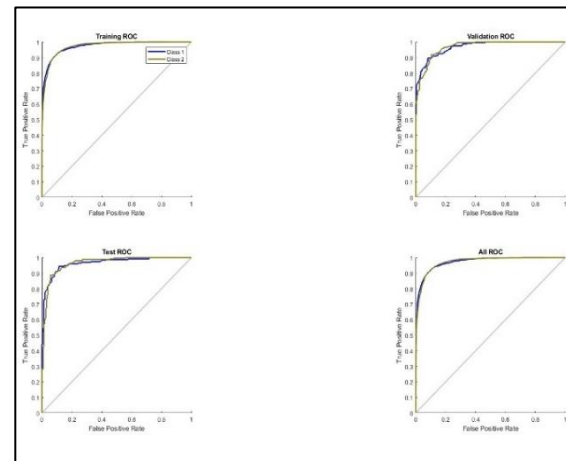
Berdasarkan Tabel 4.15 diketahui bahwa rata-rata akurasi *training* menggunakan model OSS dengan *hidden neuron 20* adalah 90-92%. Hasil akurasi *training* terbaik terdapat pada *run 4* dengan akurasi *training* sebesar 92,9%, akurasi validasi 92,5%, dan akurasi *testing* 93,3%. Akurasi *training* terbaik ini diperoleh pada *epoch* ke 110 dari total 116 *epoch* dengan waktu *training* selama 5 detik. Sementara itu, hasil akurasi *training* terendah terdapat pada *run 10* dengan akurasi *training* 90,0%, akurasi validasi 89,3%, dan akurasi *testing* 89,9%. Dari pengujian diatas, diperoleh hasil bahwa rata-rata akurasi *training* adalah 91,3%, akurasi validasi 91,6%, dan akurasi *testing* 90,8%. Rata-rata waktu *training* menggunakan model OSS *hidden neuron 20* adalah 4 detik. Visualisasi hasil *training* dapat dilihat pada Gambar 4.12.



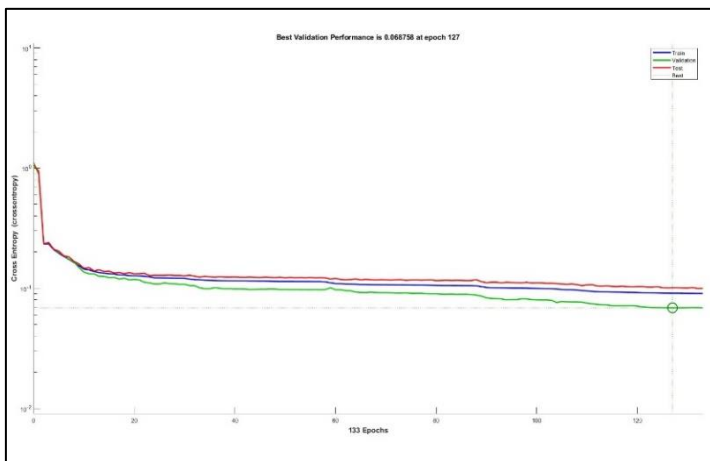
(a1)



(a2)



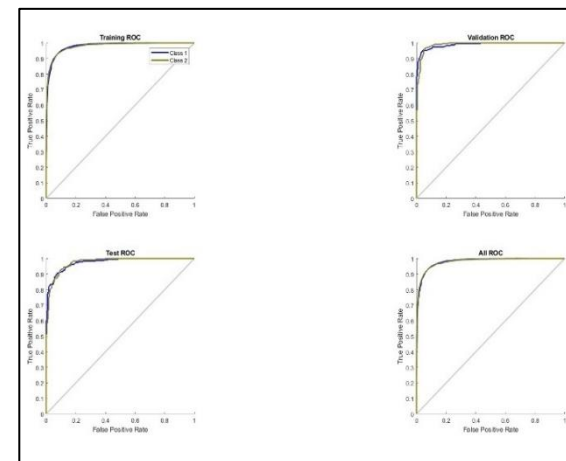
(a3)



(b1)

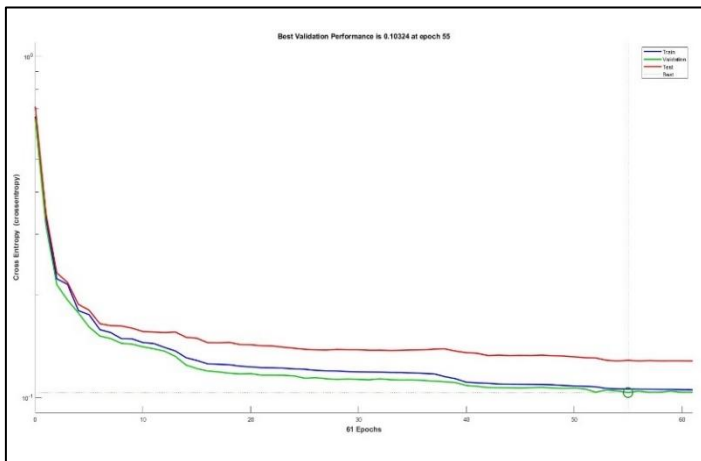


(b2)



(b3)

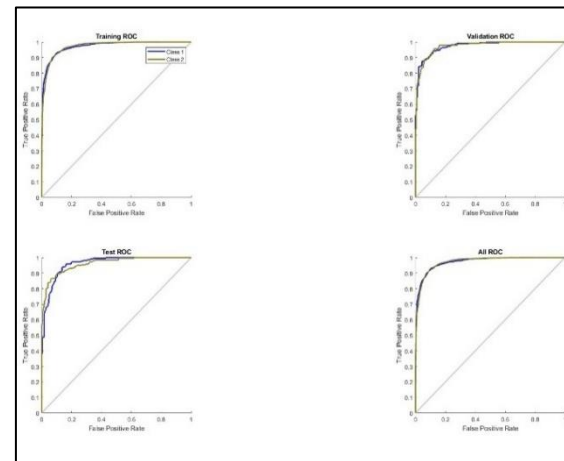
Gambar 4. 12 (a1) (b1) Performance Run 1 dan 2 Training OSS HN 20, (a2) (b2) Confusion Matrix Run 1 dan 2 Training OSS HN 20, (a3) (b3) ROC Run 1 dan 2 Training OSS HN 20



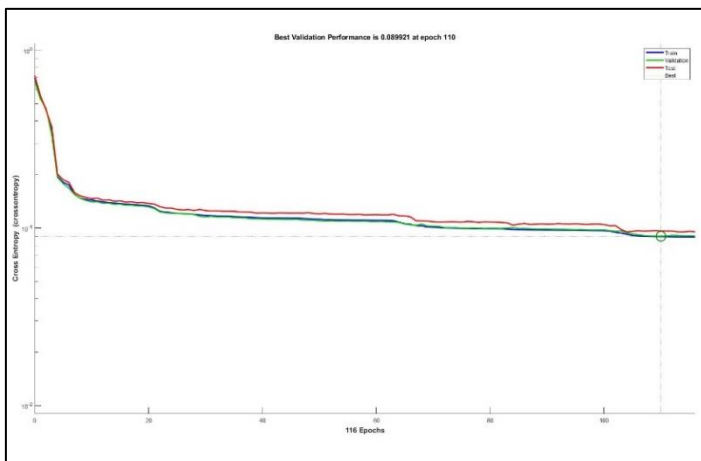
(c1)



(c2)



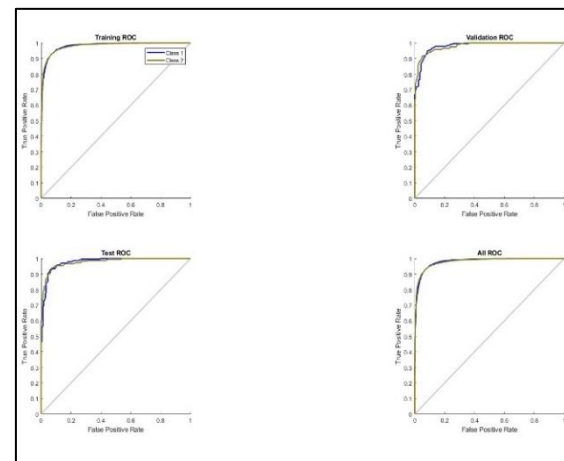
(c3)



(d1)

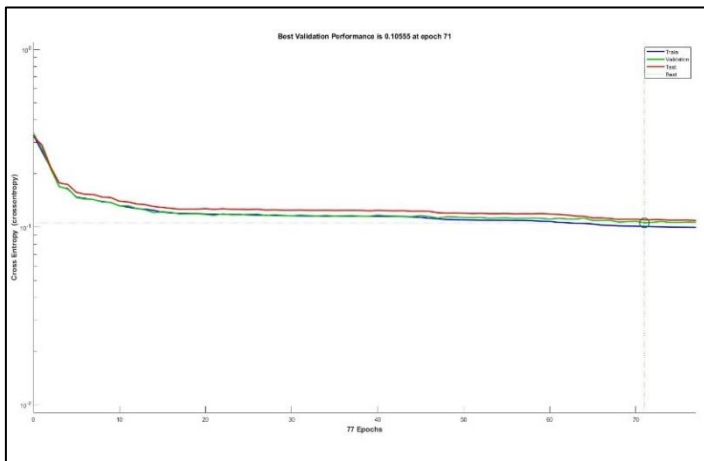


(d2)



(d3)

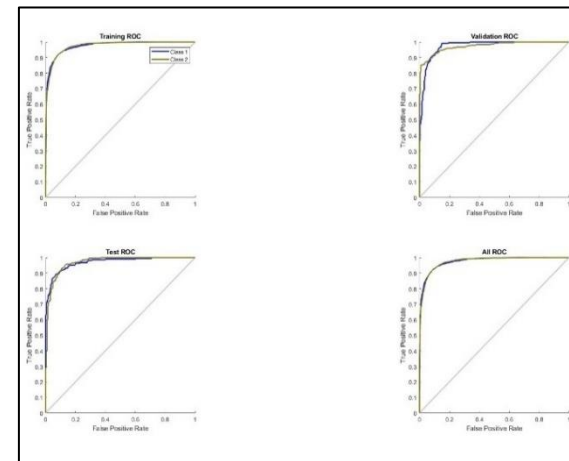
Gambar 4.12 (c1) (d1) Performance Run 3 dan 4 Training OSS HN 20, (c2) (d2) Confusion Matrix Run 3 dan 4 Training OSS HN 20, (c3) (d3) ROC Run 3 dan 4 Training OSS HN 20



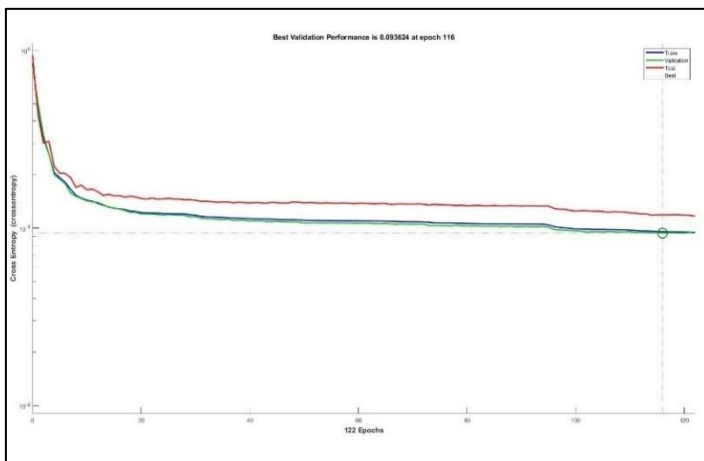
(e1)



(e2)



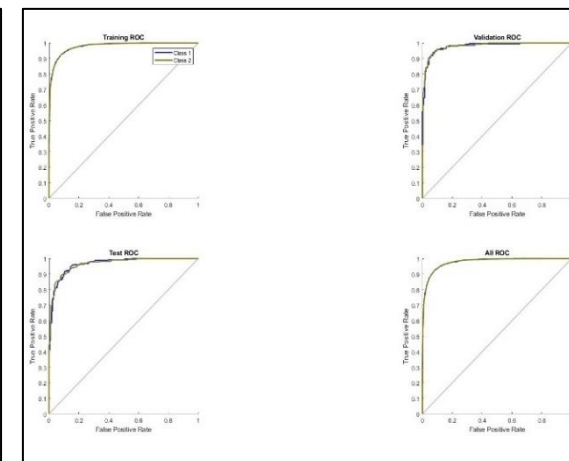
(e3)



(f1)

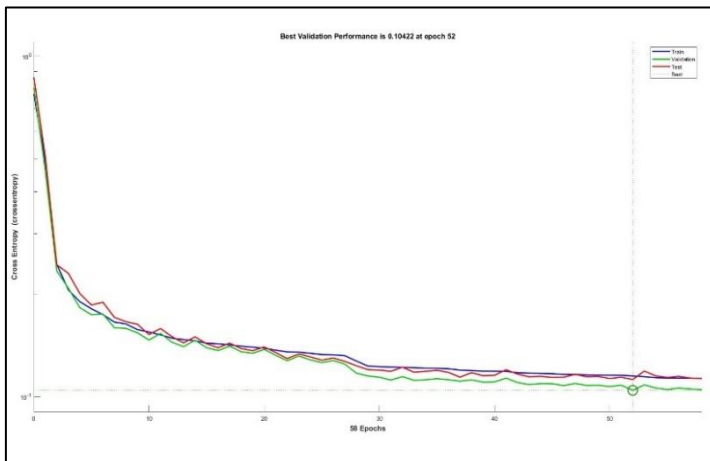


(f2)



(f3)

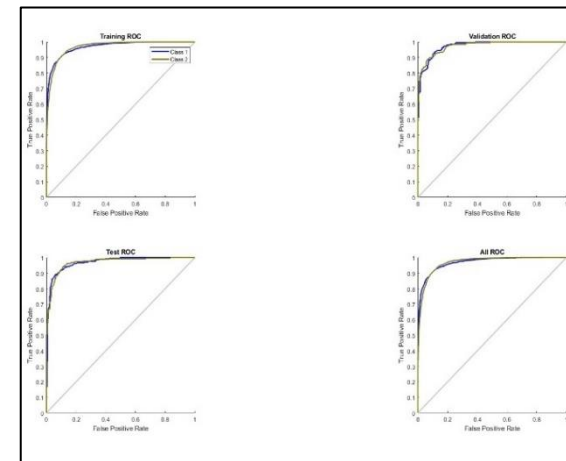
Gambar 4.12 (e1) (f1) Performance Run 5 dan 6 Training OSS HN 20, (e2) (f2) Confusion Matrix Run 5 dan 6 Training OSS HN 20, (e3) (f3) ROC Run 5 dan 6 Training OSS HN 20



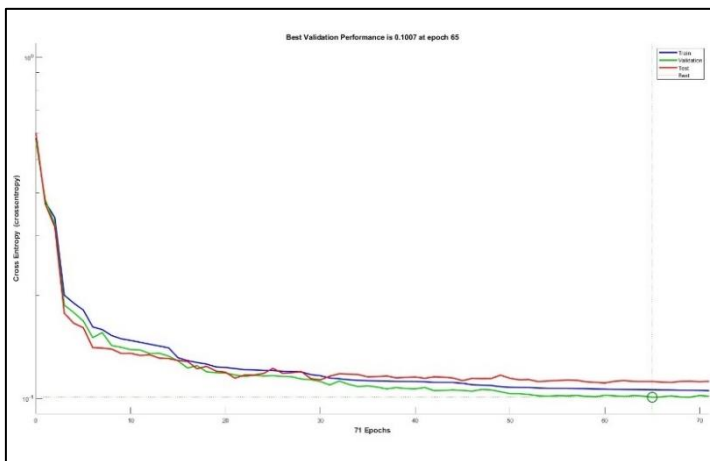
(g1)



(g2)



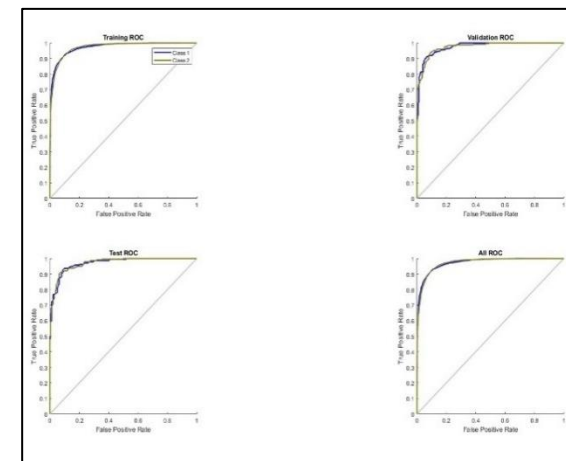
(g3)



(h1)

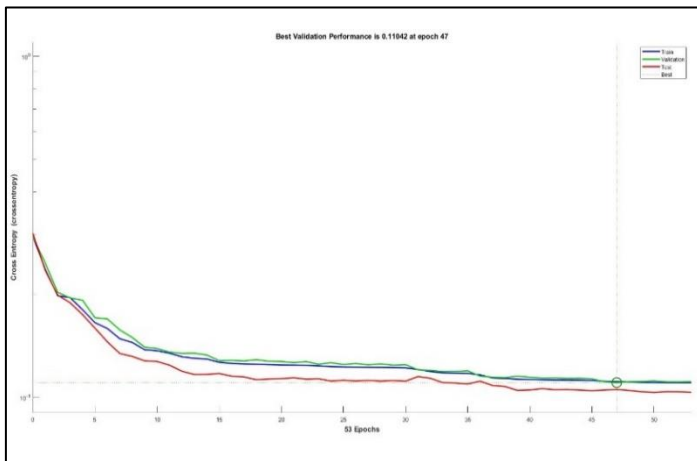


(h2)



(h3)

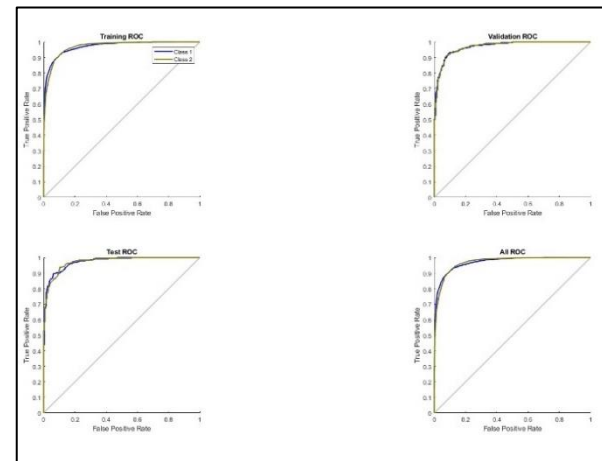
Gambar 4.12 (g1) (h1) Performance Run 7 dan 8 Training OSS HN 20, (g2) (h2) Confusion Matrix Run 7 dan 8 Training OSS HN 20, (g3) (h3) ROC Run 7 dan 8 Training OSS HN 20



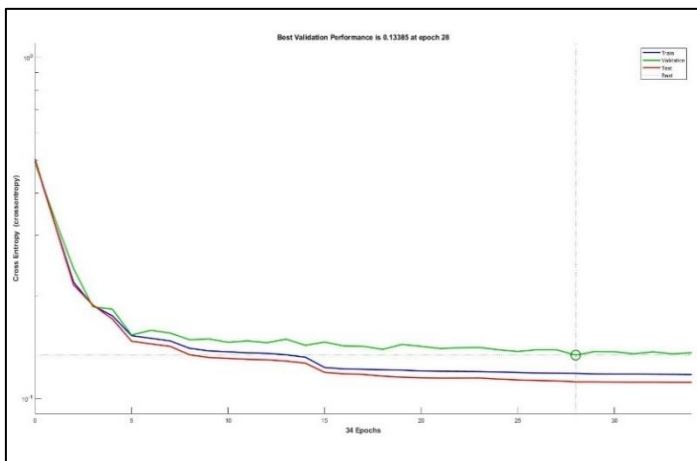
(i1)



(i2)



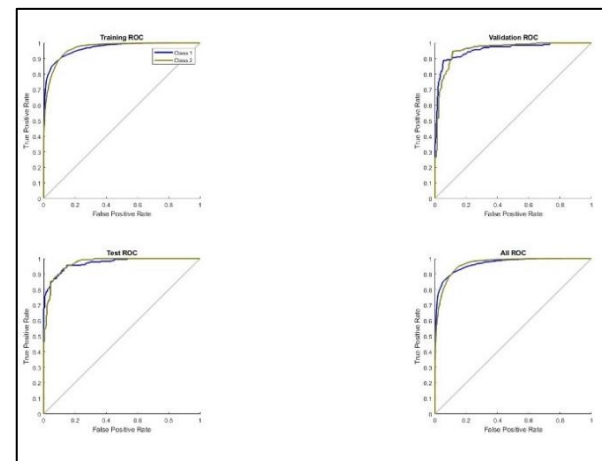
(i3)



(j1)



(j2)



(j3)

Gambar 4.12 (i1) (j1) Performance Run 9 dan 10 Training OSS HN 20, (i2) (j2) Confusion Matrix Run 9 dan 10 Training OSS HN 20, (i3) (j3) ROC Run 9 dan 10 Training OSS HN 20

Berdasarkan visualisasi hasil training di atas, dapat diketahui bahwa grafik *performance* terdiri dari 3 grafik, yaitu grafik biru merepresentasikan hasil *training*, grafik hijau merepresentasikan hasil validasi, dan grafik merah merepresentasikan hasil *testing*. Gambar 4.12 menampilkan sepuluh grafik *performane training* OSS *hidden neuron* 20 dimana grafik mengalami penurunan signifikan pada *range epoch* 0-3 kecuali *run* 2, 5, dan 9. Penurunan *error* melandai terjadi pada *range epoch* 5 hingga maksimal pada *run* 2, 5, dan 9. Semakin rendah grafik menunjukkan bahwa semakin bagus hasilnya dan nilai *error* semakin kecil. Dari uraian di atas, dapat disimpulkan bahwa semakin rendah nilai entropi dari grafik (nilai *error*) maka semakin baik performa dari model.

Hasil *training* yang menunjukkan akurasi dari sistem ditunjukkan oleh *confusion matrix* dimana warna hijau merepresentasikan citra yang berhasil diklasifikasikan ke dalam kelas secara benar dan warna merah adalah citra yang gagal diklasifikasikan dalam kelasnya. Perbandingan data yang digunakan untuk *10-run cross validation* yaitu 90% dari total data (8.591) digunakan sebagai data *training*, 5% dari total data (477) digunakan sebagai data validasi, dan 5% lainnya (477) digunakan sebagai data *testing*. Dari *confusion matrix* tersebut, diketahui bahwa akurasi *training*, validasi dan *testing* $\geq 90\%$. Nilai rata-rata *error training* adalah $\leq 10\%$ (859) dan rata-rata nilai *error testing* adalah $\leq 10\%$ (47) kecuali *run* 3. Meskipun demikian, model tetap memiliki performa yang cukup baik.

Visualisasi grafik lainnya adalah grafik *Receiver Operating Characteristic* (ROC). Fungsi dari grafik ROC ini adalah untuk mengetahui hasil evaluasi performa *machine learning*. Pada grafik ROC, terdapat dua warna grafik yaitu biru yang menunjukkan klasifikasi citra kelas 1 dan grafik hijau menunjukkan klasifikasi citra kelas 2. Kurva *training*, validasi, dan *testing* terlihat mendekati angka 1 dan semakin menjauhi garis *threshold*. Dengan demikian, dapat disimpulkan bahwa performa *machine learning* (*Multilayer Perceptron* model OSS *hidden neuron* 20) menunjukkan hasil performa *perfect*.

4.3.1.3 Gradient Descent with Momentum (GDM)

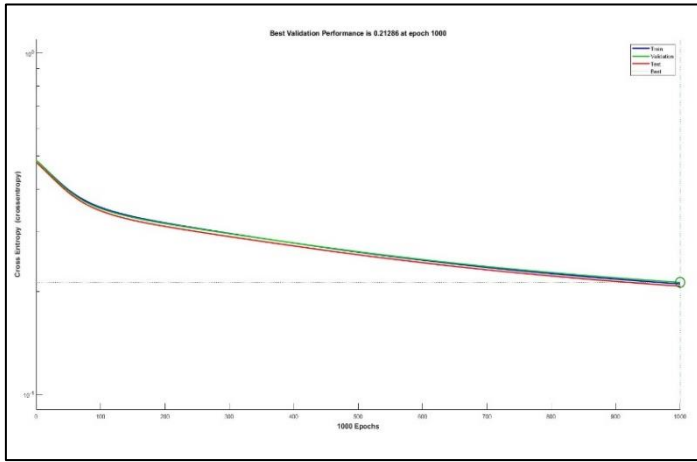
a. Hidden Neuron 1

Tabel 4.16 merupakan hasil akurasi *training* citra *body fluids* menggunakan model *Gradient Descent with Momentum* (GDM) *hidden neuron 1*.

Tabel 4. 16 Hasil Akurasi *Training* Menggunakan Model GDM *Hidden Neuron 1*

<i>Run</i>	<i>Epoch</i>		MLP-TrainGDM Hidden Neuron 1			<i>Elapsed Time</i>
	<i>All</i>	<i>Best</i>	<i>Acc Training</i>	<i>Acc Validation</i>	<i>Acc Testing</i>	
1	1000	1000	83,1%	82,4%	85,3%	00:00:09
2	1000	1000	82,9%	85,1%	83,4%	00:00:05
3	1000	1000	62,7%	62,9%	60,6%	00:00:05
4	1000	1000	68,6%	70,9%	68,3%	00:00:06
5	1000	1000	82,8%	86,0%	83,4%	00:00:05
6	1000	1000	83,4%	85,3%	81,6%	00:00:06
7	1000	1000	84,6%	84,1%	84,5%	00:00:05
8	1000	1000	73,8%	71,3%	75,5%	00:00:05
9	1000	1000	83,9%	81,8%	84,3%	00:00:05
10	1000	1000	77,4%	78,4%	75,3%	00:00:05
Average			78,3%	78,8%	78,2%	00:00:06
Standar Deviasi			0,0760129	0,0785604	0,082759	

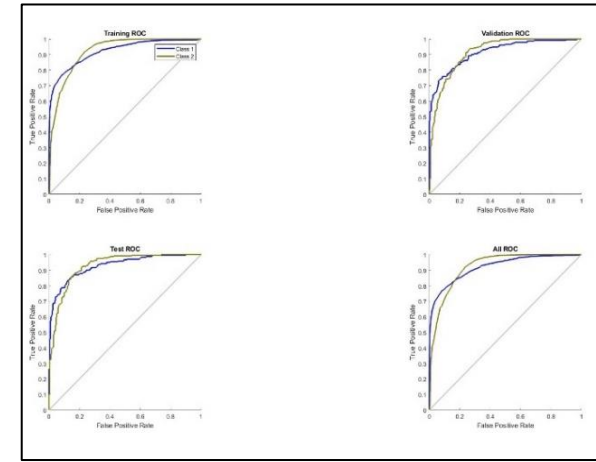
Berdasarkan Tabel 4.16 diketahui bahwa rata-rata akurasi *training* menggunakan model GDM dengan *hidden neuron 1* adalah 62-84%. Hasil akurasi *training* terbaik terdapat pada *run 7* dengan akurasi *training* sebesar 84,6%, akurasi validasi 84,1%, dan akurasi *testing* 84,5%. Akurasi *training* terbaik ini diperoleh pada *epoch* ke 1000 dari total 1000 *epoch* dengan waktu *training* selama 5 detik. Sementara itu, hasil akurasi *training* terendah terdapat pada *run 3* dengan akurasi *training* 62,7%, akurasi validasi 62,9%, dan akurasi *testing* 60,6%. Dari pengujian diatas, diperoleh hasil bahwa rata-rata akurasi *training* adalah 78,3%, akurasi validasi 78,8%, dan akurasi *testing* 78,2%. Rata-rata waktu *training* menggunakan model GDM *hidden neuron 1* adalah 6 detik. Visualisasi hasil *training* dapat dilihat pada Gambar 4.13.



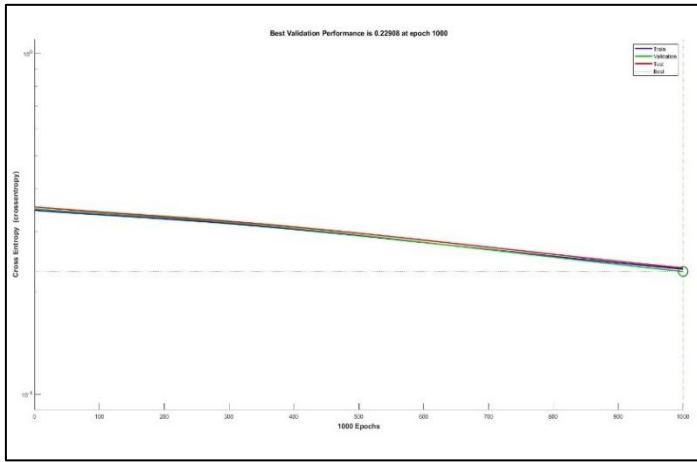
(a1)



(a2)



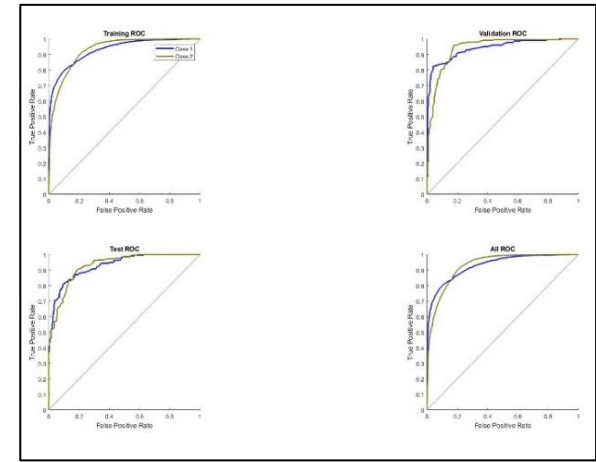
(a3)



(b1)

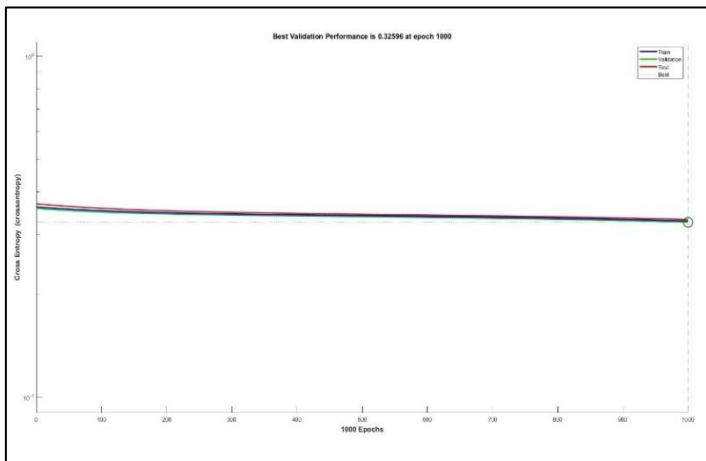


(b2)



(b3)

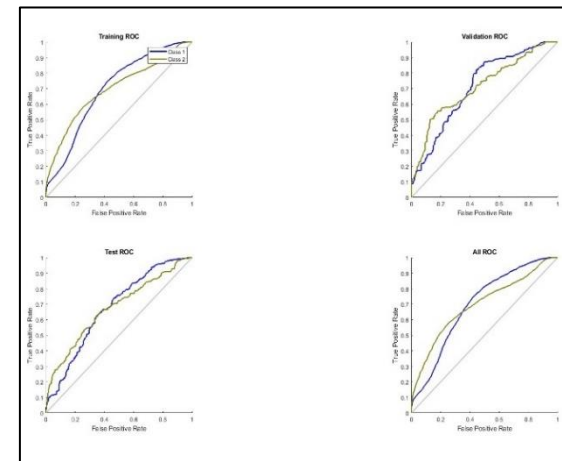
Gambar 4. 13 (a1) (b1) Performance Run 1 dan 2 Training GDM HN 1, (a2) (b2) Confusion Matrix Run 1 dan 2 Training GDM HN 1, (a3) (b3) ROC Run 1 dan 2 Training GDM HN 1



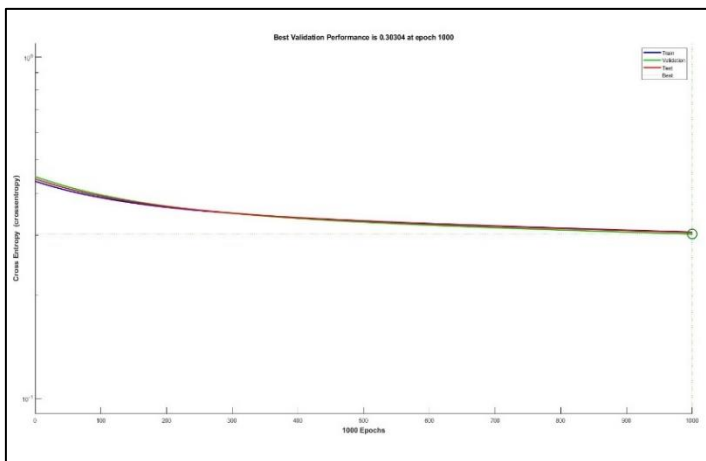
(c1)



(c2)



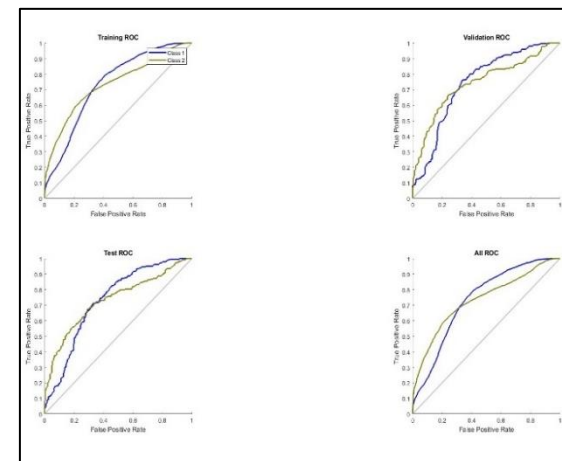
(c3)



(d1)

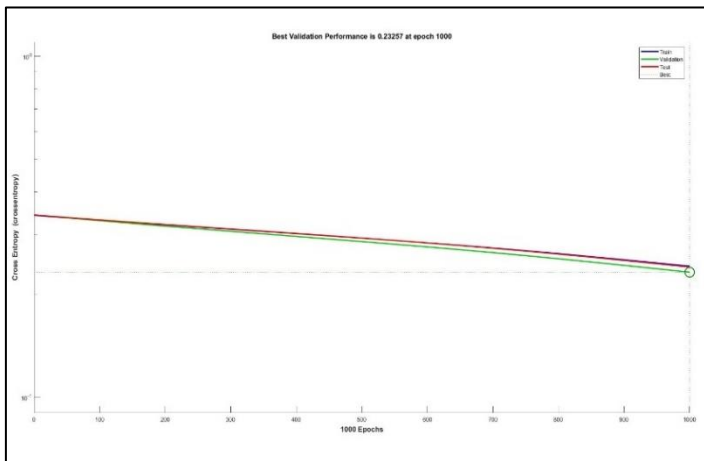


(d2)



(d3)

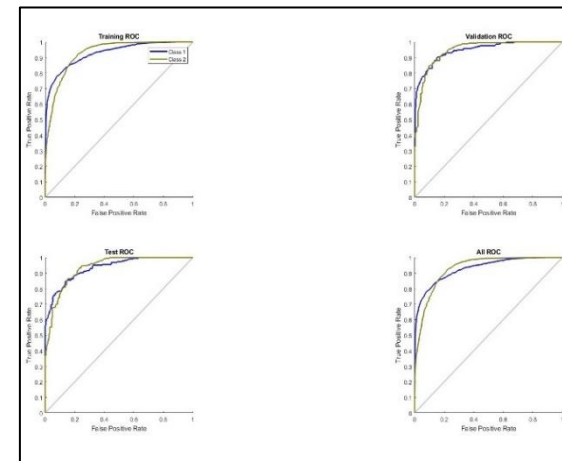
Gambar 4.13 (c1) (d1) Performance Run 3 dan 4 Training GDM HN 1, (c2) (d2) Confusion Matrix Run 3 dan 4 Training GDM HN 1, (c3) (d3) ROC Run 3 dan 4 Training GDM HN 1



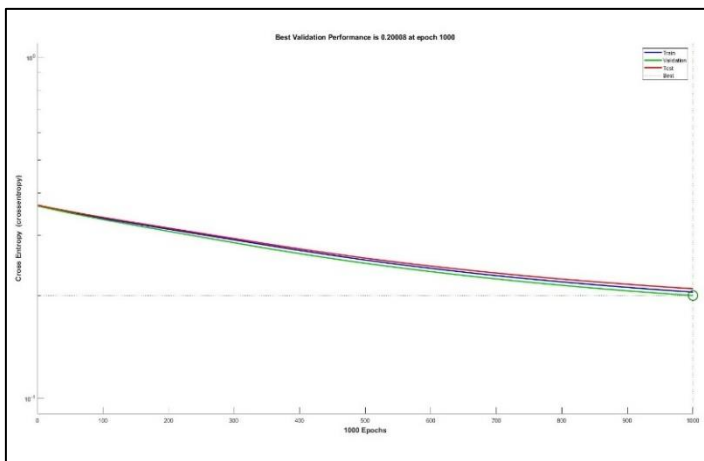
(e1)



(e2)



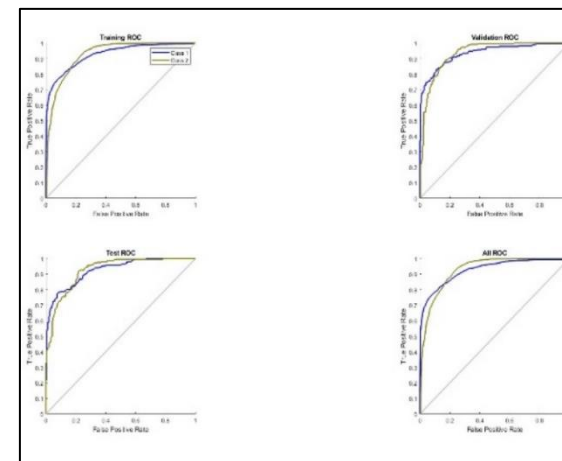
(e3)



(f1)

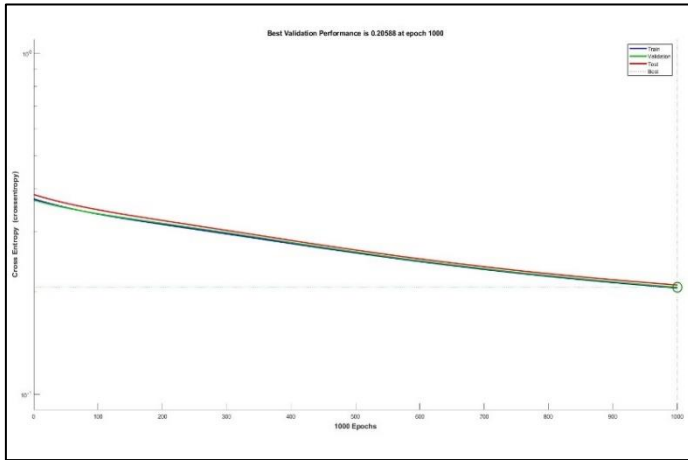


(f2)



(f3)

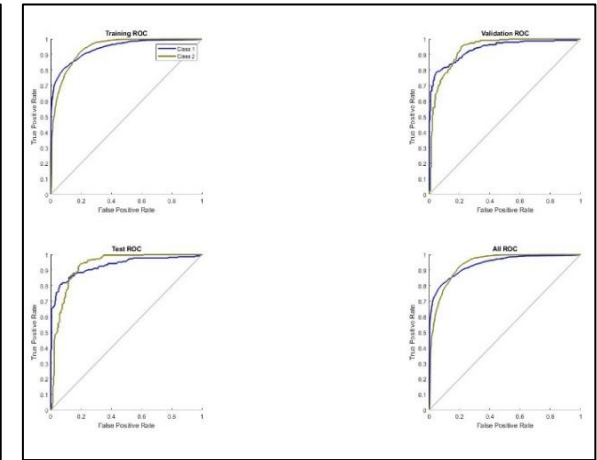
Gambar 4.13 (e1) (f1) Performance Run 5 dan 6 Training GDM HN 1, (e2) (f2) Confusion Matrix Run 5 dan 6 Training GDM HN 1, (e3) (f3) ROC Run 5 dan 6 Training GDM HN 1



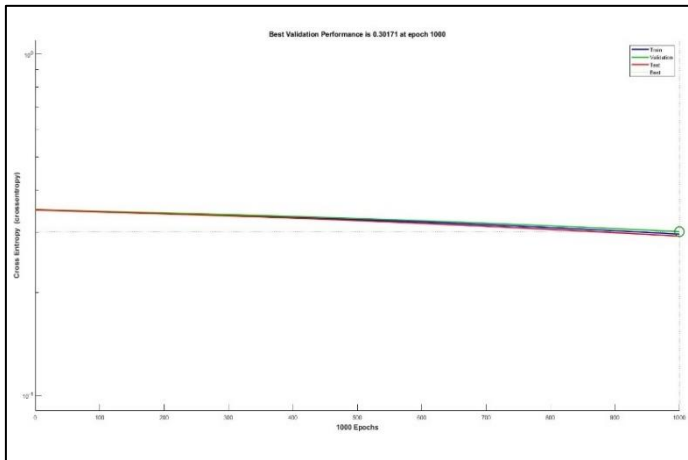
(g1)



(g2)



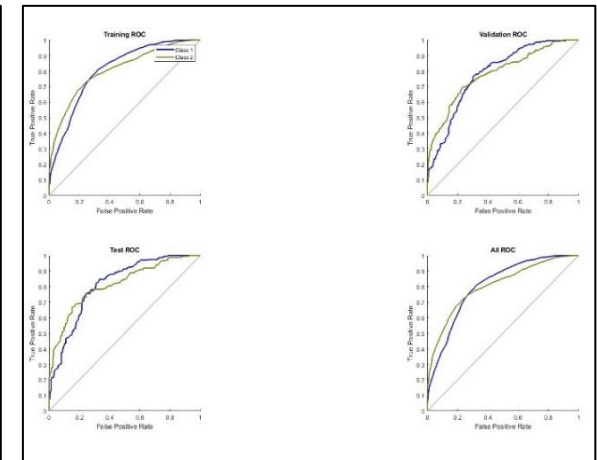
(g3)



(h1)

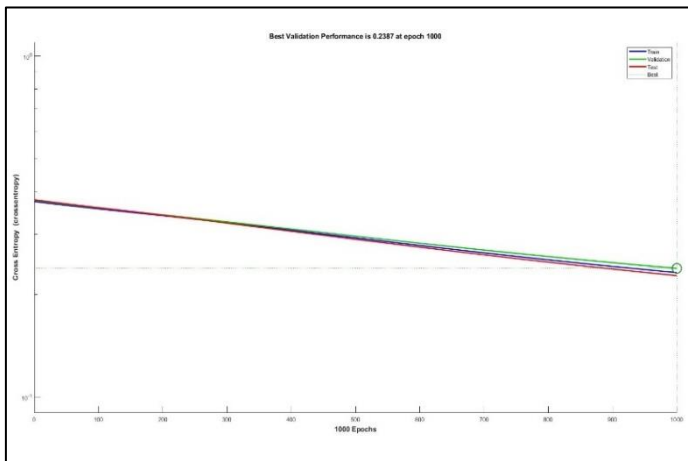


(h2)



(h3)

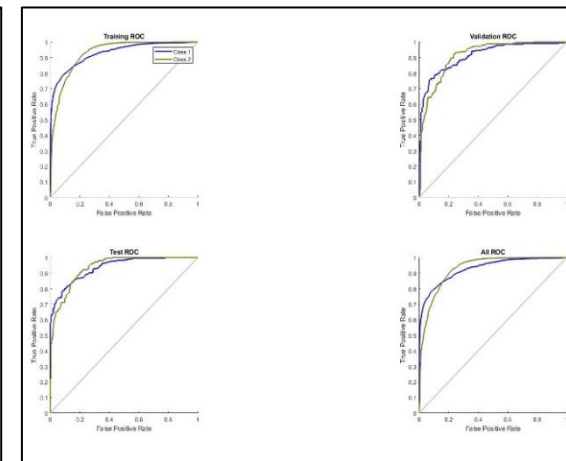
Gambar 4.13 (g1) (h1) Performance Run 7 dan 8 Training GDM HN 1, (g2) (h2) Confusion Matrix Run 7 dan 8 Training GDM HN 1, (g3) (h3) ROC Run 7 dan 8 Training GDM HN 1



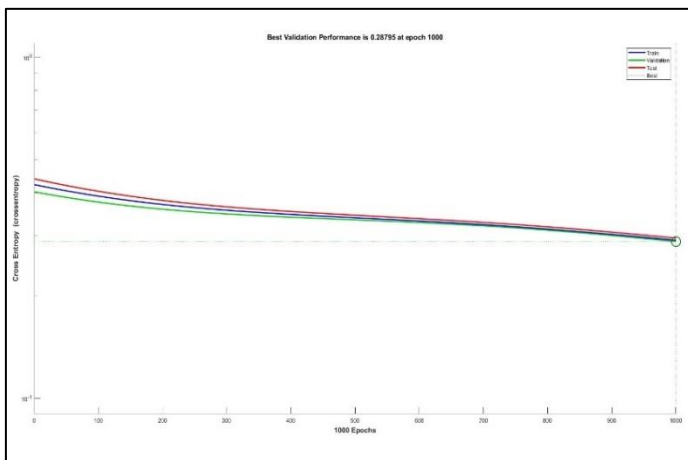
(i1)



(i2)



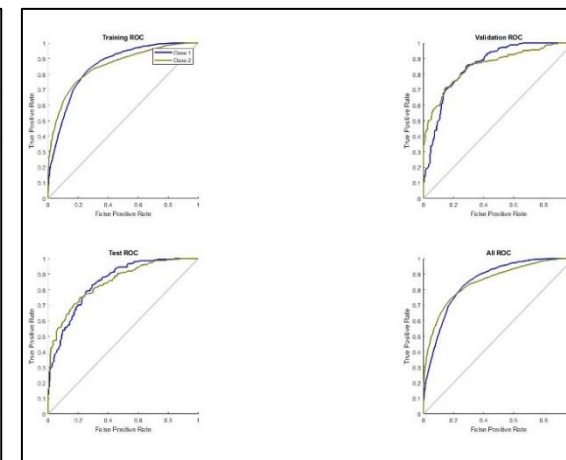
(i3)



(j1)



(j2)



(j3)

Gambar 4.13 (i1) (j1) Performance Run 9 dan 10 Training GDM HN 1, (i2) (j2) Confusion Matrix Run 9 dan 10 Training GDM HN 1, (i3) (j3) ROC Run 9 dan 10 Training GDM HN 1

Berdasarkan visualisasi hasil *training* di atas, dapat diketahui bahwa grafik *performance* terdiri dari 3 grafik, yaitu grafik biru adalah hasil *training*, grafik hijau adalah hasil validasi, dan grafik merah adalah hasil *testing*. Gambar 4.13 menampilkan sepuluh grafik *performane training* GDM *hidden neuron* 1 dimana rata-rata grafik tidak mengalami penurunan signifikan. Namun terdapat beberapa grafik yang penurunan errornya sangat landai, seperti pada *run* 2, 3, 5, dan 8. Dari uraian di atas, dapat disimpulkan bahwa semakin rendah nilai entropi dari grafik (nilai error) maka semakin baik performa dari model.

Hasil *training* yang menunjukkan akurasi dari sistem ditunjukkan oleh *confusion matrix* dimana warna hijau merepresentasikan citra yang berhasil diklasifikasikan ke dalam kelas secara benar dan warna merah adalah citra yang gagal diklasifikasikan dalam kelasnya. Perbandingan data yang digunakan untuk *10-run cross validation* yaitu 90% dari total data (8.591) digunakan sebagai data *training*, 5% dari total data (477) digunakan sebagai data validasi, dan 5% lainnya (477) digunakan sebagai data *testing*. Dari *confusion matrix* tersebut, diketahui bahwa akurasi *training* mencapai 60-80% sedangkan range akurasi validasi dan *testing* adalah 70-80%. Nilai *error training* > 16% (1.374) serta error validasi dan *testing* 16-30% (78-143). Hal ini menunjukkan bahwa model memiliki performa yang kurang baik

Visualisasi grafik lainnya adalah grafik *Receiver Operating Characteristic* (ROC). Fungsi dari grafik ROC ini adalah untuk mengetahui hasil evaluasi performa *machine learning*. Pada grafik ROC, terdapat dua warna grafik yaitu biru yang menunjukkan klasifikasi citra kelas 1 dan grafik hijau menunjukkan klasifikasi citra kelas 2. Pada *run* 3 dan 4 kurva *training*, validasi, dan *testing* menjauhi angka 1 dan semakin mendekati garis *threshold* sedangkan kurva *run* lainnya validasi dan *testing* berada pada angka 0,7. Dengan demikian, dapat disimpulkan bahwa performa *machine learning* (*Multilayer Perceptron* model GDM *hidden neuron* 1) menunjukkan performa tidak perfect.

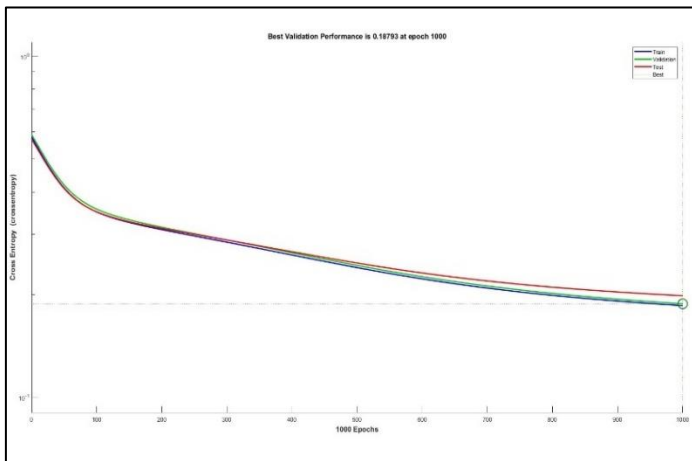
b. *Hidden Neuron 5*

Tabel 4.17 merupakan hasil akurasi training citra body fluids menggunakan model *Gradient Descent with Momentum* (GDM) *hidden neuron 5*.

Tabel 4. 17 Hasil Akurasi *Training* Menggunakan Model GDM *Hidden Neuron 5*

<i>Run</i>	<i>Epoch</i>		MLP-TrainGDM Hidden Neuron 5			<i>Elapsed Time</i>
	<i>All</i>	<i>Best</i>	<i>Acc Training</i>	<i>Acc Validation</i>	<i>Acc Testing</i>	
1	1000	1000	83,8%	82,8%	82,6%	00:00:07
2	1000	1000	84,8%	84,3%	84,3%	00:00:07
3	1000	1000	80,7%	82,0%	79,2%	00:00:07
4	1000	1000	81,1%	81,6%	83,9%	00:00:07
5	1000	1000	70,9%	69,8%	71,7%	00:00:07
6	1000	1000	85,4%	84,9%	86,4%	00:00:07
7	1000	1000	82,1%	83,0%	82,4%	00:00:07
8	1000	1000	84,9%	86,0%	86,0%	00:00:07
9	1000	1000	84,9%	87,4%	83,6%	00:00:07
10	1000	1000	86,1%	86,8%	82,4%	00:00:07
Average			82,5%	82,9%	82,3%	00:00:07
Standar Deviasi			0,0446941	0,050016	0,04227	

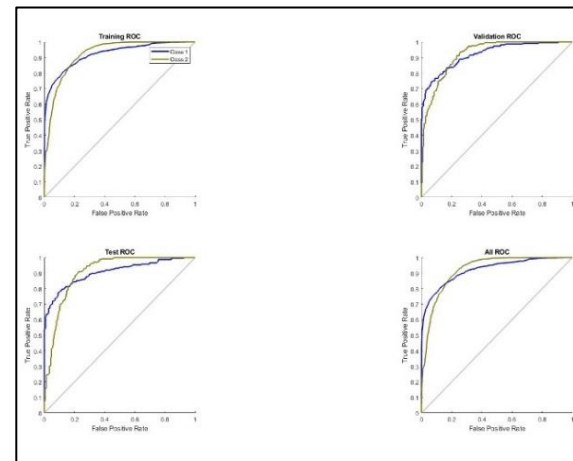
Berdasarkan Tabel 4.17 diketahui bahwa rata-rata akurasi *training* menggunakan model GDM dengan *hidden neuron 5* adalah 70-86%. Hasil akurasi *training* terbaik terdapat pada *run* 10 dengan akurasi *training* sebesar 86,1%, akurasi validasi 86,8%, dan akurasi *testing* 82,4%. Akurasi *training* terbaik ini diperoleh pada *epoch* ke 1000 dari total 1000 *epoch* dengan waktu *training* selama 7 detik. Sementara itu, hasil akurasi *training* terendah terdapat pada *run* 5 dengan akurasi *training* 70,9%, akurasi validasi 69,8%, dan akurasi *testing* 71,7%. Dari pengujian diatas, diperoleh hasil bahwa rata-rata akurasi *training* adalah 82,5%, akurasi validasi 82,9%, dan akurasi *testing* 82,3%. Rata-rata waktu *training* menggunakan model GDM *hidden neuron 5* adalah 7 detik. Visualisasi hasil *training* dapat dilihat pada Gambar 4.14.



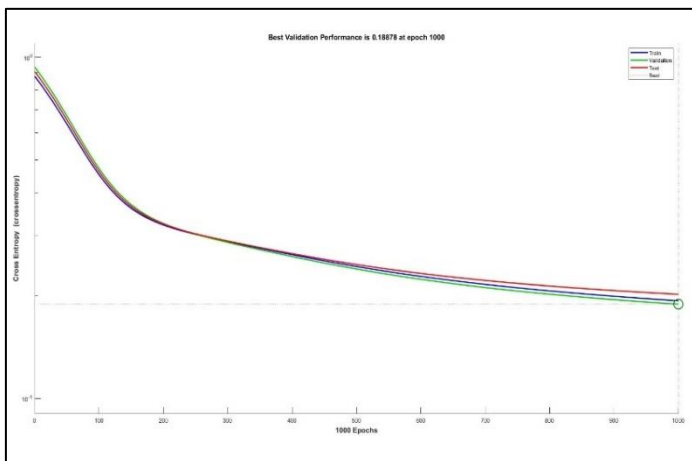
(a1)



(a2)



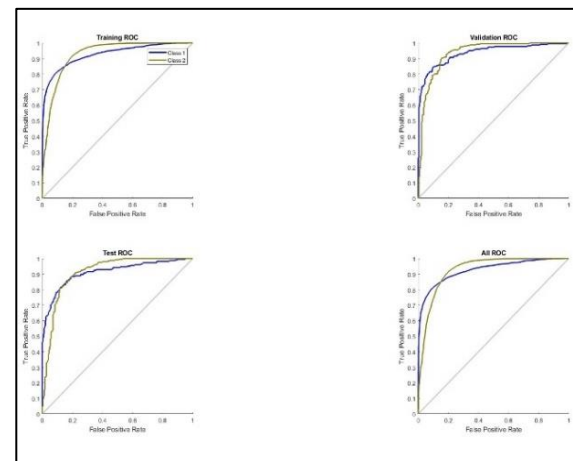
(a3)



(b1)

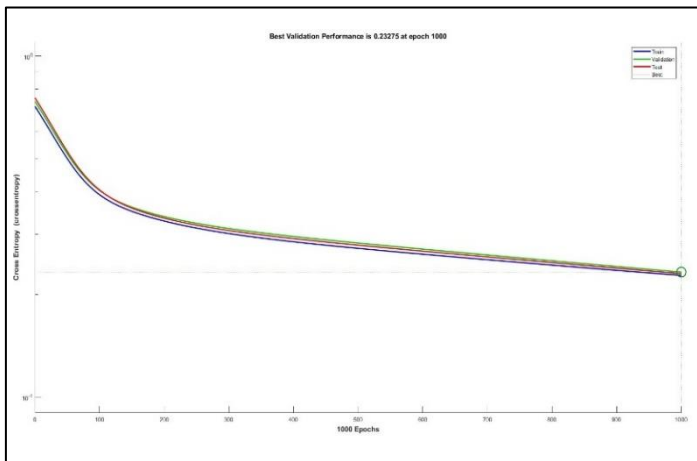


(b2)



(b3)

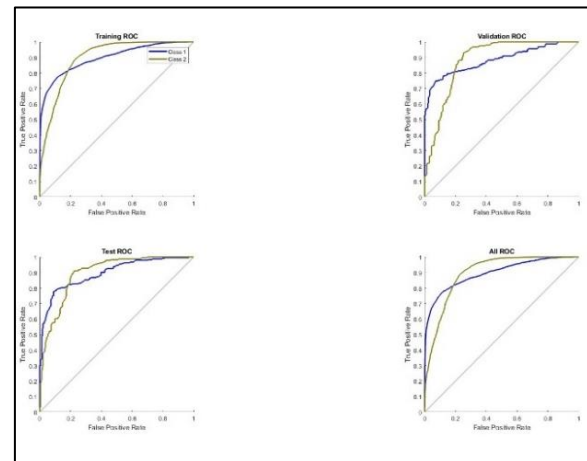
Gambar 4. 14 (a1) (b1) Performance Run 1 dan 2 Training GDM HN 5, (a2) (b2) Confusion Matrix Run 1 dan 2 Training GDM HN 5, (a3) (b3) ROC Run 1 dan 2 Training GDM HN 5



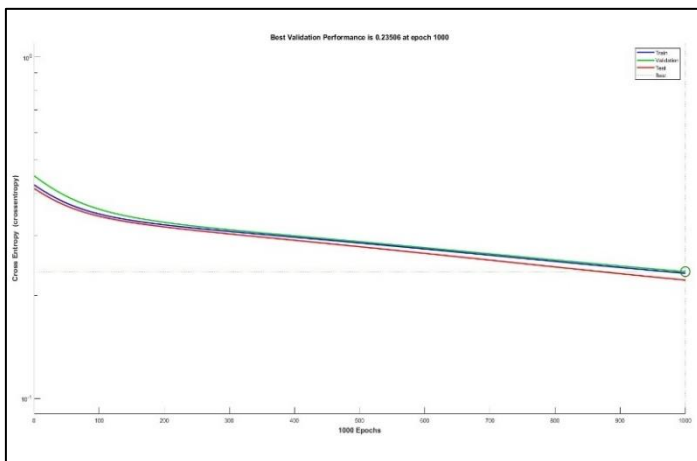
(c1)



(c2)



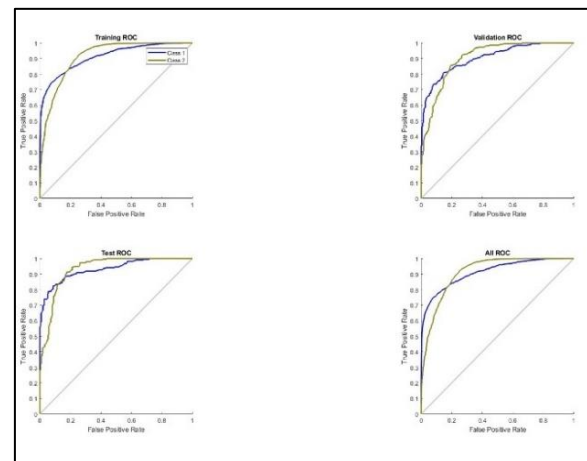
(c3)



(d1)

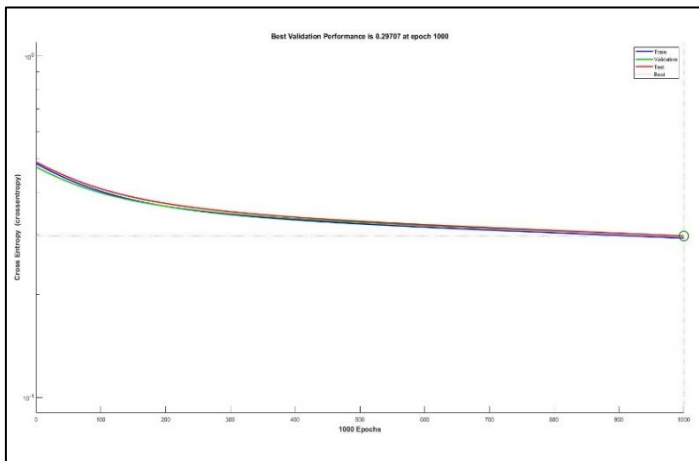


(d2)



(d3)

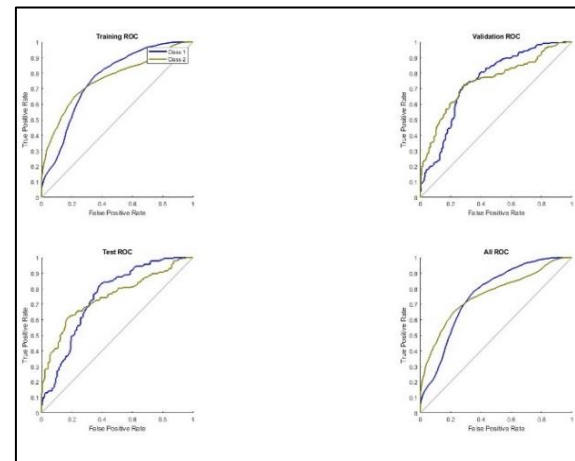
Gambar 4.14 (c1) (d1) Performance Run 3 dan 4 Training GDM HN 5, (c2) (d2) Confusion Matrix Run 3 dan 4 Training GDM HN 5, (c3) (d3) ROC Run 3 dan 4 Training GDM HN 5



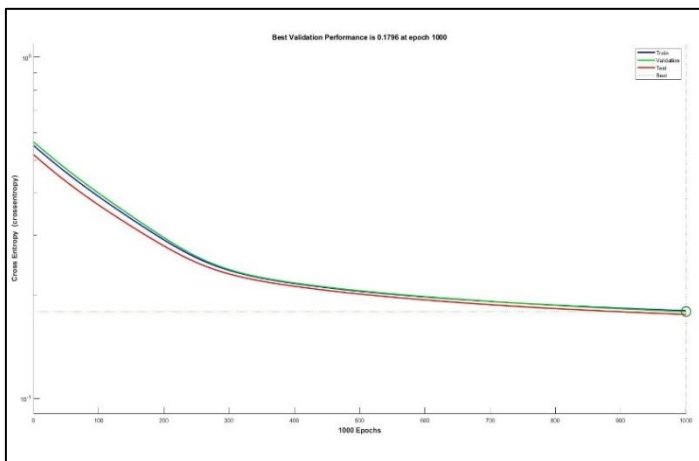
(e1)



(e2)



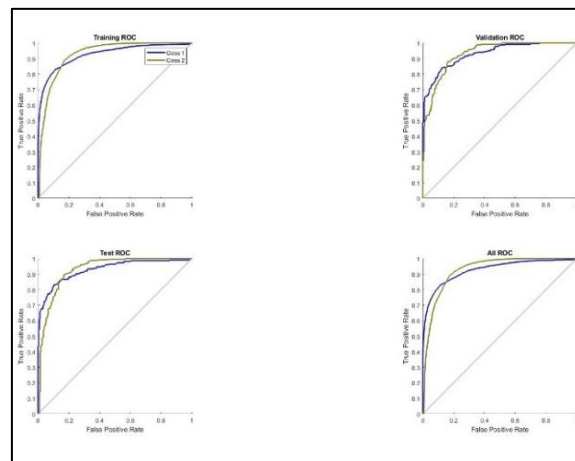
(e3)



(f1)

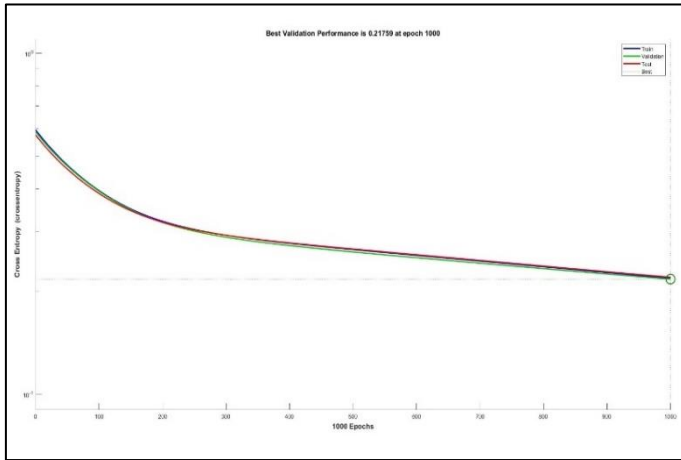


(f2)



(f3)

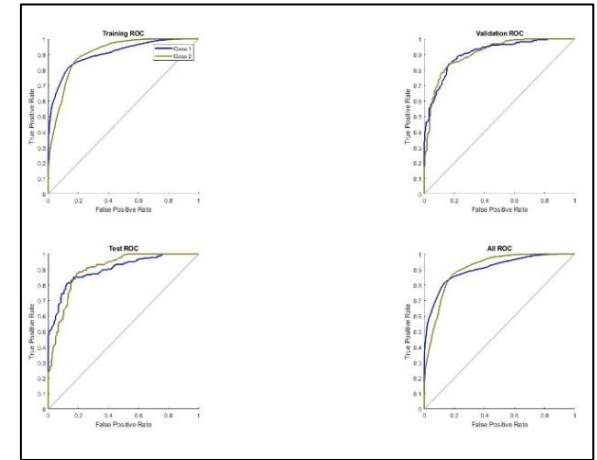
Gambar 4.14 (e1) (f1) Performance Run 5 dan 6 Training GDM HN 5, (e2) (d2) Confusion Matrix Run 5 dan 6 Training GDM HN 5, (e3) (f3) ROC Run 5 dan 6 Training GDM HN 5



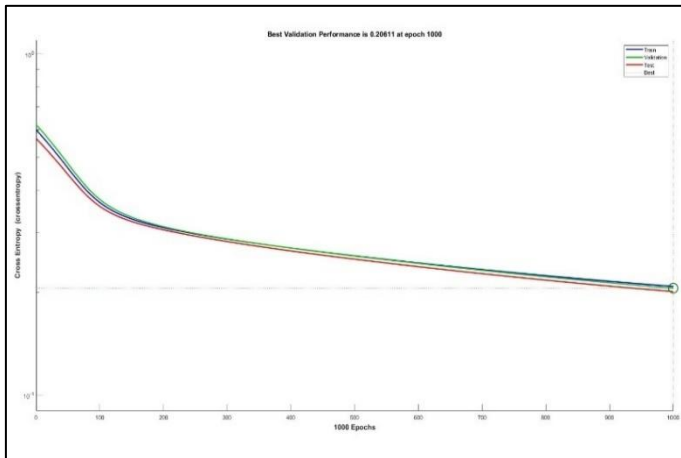
(g1)



(g2)



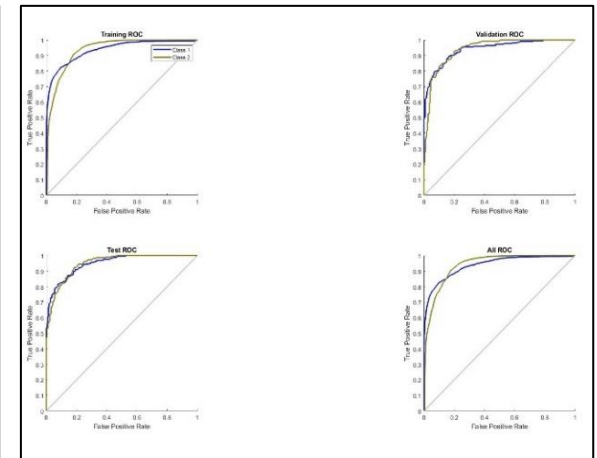
(g3)



(h1)

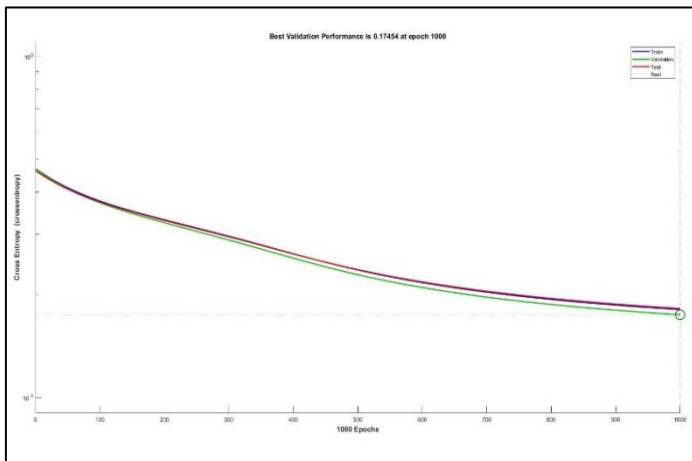


(h2)



(h3)

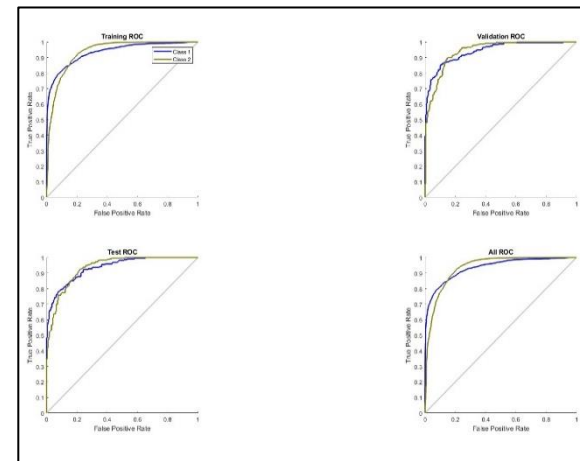
Gambar 4.14 (g1) (h1) Performance Run 7 dan 8 Training GDM HN 5, (g2) (h2) Confusion Matrix Run 7 dan 8 Training GDM HN 5, (g3) (h3) ROC Run 7 dan 8 Training GDM HN 5



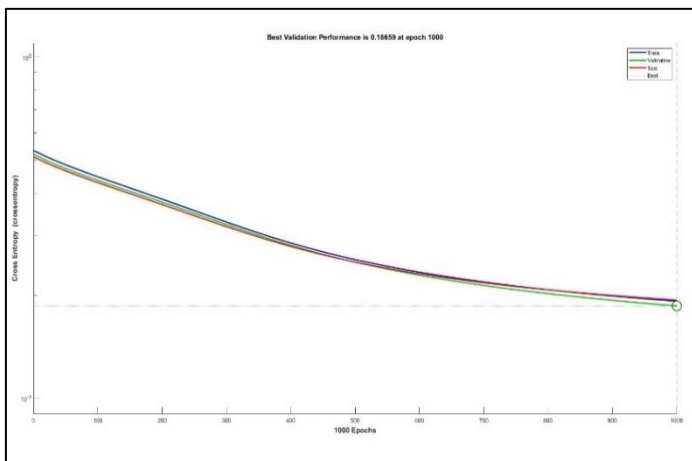
(i1)



(i2)



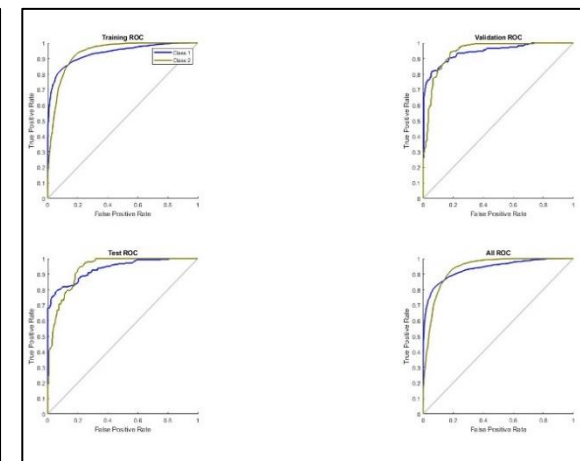
(i3)



(j1)



(j2)



(j3)

Gambar 4.14 (i1) (j1) Performance Run 9 dan 10 Training GDM HN 5, (i2) (j2) Confusion Matrix Run 9 dan 10 Training GDM HN 5, (i3) (j3) ROC Run 9 dan 10 Training GDM HN 5

Berdasarkan visualisasi hasil *training* di atas, dapat diketahui bahwa grafik *performance* terdiri dari 3 grafik, yaitu grafik biru adalah hasil *training*, grafik hijau adalah hasil validasi, dan grafik merah adalah hasil *testing*. Gambar 4.14 menampilkan sepuluh grafik *performane training* GDM *hidden neuron* 5 dimana grafik tersebut menunjukkan penurunan error terjadi signifikan pada *range epoch* 0-100, kecuali *run* 4,5,9, dan 10. Penurunan error landai terjadi pada *range epoch* 200 sampai maksimalnya, yaitu terlihat pada *run* 1 sampai 3 dan *run* 6 sampai 8. Pada *run* 4, 5, 9, dan 10 grafik tidak mengalami penurunan signifikan dari awal hingga *epoch* maksimalnya.

Hasil *training* yang menunjukkan akurasi dari sistem ditunjukkan oleh *confusion matrix* dimana warna hijau merepresentasikan citra yang berhasil diklasifikasikan ke dalam kelas secara benar dan warna merah adalah citra yang gagal diklasifikasikan dalam kelasnya. Perbandingan data yang digunakan untuk *10-run cross validation* yaitu 90% dari total data (8.591) digunakan sebagai data *training*, 5% dari total data (477) digunakan sebagai data validasi, dan 5% lainnya (477) digunakan sebagai data *testing*. Dari *confusion matrix* diketahui akurasi *training* $\geq 80\%$ dan akurasi validasi dan *testing* 70-80%. Nilai *error training* $\leq 30\%$ (2.577) serta rata-rata *error* validasi dan *testing* adalah $\leq 30\%$ (143). Hal ini menunjukkan bahwa model memiliki performa yang kurang baik.

Visualisasi grafik lainnya adalah grafik *Receiver Operating Characteristic* (ROC). Fungsi dari grafik ROC ini adalah untuk mengetahui hasil evaluasi performa *machine learning*. Pada grafik ROC, terdapat dua warna grafik yaitu biru yang menunjukkan klasifikasi citra kelas 1 dan grafik hijau menunjukkan klasifikasi citra kelas 2. Kurva grafik terlihat mendekati angka 1 kecuali pada *run* 3 dan 5. Pada *run* 3 dan 5, grafik berada di tengah atau mendekati garis *threshold*. Dengan demikian, dapat disimpulkan bahwa performa *machine learning* (*Multilayer Perceptron* model GDM *hidden neuron* 5) menunjukkan hasil performa tidak *perfect*.

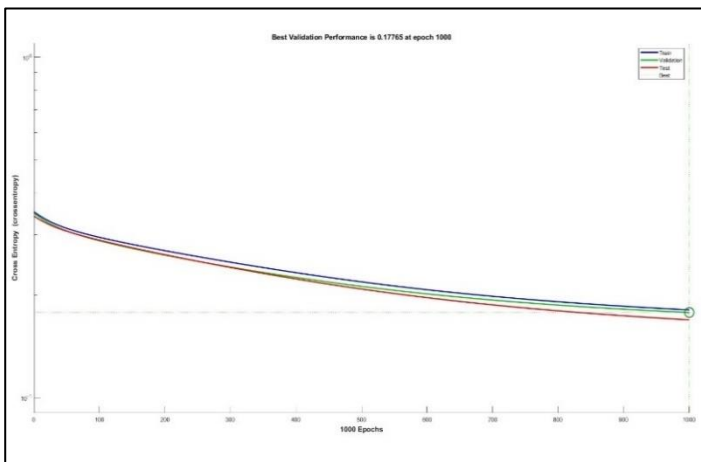
c. *Hidden Neuron 10*

Tabel 4.18 merupakan hasil akurasi *training* citra *body fluids* menggunakan model *Gradient Descent with Momentum* (GDM) *hidden neuron 10*.

Tabel 4. 18 Hasil Akurasi *Training* Menggunakan Model GDM *Hidden Neuron 10*

<i>Run</i>	<i>Epoch</i>		MLP-TrainGDM <i>Hidden Neuron 10</i>			<i>Elapsed Time</i>
	<i>All</i>	<i>Best</i>	<i>Acc Training</i>	<i>Acc Validation</i>	<i>Acc Testing</i>	
1	1000	1000	85,4%	86,2%	88,5%	00:00:08
2	1000	1000	88,1%	87,0%	85,7%	00:00:08
3	1000	1000	87,0%	89,7%	85,5%	00:00:08
4	1000	1000	84,5%	86,0%	85,5%	00:00:08
5	1000	1000	84,9%	85,5%	83,9%	00:00:08
6	1000	1000	84,7%	84,1%	84,1%	00:00:08
7	1000	1000	83,8%	82,8%	80,3%	00:00:08
8	1000	1000	84,8%	84,1%	84,7%	00:00:08
9	1000	1000	87,3%	89,3%	85,7%	00:00:08
10	1000	1000	82,8%	80,9%	84,3%	00:00:08
Average			85,3%	85,6%	84,8%	00:00:08
Standar Deviasi			0,0165466	0,0273666	0,020574	

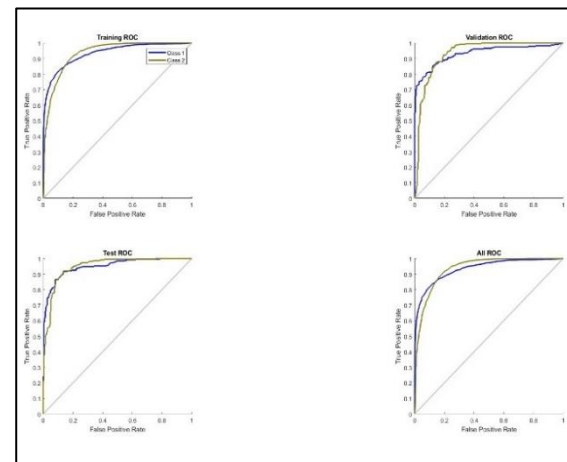
Berdasarkan Tabel 4.18 diketahui bahwa rata-rata akurasi *training* menggunakan model GDM dengan *hidden neuron 10* adalah 82-88%. Hasil akurasi *training* terbaik terdapat pada *run 2* dengan akurasi *training* sebesar 88,1%, akurasi validasi 87,0%, dan akurasi *testing* 85,7%. Akurasi *training* terbaik ini diperoleh pada *epoch* ke 1000 dari total 1000 *epoch* dengan waktu *training* selama 8 detik. Sementara itu, hasil akurasi *training* terendah terdapat pada *run 10* dengan akurasi *training* 82,8%, akurasi validasi 80,9%, dan akurasi *testing* 84,3%. Dari pengujian diatas, diperoleh hasil bahwa rata-rata akurasi *training* adalah 85,3%, akurasi validasi 85,6%, dan akurasi *testing* 84,8%. Rata-rata waktu *training* menggunakan model GDM *hidden neuron 10* adalah 8 detik. Visualisasi hasil *training* dapat dilihat pada Gambar 4.13.



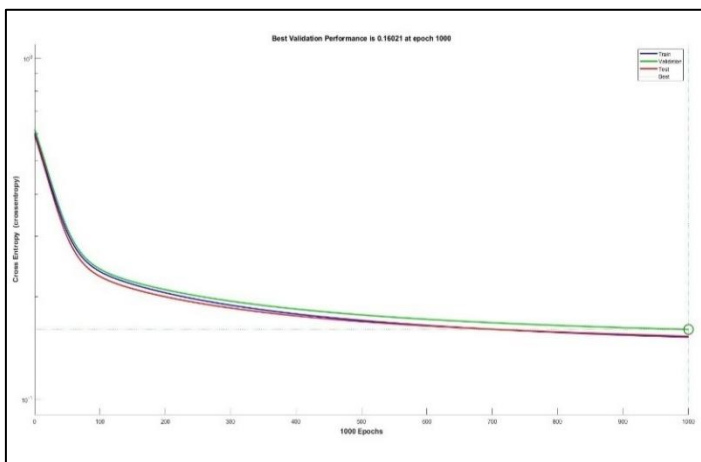
(a1)



(a2)



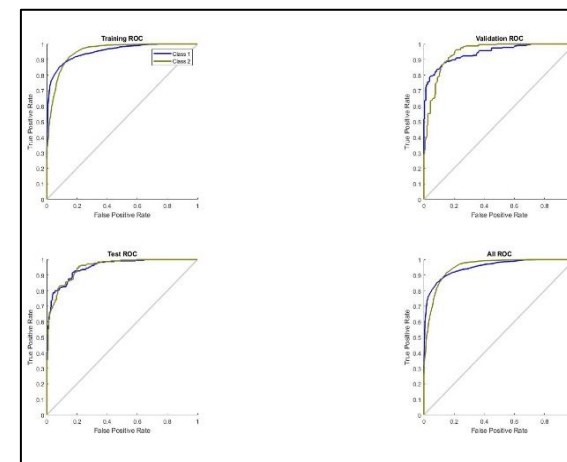
(a3)



(b1)

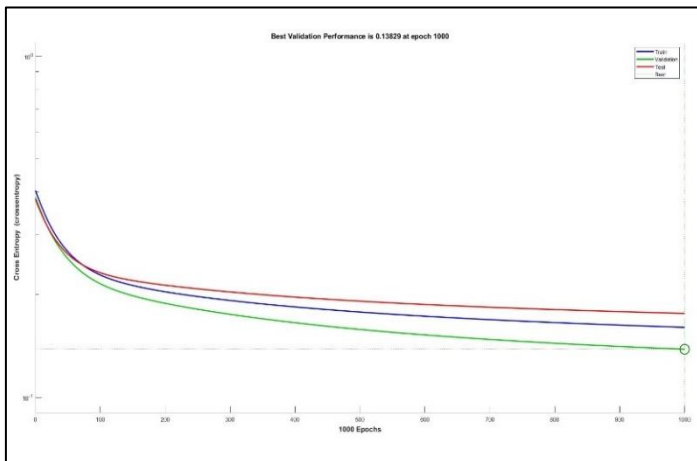


(b2)



(b3)

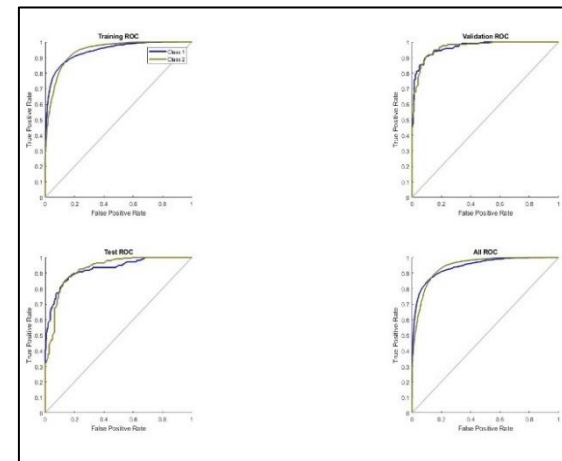
Gambar 4. 15 (a1) (b1) Performance Run 1 dan 2 Training GDM HN 10, (a2) (b2) Confusion Matrix Run 1 dan 2 Training GDM HN 10, (a3) (b3) ROC Run 1 dan 2 Training GDM HN 10



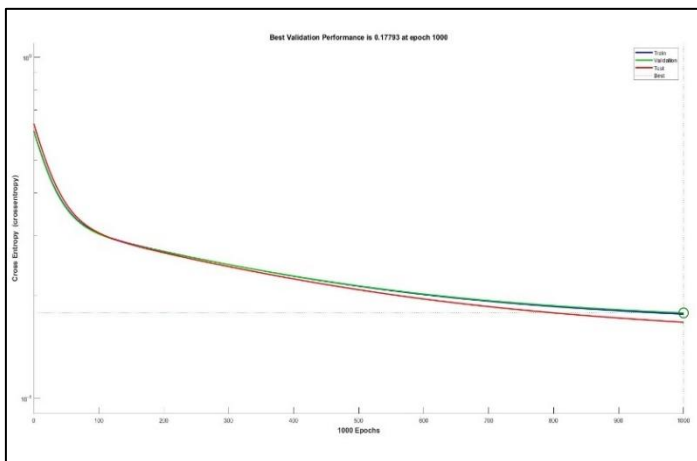
(c1)



(c2)



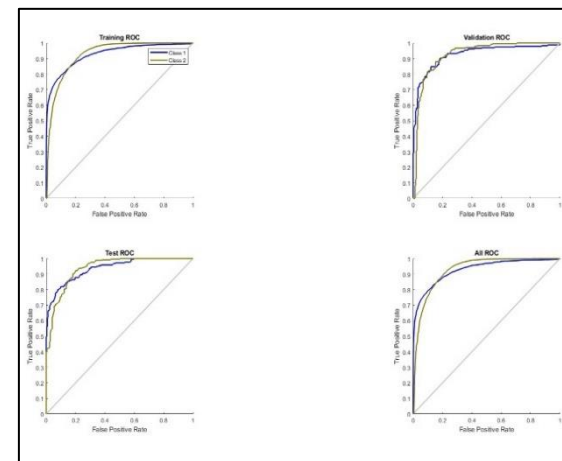
(c3)



(d1)

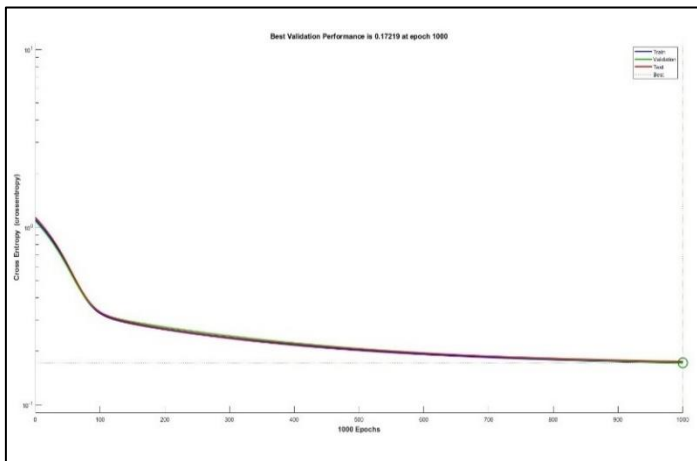


(d2)



(d3)

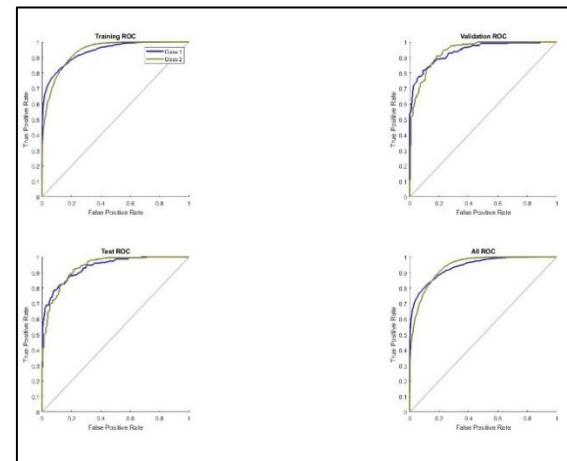
Gambar 4.15 (c1) (d1) Performance Run 3 dan 4 Training GDM HN 10, (c2) (d2) Confusion Matrix Run 3 dan 4 Training GDM HN 10, (c3) (d3) ROC Run 3 dan 4 Training GDM HN 10



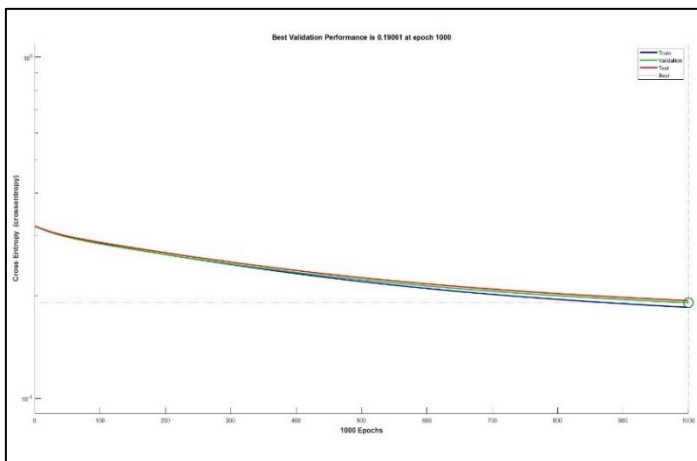
(e1)



(e2)



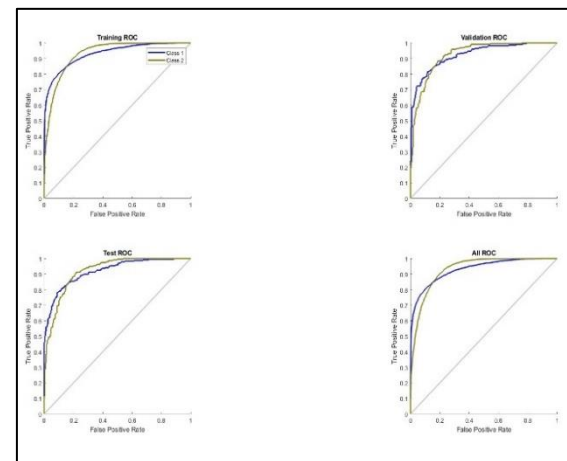
(e3)



(f1)

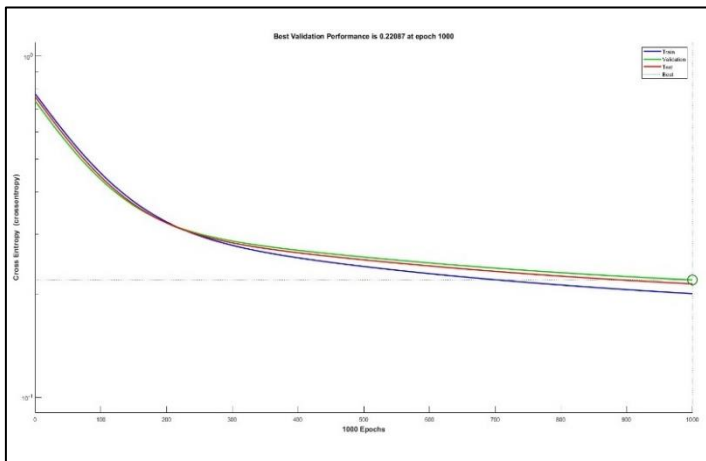


(f2)



(f3)

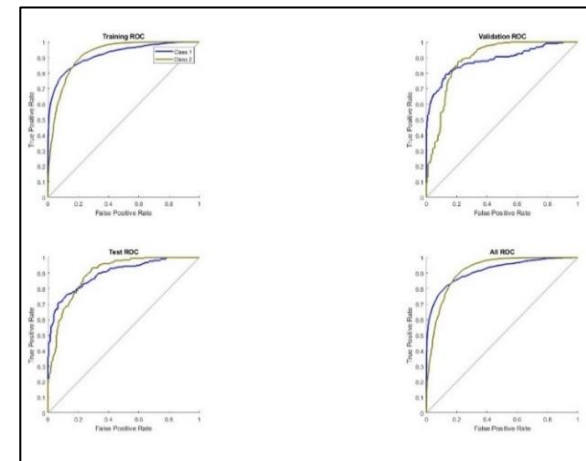
Gambar 4.15 (e1) (f1) Performance Run 5 dan 6 Training GDM HN 10, (e2) (f2) Confusion Matrix Run 5 dan 6 Training GDM HN 10, (e3) (f3) ROC Run 5 dan 6 Training GDM HN 10



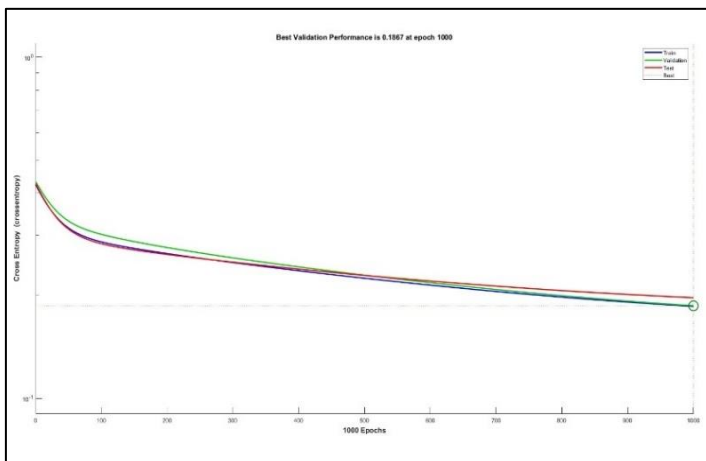
(g1)



(g2)



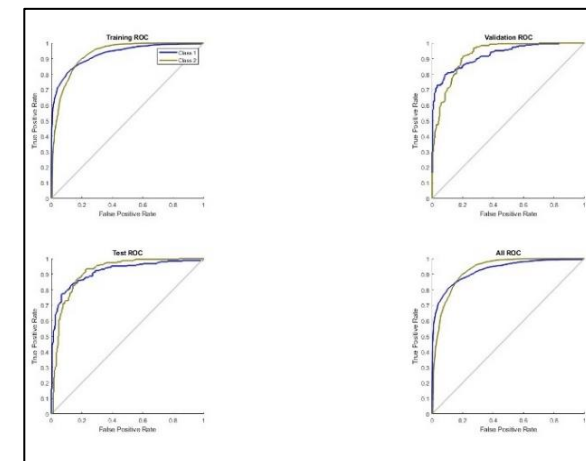
(g3)



(h1)

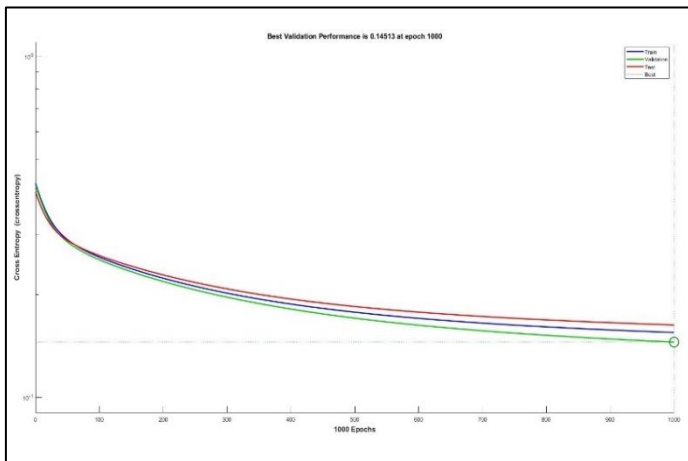


(h2)



(h3)

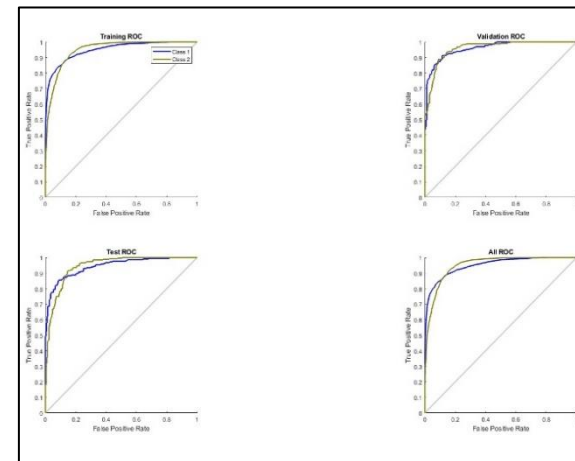
Gambar 4.15 (g1) (h1) Performance Run 7 dan 8 Training GDM HN 10, (g2) (h2) Confusion Matrix Run 7 dan 8 Training GDM HN 10, (g3) (h3) ROC Run 7 dan 8 Training GDM HN 10



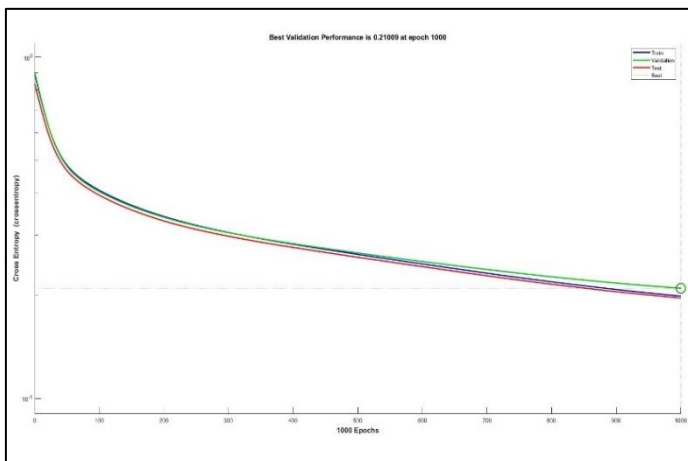
(i1)



(i2)



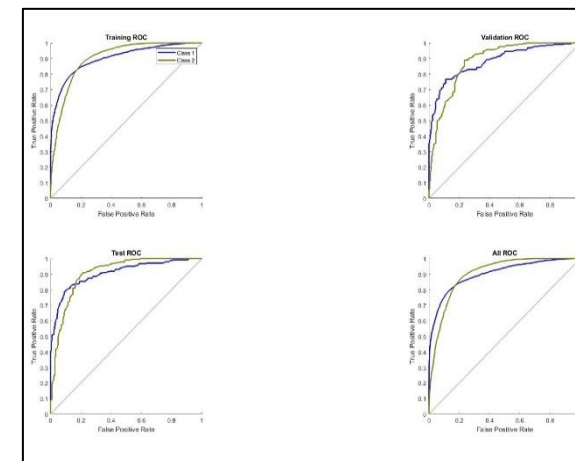
(i3)



(j1)



(j2)



(j3)

Gambar 4.15 (i1) (j1) Performance Run 9 dan 10 Training GDM HN 10, (i2) (j2) Confusion Matrix Run 9 dan 10 Training GDM HN 10, (i3) (j3) ROC Run 9 dan 10 Training GDM HN 10

Berdasarkan visualisasi hasil *training* di atas, dapat diketahui bahwa grafik *performance* terdiri dari 3 grafik, yaitu grafik biru merepresentasikan hasil *training*, grafik hijau merepresentasikan hasil validasi, dan grafik merah merepresentasikan hasil *testing*. Gambar 4.15 menampilkan sepuluh grafik *performance training* GDM *hidden neuron* 10 dimana grafik tersebut menunjukkan penurunan error terjadi signifikan pada *range epoch* 0-100, kecuali *run* 1 dan 6. Penurunan error yang landai terjadi pada *range epoch* 150 sampai *epoch* maksimalnya, yaitu terlihat pada *run* 1 sampai 5 dan *run* 7 sampai 10. Pada *run* 1 dan 6, grafik tidak mengalami penurunan signifikan dari awal hingga *epoch* maksimalnya.

Hasil *training* yang menunjukkan akurasi dari sistem ditunjukkan oleh *confusion matrix* dimana warna hijau merepresentasikan citra yang berhasil diklasifikasikan ke dalam kelas secara benar dan warna merah adalah citra yang gagal diklasifikasikan dalam kelasnya. Perbandingan data yang digunakan untuk *10-run cross validation* yaitu 90% dari total data (8.591) digunakan sebagai data *training*, 5% dari total data (477) digunakan sebagai data validasi, dan 5% lainnya (477) digunakan sebagai data *testing*. Dari *confusion matrix* tersebut, diketahui bahwa akurasi *training*, validasi, dan *testing* $\geq 80\%$. Nilai *error training* $\leq 17\%$ (1.460) serta *error* validasi dan *testing* adalah $< 19\%$ dari total data (90). Hal ini menunjukkan bahwa model memiliki performa yang tidak cukup baik

Visualisasi grafik lainnya adalah grafik *Receiver Operating Characteristic* (ROC). Fungsi dari grafik ROC ini adalah untuk mengetahui hasil evaluasi performa *machine learning*. Pada grafik ROC, terdapat dua warna grafik yaitu biru yang menunjukkan klasifikasi citra kelas 1 dan grafik hijau menunjukkan klasifikasi citra kelas 2. Kurva grafik berada pada angka 0,8 dan menjauhi garis *threshold*. Dengan demikian, dapat disimpulkan bahwa performa *machine learning* (*Multilayer Perceptron* model GDM *hidden neuron* 10) menunjukkan performa tidak perfect.

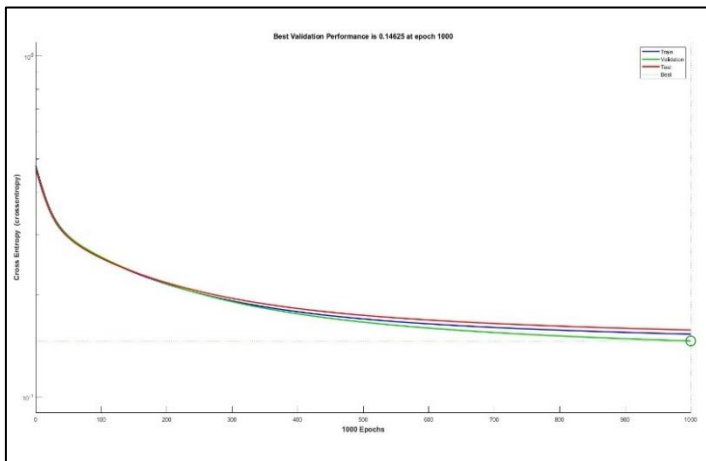
d. *Hidden Neuron 15*

Tabel 4.19 merupakan hasil akurasi *training* citra *body fluids* menggunakan model *Gradient Descent with Momentum* (GDM) *hidden neuron 15*.

Tabel 4. 19 Hasil Akurasi *Training* Menggunakan Model GDM *Hidden Neuron 15*

<i>Run</i>	<i>Epoch</i>		MLP-TrainGDM <i>Hidden Neuron 15</i>			<i>Elapsed Time</i>
	<i>All</i>	<i>Best</i>	<i>Acc Training</i>	<i>Acc Validation</i>	<i>Acc Testing</i>	
1	1000	1000	87,2%	87,8%	85,1%	00:00:09
2	1000	1000	85,5%	87,2%	85,3%	00:00:09
3	1000	1000	85,3%	87,8%	86,0%	00:00:09
4	1000	1000	83,3%	83,4%	81,3%	00:00:09
5	1000	1000	86,0%	87,2%	87,0%	00:00:10
6	1000	1000	86,9%	86,2%	84,9%	00:00:09
7	1000	1000	86,3%	85,3%	85,7%	00:00:09
8	1000	1000	87,5%	87,6%	87,6%	00:00:09
9	1000	1000	85,1%	86,8%	87,6%	00:00:09
10	1000	1000	84,9%	85,5%	85,3%	00:00:09
Average			85,8%	86,5%	85,6%	00:00:09
Standar Deviasi			0,0125786	0,0140618	0,018116	

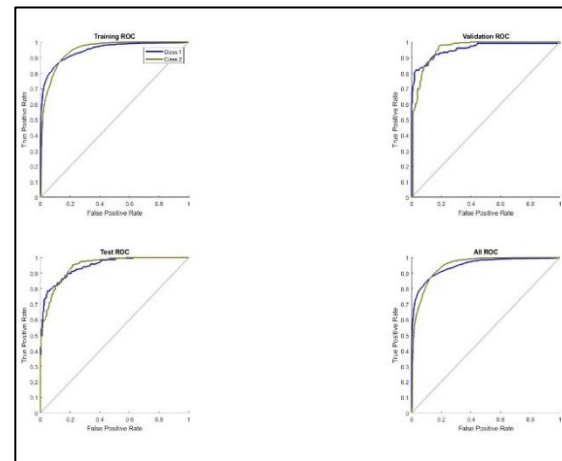
Berdasarkan Tabel 4.19 diketahui bahwa rata-rata akurasi *training* menggunakan model GDM dengan *hidden neuron 15* adalah 83-87%. Hasil akurasi *training* terbaik terdapat pada *run 8* dengan akurasi *training* sebesar 87,5%, akurasi validasi 87,6%, dan akurasi *testing* 87,6%. Akurasi *training* terbaik ini diperoleh pada *epoch* ke 1000 dari total 1000 *epoch* dengan waktu *training* selama 9 detik. Sementara itu, hasil akurasi *training* terendah terdapat pada *run 4* dengan akurasi *training* 83,3%, akurasi validasi 83,4%, dan akurasi *testing* 81,3%. Dari pengujian diatas, diperoleh hasil bahwa rata-rata akurasi *training* adalah 85,8%, akurasi validasi 86,5%, dan akurasi *testing* 85,6%. Rata-rata waktu *training* menggunakan model GDM *hidden neuron 15* adalah 9 detik. Visualisasi hasil *training* dapat dilihat pada Gambar 4.16.



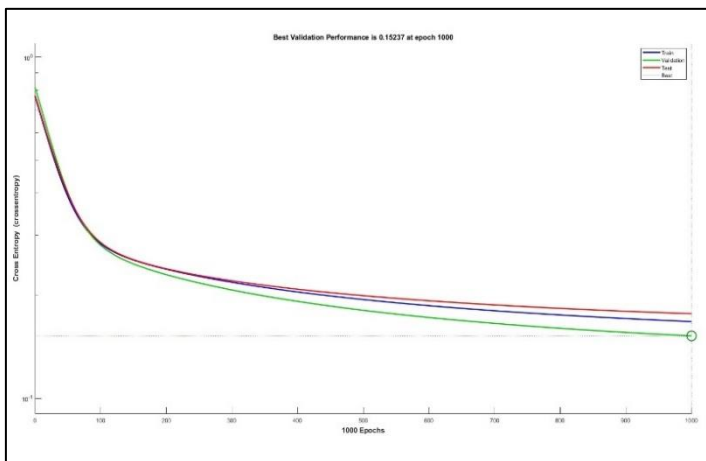
(a1)



(a2)



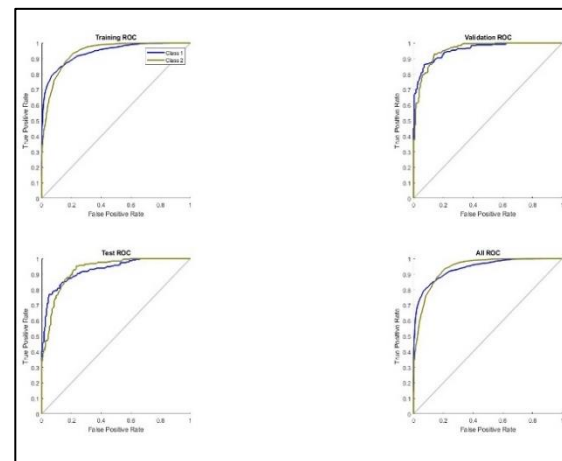
(a3)



(b1)

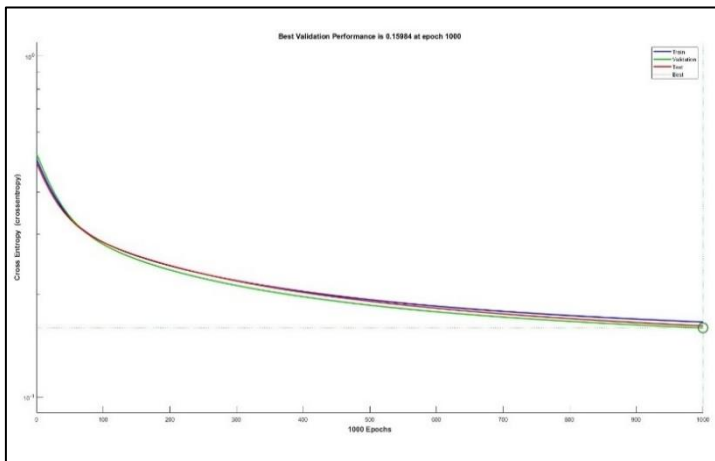


(b2)



(b3)

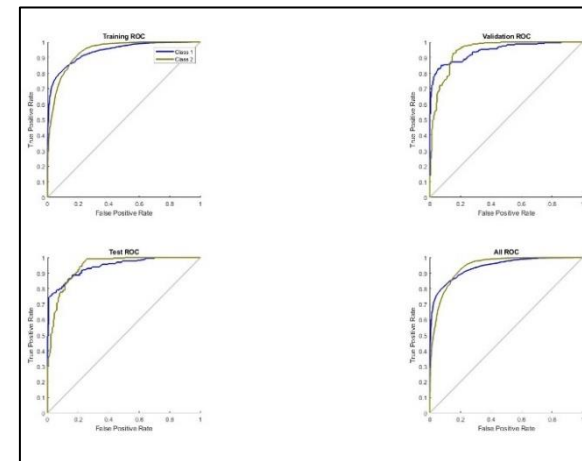
Gambar 4. 16 (a1) (b1) Performance Run 1 dan 2 Training GDM HN 15, (a2) (b2) Confusion Matrix Run 1 dan 2 Training GDM HN 15, (a3) (b3) ROC Run 1 dan 2 Training GDM HN 15



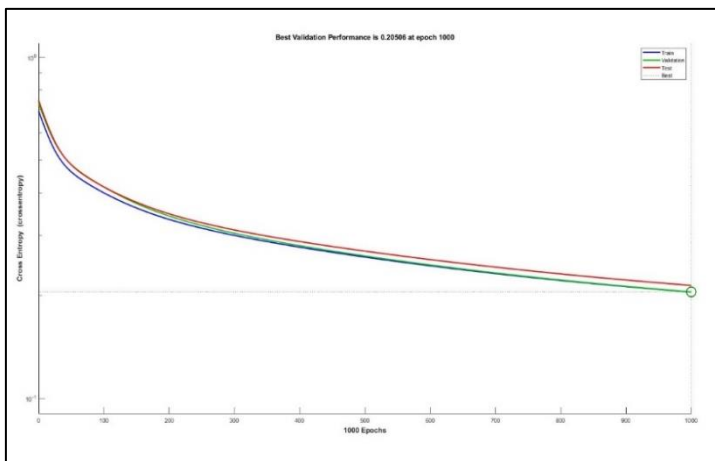
(c1)



(c2)



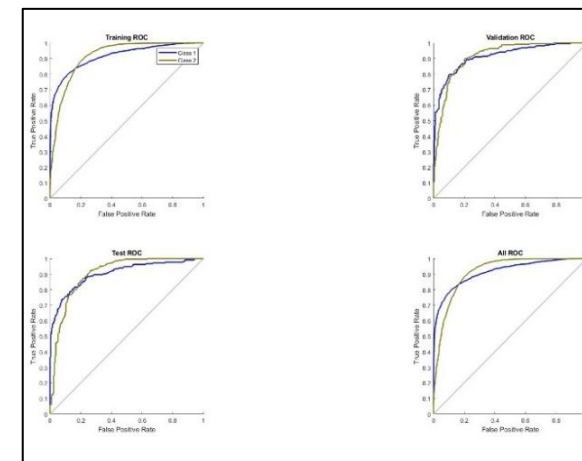
(c3)



(d1)

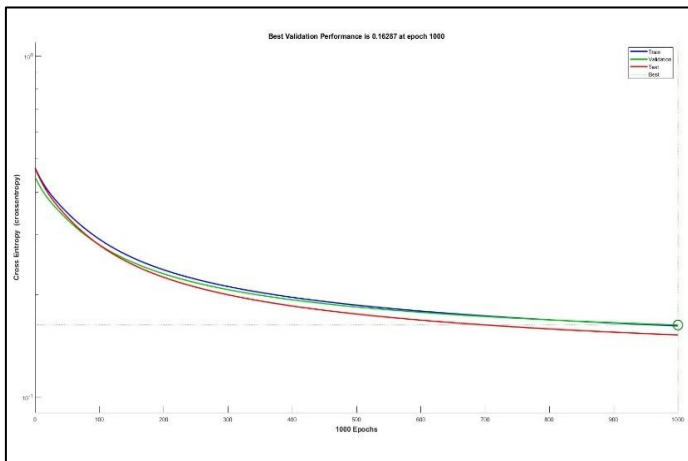


(d2)



(d3)

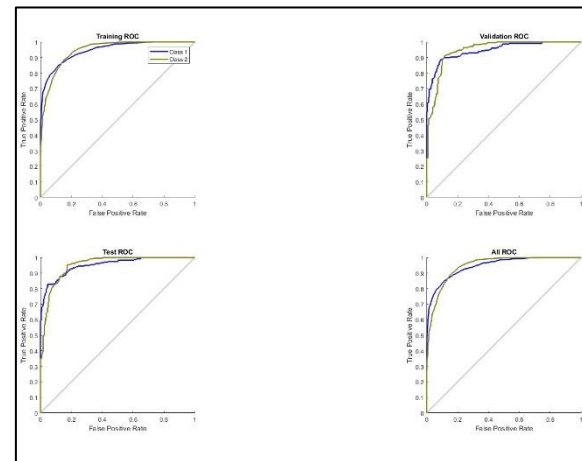
Gambar 4.16 (c1) (d1) Performance Run 3 dan 4 Training GDM HN 15, (c2) (d2) Confusion Matrix Run 3 dan 4 Training GDM HN 15, (c3) (d3) ROC Run 3 dan 4 Training GDM HN 15



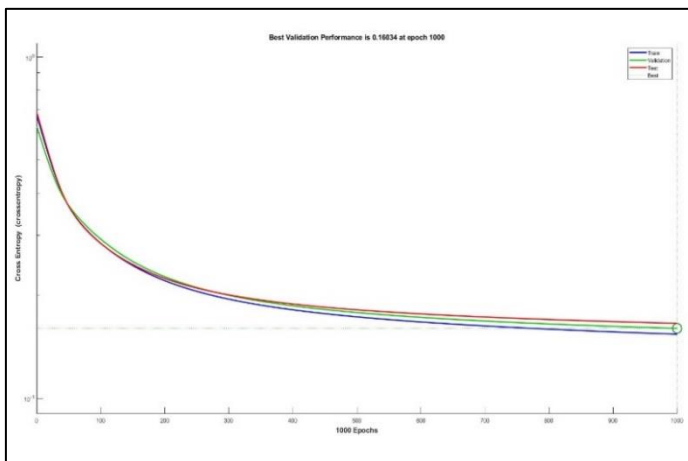
(e1)



(e2)



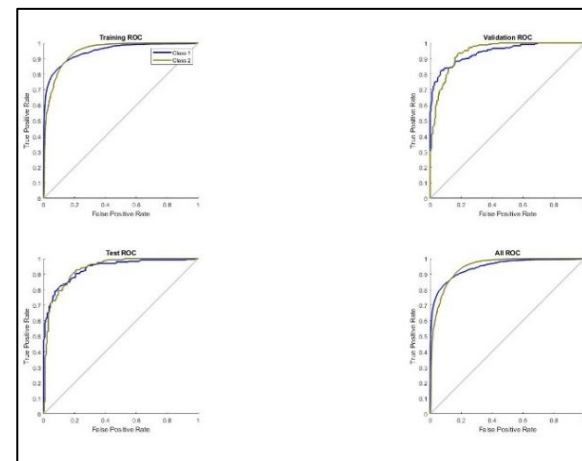
(e3)



(f1)

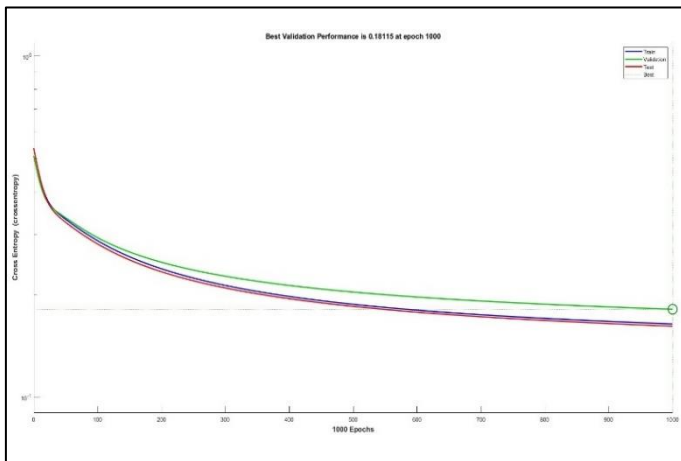


(f2)



(f3)

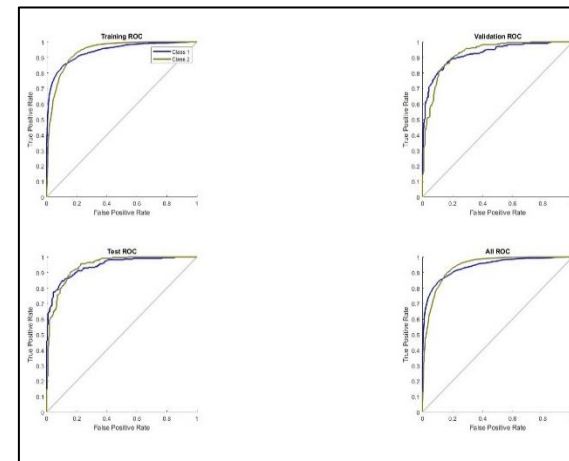
Gambar 4.16 (e1) (f1) Performance Run 5 dan 6 Training GDM HN 15, (e2) (f2) Confusion Matrix Run 5 dan 6 Training GDM HN 15, (e3) (f3) ROC Run 5 dan 6 Training GDM HN 15



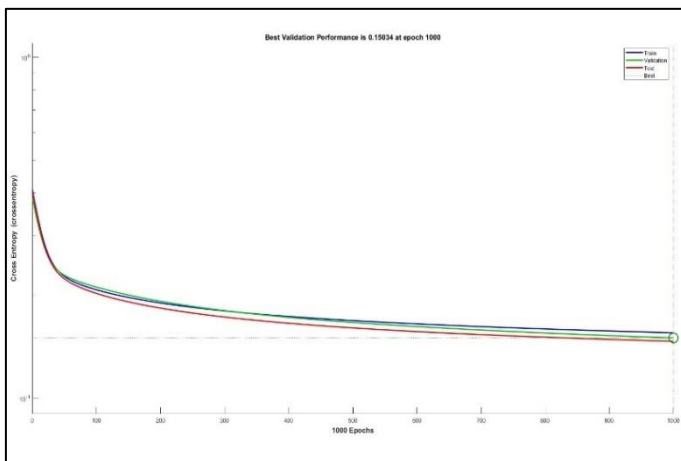
(g1)



(g2)



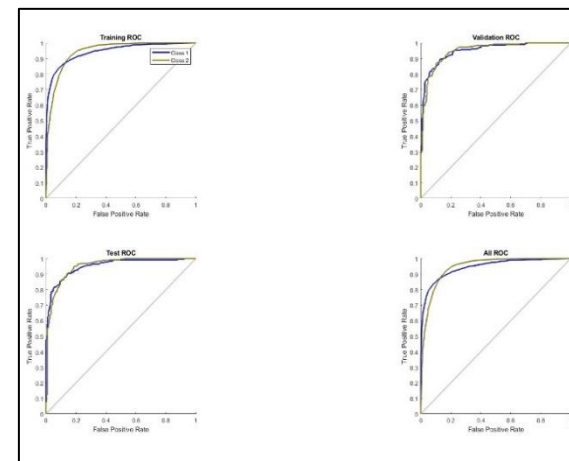
(g3)



(h1)

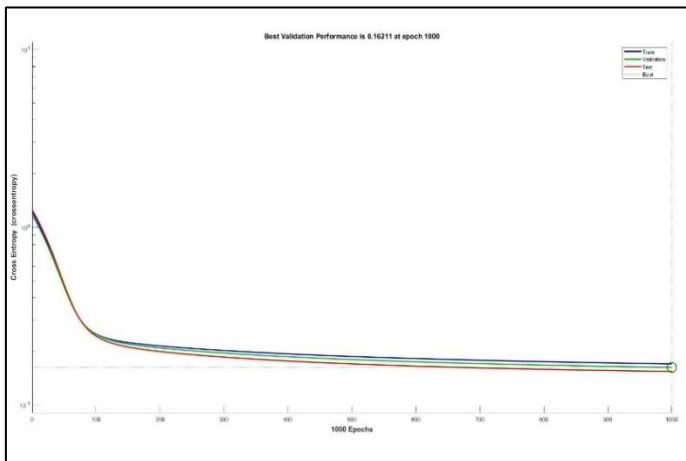


(h2)



(h3)

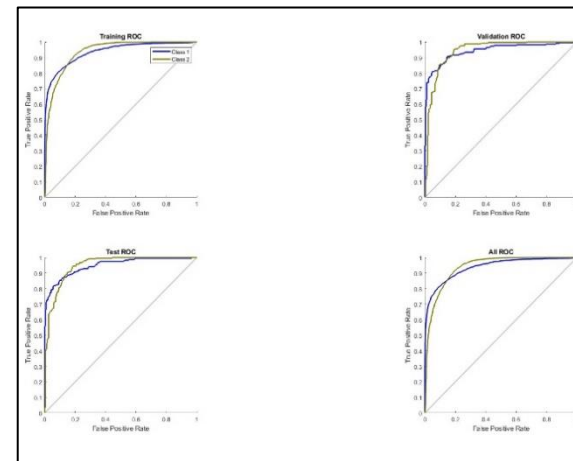
Gambar 4.16 (g1) (h1) Performance Run 7 dan 8 Training GDM HN 15, (g2) (h2) Confusion Matrix Run 7 dan 8 Training GDM HN 15, (g3) (h3) ROC Run 7 dan 8 Training GDM HN 15



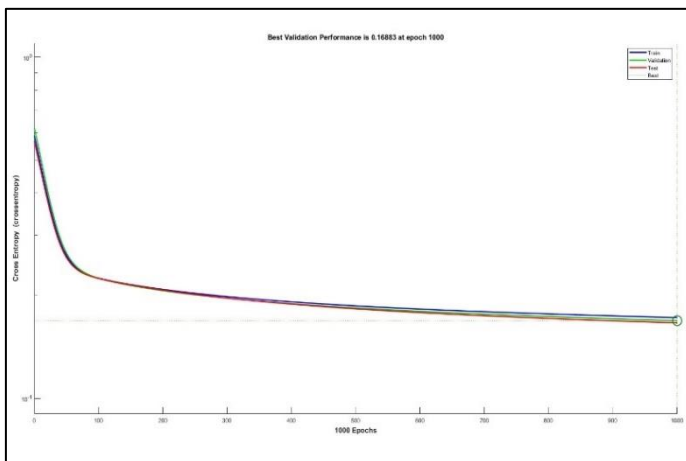
(i1)



(i2)



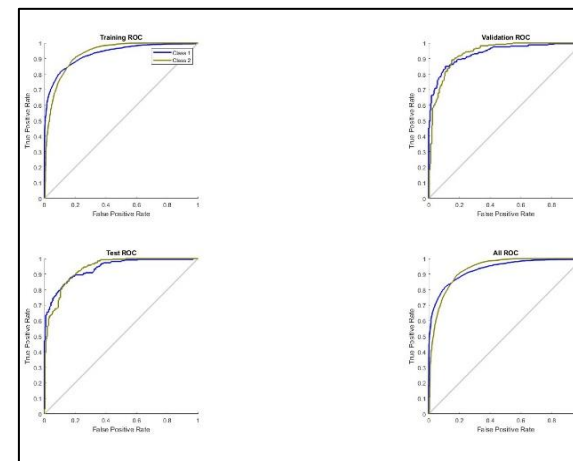
(i3)



(j1)



(j2)



(j3)

Gambar 4.16 (i1) (j1) Performance Run 9 dan 10 Training GDM HN 15, (i2) (j2) Confusion Matrix Run 9 dan 10 Training GDM HN 15, (i3) (j3) ROC Run 9 dan 10 Training GDM HN 15

Berdasarkan visualisasi hasil *training* di atas, dapat diketahui bahwa grafik *performance* terdiri dari 3 grafik, yaitu grafik biru merepresentasikan hasil *training*, grafik hijau merepresentasikan hasil validasi, dan grafik merah merepresentasikan hasil *testing*. Gambar 4.16 menampilkan sepuluh grafik *performance training* GDM *hidden neuron* 15 dimana grafik tersebut rata-rata mengalami penurunan error signifikan pada *range epoch* 0-100, kecuali *run* 3 dan 5. Penurunan error yang landai terjadi pada *range epoch* 110 sampai *epoch* maksimalnya, seperti terlihat pada *run* 1, *run* 2, *run* 4, *run* 6 hingga 10. Pada *run* 3 dan 5, grafik tidak mengalami penurunan signifikan dari awal hingga *epoch* maksimalnya.

Hasil *training* yang menunjukkan akurasi dari sistem ditunjukkan oleh *confusion matrix* dimana warna hijau merepresentasikan citra yang berhasil diklasifikasikan ke dalam kelas secara benar dan warna merah adalah citra yang gagal diklasifikasikan dalam kelasnya. Perbandingan data yang digunakan untuk *10-run cross validation* yaitu 90% dari total data (8.591) digunakan sebagai data *training*, 5% dari total data (477) digunakan sebagai data validasi, dan 5% lainnya (477) digunakan sebagai data *testing*. Dari *confusion matrix* tersebut, diketahui bahwa akurasi *training*, validasi, dan *testing* $\geq 85\%$. Nilai *error training* $\leq 16\%$ (1.374) serta *error validasi* dan *testing* adalah $\leq 18\%$ (86). Hal ini menunjukkan bahwa model memiliki performa yang tidak cukup baik

Visualisasi grafik lainnya adalah grafik *Receiver Operating Characteristic* (ROC) model *Descent with Momentum* (GDM) *hidden neuron* 15. Fungsi dari grafik ROC ini adalah untuk mengetahui hasil evaluasi performa *machine learning*. Pada grafik ROC, terdapat dua warna grafik yaitu biru yang menunjukkan klasifikasi citra kelas 1 dan grafik hijau menunjukkan klasifikasi citra kelas 2. Kurva grafik berada pada angka 0,8 dan menjauhi garis *threshold*. Dengan demikian, dapat disimpulkan bahwa performa *machine learning* (*Multilayer Perceptron* model GDM *hidden neuron* 15) menunjukkan performa tidak perfect.

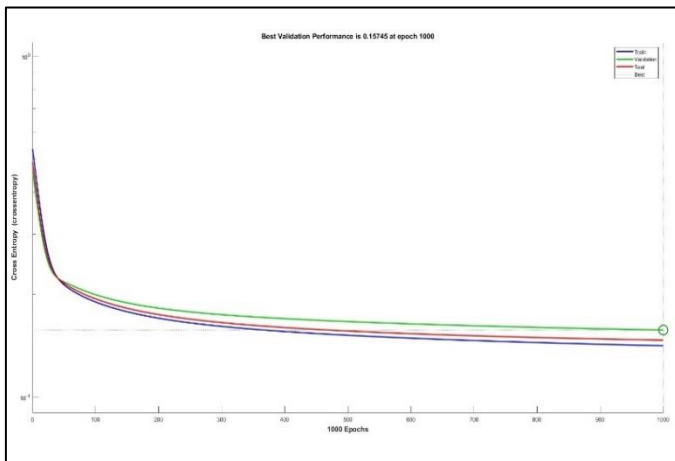
e. *Hidden Neuron 20*

Tabel 4.20 merupakan hasil akurasi *training* citra *body fluids* menggunakan model *Gradient Descent with Momentum* (GDM) *hidden neuron 20*.

Tabel 4. 20 Hasil Akurasi *Training* Menggunakan Model GDM *Hidden Neuron 20*

<i>Run</i>	<i>Epoch</i>		MLP-TrainGDM <i>Hidden Neuron 20</i>			<i>Elapsed Time</i>
	<i>All</i>	<i>Best</i>	<i>Acc Training</i>	<i>Acc Validation</i>	<i>Acc Testing</i>	
1	1000	1000	88,7%	85,7%	87,4%	00:00:10
2	1000	1000	87,1%	85,3%	85,1%	00:00:10
3	1000	1000	87,7%	90,1%	87,4%	00:00:10
4	1000	1000	86,3%	85,5%	86,2%	00:00:10
5	1000	1000	87,5%	86,2%	88,3%	00:00:10
6	1000	1000	86,2%	86,4%	82,0%	00:00:10
7	1000	1000	86,9%	86,4%	86,8%	00:00:10
8	1000	1000	88,1%	88,9%	87,6%	00:00:10
9	1000	1000	89,0%	87,2%	86,4%	00:00:10
10	1000	1000	85,8%	83,4%	83,9%	00:00:10
Average			87,3%	86,5%	86,1%	00:00:10
Standar Deviasi			0,0107191	0,0188588	0,019353	

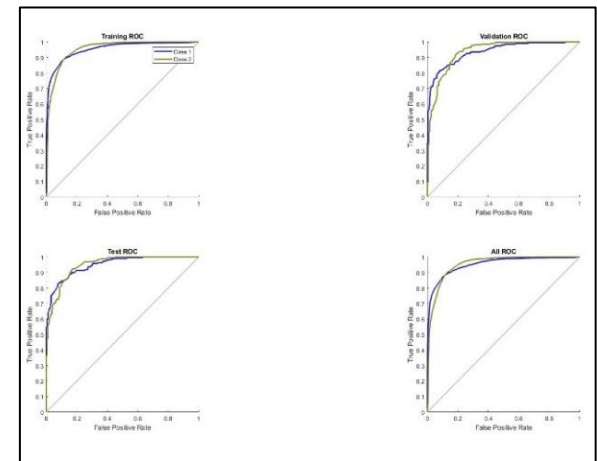
Berdasarkan Tabel 4.20 diketahui bahwa rata-rata akurasi *training* menggunakan model GDM dengan *hidden neuron 20* adalah 85-89%. Hasil akurasi *training* terbaik terdapat pada *run 9* dengan akurasi *training* sebesar 89,0%, akurasi validasi 87,2%, dan akurasi *testing* 86,4%. Akurasi *training* terbaik ini diperoleh pada *epoch* ke 1000 dari total 1000 *epoch* dengan waktu *training* selama 10 detik. Sementara itu, hasil akurasi *training* terendah terdapat pada *run 10* dengan akurasi *training* 85,8%, akurasi validasi 83,4%, dan akurasi *testing* 83,9%. Dari pengujian diatas, diperoleh hasil bahwa rata-rata akurasi *training* adalah 87,3%, akurasi validasi 86,5%, dan akurasi *testing* 86,1%. Rata-rata waktu *training* menggunakan model GDM *hidden neuron 20* adalah 10 detik. Visualisasi hasil *training* dapat dilihat pada Gambar 4.17.



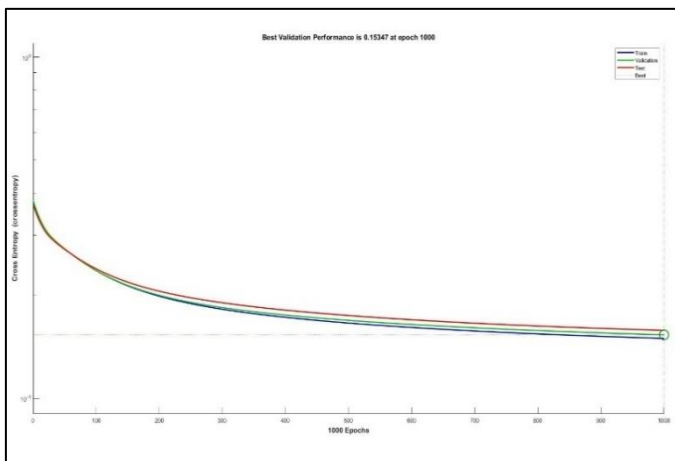
(a1)



(a2)



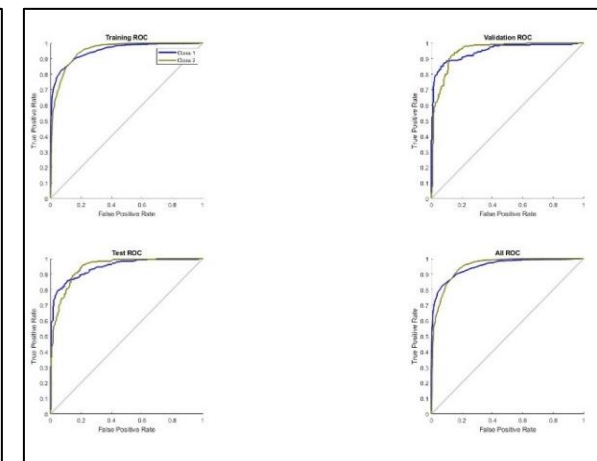
(a3)



(b1)

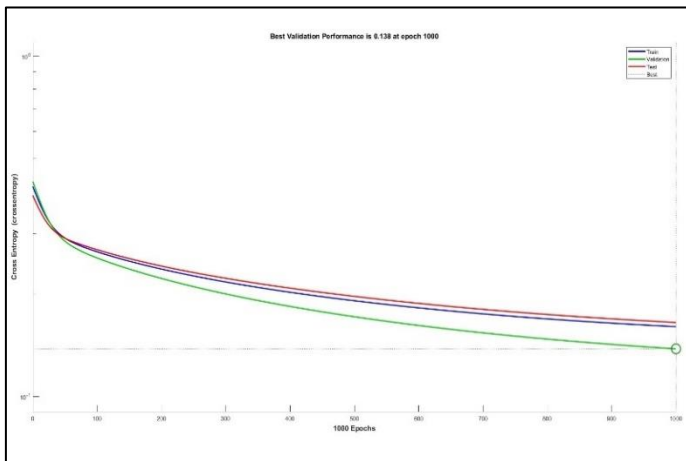


(b2)



(b3)

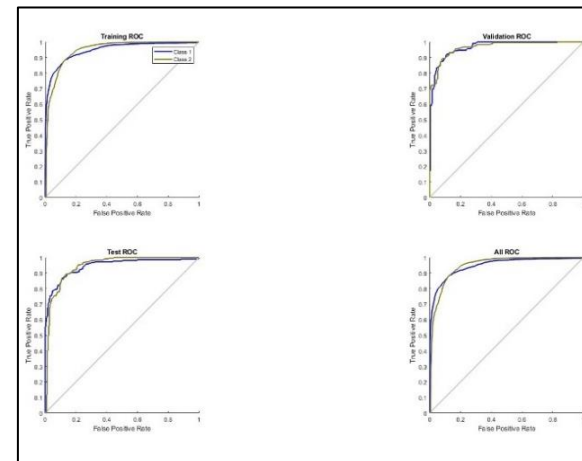
Gambar 4. 17 (a1) (b1) Performance Run 1 dan 2 Training GDM HN 20, (a2) (b2) Confusion Matrix Run 1 dan 2 Training GDM HN 20, (a3) (b3) ROC Run 1 dan 2 Training GDM HN 20



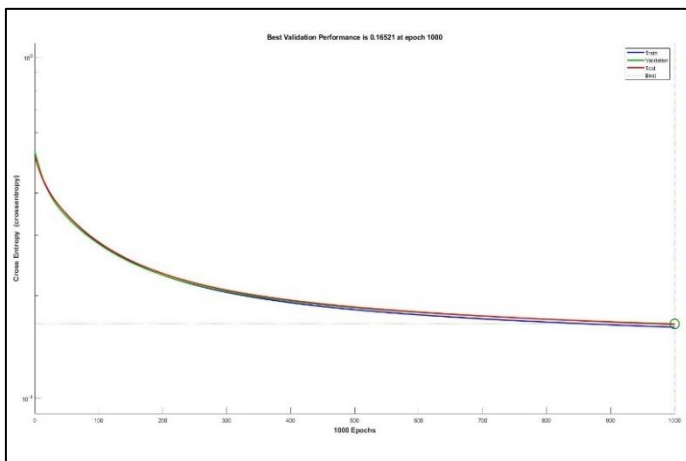
(c1)



(c2)



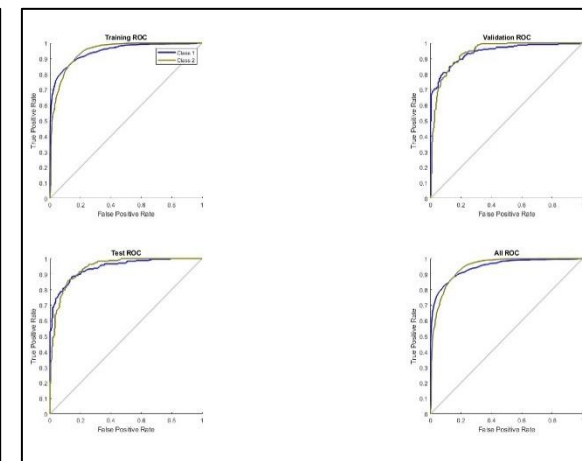
(c3)



(d1)

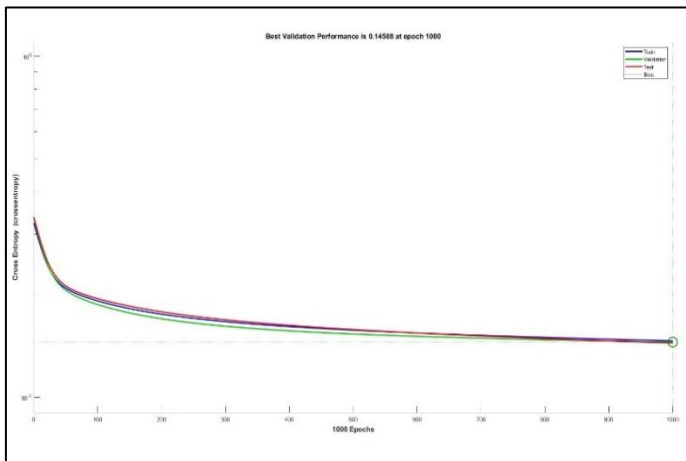


(d2)



(d3)

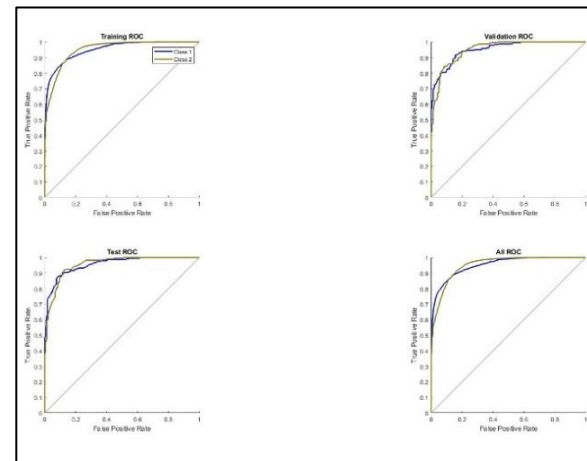
Gambar 4.17 (c1) (d1) Performance Run 3 dan 4 Training GDM HN 20, (c2) (d2) Confusion Matrix Run 3 dan 4 Training GDM HN 20, (c3) (d3) ROC Run 3 dan 4 Training GDM HN 20



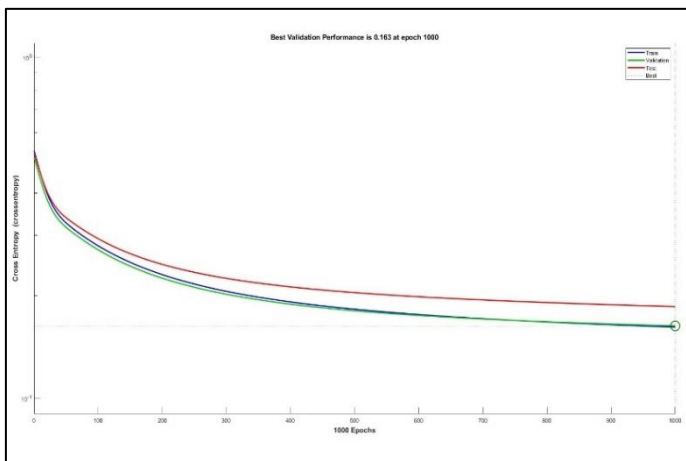
(e1)



(e2)



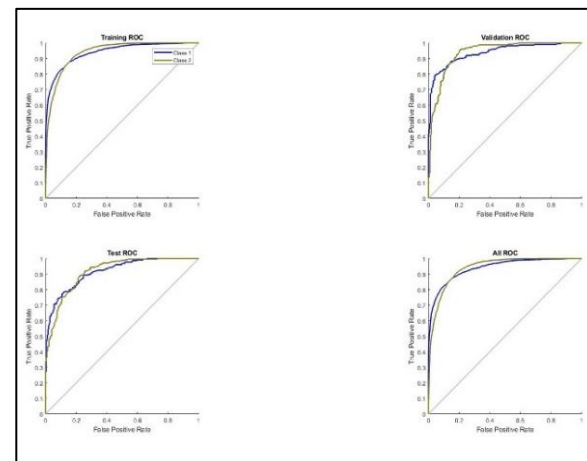
(e3)



(f1)

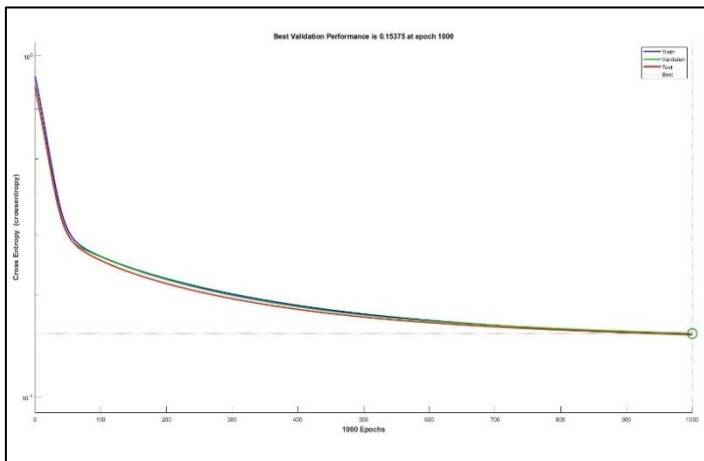


(f2)



(f3)

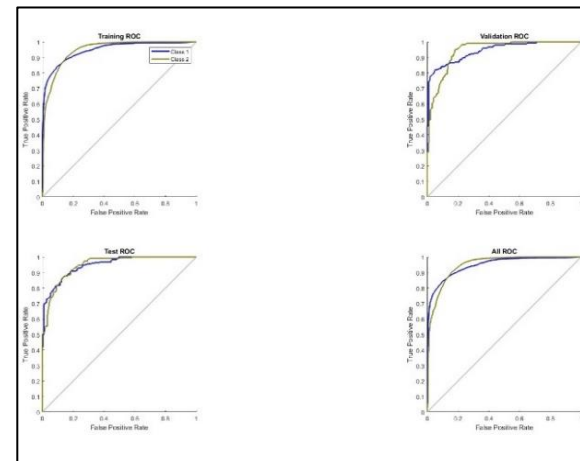
Gambar 4.17 (e1) (f1) Performance Run 5 dan 6 Training GDM HN 20, (e2) (f2) Confusion Matrix Run 5 dan 6 Training GDM HN 20, (e3) (f3) ROC Run 5 dan 6 Training GDM HN 20



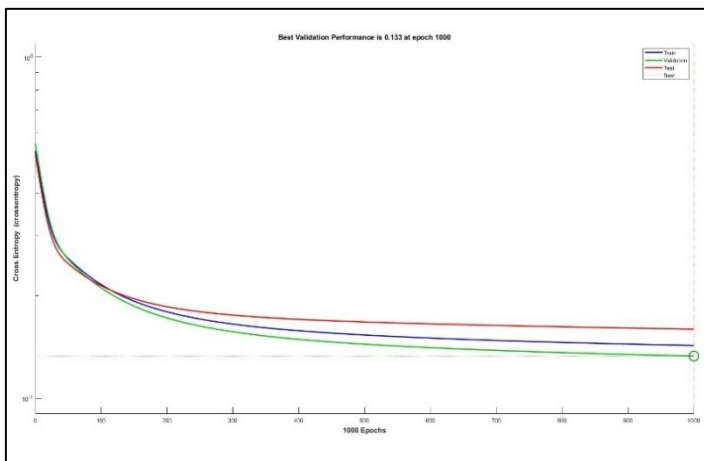
(g1)



(g2)



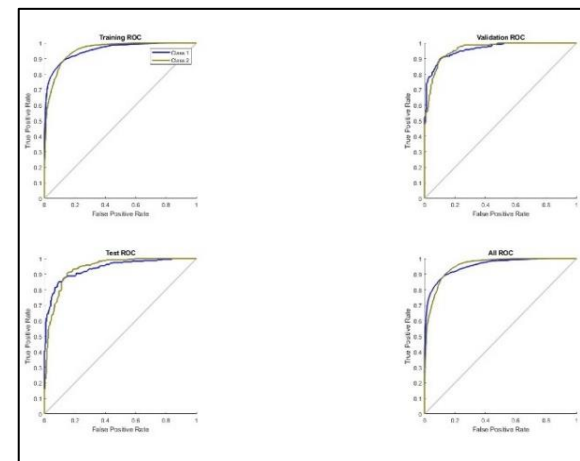
(g3)



(h1)

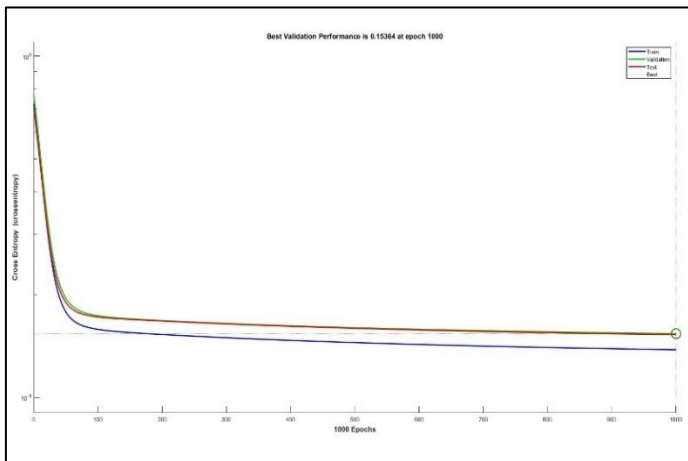


(h2)



(h3)

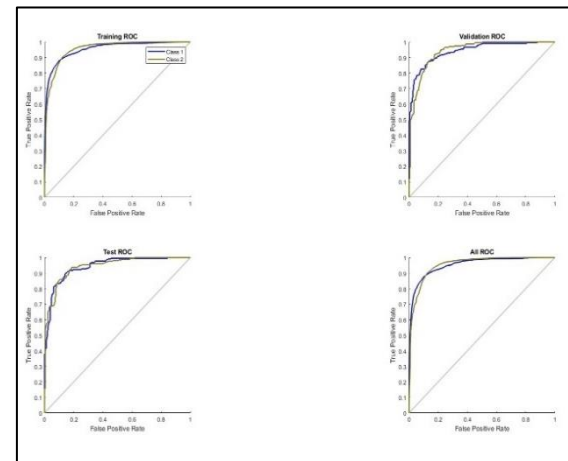
Gambar 4.17 (g1) (h1) Performance Run 7 dan 8 Training GDM HN 20, (g2) (h2) Confusion Matrix Run 7 dan 8 Training GDM HN 20, (g3) (h3) ROC Run 7 dan 8 Training GDM HN 20



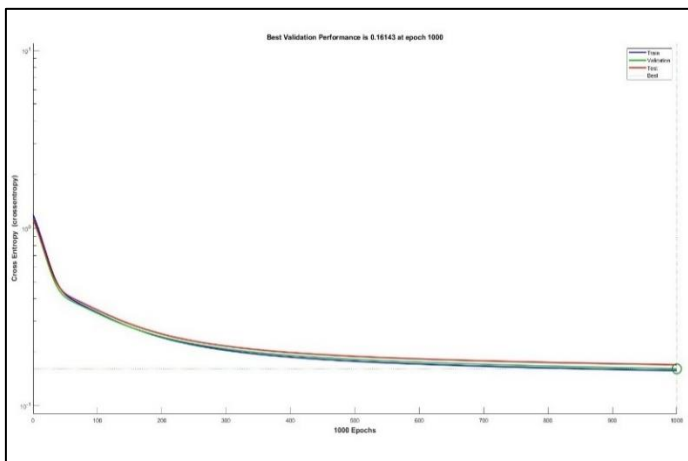
(i1)



(i2)



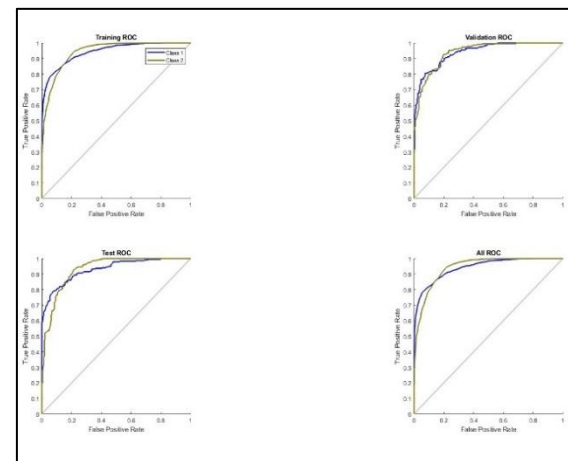
(i3)



(j1)



(j2)



(j3)

Gambar 4.17 (i1) (j1) Performance Run 9 dan 10 Training GDM HN 20, (i2) (j2) Confusion Matrix Run 9 dan 10 Training GDM HN 20, (i3) (j3) ROC Run 9 dan 10 Training GDM HN 20

Berdasarkan visualisasi hasil *training* di atas, dapat diketahui bahwa grafik *performance* terdiri dari 3 grafik, yaitu grafik biru merepresentasikan hasil *training*, grafik hijau merepresentasikan hasil validasi, dan grafik merah merepresentasikan hasil *testing*. Gambar 4.17 menampilkan sepuluh grafik *performance training* GDM *hidden neuron* 20 dimana grafik tersebut menunjukkan penurunan *error* signifikan pada *range epoch* 0-50, kecuali *run* 4. Penurunan *error* yang landai terjadi pada *range epoch* 60 sampai *epoch* maksimalnya, seperti terlihat pada *run* 1 sampai *run* 3, dan *run* 5 sampai *run* 10. Pada *run* 4, grafik tidak mengalami penurunan signifikan dari awal hingga *epoch* maksimalnya.

Hasil *training* yang menunjukkan akurasi dari sistem ditunjukkan oleh *confusion matrix* dimana warna hijau merepresentasikan citra yang berhasil diklasifikasikan ke dalam kelas secara benar dan warna merah adalah citra yang gagal diklasifikasikan dalam kelasnya. Perbandingan data yang digunakan untuk *10-run cross validation* yaitu 90% dari total data (8.591) digunakan sebagai data *training*, 5% dari total data (477) digunakan sebagai data validasi, dan 5% lainnya (477) digunakan sebagai data *testing*. Dari *confusion matrix* tersebut, diketahui bahwa akurasi *training*, validasi, dan *testing* $\geq 85\%$. Nilai *error* dari *training* $\leq 14\%$ (1.202) serta *error testing* $\leq 16\%$ dari data *testing* (76). Hal ini menunjukkan bahwa model memiliki performa yang tidak cukup baik.

Visualisasi grafik lainnya adalah grafik *Receiver Operating Characteristic* (ROC) model *Descent with Momentum* (GDM) *hidden neuron* 15. Fungsi dari grafik ROC ini adalah untuk mengetahui hasil evaluasi performa *machine learning*. Pada grafik ROC, terdapat dua warna grafik yaitu biru yang menunjukkan klasifikasi citra kelas 1 dan grafik hijau menunjukkan klasifikasi citra kelas 2. Kurva grafik berada pada angka 0,8 dan menjauhi garis *threshold*. Dengan demikian, dapat disimpulkan bahwa performa *machine learning* (*Multilayer Perceptron* model GDM *hidden neuron* 20) menunjukkan performa tidak *perfect*.

4.3.2 Klasifikasi dari Metode *Haar Wavelet*

Sebanyak 9.545 citra yang telah diekstraksi menggunakan metode *Haar Wavelet* dilatih dan diuji menggunakan tiga model *Multilayer Perceptron* (MLP), yaitu *Scaled Conjugate Gradient Backpropagation* (SCG), *One Step Secant Backpropagation* (OSS), dan *Gradient Descent with Momentum* (GDM) dengan *hidden neuron* 1, 5, 10, 15, dan 20. Setiap *training* akan menghasilkan *Pattern Recognition Neural Network* yang menggambarkan proses input hingga output. *Pattern Recognition Neural Network* masing-masing *hidden neuron* dapat dilihat pada Gambar 4.18.



Gambar 4. 18 (a) Pattern Recognition Neural Network Hidden Neuron 1, (b) Pattern Recognition Neural Network Hidden Neuron 5, (c) Pattern Recognition Neural Network Hidden Neuron 10, (d) Pattern Recognition Neural Network Hidden Neuron 15, (e) Pattern Recognition Neural Network Hidden Neuron 20

4.3.2.1 Scaled Conjugate Gradient Backpropagation (SCG)

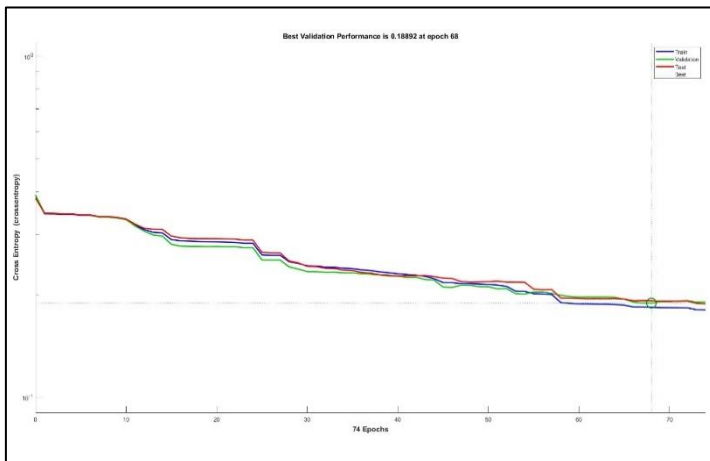
a. Hidden Neuron 1

Hasil akurasi *training* citra *body fluids* menggunakan model *Scaled Conjugate Gradient Backpropagation* (SCG) *hidden neuron* 1 dapat dilihat pada Tabel 4.21.

Tabel 4. 21 Hasil Akurasi *Training* Menggunakan Model SCG *Hidden Neuron* 1

<i>Run</i>	<i>Epoch</i>		MLP-TrainSCG <i>Hidden Neuron</i> 1			<i>Elapsed Time</i>
	<i>All</i>	<i>Best</i>	<i>Acc Training</i>	<i>Acc Validation</i>	<i>Acc Testing</i>	
1	74	68	85,3%	86,2%	83,9%	00:00:04
2	73	67	90,2%	85,3%	85,5%	00:00:04
3	26	20	77,8%	80,5%	75,1%	00:00:03
4	31	25	75,5%	75,3%	71,9%	00:00:03
5	16	10	69,0%	70,0%	66,9%	00:00:02
6	55	49	82,0%	78,8%	74,4%	00:00:04
7	71	65	82,4%	78,6%	76,7%	00:00:04
8	47	41	84,3%	82,8%	79,5%	00:00:03
9	39	33	81,8%	79,9%	78,4%	00:00:03
10	26	20	72,4%	67,9%	74,4%	00:00:03
Average			80,1%	78,5%	76,7%	00:00:03
Standar Deviasi			0,063894	0,0600852	0,054997	

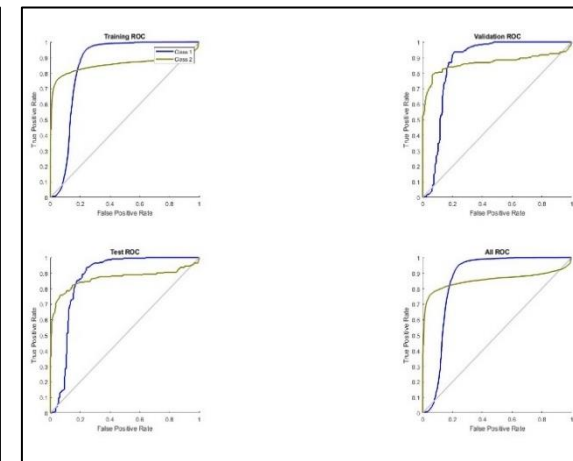
Berdasarkan Tabel 4.21 diketahui bahwa rata-rata akurasi *training* menggunakan model SCG dengan *hidden neuron* 1 adalah 69-85%. Hasil akurasi *training* terbaik terdapat pada *run* 1 dengan akurasi *training* sebesar 85,3%, akurasi validasi 86,2%, dan akurasi *testing* 83,9%. Akurasi *training* terbaik ini diperoleh pada *epoch* ke 68 dari total 74 *epoch* dengan waktu *training* selama 4 detik. Sementara itu, hasil akurasi *training* terendah terdapat pada *run* 5 dengan akurasi *training* 69,0%, akurasi validasi 70,0%, dan akurasi *testing* 66,9%. Dari pengujian diatas, diperoleh hasil bahwa rata-rata akurasi *training* adalah 80,1%, akurasi validasi 78,5%, dan akurasi *testing* 76,7%. Rata-rata waktu *training* menggunakan model SCG *hidden neuron* 1 adalah 3 detik. Visualisasi hasil *training* dapat dilihat pada Gambar 4.19.



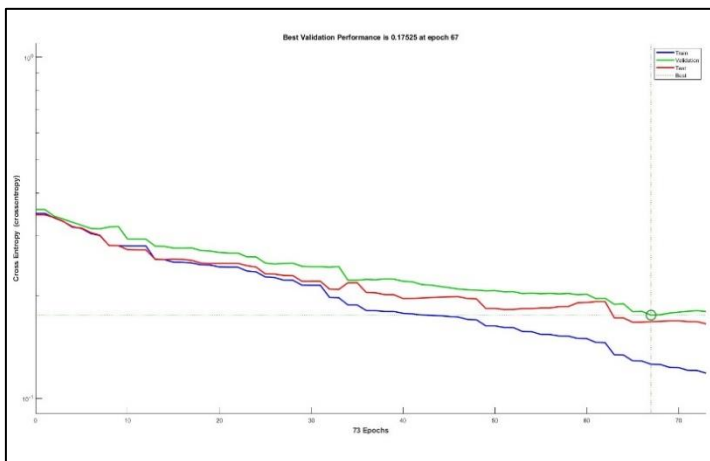
(a1)



(a2)



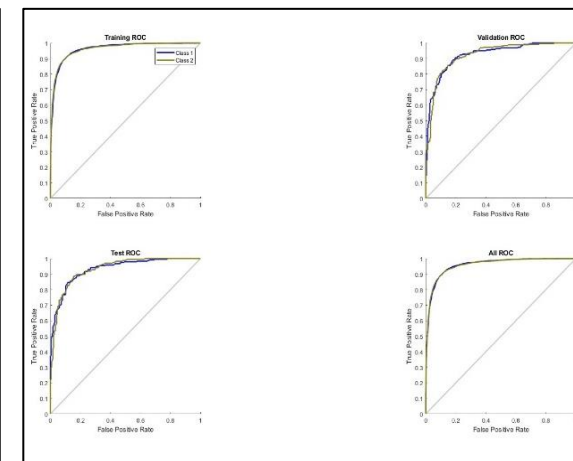
(a3)



(b1)

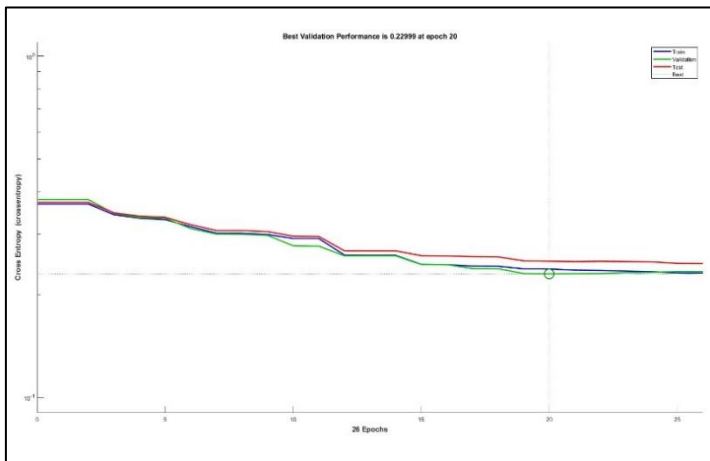


(b2)



(b3)

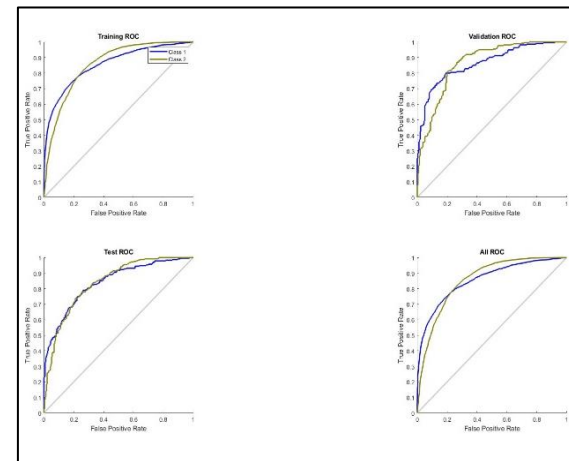
Gambar 4. 19 (a1) (b1) Performance Run 1 dan 2 Training SCG HN 1, (a2) (b2) Confusion Matrix Run 1 dan 2 Training SCG HN 1, (a3) (b3) ROC Run 1 dan 2 Training SCG HN 1



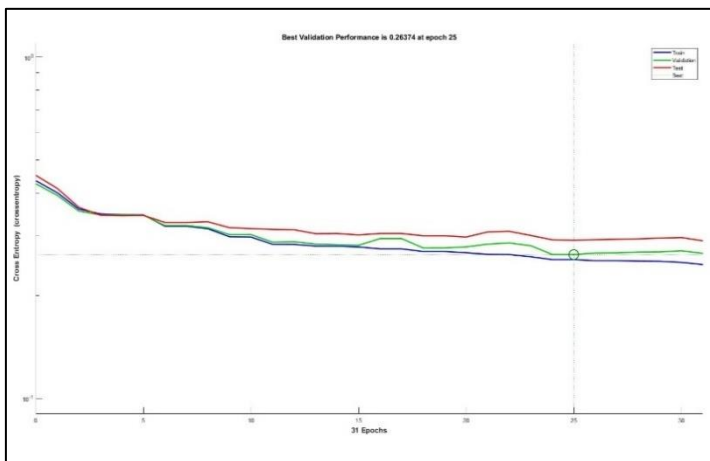
(c1)



(c2)



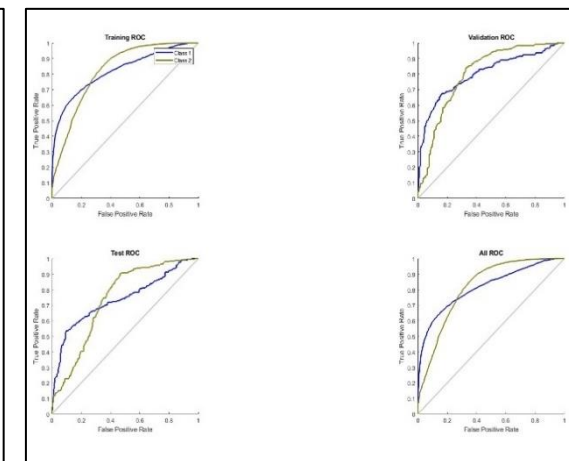
(c3)



(d1)

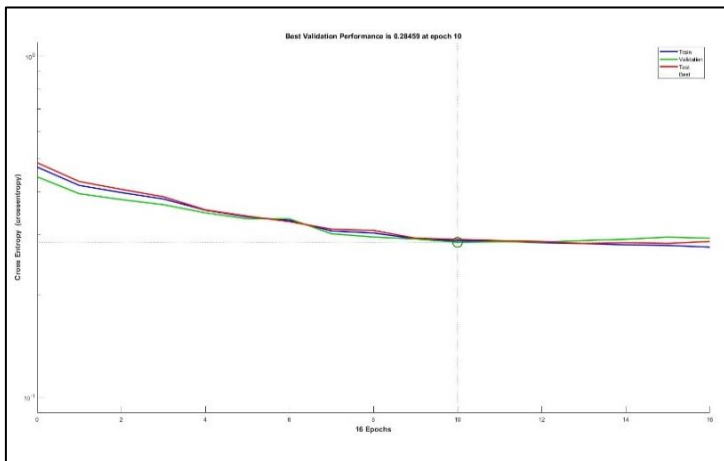


(d2)



(d3)

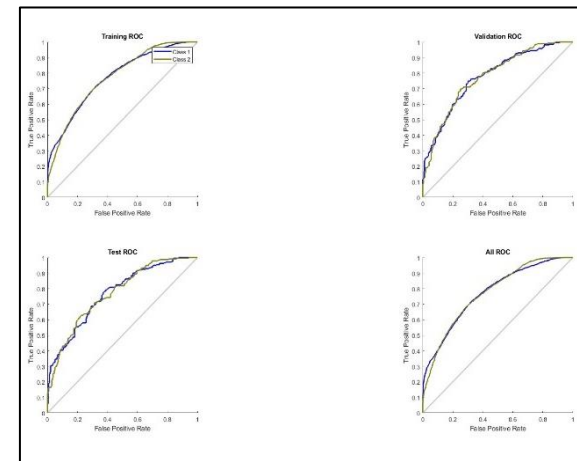
Gambar 4.19 (c1) (d1) Performance Run 3 dan 4 Training SCG HN 1, (c2) (d2) Confusion Matrix Run 3 dan 4 Training SCG HN 1, (c3) (d3) ROC Run 3 dan 4 Training SCG HN 1



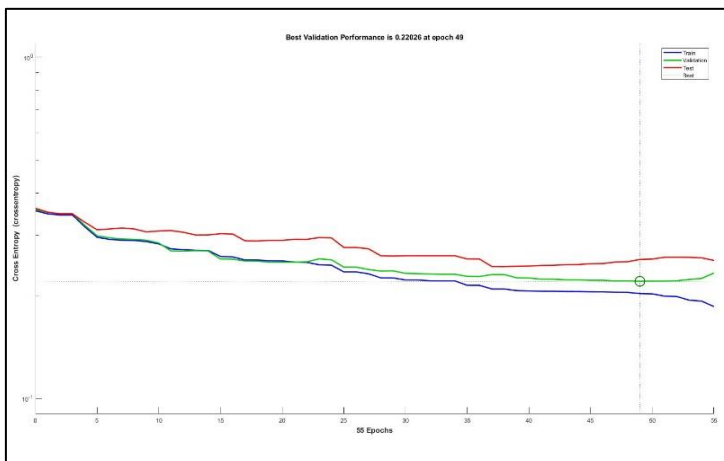
(e1)



(e2)



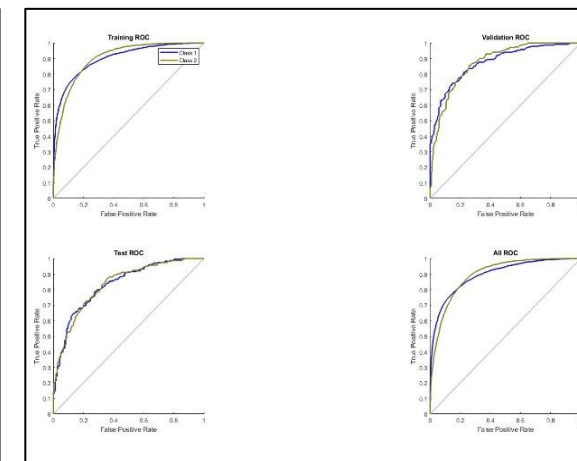
(e3)



(f1)

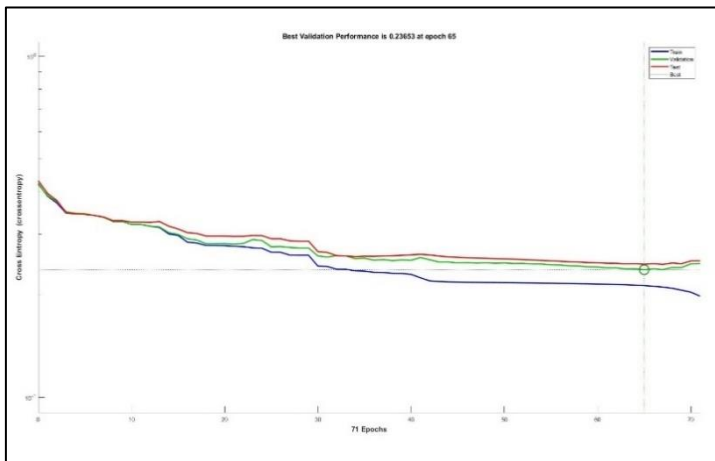


(f2)



(f3)

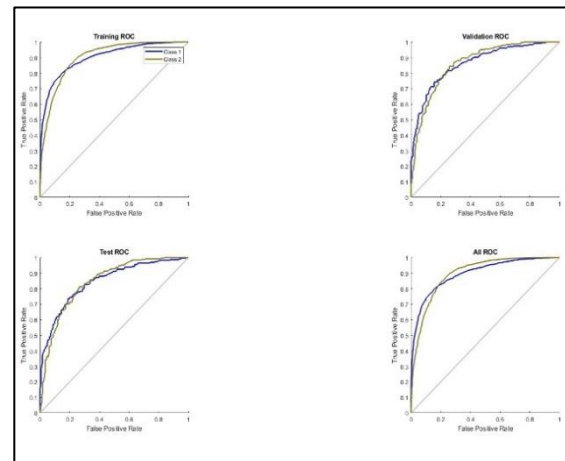
Gambar 4.19 (e1) (f1) Performance Run 5 dan 6 Training SCG HN 1, (e2) (f2) Confusion Matrix Run 5 dan 6 Training SCG HN 1, (e3) (f3) ROC Run 5 dan 6 Training SCG HN 1



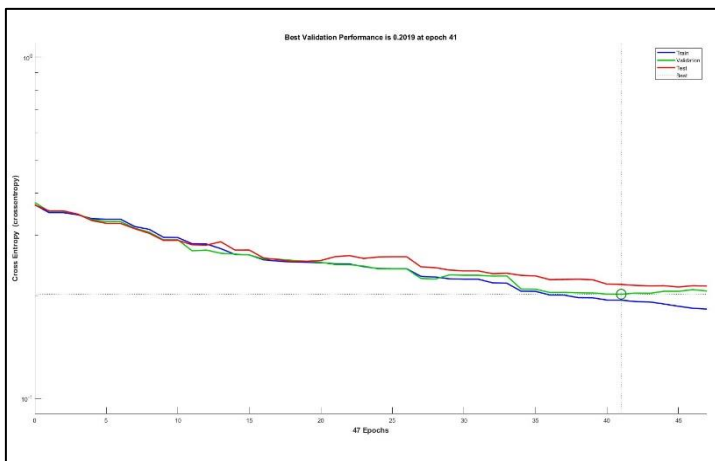
(g1)



(g2)



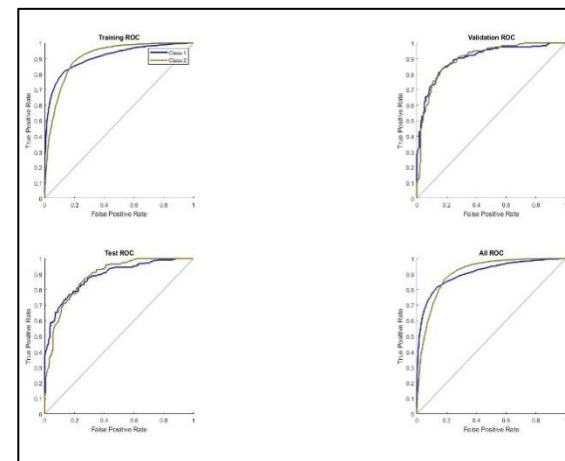
(g3)



(h1)

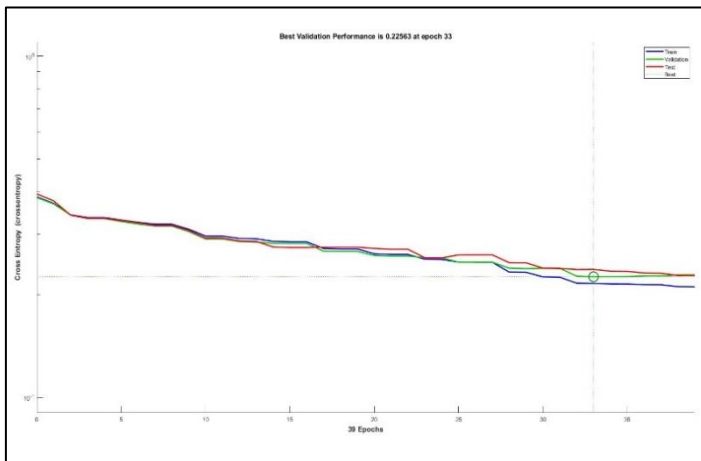


(h2)



(h3)

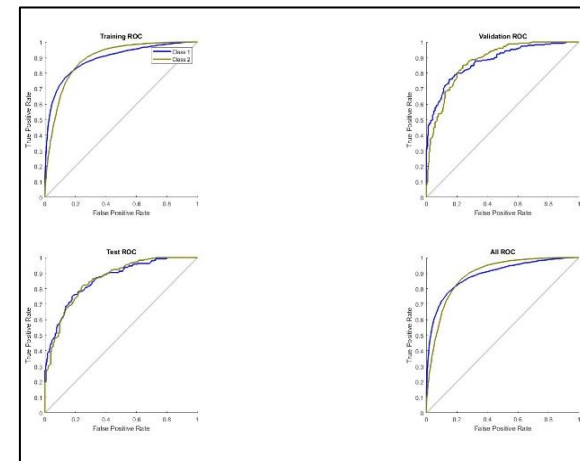
Gambar 4.19 (g1) (h1) Performance Run 7 dan 8 Training SCG HN 1, (g2) (h2) Confusion Matrix Run 7 dan 8 Training SCG HN 1, (g3) (h3) ROC Run 7 dan 8 Training SCG HN 1



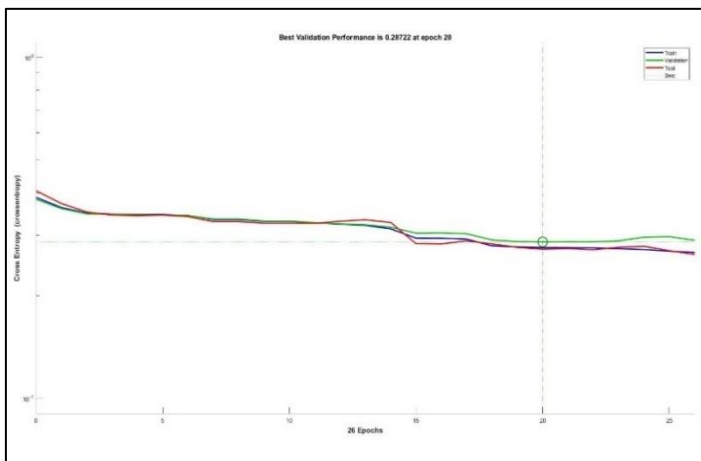
(i1)



(i2)



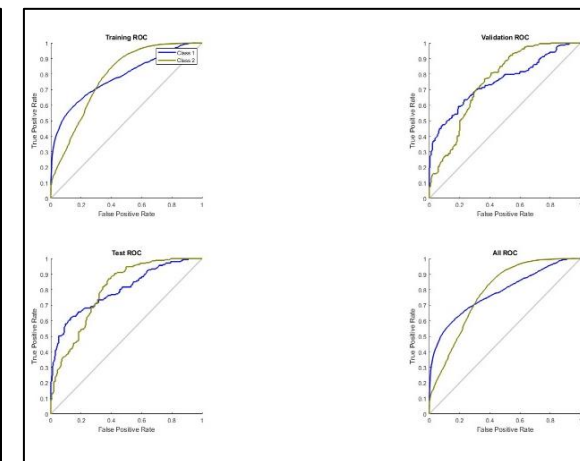
(i3)



(j1)



(j2)



(j3)

Gambar 4.19 (i1) (j1) Performance Run 9 dan 10 Training SCG HN 1, (i2) (j2) Confusion Matrix Run 9 dan 10 Training SCG HN 1, (i3) (j3) ROC Run 9 dan 10 Training SCG HN 1

Berdasarkan visualisasi hasil *training* di atas, dapat diketahui bahwa grafik *performance* terdiri dari 3 grafik, yaitu grafik biru merepresentasikan hasil *training*, grafik hijau merepresentasikan hasil validasi, dan grafik merah merepresentasikan hasil *testing*. Gambar 4.19 menampilkan sepuluh grafik *performane training* SCG hidden neuron 1 dimana rata-rata grafik mengalami penurunan *error* landai dari *epoch* 5 hingga *epoch* maksimal. Pada *run* 2, 6, dan 7, penurunan grafik melandai antara *training*, validasi, dan *testing* tidak sama sedangkan *run* 10 tidak mengalami penurunan signifikan dari awal *epoch* hingga *epoch* maksimalnya. Semakin rendah grafik menunjukkan bahwa nilai *error* semakin kecil.

Hasil *training* yang menunjukkan akurasi dari sistem ditunjukkan oleh *confusion matrix* dimana warna hijau merepresentasikan citra yang berhasil diklasifikasikan ke dalam kelas secara benar dan warna merah adalah citra yang gagal diklasifikasikan dalam kelasnya. Perbandingan data yang digunakan untuk *10-run cross validation* yaitu 90% dari total data (8.591) digunakan sebagai data *training*, 5% dari total data (477) digunakan sebagai data validasi, dan 5% lainnya (477) digunakan sebagai data *testing*. Dari *confusion matrix* tersebut, diketahui bahwa akurasi *training*, validasi, dan *testing* kurang dari 90%. Nilai rata-rata *error training* adalah 15% (1.288) dan rata-rata *error testing* 20% (95). Hal ini menunjukkan bahwa model memiliki performa yang tidak cukup baik.

Visualisasi grafik lainnya adalah grafik *Receiver Operating Characteristic* (ROC). Fungsi dari grafik ROC ini adalah untuk mengetahui hasil evaluasi performa *machine learning*. Pada grafik ROC, terdapat dua warna grafik yaitu biru yang menunjukkan klasifikasi citra kelas 1 dan grafik hijau menunjukkan klasifikasi citra kelas 2. Rata-rata kurva validasi dan *testing* mendekati garis *threshold* kecuali pada *run* 2. Dengan demikian, dapat disimpulkan bahwa performa *machine learning* (*Multilayer Perceptron* model SCG hidden neuron 1) menunjukkan performa tidak *perfect*.

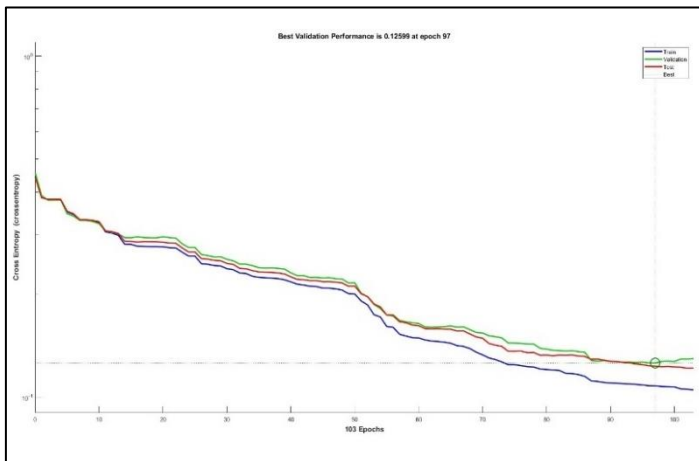
b. *Hidden Neuron 5*

Hasil akurasi *training* citra *body fluids* menggunakan model *Scaled Conjugate Gradient Backpropagation* (SCG) *hidden neuron 5* dapat dilihat pada Tabel 4.22.

Tabel 4. 22 Hasil Akurasi *Training* Menggunakan Model SCG *Hidden Neuron 5*

<i>Run</i>	<i>Epoch</i>		MLP-TrainSCG <i>Hidden Neuron 5</i>			<i>Elapsed Time</i>
	<i>All</i>	<i>Best</i>	<i>Acc Training</i>	<i>Acc Validation</i>	<i>Acc Testing</i>	
1	103	97	92,8%	89,1%	91,4%	00:00:08
2	71	65	88,2%	87,0%	85,5%	00:00:06
3	118	112	89,7%	84,9%	86,4%	00:00:09
4	86	80	87,4%	82,0%	82,6%	00:00:07
5	18	12	73,7%	70,0%	75,3%	00:00:03
6	93	87	92,2%	87,8%	90,6%	00:00:07
7	120	114	87,8%	86,2%	83,9%	00:00:09
8	117	111	93,2%	90,1%	87,2%	00:00:09
9	46	40	83,2%	82,4%	77,6%	00:00:05
10	87	81	91,9%	88,7%	89,5%	00:00:07
Average			88,0%	84,8%	85,0%	00:00:07
Standar Deviasi			0,059047	0,0587117	0,053275	

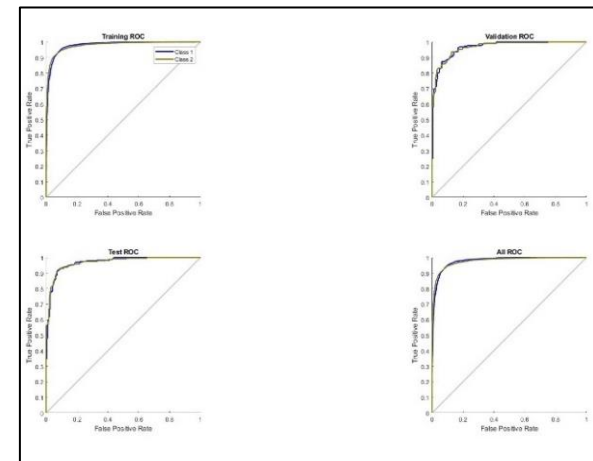
Berdasarkan Tabel 4.22 diketahui bahwa rata-rata akurasi *training* menggunakan model SCG dengan *hidden neuron 5* adalah 73-93%. Hasil akurasi *training* terbaik terdapat pada *run 8* dengan akurasi *training* sebesar 93,2%, akurasi validasi 90,1%, dan akurasi *testing* 87,2%. Akurasi *training* terbaik ini diperoleh pada *epoch* ke 111 dari total 117 *epoch* dengan waktu *training* selama 9 detik. Sementara itu, hasil akurasi *training* terendah terdapat pada *run 5* dengan akurasi *training* 73,7%, akurasi validasi 70,0%, dan akurasi *testing* 75,3%. Dari pengujian diatas, diperoleh hasil bahwa rata-rata akurasi *training* adalah 88,0%, akurasi validasi 84,8%, dan akurasi *testing* 85,0%. Rata-rata waktu *training* menggunakan model SCG *hidden neuron 5* adalah 9 detik. Visualisasi hasil *training* dapat dilihat pada Gambar 4.20.



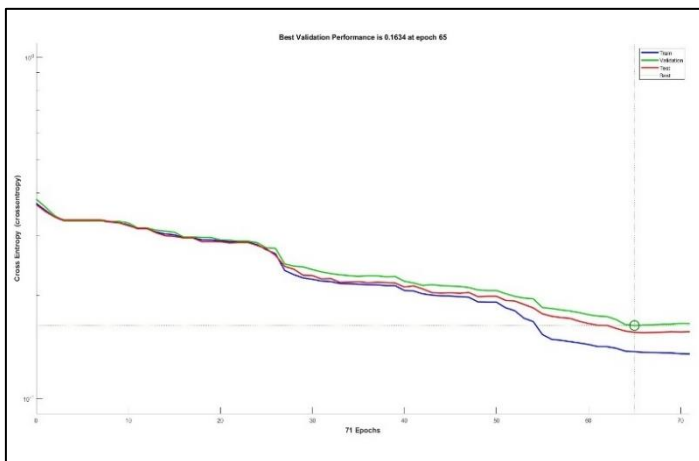
(a1)



(a2)



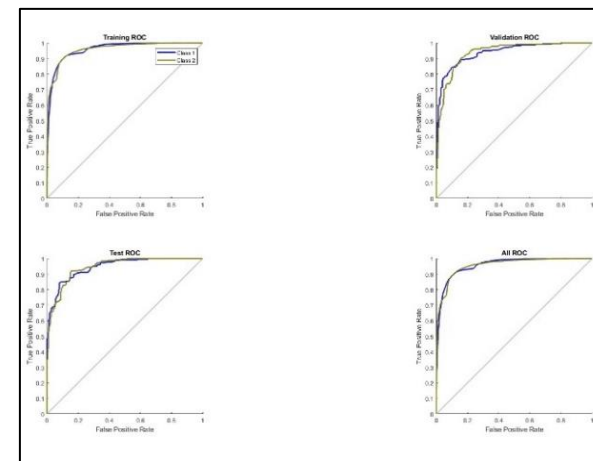
(a3)



(b1)

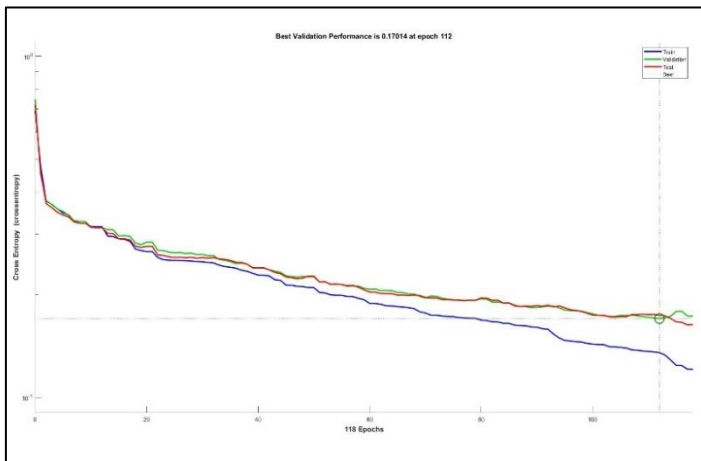


(b2)



(b3)

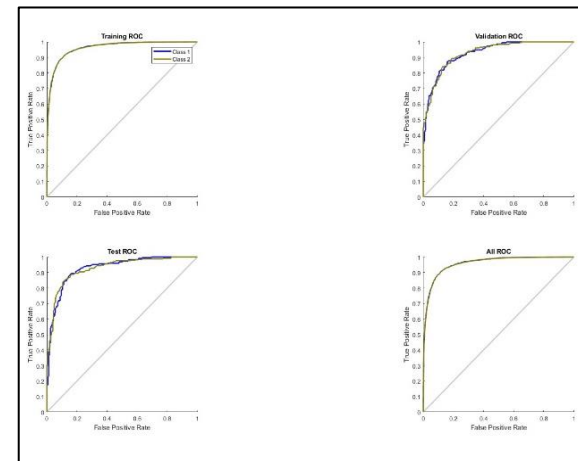
Gambar 4. 20 (a1) (b1) Performance Run 1 dan 2 Training SCG HN 5, (a2) (b2) Confusion Matrix Run 1 dan 2 Training SCG HN 5, (a3) (b3) ROC Run 1 dan 2 Training SCG HN 5



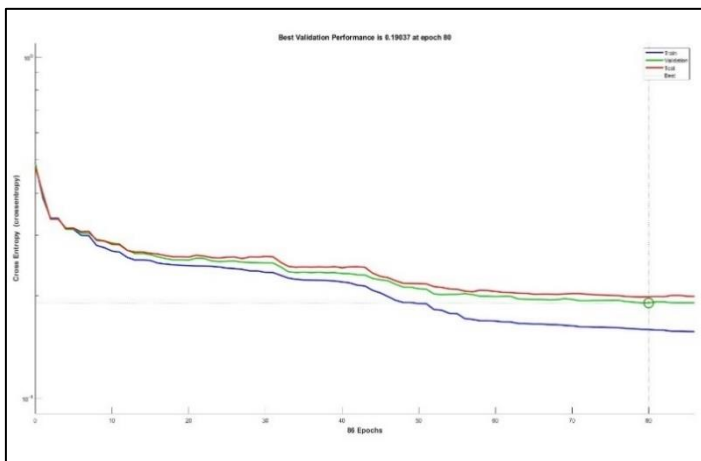
(c1)



(c2)



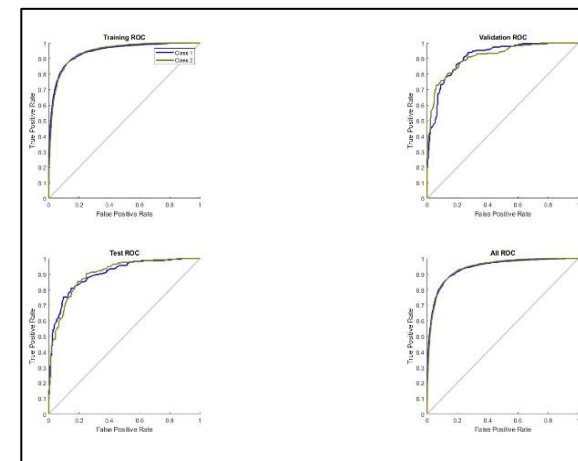
(c3)



(d1)

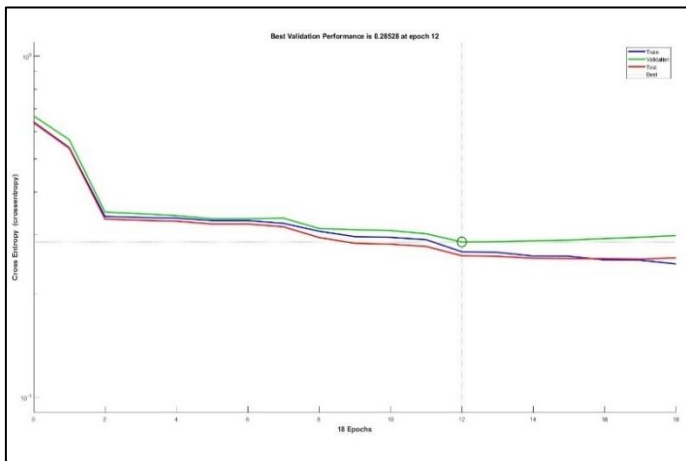


(d2)



(d3)

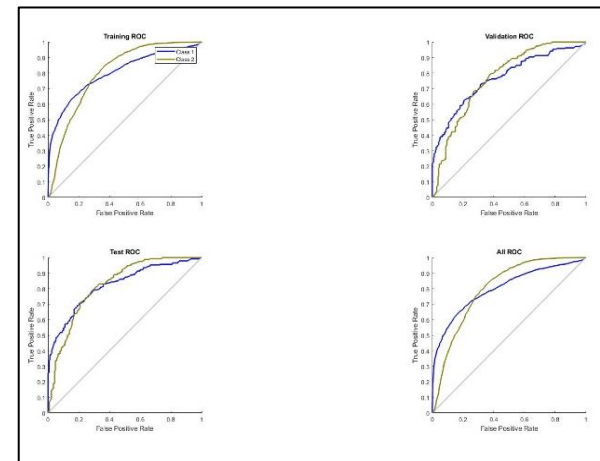
Gambar 4.20 (c1) (d1) Performance Run 3 dan 4 Training SCG HN 5, (c2) (d2) Confusion Matrix Run 3 dan 4 Training SCG HN 5, (c3) (d3) ROC Run 3 dan 4 Training SCG HN 5



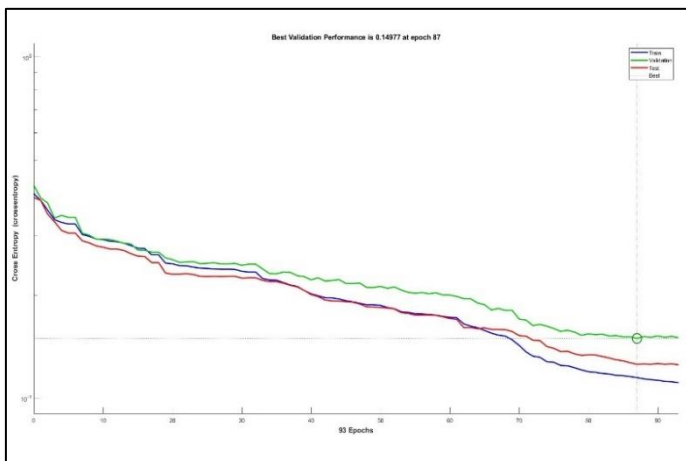
(e1)



(e2)



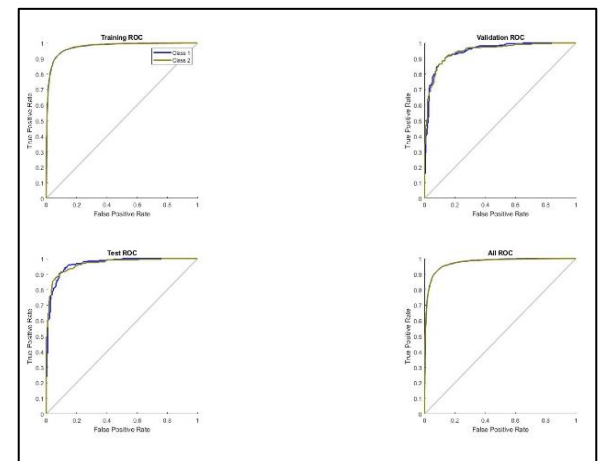
(e3)



(f1)

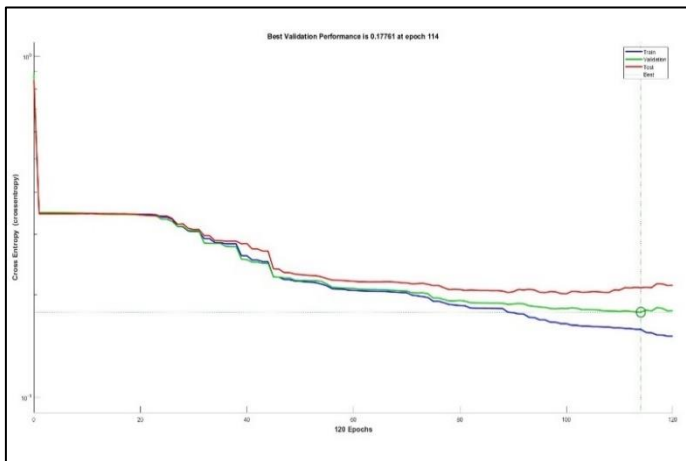


(f2)



(f3)

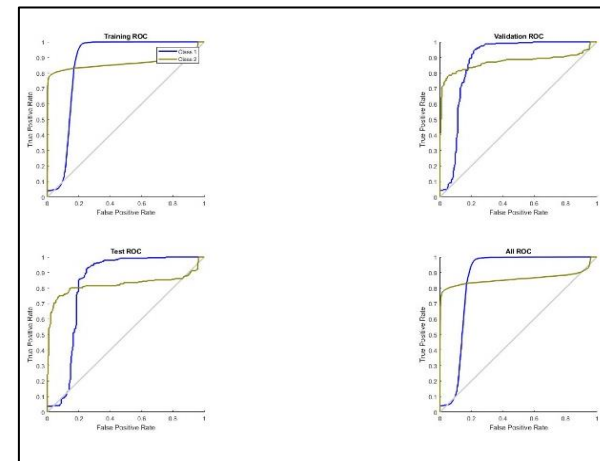
Gambar 4.20 (e1) (f1) Performance Run 5 dan 6 Training SCG HN 5, (e2) (f2) Confusion Matrix Run 5 dan 6 Training SCG HN 5, (e3) (f3) ROC Run 5 dan 6 Training SCG HN 5



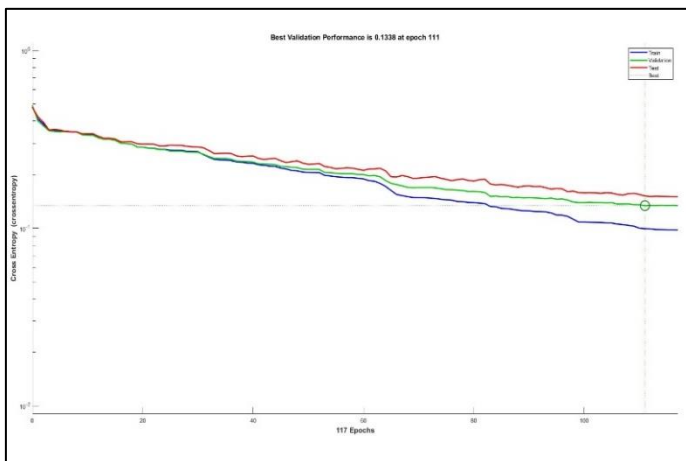
(g1)



(g2)



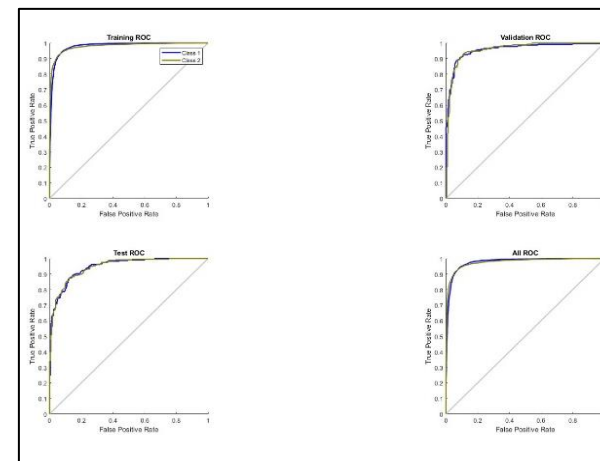
(g3)



(h1)

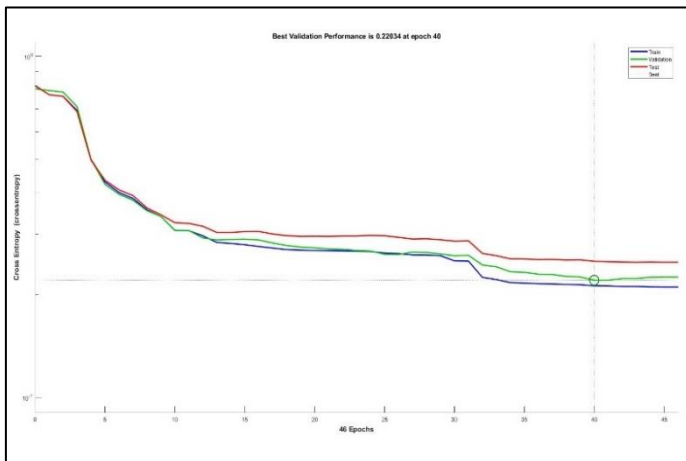


(h2)



(h3)

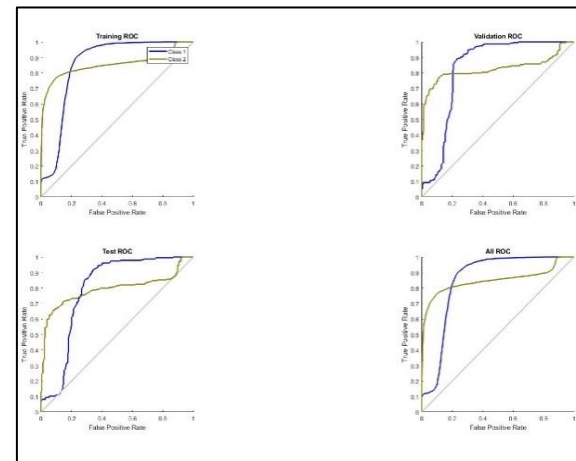
Gambar 4.20 (g1) (h1) Performance Run 7 dan 8 Training SCG HN 5, (g2) (h2) Confusion Matrix Run 7 dan 8 Training SCG HN 5, (g3) (h3) ROC Run 7 dan 8 Training SCG HN 5



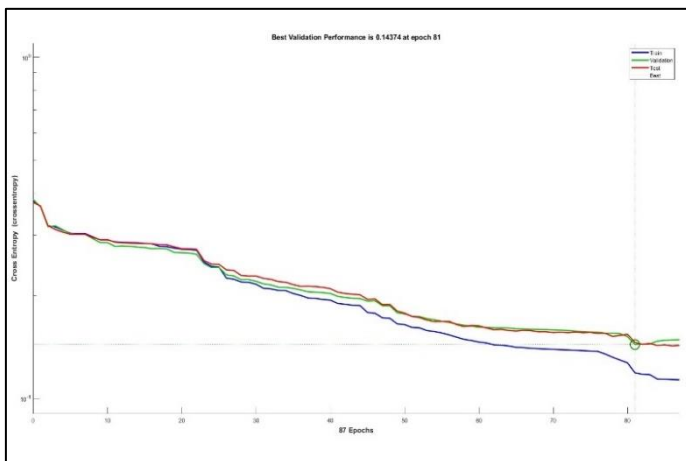
(i1)



(i2)



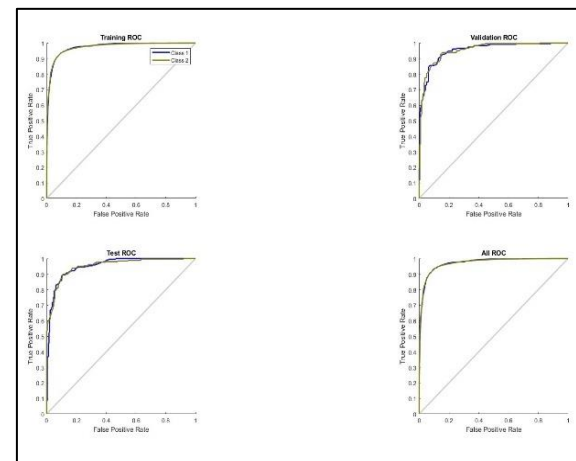
(i3)



(j1)



(j2)



(j3)

Gambar 4.20 (i1) (j1) Performance Run 9 dan 10 Training SCG HN 5, (i2) (j2) Confusion Matrix Run 9 dan 10 Training SCG HN 5, (i3) (j3) ROC Run 9 dan 10 Training SCG HN 5

Berdasarkan visualisasi hasil training di atas, dapat diketahui bahwa grafik *performance* terdiri dari 3 grafik, yaitu grafik biru merepresentasikan hasil *training*, grafik hijau merepresentasikan hasil validasi, dan grafik merah merepresentasikan hasil *testing*. Gambar 4.20 menampilkan sepuluh grafik *performane training SCG hidden neuron 5* dimana grafik mengalami penurunan signifikan pada *range epoch* 0-10 pada run 1 dan 3. Run 5 penurunan *error* signifikan terjadi pada *range epoch* 0-2 sedangkan run 8 tidak mengalami penurunan signifikan dari awal hingga *epoch* maksimal. Penurunan *error* melandai dari *epoch* 10 hingga maksimal terjadi pada run 2, 4, 6, 7, 9, dan 10.

Hasil *training* yang menunjukkan akurasi dari sistem ditunjukkan oleh *confusion matrix* dimana warna hijau merepresentasikan citra yang berhasil diklasifikasikan ke dalam kelas secara benar dan warna merah adalah citra yang gagal diklasifikasikan dalam kelasnya. Perbandingan data yang digunakan untuk *10-run cross validation* yaitu 90% dari total data (8.591) digunakan sebagai data *training*, 5% dari total data (477) digunakan sebagai data validasi, dan 5% lainnya (477) digunakan sebagai data *testing*. Dari *confusion matrix* tersebut, diketahui bahwa akurasi *training*, validasi, dan *testing* 80% dengan *error* lebih dari 10% dari total data (47). Hal tersebut menandakan bahwa performa model tidak cukup baik.

Visualisasi grafik lainnya adalah grafik *Receiver Operating Characteristic* (ROC). Fungsi dari grafik ROC ini adalah untuk mengetahui hasil evaluasi performa *machine learning*. Pada grafik ROC, terdapat dua warna grafik yaitu biru yang menunjukkan klasifikasi citra kelas 1 dan grafik hijau menunjukkan klasifikasi citra kelas 2. Kurva validasi dan *testing* berada pada angka 0,8 dan menjauhi garis *threshold* kecuali pada run 5, 7, dan 9. Dengan demikian, dapat disimpulkan bahwa performa *machine learning* (*Multilayer Perceptron* model *SCG hidden neuron 5*) menunjukkan performa tidak pefect.

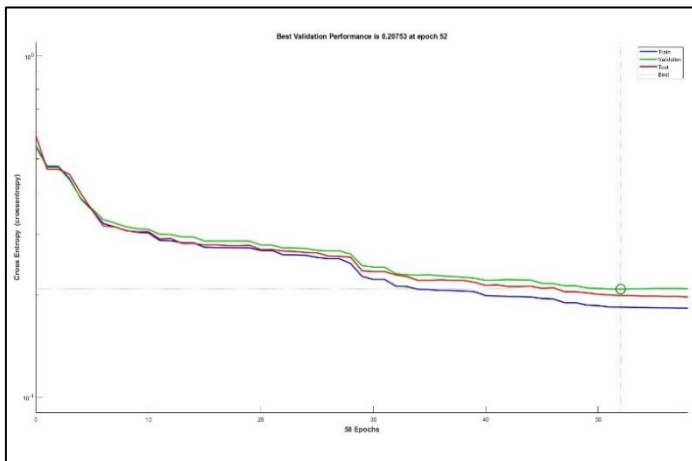
c. *Hidden Neuron 10*

Hasil akurasi *training* citra *body fluids* menggunakan model *Scaled Conjugate Gradient Backpropagation* (SCG) *hidden neuron 10* dapat dilihat pada Tabel 4.23.

Tabel 4. 23 Hasil Akurasi *Training* Menggunakan Model SCG *Hidden Neuron 10*

<i>Run</i>	<i>Epoch</i>		MLP-TrainSCG Hidden Neuron 10			<i>Elapsed Time</i>
	<i>All</i>	<i>Best</i>	<i>Acc Training</i>	<i>Acc Validation</i>	<i>Acc Testing</i>	
1	58	52	85,4%	80,7%	83,6%	00:00:06
2	45	39	83,2%	83,6%	83,9%	00:00:06
3	107	101	91,8%	88,1%	90,4%	00:00:09
4	161	155	94,8%	90,8%	89,3%	00:00:13
5	94	88	93,6%	88,1%	89,7%	00:00:08
6	45	39	82,5%	82,0%	79,9%	00:00:05
7	125	119	93,6%	90,6%	91,2%	00:00:10
8	125	119	95,0%	88,9%	94,3%	00:00:10
9	58	52	85,7%	83,9%	84,3%	00:00:06
10	68	62	83,8%	82,4%	80,7%	00:00:06
Average			88,9%	85,9%	86,7%	00:00:08
Standar Deviasi			0,052339	0,0377961	0,048637	

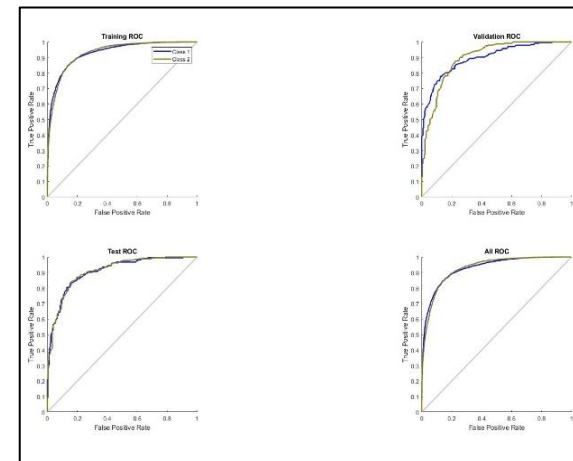
Berdasarkan Tabel 4.23 diketahui bahwa rata-rata akurasi *training* menggunakan model SCG dengan *hidden neuron 10* adalah 82-95%. Hasil akurasi *training* terbaik terdapat pada *run 8* dengan akurasi *training* sebesar 95,0%, akurasi validasi 88,9%, dan akurasi *testing* 94,3%. Akurasi *training* terbaik ini diperoleh pada *epoch* ke 119 dari total 125 *epoch* dengan waktu *training* selama 10 detik. Sementara itu, hasil akurasi *training* terendah terdapat pada *run 6* dengan akurasi *training* 82,5%, akurasi validasi 82,0%, dan akurasi *testing* 79,9%. Dari pengujian diatas, diperoleh hasil bahwa rata-rata akurasi *training* adalah 88,9%, akurasi validasi 85,9%, dan akurasi *testing* 86,7%. Rata-rata waktu *training* menggunakan model SCG *hidden neuron 10* adalah 10 detik. Visualisasi hasil *training* dapat dilihat pada Gambar 4.21.



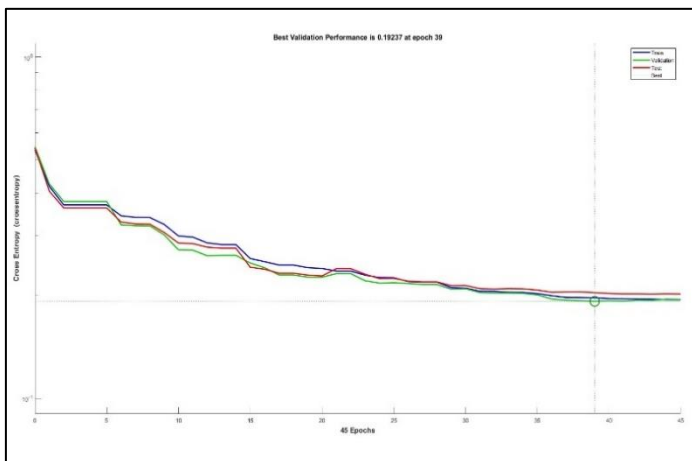
(a1)



(a2)



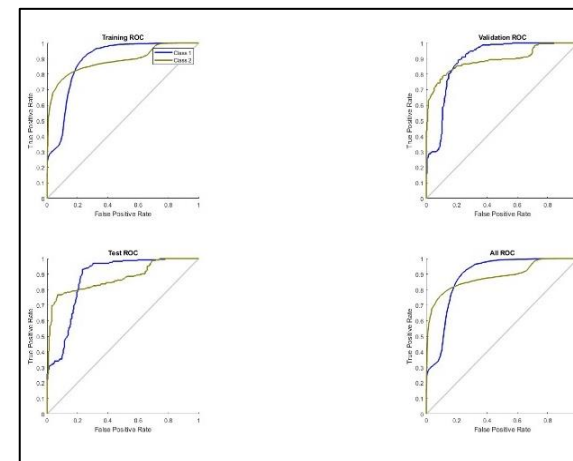
(a3)



(b1)

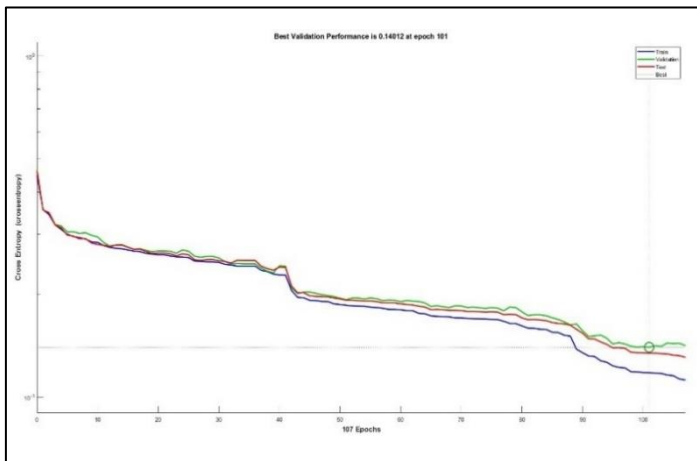


(b2)



(b3)

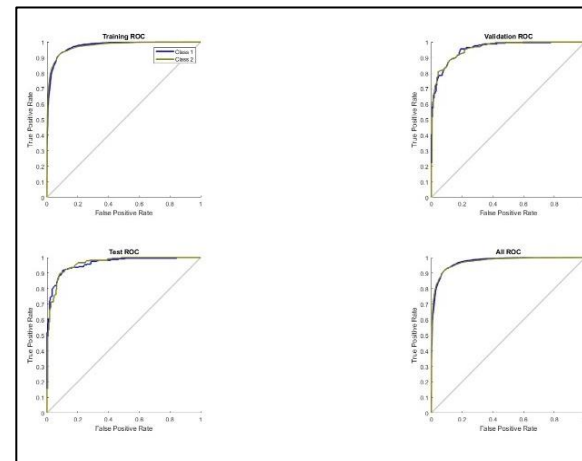
Gambar 4. 21 (a1) (b1) Performance Run 1 dan 2 Training SCG HN 10, (a2) (b2) Confusion Matrix Run 1 dan 2 Training SCG HN 10, (i3) (j3) ROC Run 1 dan 2 Training SCG HN 10



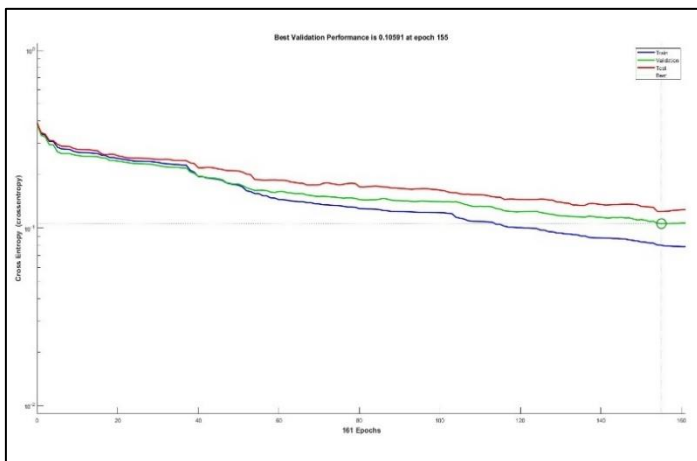
(c1)



(c2)



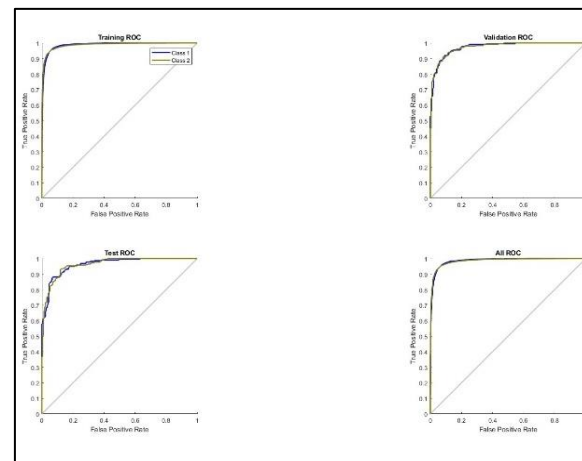
(c3)



(d1)

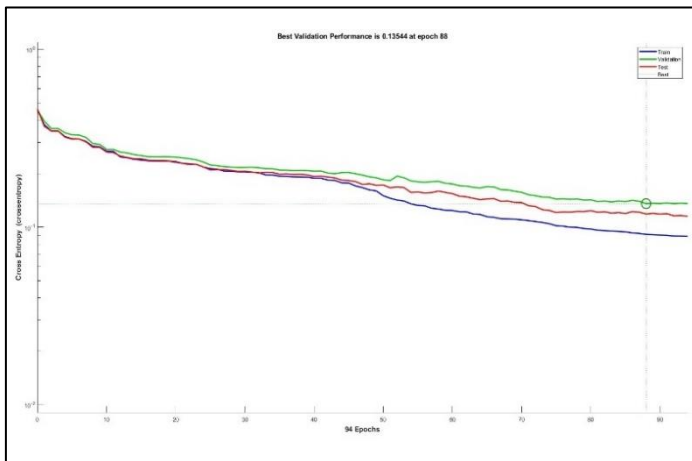


(d2)



(d3)

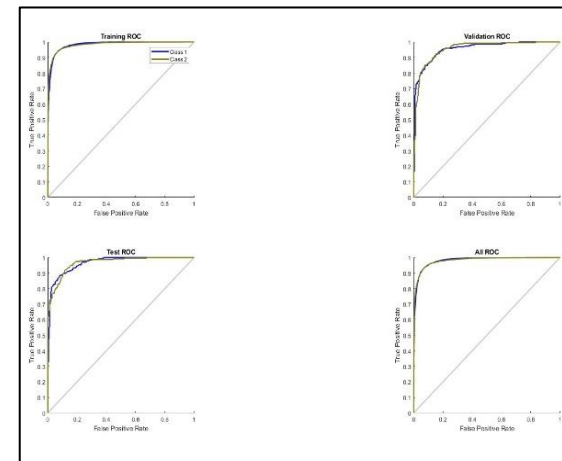
Gambar 4.21 (c1) (d1) Performance Run 3 dan 4 Training SCG HN 10, (c2) (d2) Confusion Matrix Run 3 dan 4 Training SCG HN 10, (c3) (d3) ROC Run 3 dan 4 Training SCG HN 10



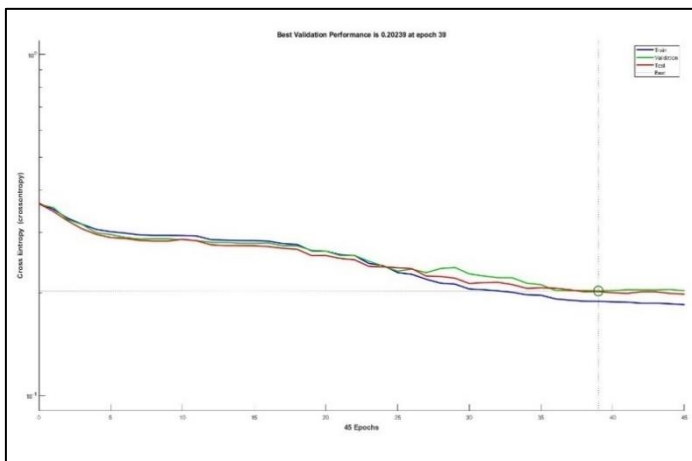
(e1)



(e2)



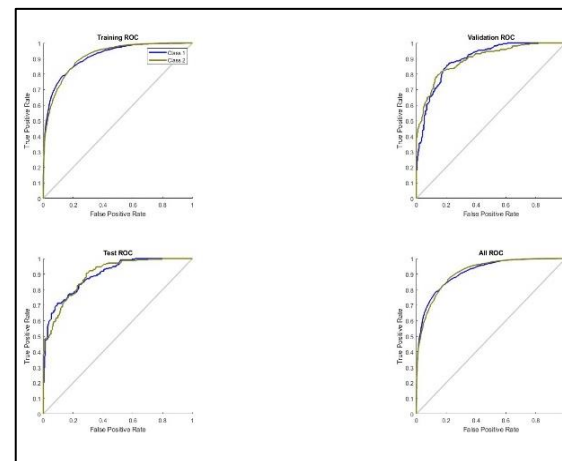
(e3)



(f1)

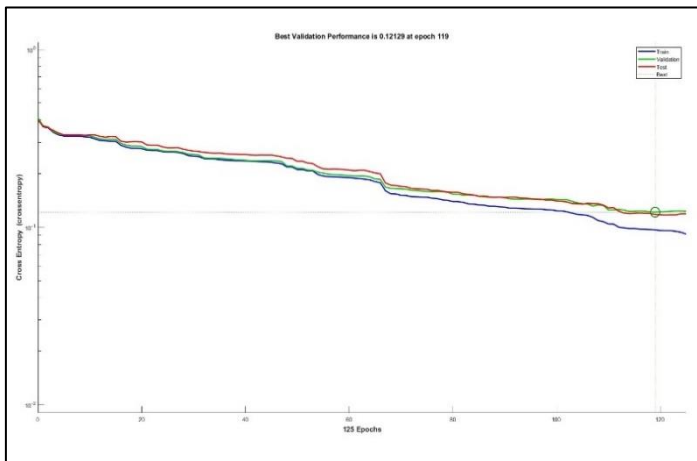


(f2)



(f3)

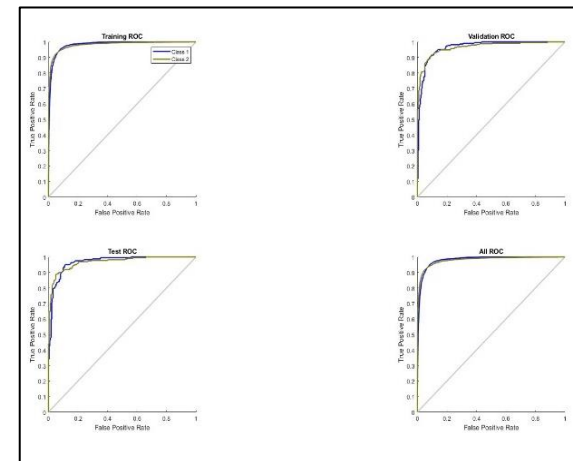
Gambar 4.21 (e1) (f1) Performance Run 5 dan 6 Training SCG HN 10, (e2) (f2) Confusion Matrix Run 5 dan 6 Training SCG HN 10, (e3) (f3) ROC Run 5 dan 6 Training SCG HN 10



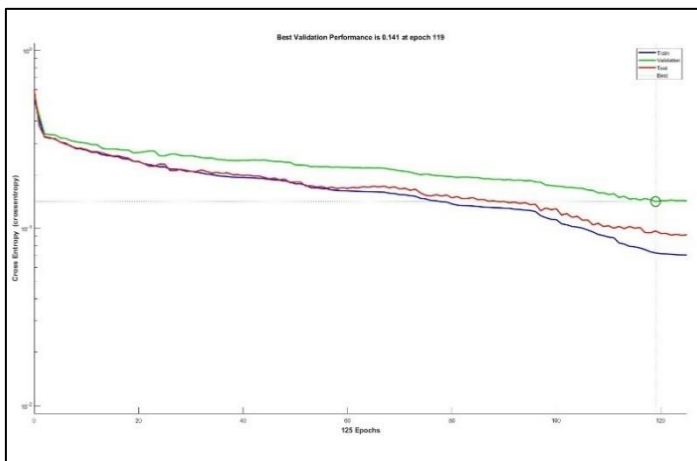
(g1)



(g2)



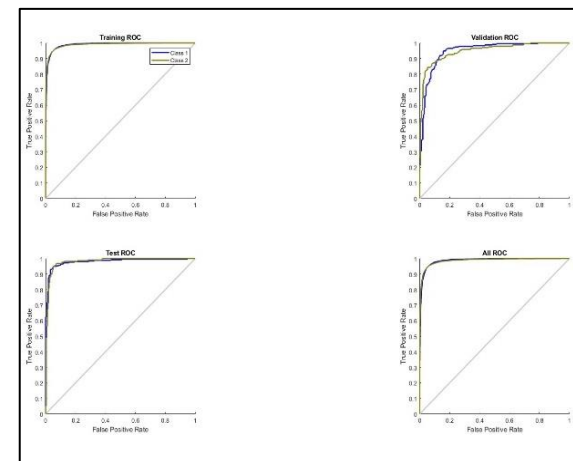
(g3)



(h1)

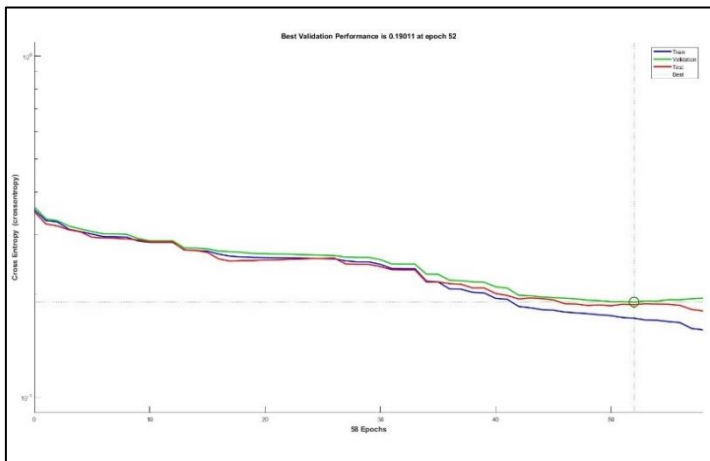


(h2)



(h3)

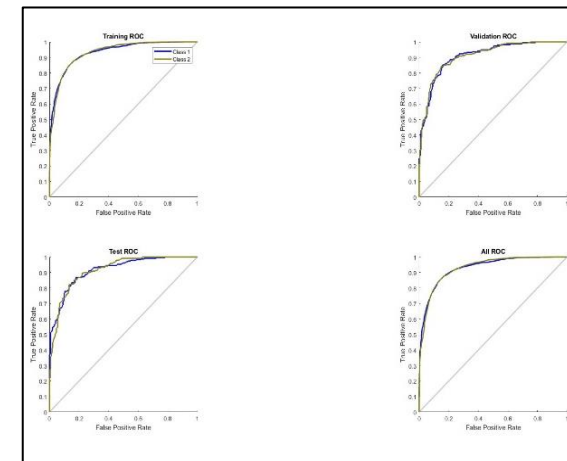
Gambar 4.21 (g1) (h1) Performance Run 7 dan 8 Training SCG HN 10, (g2) (h2) Confusion Matrix Run 7 dan 8 Training SCG HN 10, (g3) (h3) ROC Run 7 dan 8 Training SCG HN 10



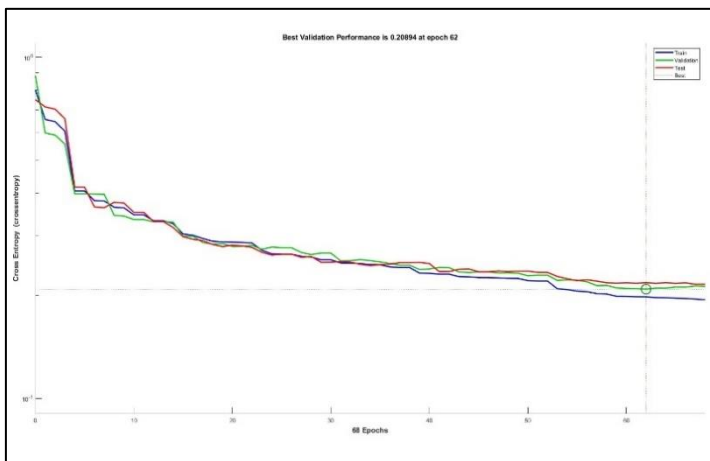
(i1)



(i2)



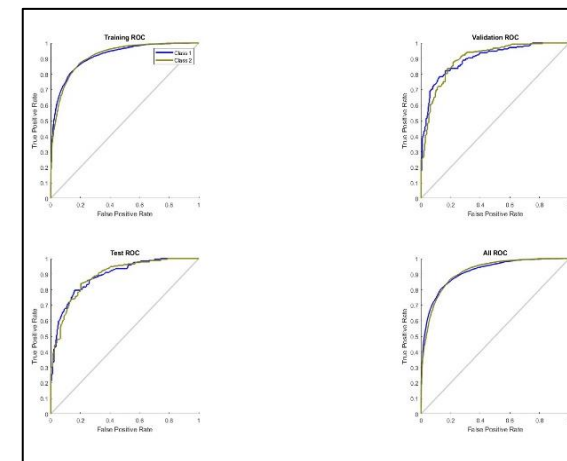
(i3)



(j1)



(j2)



(j3)

Gambar 4.21 (i1) (j1) Performance Run 9 dan 10 Training SCG HN 10, (i2) (j2) Confusion Matrix Run 9 dan 10 Training SCG HN 10, (i3) (j3) ROC Run 9 dan 10 Training SCG HN 10

Berdasarkan visualisasi hasil *training* di atas, dapat diketahui bahwa grafik *performance* terdiri dari 3 grafik, yaitu grafik biru merepresentasikan hasil *training*, grafik hijau merepresentasikan hasil validasi, dan grafik merah merepresentasikan hasil *testing*. Gambar 4.21 menampilkan sepuluh grafik *performane training SCG hidden neuron 10* dimana performa *error* menurun signifikan pada *range epoch* 0-5 pada *run* 1, 2, dan 10. Grafik performa *error run* 3 sampai 9 mengalami penurunan landai dari *epoch* 3 hingga *epoch* maksimalnya. Dari uraian di atas, dapat disimpulkan bahwa semakin rendah nilai entropi dari grafik (nilai error) maka semakin baik performa dari model.

Hasil *training* yang menunjukkan akurasi dari sistem ditunjukkan oleh *confusion matrix* dimana warna hijau merepresentasikan citra yang berhasil diklasifikasikan ke dalam kelas secara benar dan warna merah adalah citra yang gagal diklasifikasikan dalam kelasnya. Perbandingan data yang digunakan untuk *10-run cross validation* yaitu 90% dari total data (8.591) digunakan sebagai data *training*, 5% dari total data (477) digunakan sebagai data validasi, dan 5% lainnya (477) digunakan sebagai data *testing*. Dari *confusion matrix* tersebut, diketahui bahwa rata-rata akurasi *training*, validasi, dan *testing* berada pada $\geq 90\%$. Nilai rata-rata *error training* adalah $\leq 10\%$ (859) dan rata-rata nilai *error testing* adalah $\leq 10\%$ (47). Hal ini menunjukkan bahwa model memiliki performa yang baik.

Visualisasi grafik lainnya adalah grafik *Receiver Operating Characteristic* (ROC). Fungsi dari grafik ROC ini adalah untuk mengetahui hasil evaluasi performa *machine learning*. Pada grafik ROC, terdapat dua warna grafik yaitu biru yang menunjukkan klasifikasi citra kelas 1 dan grafik hijau menunjukkan klasifikasi citra kelas 2. Kurva validasi dan *testing* berada pada angka 0,9 dan menjauhi garis *threshold* kecuali *run* 2, 6, dan 10. Pada *run* 6 dan 10 kurva berada pada angka 0,7 sedangkan kurva *run* 2 mendekati garis *threshold*. Dengan demikian, dapat disimpulkan bahwa performa *machine learning* (*Multilayer Perceptron* model *SCG hidden neuron 10*) menunjukkan performa *perfect*.

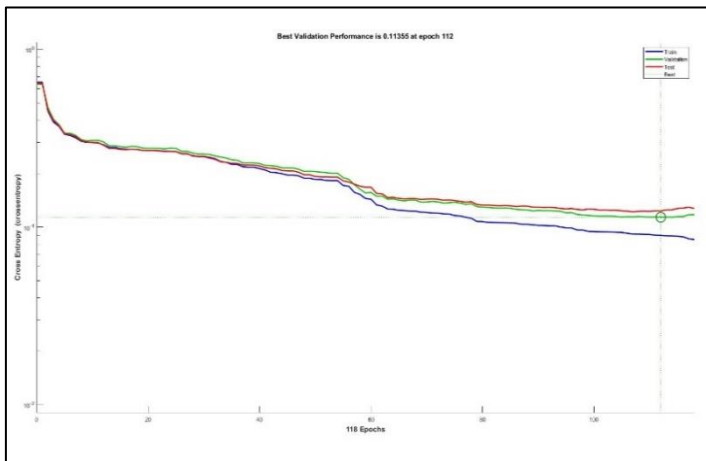
d. *Hidden Neuron 15*

Hasil akurasi *training* citra *body fluids* menggunakan model *Scaled Conjugate Gradient Backpropagation* (SCG) *hidden neuron 15* dapat dilihat pada Tabel 4.24.

Tabel 4. 24 Hasil Akurasi *Training* Menggunakan Model SCG *Hidden Neuron 15*

<i>Run</i>	<i>Epoch</i>		MLP-TrainSCG Hidden Neuron 15			<i>Elapsed Time</i>
	<i>All</i>	<i>Best</i>	<i>Acc Training</i>	<i>Acc Validation</i>	<i>Acc Testing</i>	
1	118	112	93,8%	91,4%	90,1%	00:00:11
2	145	139	94,4%	90,8%	92,7%	00:00:13
3	107	101	93,4%	90,1%	88,7%	00:00:10
4	57	51	88,1%	85,3%	88,9%	00:00:06
5	74	68	90,3%	86,6%	86,8%	00:00:07
6	129	123	95,6%	91,0%	90,8%	00:00:12
7	105	99	94,3%	89,7%	91,4%	00:00:09
8	119	113	94,3%	91,4%	90,6%	00:00:11
9	111	105	92,0%	91,2%	89,1%	00:00:10
10	81	75	92,6%	91,0%	88,5%	00:00:08
Average			92,9%	89,9%	89,8%	00:00:10
Standar Deviasi			0,022414	0,02148	0,016959	

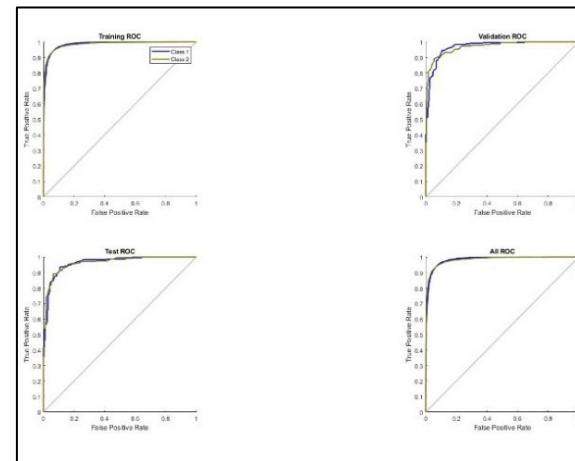
Berdasarkan Tabel 4.24 diketahui bahwa rata-rata akurasi *training* menggunakan model SCG dengan *hidden neuron 15* adalah 88-95%. Hasil akurasi *training* terbaik terdapat pada *run 6* dengan akurasi *training* sebesar 95,6%, akurasi validasi 91,0%, dan akurasi *testing* 90,8%. Akurasi *training* terbaik ini diperoleh pada *epoch* ke 123 dari total 129 *epoch* dengan waktu *training* selama 12 detik. Sementara itu, hasil akurasi *training* terendah terdapat pada *run 4* dengan akurasi *training* 88,1%, akurasi validasi 85,3%, dan akurasi *testing* 88,9%. Dari pengujian diatas, diperoleh hasil bahwa rata-rata akurasi *training* adalah 92,9%, akurasi validasi 89,9%, dan akurasi *testing* 89,8%. Rata-rata waktu *training* menggunakan model SCG *hidden neuron 15* adalah 10 detik. Visualisasi hasil *training* dapat dilihat pada Gambar 4.22.



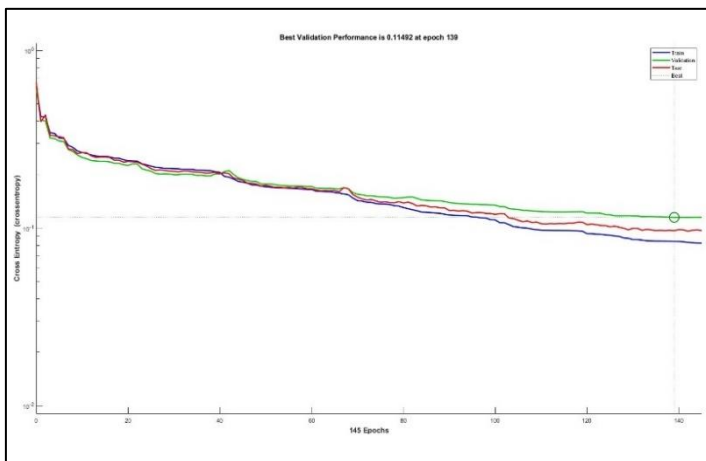
(a1)



(a2)



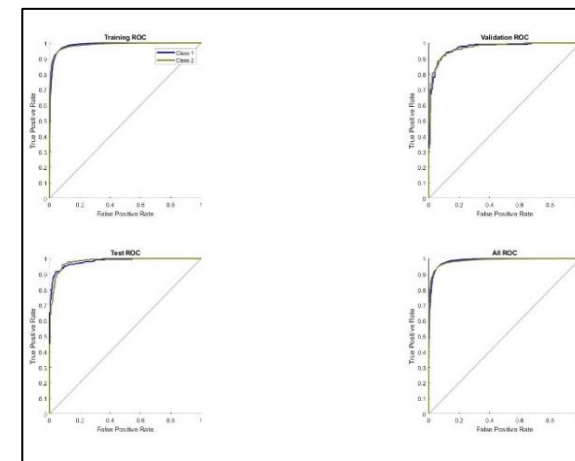
(a3)



(b1)

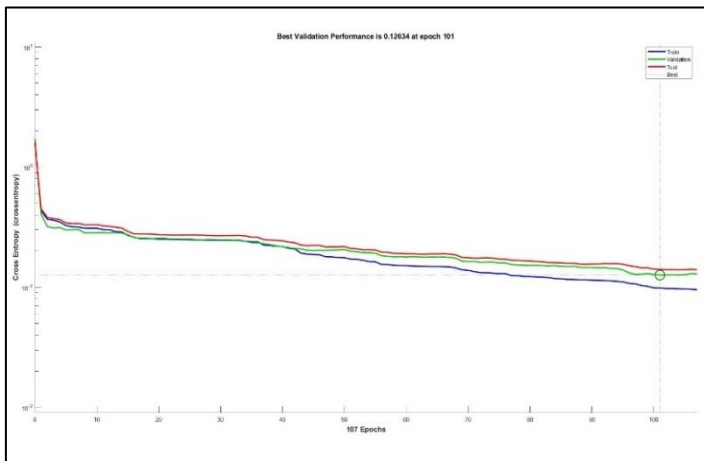


(b2)



(b3)

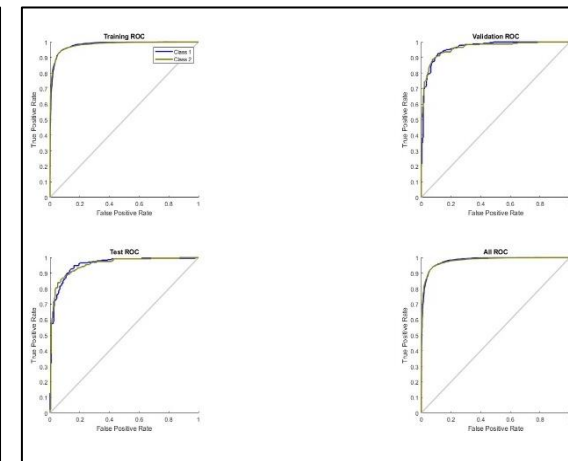
Gambar 4. 22 (a1) (b1) Performance Run 1 dan 2 Training SCG HN 15, (a2) (b2) Confusion Matrix Run 1 dan 2 Training SCG HN 15, (a3) (b3) ROC Run 1 dan 2 Training SCG HN 15



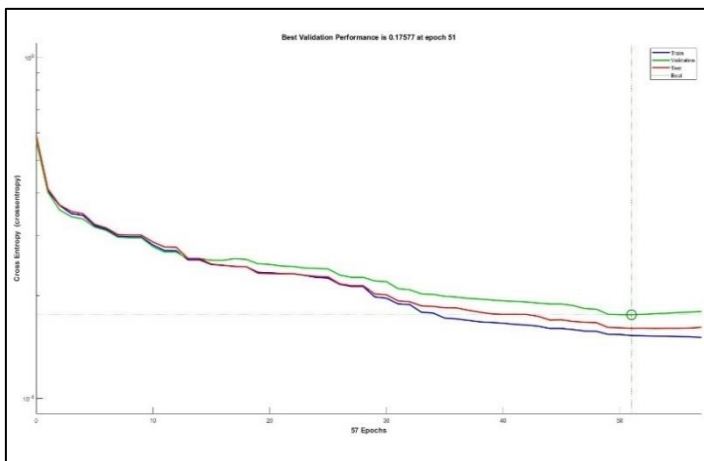
(c1)



(c2)



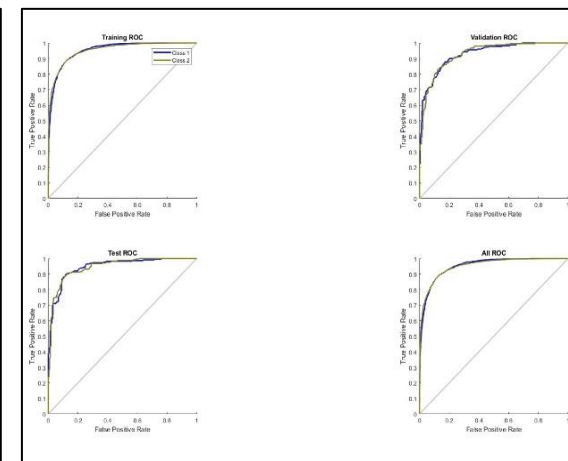
(c3)



(d1)

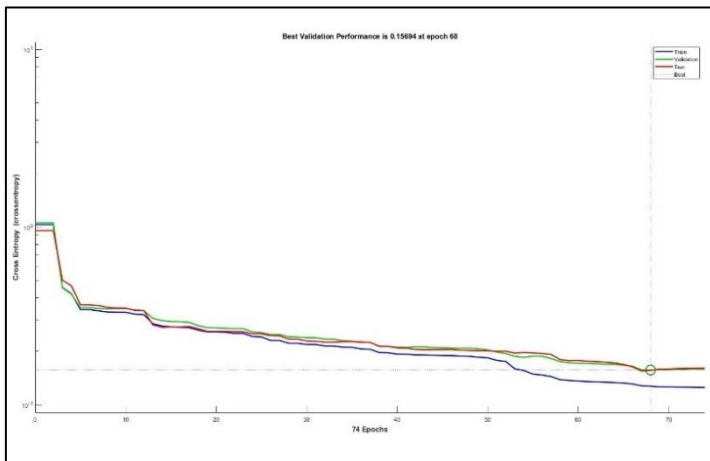


(d2)



(d3)

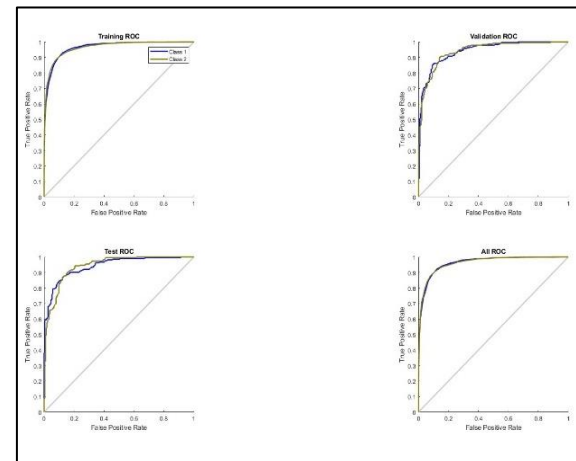
Gambar 4.22 (c1) (d1) Performance Run 3 dan 4 Training SCG HN 15, (c2) (d2) Confusion Matrix Run 3 dan 4 Training SCG HN 15, (c3) (d3) ROC Run 3 dan 4 Training SCG HN 15



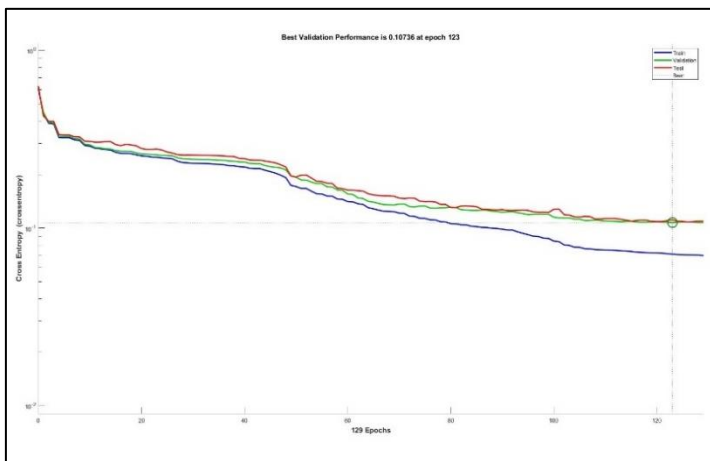
(e1)



(e2)



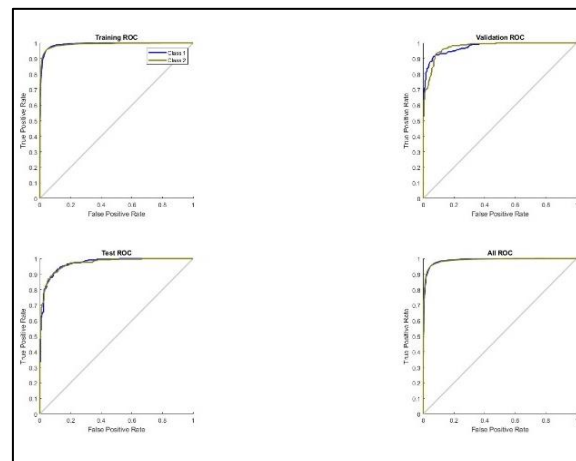
(e3)



(f1)

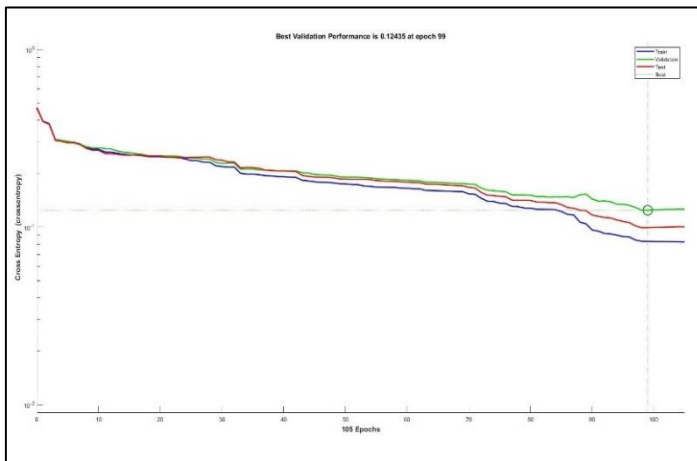


(f2)



(f3)

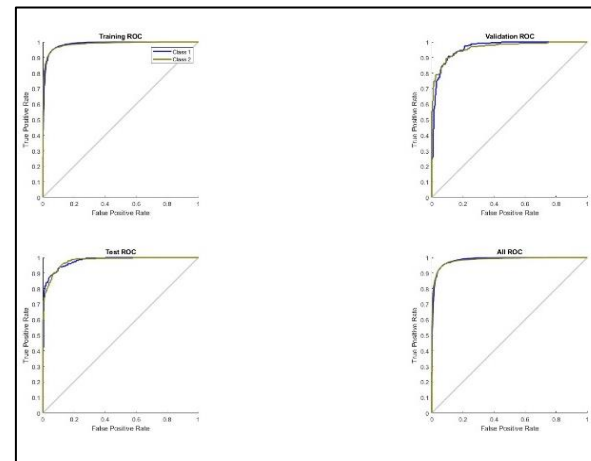
Gambar 4.22 (e1) (f1) Performance Run 5 dan 6 Training SCG HN 15, (e2) (f2) Confusion Matrix Run 5 dan 6 Training SCG HN 15, (e3) (f3) ROC Run 5 dan 6 Training SCG HN 15



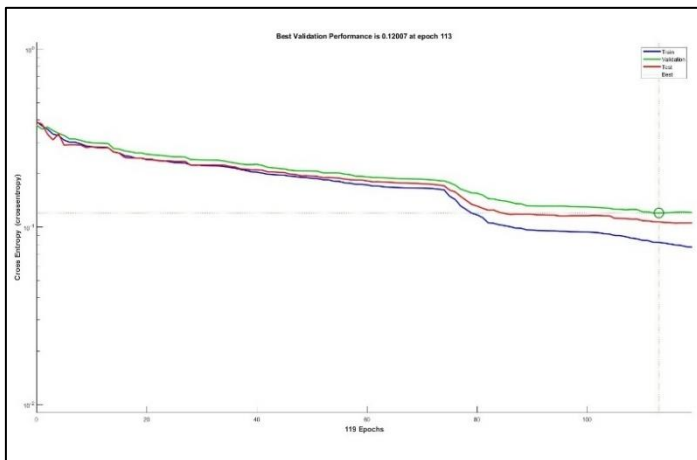
(g1)



(g2)



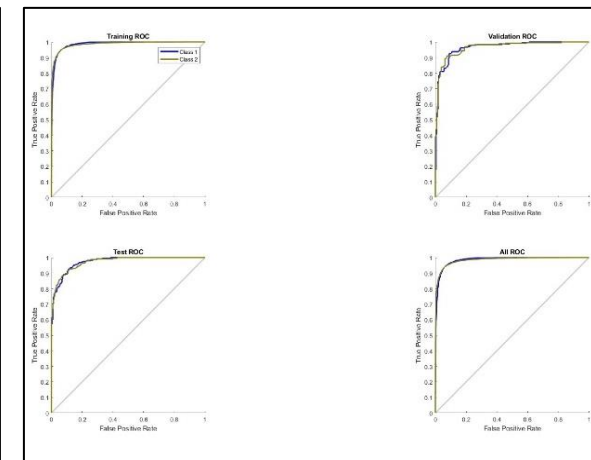
(g3)



(h1)

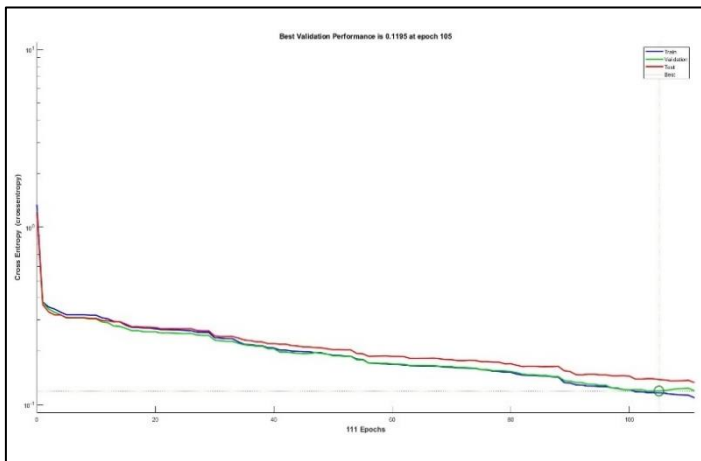


(h2)



(h3)

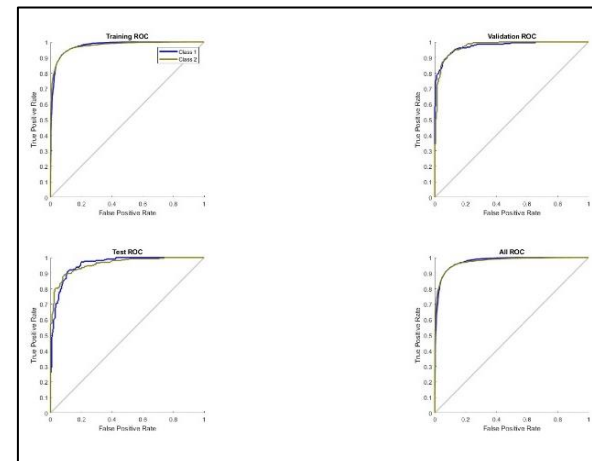
Gambar 4.22 (g1) (h1) Performance Run 7 dan 8 Training SCG HN 15, (g2) (h2) Confusion Matrix Run 7 dan 8 Training SCG HN 15, (g3) (h3) ROC Run 7 dan 8 Training SCG HN 15



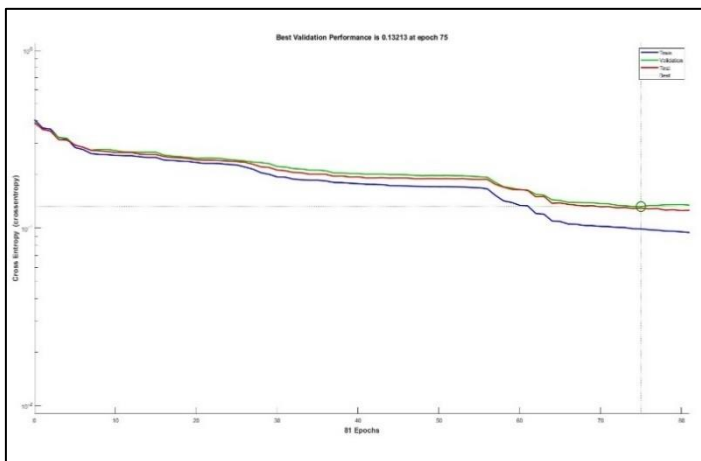
(i1)



(i2)



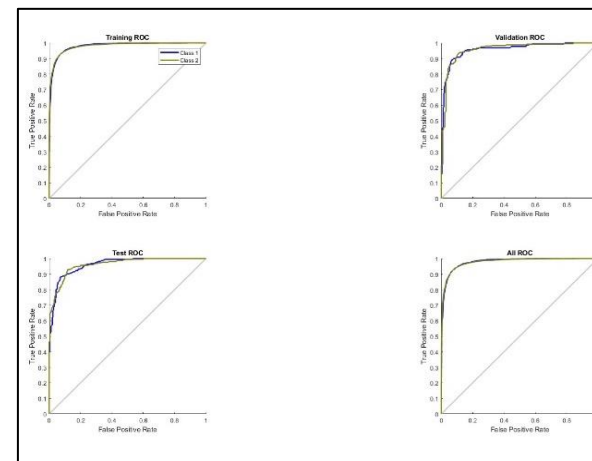
(i3)



(j1)



(j2)



(j3)

Gambar 4.22 (i1) (j1) Performance Run 9 dan 10 Training SCG HN 15, (i2) (j2) Confusion Matrix Run 9 dan 10 Training SCG HN 15, (i3) (j3) ROC Run 9 dan 10 Training SCG HN 15

Berdasarkan visualisasi hasil *training* di atas, dapat diketahui bahwa grafik *performance* terdiri dari 3 grafik, yaitu grafik biru merepresentasikan hasil *training*, grafik hijau merepresentasikan hasil validasi, dan grafik merah merepresentasikan hasil *testing*. Gambar 4.22 menampilkan sepuluh grafik *performane training SCG hidden neuron 15* dimana grafik mengalami penurunan *error* signifikan pada *range epoch* 0-5 kecuali *run* 8 dan 10. Performa grafik *error* menurun landai dari *epoch* 3 hingga *epoch* maksimalnya pada *run* 8 dan 10. Dari uraian di atas, dapat disimpulkan bahwa semakin rendah nilai entropi dari grafik (nilai *error*) maka semakin baik performa dari model.

Hasil *training* yang menunjukkan akurasi dari sistem ditunjukkan oleh *confusion matrix* dimana warna hijau merepresentasikan citra yang berhasil diklasifikasikan ke dalam kelas secara benar dan warna merah adalah citra yang gagal diklasifikasikan dalam kelasnya. Perbandingan data yang digunakan untuk *10-run cross validation* yaitu 90% dari total data (8.591) digunakan sebagai data *training*, 5% dari total data (477) digunakan sebagai data validasi, dan 5% lainnya (477) digunakan sebagai data *testing*. Dari *confusion matrix* tersebut, diketahui bahwa akurasi *training*, validasi, dan *testing* $\geq 90\%$. Nilai rata-rata *error training* adalah $\leq 10\%$ (859) dan rata-rata nilai *error testing* adalah $\leq 10\%$ (47). Hal ini menunjukkan bahwa model memiliki performa yang baik.

Visualisasi grafik lainnya adalah grafik *Receiver Operating Characteristic* (ROC). Fungsi dari grafik ROC ini adalah untuk mengetahui hasil evaluasi performa *machine learning*. Pada grafik ROC, terdapat dua warna grafik yaitu biru yang menunjukkan klasifikasi citra kelas 1 dan grafik hijau menunjukkan klasifikasi citra kelas 2. Kurva validasi dan *testing* mendekati angka 1 dan semakin menjauhi garis *threshold*. Dengan demikian, dapat disimpulkan bahwa performa *machine learning* (*Multilayer Perceptron* model *SCG hidden neuron 15*) menunjukkan performa perfect.

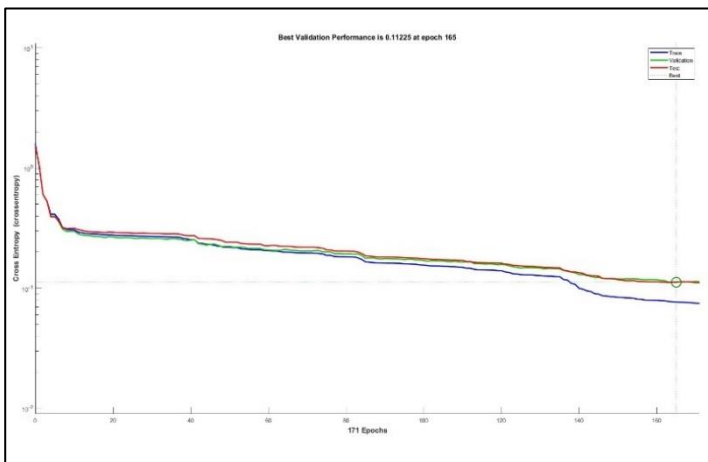
e. *Hidden Neuron 20*

Hasil akurasi *training* citra *body fluids* menggunakan model *Scaled Conjugate Gradient Backpropagation* (SCG) *hidden neuron 20* dapat dilihat pada Tabel 4.25.

Tabel 4. 25 Hasil Akurasi *Training* Menggunakan Model SCG Hidden Neuron 20

<i>Run</i>	<i>Epoch</i>		MLP-TrainSCG Hidden Neuron 20			<i>Elapsed Time</i>
	<i>All</i>	<i>Best</i>	<i>Acc Training</i>	<i>Acc Validation</i>	<i>Acc Testing</i>	
1	171	165	94,5%	90,8%	91,0%	00:00:16
2	62	56	87,1%	83,2%	83,2%	00:00:07
3	129	123	95,4%	91,8%	91,0%	00:00:13
4	151	145	96,3%	91,8%	93,1%	00:00:14
5	127	121	96,2%	92,0%	91,4%	00:00:12
6	112	106	93,1%	91,0%	91,2%	00:00:11
7	103	97	94,8%	91,0%	88,9%	00:00:10
8	107	101	94,1%	88,1%	90,1%	00:00:10
9	153	147	97,7%	91,0%	93,3%	00:00:14
10	94	88	91,9%	89,3%	87,4%	00:00:09
Average			94,1%	90,0%	90,1%	00:00:12
Standar Deviasi			0,0297	0,0267623	0,02983	

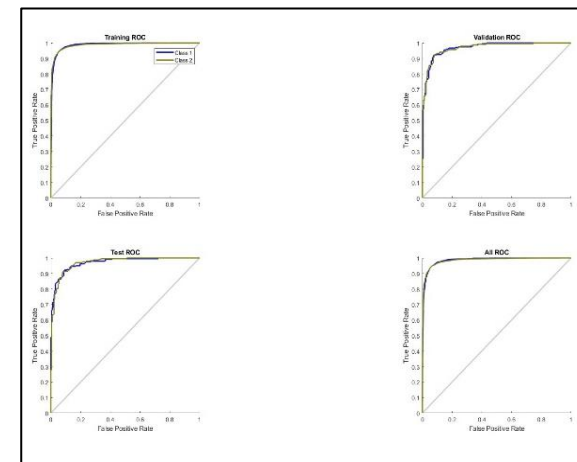
Berdasarkan Tabel 4.25 diketahui bahwa rata-rata akurasi *training* menggunakan model SCG dengan *hidden neuron 20* adalah 87-97%. Hasil akurasi *training* terbaik terdapat pada *run 9* dengan akurasi *training* sebesar 97,7%, akurasi validasi 91,0%, dan akurasi *testing* 93,3%. Akurasi *training* terbaik ini diperoleh pada *epoch* ke 147 dari total 153 *epoch* dengan waktu *training* selama 14 detik. Sementara itu, hasil akurasi *training* terendah terdapat pada *run 2* dengan akurasi *training* 87,1%, akurasi validasi 83,2%, dan akurasi *testing* 83,2%. Dari pengujian diatas, diperoleh hasil bahwa rata-rata akurasi *training* adalah 94,1%, akurasi validasi 90,0%, dan akurasi *testing* 90,1%. Rata-rata waktu *training* menggunakan model SCG *hidden neuron 20* adalah 12 detik. Visualisasi hasil *training* dapat dilihat pada Gambar 4.23.



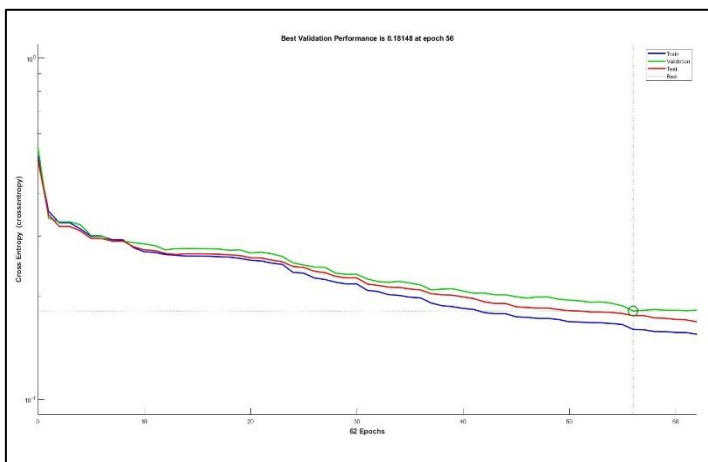
(a1)



(a2)



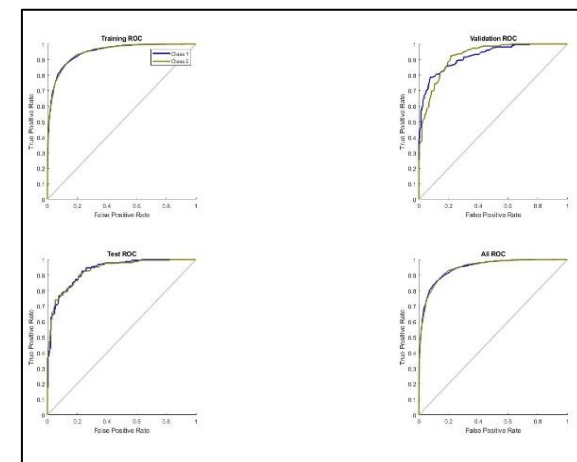
(a3)



(b1)

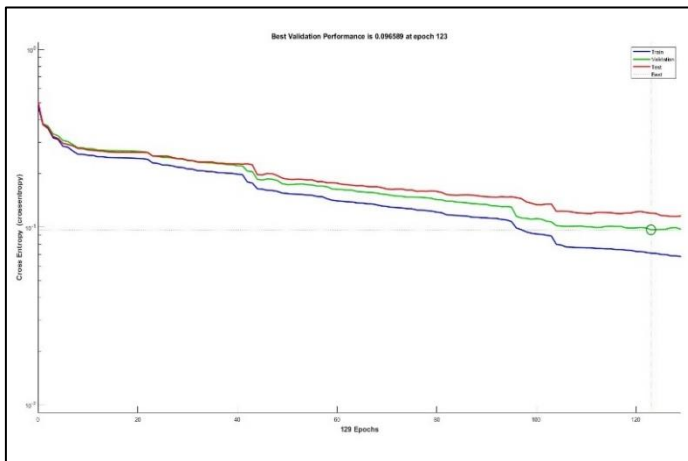


(b2)



(b3)

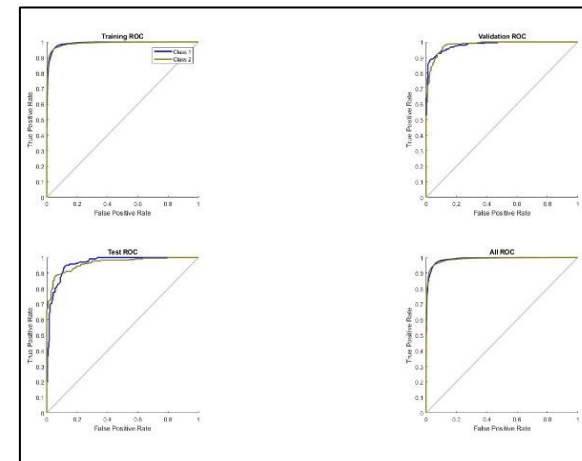
Gambar 4. 23 (a1) (b1) Performance Run 1 dan 2 Training SCG HN 20, (a2) (b2) Confusion Matrix Run 1 dan 2 Training SCG HN 20, (a3) (b3) ROC Run 1 dan 2 Training SCG HN 20



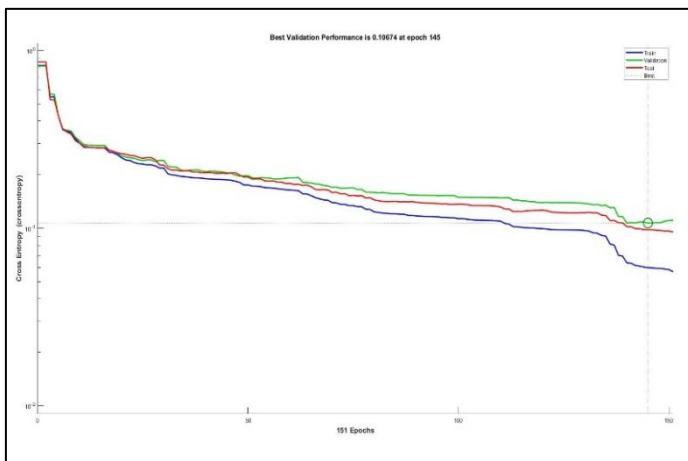
(c1)



(c2)



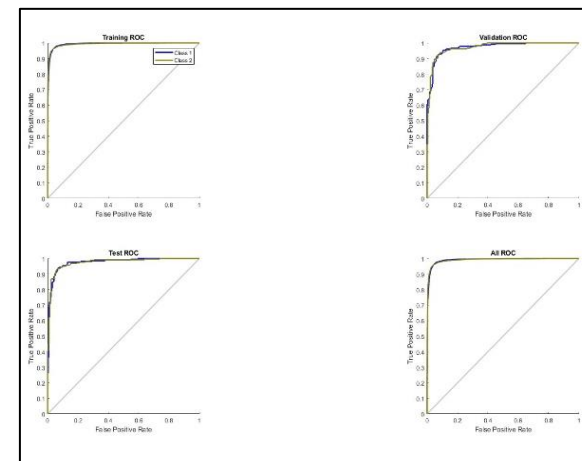
(c3)



(d1)

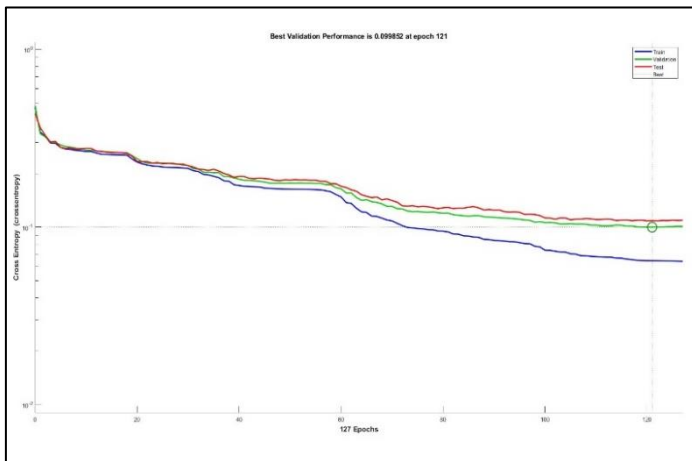


(d2)



(d3)

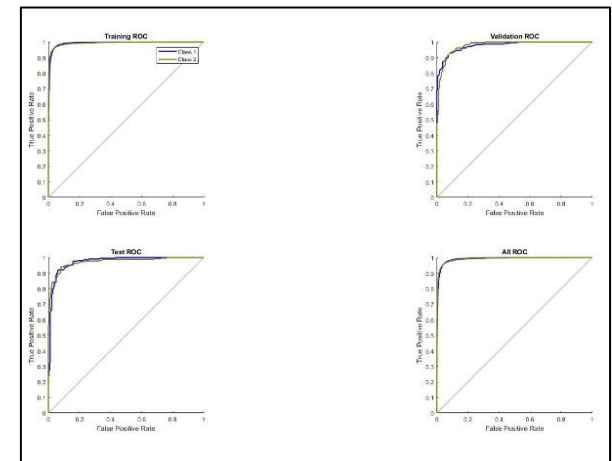
Gambar 4.23 (c1) (d1) Performance Run 3 dan 4 Training SCG HN 20, (c2) (d2) Confusion Matrix Run 3 dan 4 Training SCG HN 20, (c3) (d3) ROC Run 3 dan 4 Training SCG HN 20



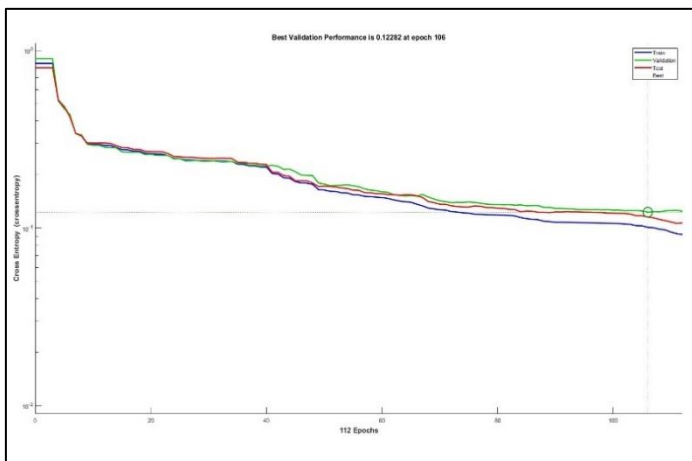
(e1)



(e2)



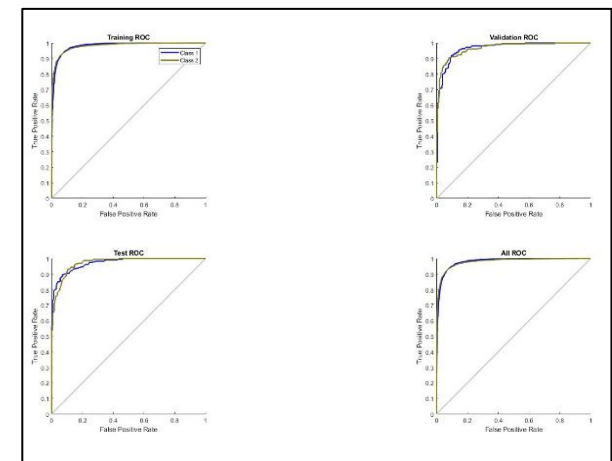
(e3)



(f1)

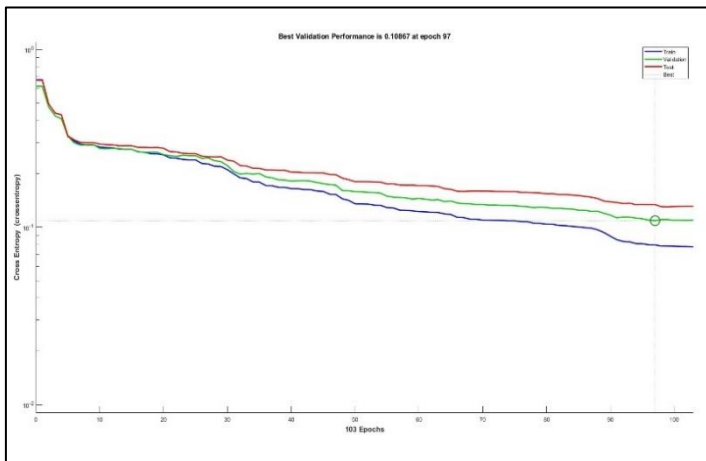


(f2)



(f3)

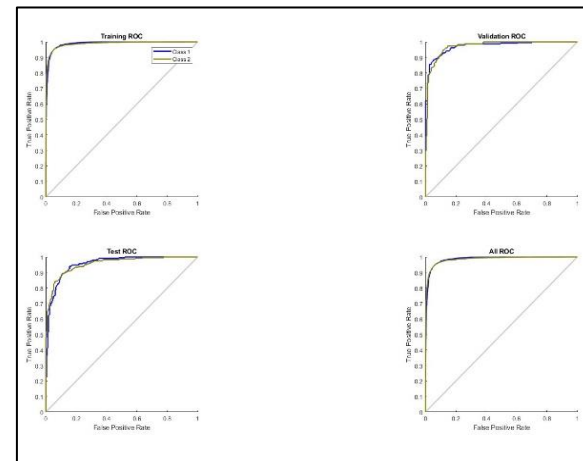
Gambar 4.23 (e1) (f1) Performance Run 5 dan 6 Training SCG HN 20, (e2) (f2) Confusion Matrix Run 5 dan 6 Training SCG HN 20, (e3) (f3) ROC Run 5 dan 6 Training SCG HN 20



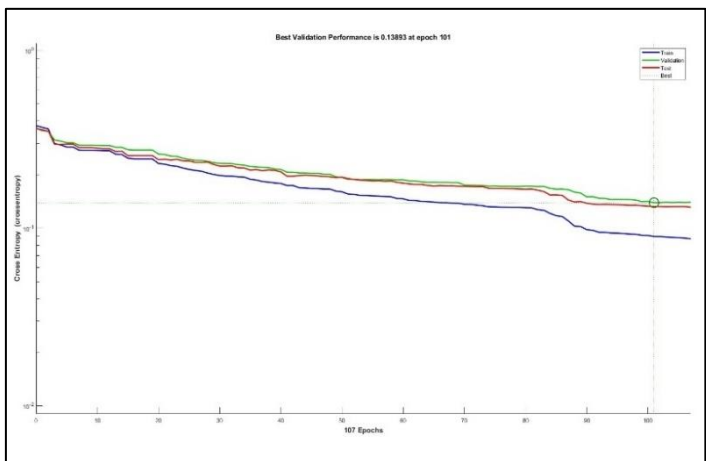
(g1)



(g2)



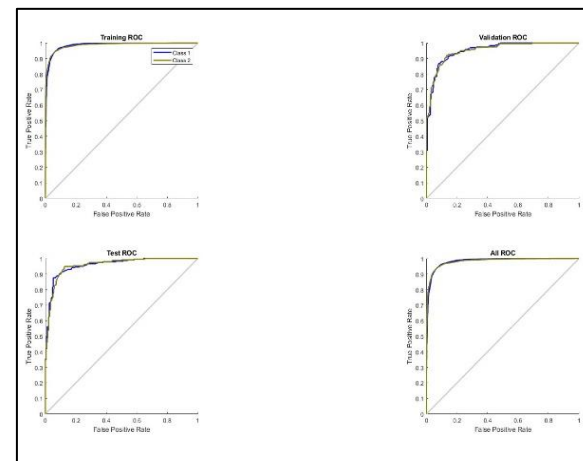
(g3)



(h1)

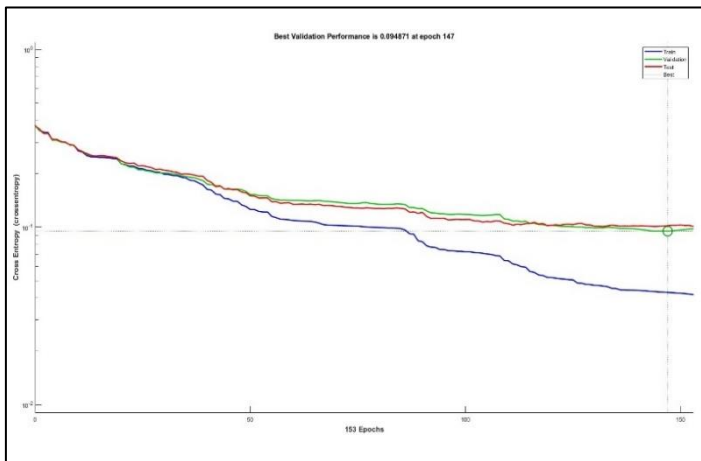


(h2)



(h3)

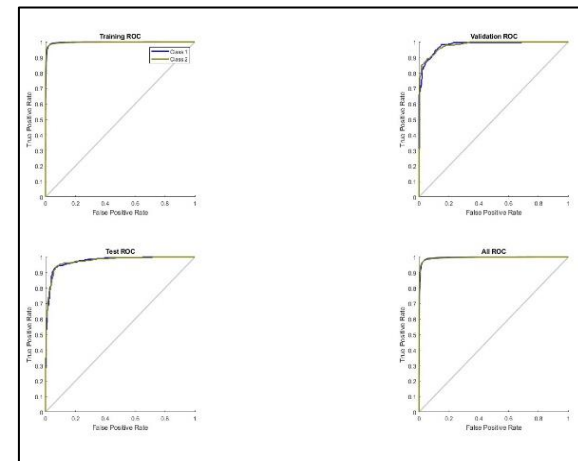
Gambar 4.23 (g1) (h1) Performance Run 7 dan 8 Training SCG HN 20, (g2) (h2) Confusion Matrix Run 7 dan 8 Training SCG HN 20, (g3) (h3) ROC Run 7 dan 8 Training SCG HN 20



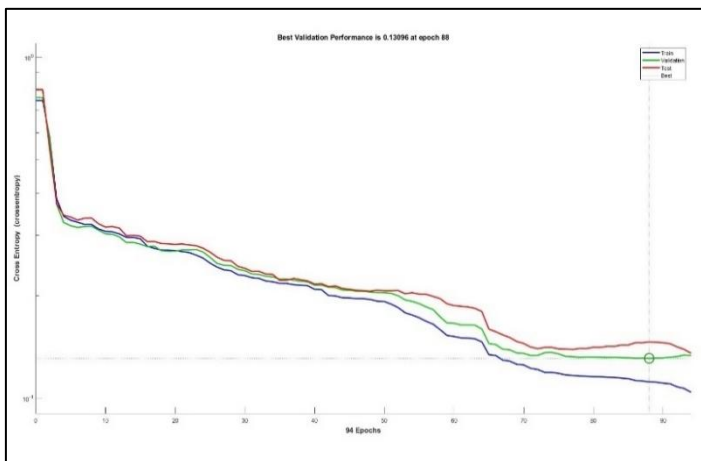
(i1)



(i2)



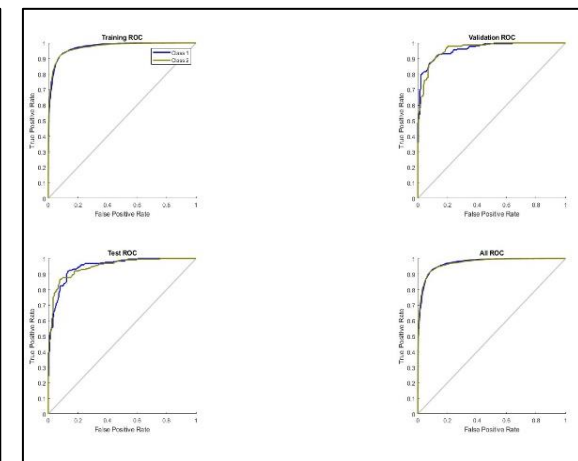
(i3)



(j1)



(j2)



(j3)

Gambar 4.23 (i1) (j1) Performance Run 9 dan 10 Training SCG HN 20, (i2) (j2) Confusion Matrix Run 9 dan 10 Training SCG HN 20, (i3) (j3) ROC Run 9 dan 10 Training SCG HN 20

Berdasarkan visualisasi hasil *training* di atas, dapat diketahui bahwa grafik *performance* terdiri dari 3 grafik, yaitu grafik biru merepresentasikan hasil *training*, grafik hijau merepresentasikan hasil validasi, dan grafik merah merepresentasikan hasil *testing*. Gambar 4.23 menampilkan sepuluh grafik *performane training SCG hidden neuron 20* dimana penurunan *error* signifikan terdapat pada *range epoch* 0-5 kecuali *run* 3, 8, dan 9. Performa grafik *run* 3, 8, dan 9 mengalami penurunan landai dari *epoch* 3 hingga *epoch* maksimal. Dari uraian di atas, dapat disimpulkan bahwa semakin rendah nilai entropi dari grafik (nilai *error*) maka semakin baik performa dari model.

Hasil *training* yang menunjukkan akurasi dari sistem ditunjukkan oleh *confusion matrix* dimana warna hijau merepresentasikan citra yang berhasil diklasifikasikan ke dalam kelas secara benar dan warna merah adalah citra yang gagal diklasifikasikan dalam kelasnya. Perbandingan data yang digunakan untuk *10-run cross validation* yaitu 90% dari total data (8.591) digunakan sebagai data *training*, 5% dari total data (477) digunakan sebagai data validasi, dan 5% lainnya (477) digunakan sebagai data *testing*. Dari *confusion matrix* tersebut, diketahui bahwa akurasi *training*, validasi, dan *testing* $\geq 90\%$. Nilai rata-rata *error training* adalah $\leq 10\%$ (859) dan rata-rata nilai *error testing* adalah $\leq 10\%$ (47). Hal ini menunjukkan bahwa model memiliki performa yang baik.

Visualisasi grafik lainnya adalah grafik *Receiver Operating Characteristic* (ROC). Fungsi dari grafik ROC ini adalah untuk mengetahui hasil evaluasi performa *machine learning*. Pada grafik ROC, terdapat dua warna grafik yaitu biru yang menunjukkan klasifikasi citra kelas 1 dan grafik hijau menunjukkan klasifikasi citra kelas 2. Kurva validasi dan *testing* mendekati angka 1 dan menjauhi garis *threshold* kecuali pada *run* 2. Meskipun demikian, performa *machine learning* (*Multilayer Perceptron* model *SCG hidden neuron 20*) tetap menunjukkan performa perfect.

4.3.2.2 One Step Secant Backpropagation (OSS)

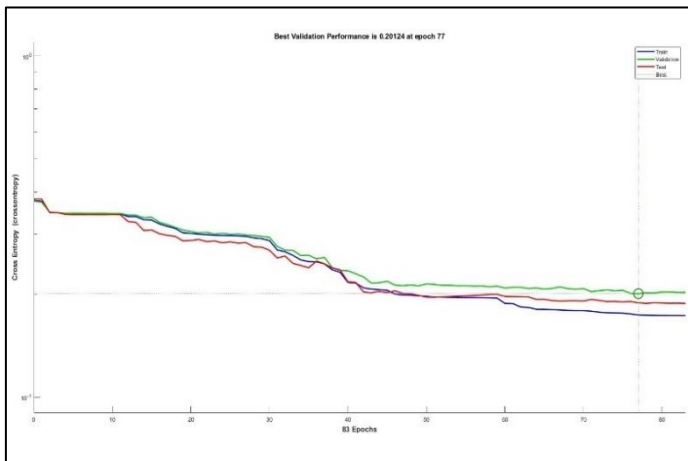
a. Hidden Neuron 1

Hasil akurasi *training* citra *body fluids* menggunakan model *One Step Secant Backpropagation (OSS) hidden neuron 1* dapat dilihat pada Tabel 4.26.

Tabel 4. 26 Hasil Akurasi *Training* Menggunakan Model OSS *Hidden Neuron 1*

<i>Run</i>	<i>Epoch</i>		MLP-TrainOSS Hidden Neuron 1			<i>Elapsed Time</i>
	<i>All</i>	<i>Best</i>	<i>Acc Training</i>	<i>Acc Validation</i>	<i>Acc Testing</i>	
1	83	77	86,6%	83,0%	84,5%	00:00:08
2	37	31	86,2%	82,6%	83,0%	00:00:04
3	51	45	85,0%	82,8%	86,2%	00:00:05
4	60	54	85,3%	82,8%	85,3%	00:00:06
5	21	15	73,8%	73,4%	73,6%	00:00:03
6	18	12	51,7%	52,6%	55,1%	00:00:03
7	54	48	85,6%	82,4%	83,6%	00:00:06
8	21	15	74,6%	73,4%	75,5%	00:00:03
9	49	43	85,4%	82,6%	82,0%	00:00:05
10	16	10	51,3%	50,7%	49,5%	00:00:03
Average			76,6%	74,6%	75,8%	00:00:05
Standar Deviasi			0,14019	0,1270154	0,131247	

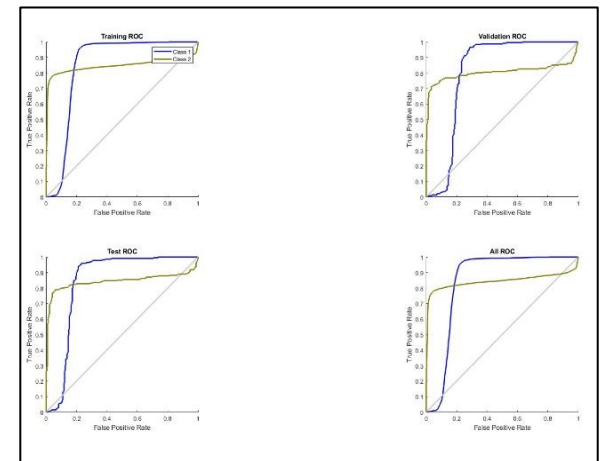
Berdasarkan Tabel 4.26 diketahui bahwa rata-rata akurasi *training* menggunakan model OSS dengan *hidden neuron 1* adalah 51-86%. Hasil akurasi *training* terbaik terdapat pada *run 1* dengan akurasi *training* sebesar 86,6%, akurasi validasi 83,0%, dan akurasi *testing* 84,5%. Akurasi *training* terbaik ini diperoleh pada *epoch* ke 77 dari total 83 *epoch* dengan waktu *training* selama 8 detik. Sementara itu, hasil akurasi *training* terendah terdapat pada *run 10* dengan akurasi *training* 51,3%, akurasi validasi 50,7%, dan akurasi *testing* 49,5%. Dari pengujian diatas, diperoleh hasil bahwa rata-rata akurasi *training* adalah 76,6%, akurasi validasi 74,6%, dan akurasi *testing* 75,8%. Rata-rata waktu *training* menggunakan model OSS *hidden neuron 1* adalah 5 detik. Visualisasi hasil *training* dapat dilihat pada Gambar 4.24.



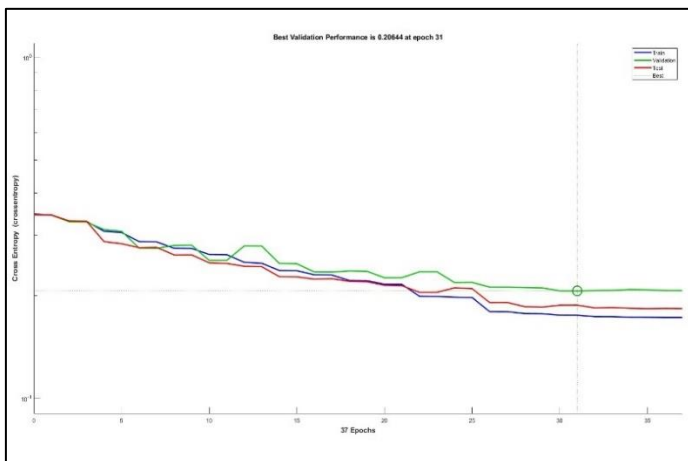
(a1)



(a2)



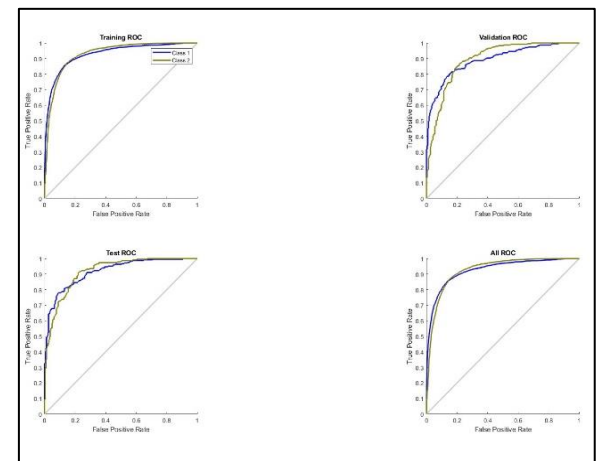
(a3)



(b1)

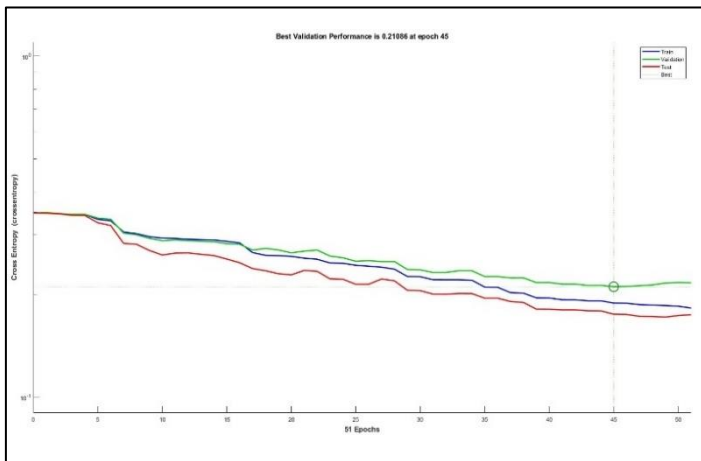


(b2)



(b3)

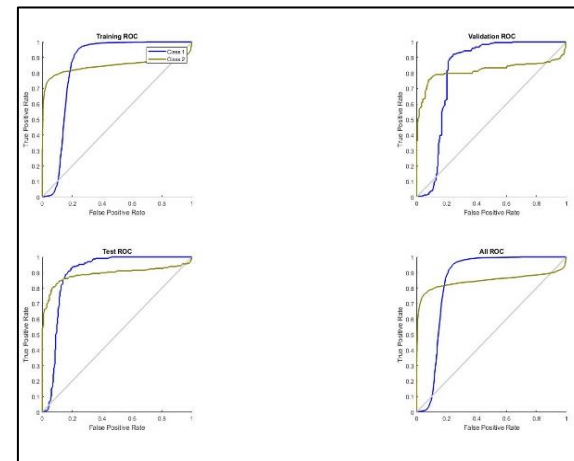
Gambar 4. 24 (a1) (b1) Performance Run 1 dan 2 Training OSS HN 1, (a2) (b2) Confusion Matrix Run 1 dan 2 Training OSS HN 1, (a3) (b3) ROC Run 1 dan 2 Training OSS HN 1



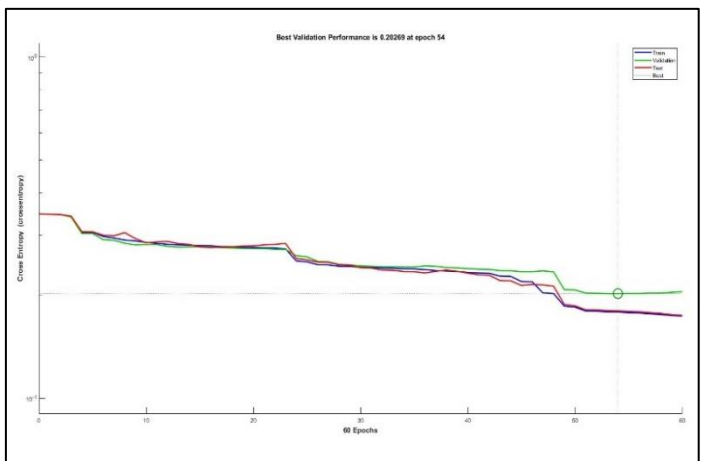
(c1)



(c2)



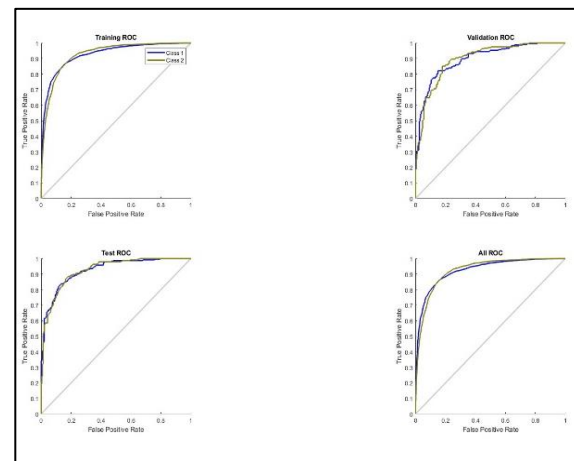
(c3)



(d1)

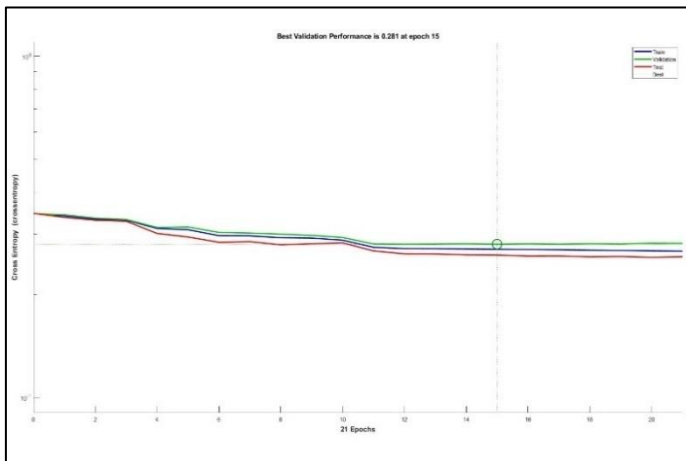


(d2)



(d3)

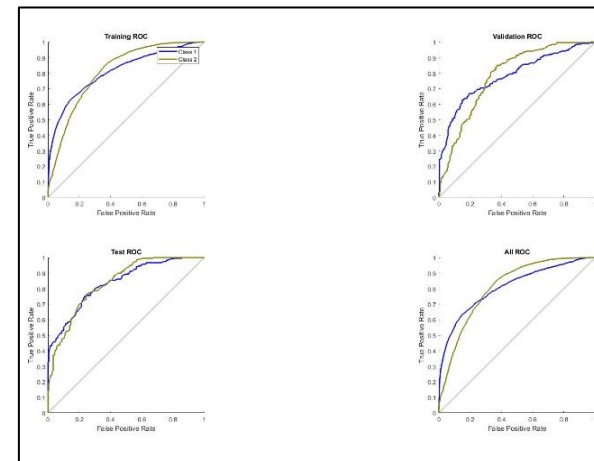
Gambar 4.24 (c1) (d1) Performance Run 3 dan 4 Training OSS HN 1, (c2) (d2) Confusion Matrix Run 3 dan 4 Training OSS HN 1, (c3) (d3) ROC Run 3 dan 4 Training OSS HN 1



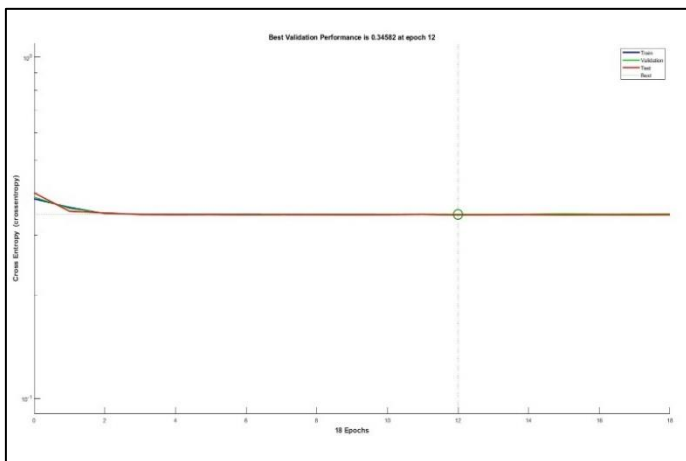
(e1)



(e2)



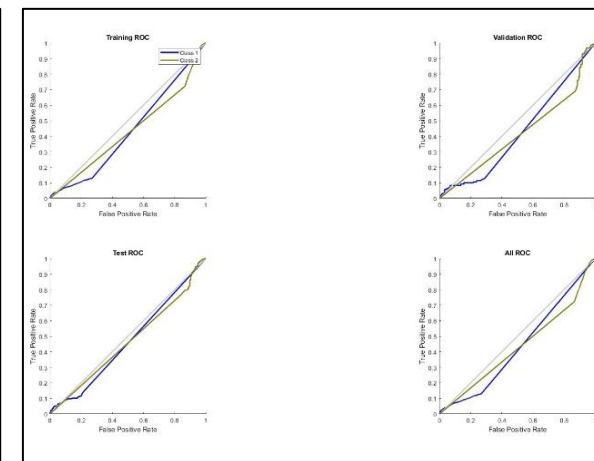
(e3)



(f1)

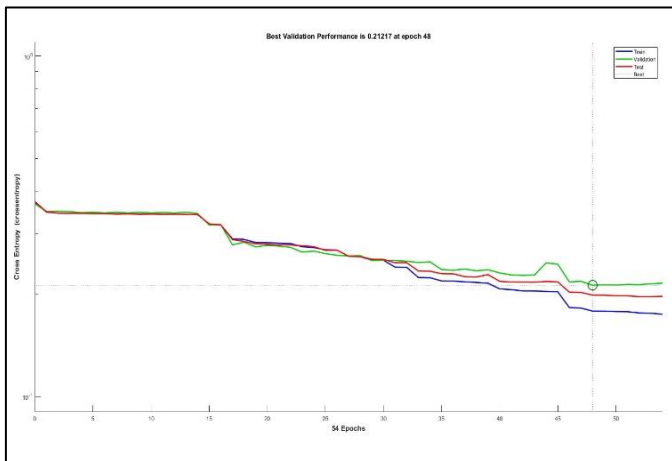


(f2)



(f3)

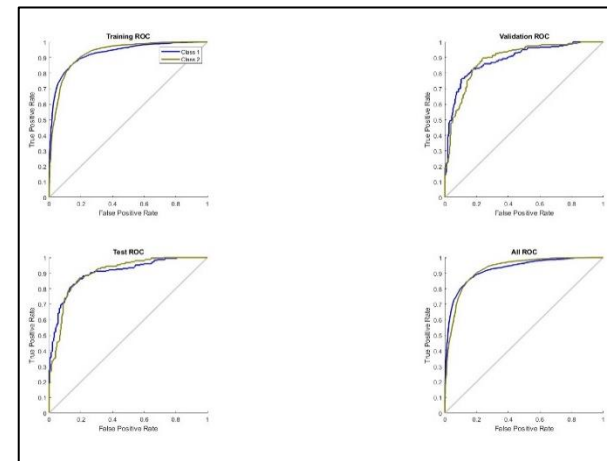
Gambar 4.24 (e1) (f1) Performance Run 5 dan 6 Training OSS HN 1, (e2) (f2) Confusion Matrix Run 5 dan 6 Training OSS HN 1, (e3) (f3) ROC Run 5 dan 6 Training OSS HN 1



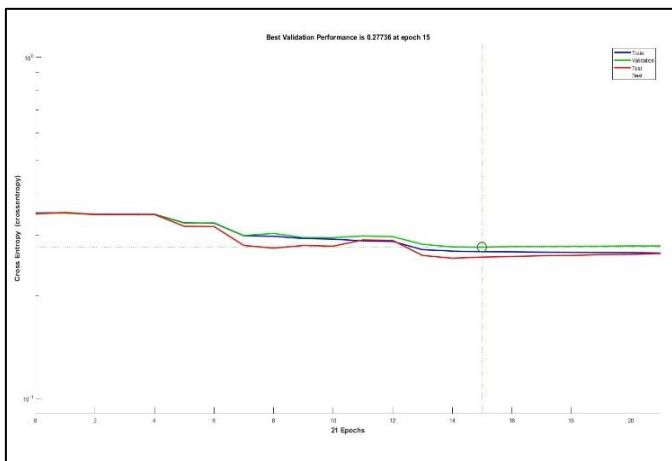
(g1)



(g2)



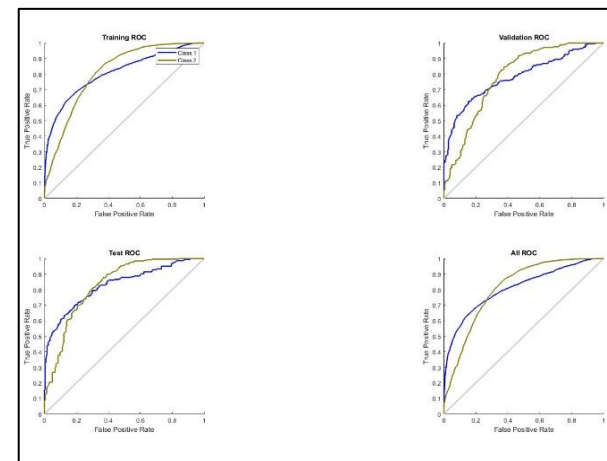
(g3)



(h1)

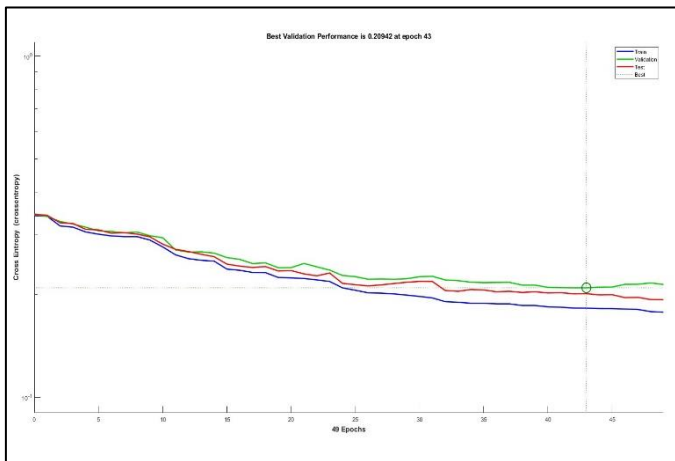


(h2)



(h3)

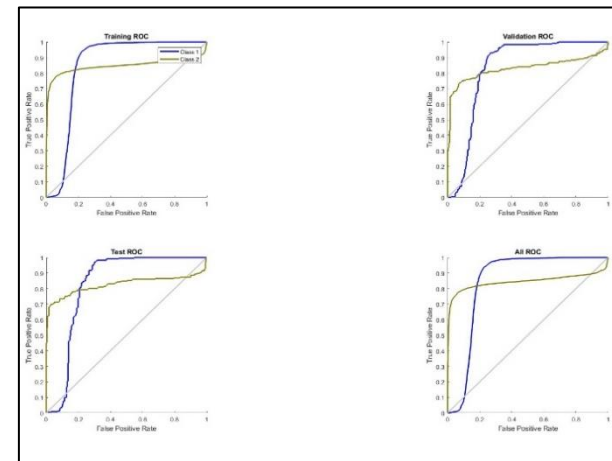
Gambar 4.24 (g1) (h1) Performance Run 7 dan 8 Training OSS HN 1, (g2) (h2) Confusion Matrix Run 7 dan 8 Training OSS HN 1, (g3) (h3) ROC Run 7 dan 8 Training OSS HN 1



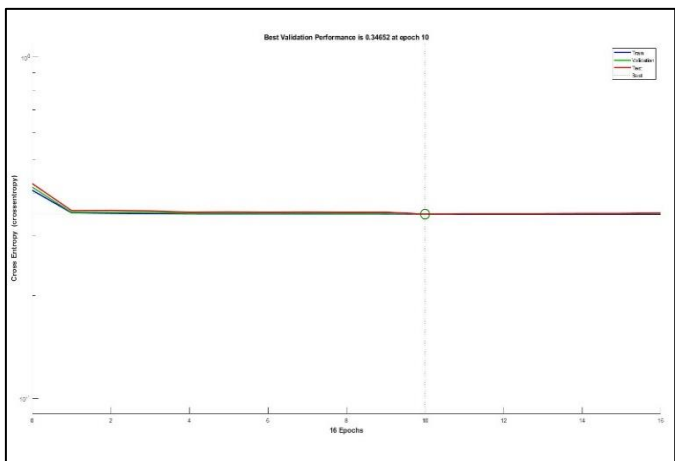
(i1)



(i2)



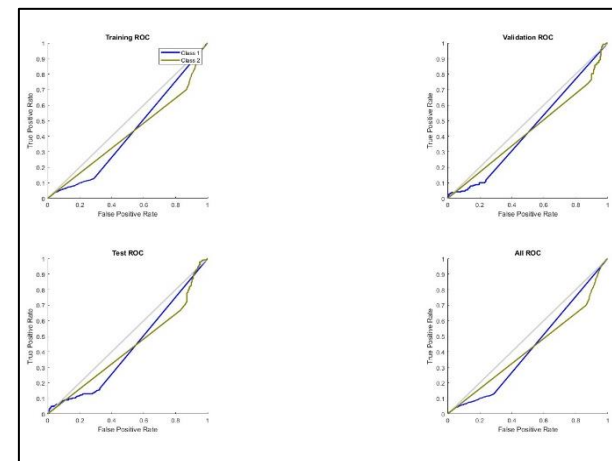
(i3)



(j1)



(j2)



(j3)

Gambar 4.24 (i1) (j1) Performance Run 9 dan 10 Training OSS HN 1, (i2) (j2) Confusion Matrix Run 9 dan 10 Training OSS HN 1, (i3) (j3) ROC Run 9 dan 10 Training OSS HN 1

Berdasarkan visualisasi hasil *training* di atas, dapat diketahui bahwa grafik *performance* terdiri dari 3 grafik, yaitu grafik biru merepresentasikan hasil *training*, grafik hijau merepresentasikan hasil validasi, dan grafik merah merepresentasikan hasil *testing*. Gambar 4.24 menampilkan sepuluh grafik *performane training OSS hidden neuron 1* dimana dari hasil tersebut tidak terdapat grafik yang mengalami penurunan signifikan. Penurunan melandai terjadi pada *epoch* 10 hingga *epoch* maksimal yaitu pada *run* 1 sampai 4 dan *run* 7. Performa grafik tidak mengalami penurunan signifikan dari awal hingga *epoch* maksimalnya terjadi pada *run* 5, 6, 8, dan 10.

Hasil *training* yang menunjukkan akurasi dari sistem ditunjukkan oleh *confusion matrix* dimana warna hijau merepresentasikan citra yang berhasil diklasifikasikan ke dalam kelas secara benar dan warna merah adalah citra yang gagal diklasifikasikan dalam kelasnya. Perbandingan data yang digunakan untuk *10-run cross validation* yaitu 90% dari total data (8.591) digunakan sebagai data *training*, 5% dari total data (477) digunakan sebagai data validasi, dan 5% lainnya (477) digunakan sebagai data *testing*. Dari *confusion matrix* tersebut, diketahui bahwa akurasi *training*, validasi, dan *testing* berada pada rentang 50-80%. Nilai *error* dari *training*, validasi, dan *testing* berada pada rentang 15-20% dari total data. Hal ini menunjukkan bahwa model memiliki performa yang tidak cukup baik.

Visualisasi grafik lainnya adalah grafik *Receiver Operating Characteristic* (ROC). Fungsi dari grafik ROC ini adalah untuk mengetahui hasil evaluasi performa *machine learning*. Pada grafik ROC, terdapat dua warna grafik yaitu biru yang menunjukkan klasifikasi citra kelas 1 dan grafik hijau menunjukkan klasifikasi citra kelas 2. Kurva validasi dan *testing* terlihat menjauhi garis *threshold* kecuali *run* 2, 4, dan 7. Ketiga kurva tersebut berada di setengah diagonal *threshold* dan mendekati garis *threshold*. Hasil *run* 6 dan 10, kurva validasi dan *testing* baerada di bawah garis *threshold*. Kurva berada di bawah garis *threshold*. Dengan demikian, dapat disimpulkan bahwa performa *machine learning* (*Multilayer Perceptron* model *OSS hidden neuron 1*) menunjukkan performa tidak perfect.

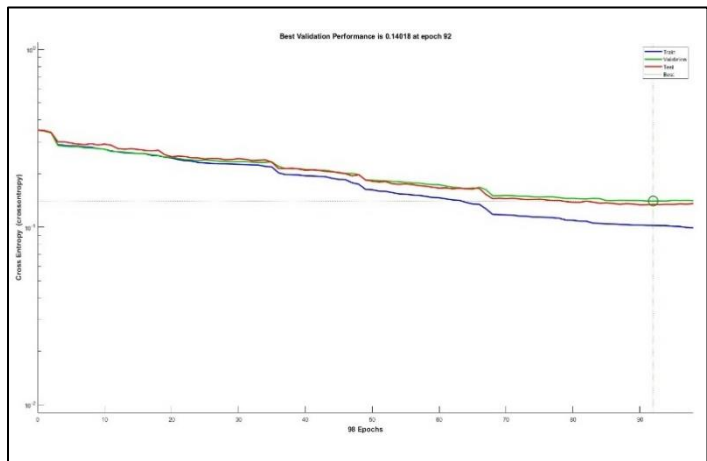
b. *Hidden Neuron 5*

Hasil akurasi *training* citra *body fluids* menggunakan model *One Step Secant Backpropagation* (OSS) *hidden neuron 5* dapat dilihat pada Tabel 4.27.

Tabel 4. 27 Hasil Akurasi *Training* Menggunakan Model OSS *Hidden Neuron 5*

<i>Run</i>	<i>Epoch</i>		MLP-TrainOSS <i>Hidden Neuron 5</i>			<i>Elapsed Time</i>
	<i>All</i>	<i>Best</i>	<i>Acc Training</i>	<i>Acc Validation</i>	<i>Acc Testing</i>	
1	98	92	93,3%	89,3%	89,7%	00:00:14
2	66	60	86,4%	83,9%	83,6%	00:00:10
3	61	55	83,5%	80,9%	78,4%	00:00:09
4	89	83	87,9%	83,6%	84,9%	00:00:13
5	57	51	82,1%	79,2%	79,9%	00:00:09
6	78	72	92,8%	90,8%	91,4%	00:00:12
7	95	89	91,9%	88,1%	88,1%	00:00:14
8	77	71	89,0%	83,2%	84,5%	00:00:12
9	35	29	82,8%	76,7%	78,8%	00:00:06
10	56	50	85,8%	80,9%	80,5%	00:00:08
Average			87,6%	83,7%	84,0%	00:00:11
Standar Deviasi			0,04148	0,0455026	0,046276	

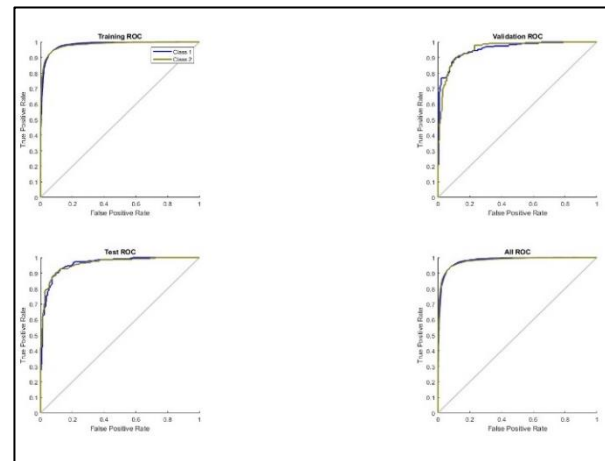
Berdasarkan Tabel 4.27 diketahui bahwa rata-rata akurasi *training* menggunakan model OSS dengan *hidden neuron 5* adalah 82-93%. Hasil akurasi *training* terbaik terdapat pada *run 1* dengan akurasi *training* sebesar 93,3%, akurasi validasi 89,3%, dan akurasi *testing* 89,7%. Akurasi *training* terbaik ini diperoleh pada *epoch* ke 92 dari total 98 *epoch* dengan waktu *training* selama 14 detik. Sementara itu, hasil akurasi *training* terendah terdapat pada *run 5* dengan akurasi *training* 82,1%, akurasi validasi 79,2%, dan akurasi *testing* 79,9%. Dari pengujian diatas, diperoleh hasil bahwa rata-rata akurasi *training* adalah 87,6%, akurasi validasi 83,7%, dan akurasi *testing* 84,0%. Rata-rata waktu *training* menggunakan model OSS *hidden neuron 5* adalah 11 detik. Visualisasi hasil *training* dapat dilihat pada Gambar 4.25.



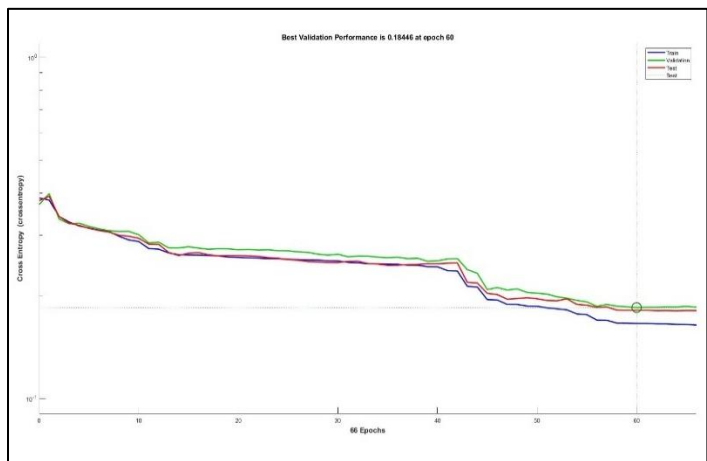
(a1)



(a2)



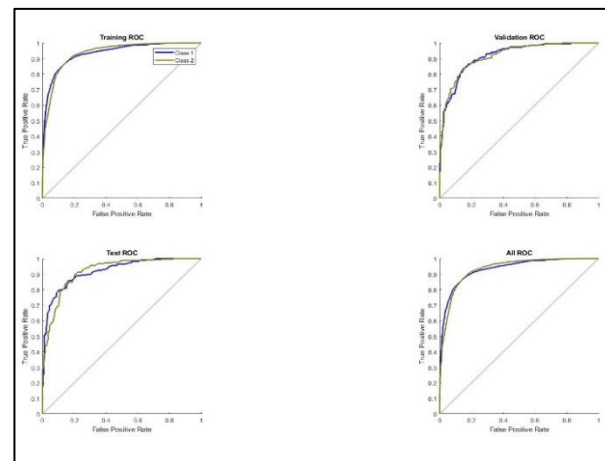
(a3)



(b1)

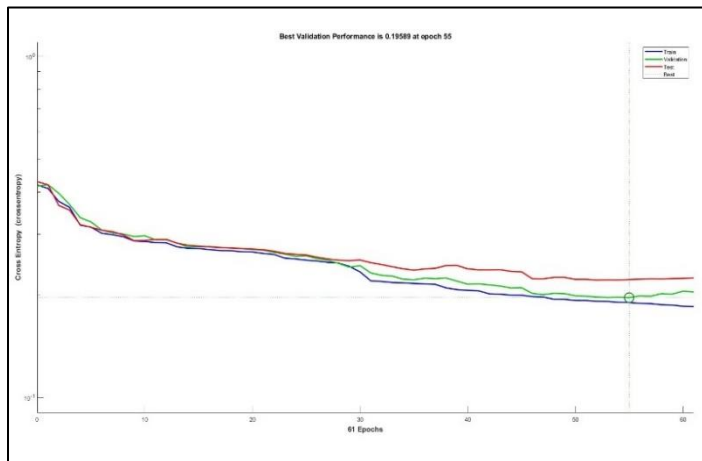


(b2)



(b3)

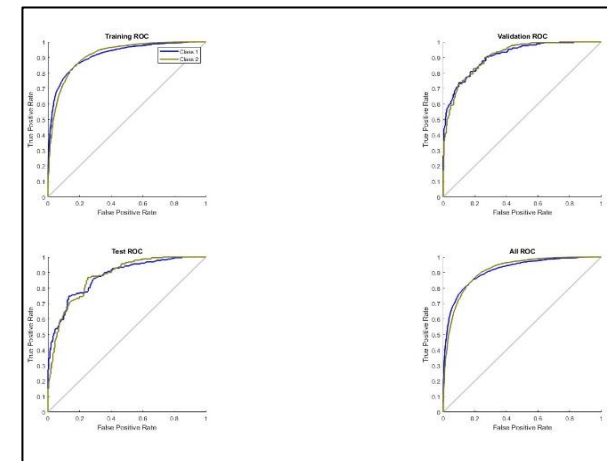
Gambar 4. 25 (a1) (b1) Performance Run 1 dan 2 Training OSS HN 5, (a2) (b2) Confusion Matrix Run 1 dan 2 Training OSS HN 5, (a3) (b3) ROC Run 1 dan 2 Training OSS HN 5



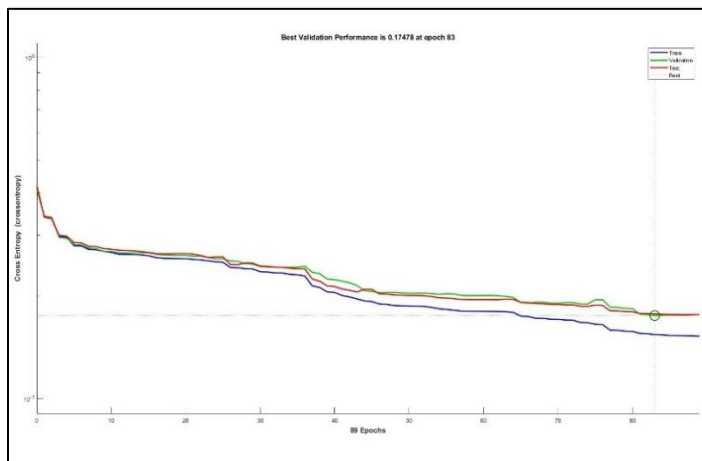
(c1)



(c2)



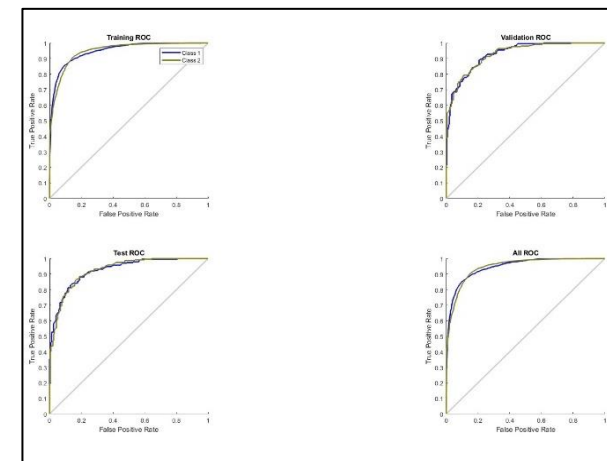
(c3)



(d1)

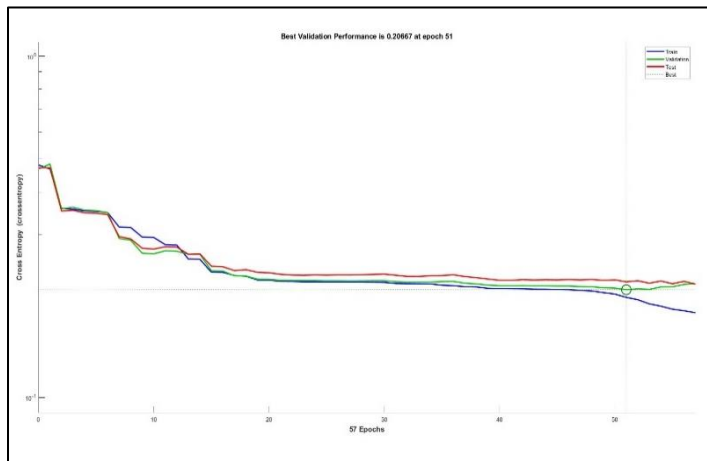


(d2)



(d3)

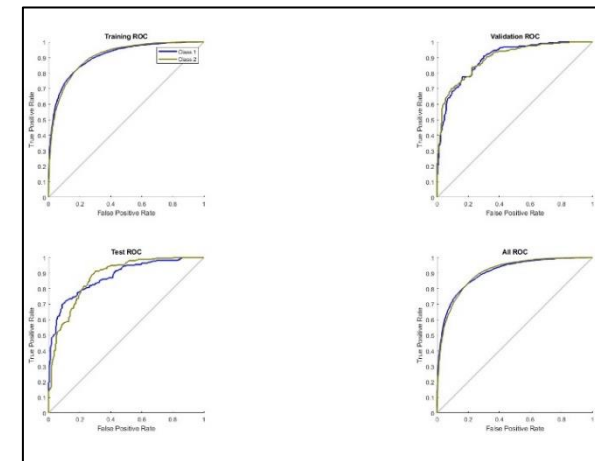
Gambar 4.25 (c1) (d1) Performance Run 3 dan 4 Training OSS HN 5, (c2) (d2) Confusion Matrix Run 3 dan 4 Training OSS HN 5, (c3) (d3) ROC Run 3 dan 4 Training OSS HN 5



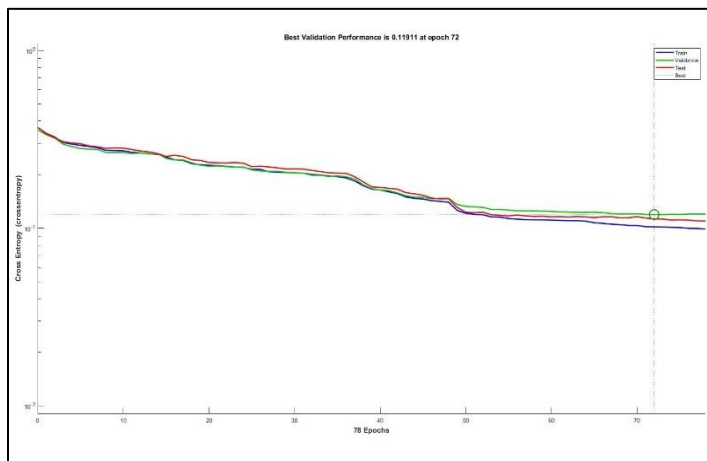
(e1)



(e2)



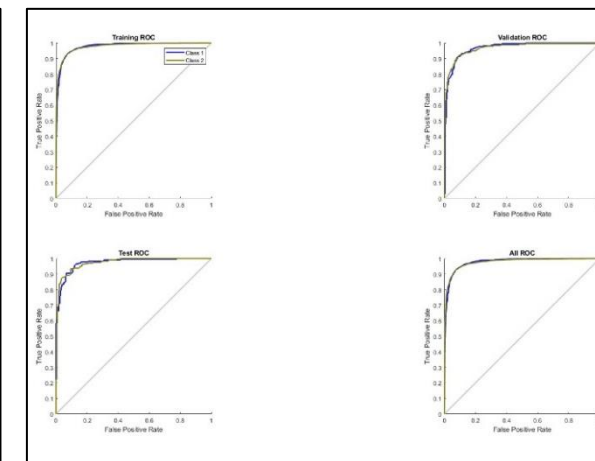
(e3)



(f1)

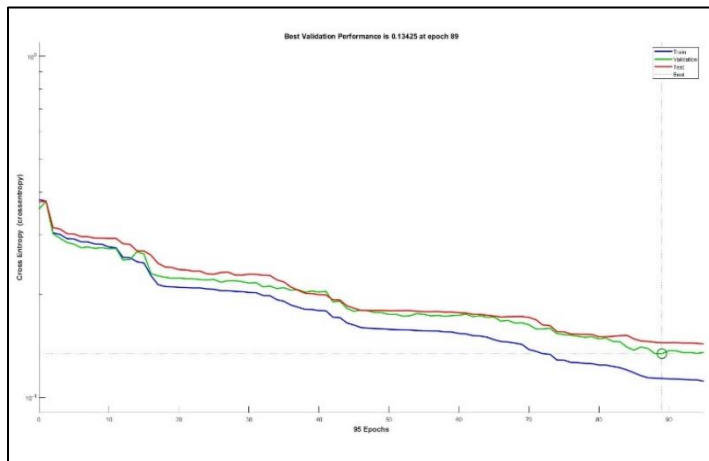


(f2)



(f3)

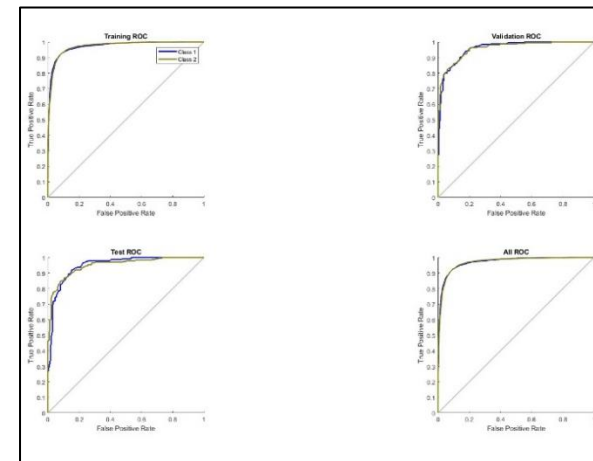
Gambar 4.25 (e1) (f1) Performance Run 5 dan 6 Training OSS HN 5, (e2) (f2) Confusion Matrix Run 5 dan 6 Training OSS HN 5, (e3) (f3) ROC Run 5 dan 6 Training OSS HN 5



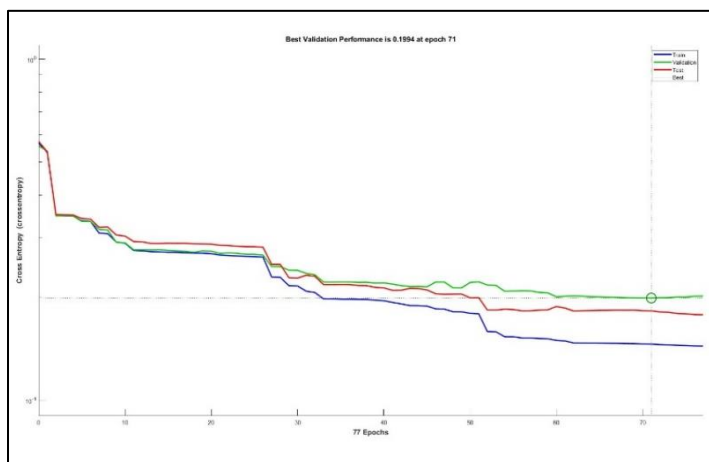
(g1)



(g2)



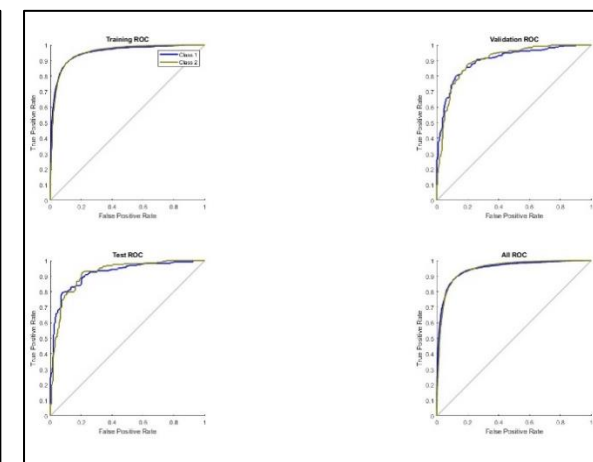
(g3)



(h1)

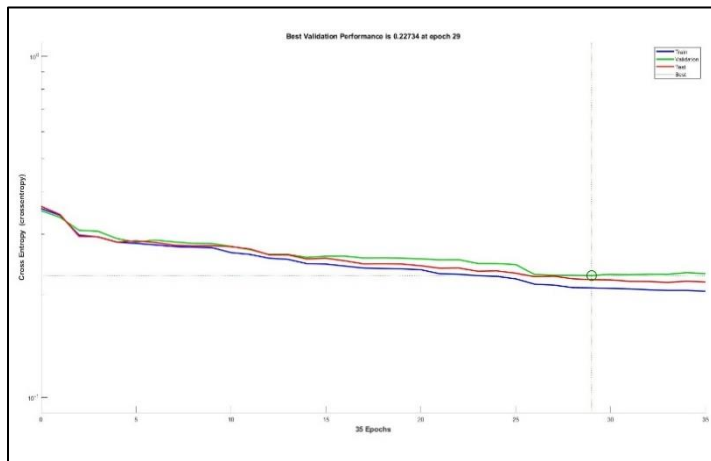


(h2)



(h3)

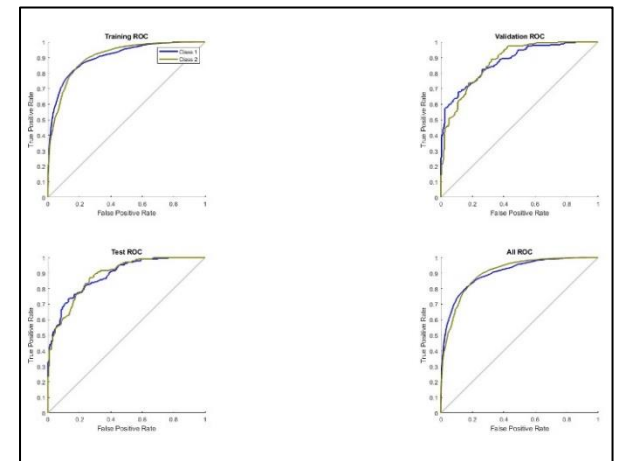
Gambar 4.25 (g1) (h1) Performance Run 7 dan 8 Training OSS HN 5, (g2) (h2) Confusion Matrix Run 7 dan 8 Training OSS HN 5, (g3) (h3) ROC Run 7 dan 8 Training OSS HN 5



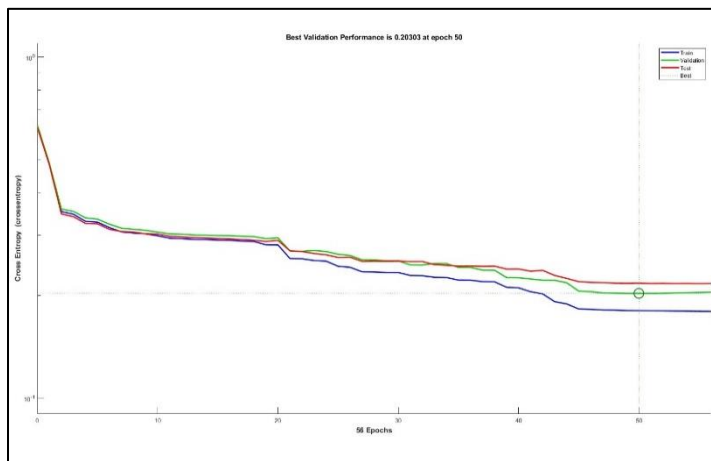
(i1)



(i2)



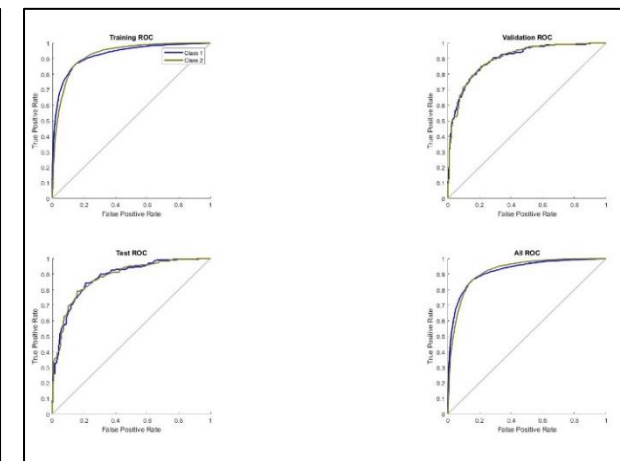
(i3)



(j1)



(j2)



(j3)

Gambar 4.25 (i1) (j1) Performance Run 9 dan 10 Training OSS HN 5, (i2) (j2) Confusion Matrix Run 9 dan 10 Training OSS HN 5, (i3) (j3) ROC Run 9 dan 10 Training OSS HN 5

Berdasarkan visualisasi hasil *training* di atas, dapat diketahui bahwa grafik *performance* terdiri dari 3 grafik, yaitu grafik biru merepresentasikan hasil *training*, grafik hijau merepresentasikan hasil validasi, dan grafik merah merepresentasikan hasil *testing*. Gambar 4.25 menampilkan sepuluh grafik *performane training* OSS *hidden neuron* 5 dimana grafik mengalami penurunan signifikan pada *epoch* 0-3. Penurunan *error* melandai dari *epoch* 10 hingga maksimal terjadi pada *run* 1, 2, dan 6 sedangkan *run* 9 grafik performa *error* tidak mengalami penurunan signifikan dari awal hingga *epoch* maksimalnya. Semakin rendah grafik menunjukkan bahwa nilai *error* semakin kecil.

Hasil *training* yang menunjukkan akurasi dari sistem ditunjukkan oleh *confusion matrix* dimana warna hijau merepresentasikan citra yang berhasil diklasifikasikan ke dalam kelas secara benar dan warna merah adalah citra yang gagal diklasifikasikan dalam kelasnya. Perbandingan data yang digunakan untuk *10-run cross validation* yaitu 90% dari total data (8.591) digunakan sebagai data *training*, 5% dari total data (477) digunakan sebagai data validasi, dan 5% lainnya (477) digunakan sebagai data *testing*. Dari *confusion matrix* tersebut, diketahui bahwa akurasi *training*, validasi, dan *testing* berada pada *range* 80-90%. Nilai *error* hasil *training*, validasi, dan *testing* $\leq 12\%$ (57). Hal ini menunjukkan bahwa model memiliki performa yang cukup baik.

Visualisasi grafik lainnya adalah grafik *Receiver Operating Characteristic* (ROC). Fungsi dari grafik ROC ini adalah untuk mengetahui hasil evaluasi performa *machine learning*. Pada grafik ROC, terdapat dua warna grafik yaitu biru yang menunjukkan klasifikasi citra kelas 1 dan grafik hijau menunjukkan klasifikasi citra kelas 2. Kurva validasi dan *testing* berada pada angka 0,8 mendekati angka 1 dan semakin menjauhi garis *threshold*. Namun, hasil validasi dan *testing run* 3, 5, dan 9 berada pada 0,7. Dengan demikian, dapat disimpulkan bahwa performa *machine learning* (*Multilayer Perceptron* model OSS *hidden neuron* 5) menunjukkan hasil performa cukup perfect.

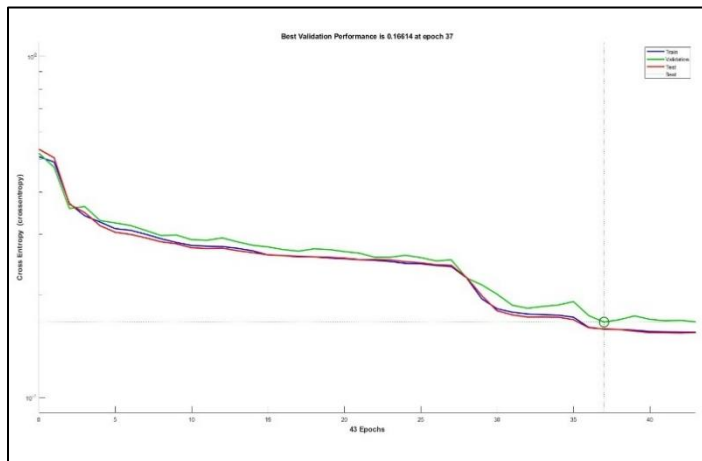
c. *Hidden Neuron 10*

Hasil akurasi *training* citra *body fluids* menggunakan model *One Step Secant Backpropagation (OSS) hidden neuron 10* dapat dilihat pada Tabel 4.28.

Tabel 4. 28 Hasil Akurasi *Training* Menggunakan Model OSS *Hidden Neuron 10*

<i>Run</i>	<i>Epoch</i>		MLP-TrainOSS Hidden Neuron 10			<i>Elapsed Time</i>
	<i>All</i>	<i>Best</i>	<i>Acc Training</i>	<i>Acc Validation</i>	<i>Acc Testing</i>	
1	43	37	88,3%	87,8%	89,3%	00:00:11
2	95	89	93,9%	91,0%	91,0%	00:00:13
3	128	122	94,1%	89,9%	90,1%	00:00:17
4	126	120	95,0%	90,4%	90,4%	00:00:17
5	137	131	96,1%	91,0%	92,7%	00:00:19
6	116	110	93,7%	89,1%	90,4%	00:00:17
7	72	66	83,7%	79,2%	81,3%	00:00:12
8	96	90	91,9%	89,3%	89,1%	00:00:15
9	45	39	82,3%	81,1%	79,2%	00:00:08
10	106	100	93,9%	89,1%	91,4%	00:00:17
Average			91,3%	87,8%	88,5%	00:00:15
Standar Deviasi			0,04853	0,0416505	0,044896	

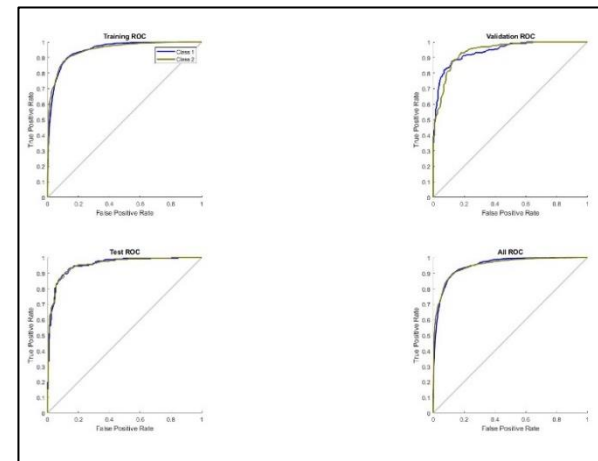
Berdasarkan Tabel 4.28 diketahui bahwa rata-rata akurasi *training* menggunakan model OSS dengan *hidden neuron 10* adalah 82-96%. Hasil akurasi *training* terbaik terdapat pada *run 5* dengan akurasi *training* sebesar 96,1%, akurasi validasi 91,0%, dan akurasi *testing* 92,7%. Akurasi *training* terbaik ini diperoleh pada *epoch* ke 131 dari total 137 *epoch* dengan waktu *training* selama 19 detik. Sementara itu, hasil akurasi *training* terendah terdapat pada *run 9* dengan akurasi *training* 82,3%, akurasi validasi 81,1%, dan akurasi *testing* 79,2%. Dari pengujian diatas, diperoleh hasil bahwa rata-rata akurasi *training* adalah 91,3%, akurasi validasi 87,8%, dan akurasi *testing* 88,5%. Rata-rata waktu *training* menggunakan model OSS *hidden neuron 5* adalah 15 detik. Visualisasi hasil *training* dapat dilihat pada Gambar 4.26.



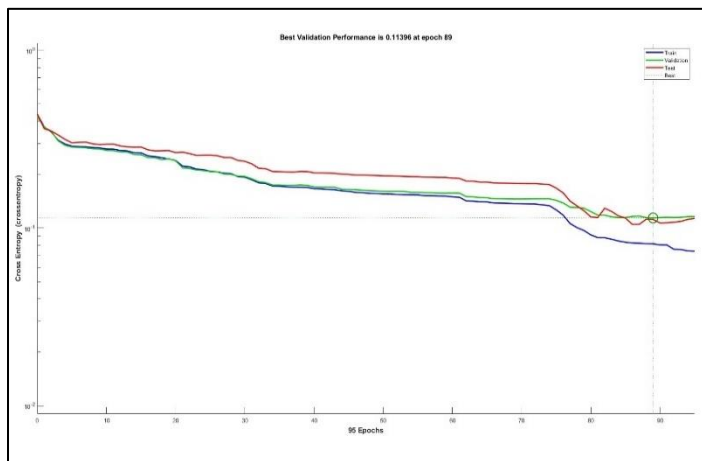
(a1)



(a2)



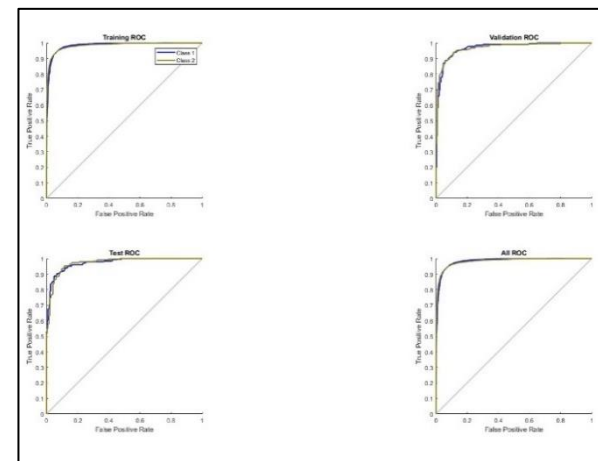
(a3)



(b1)

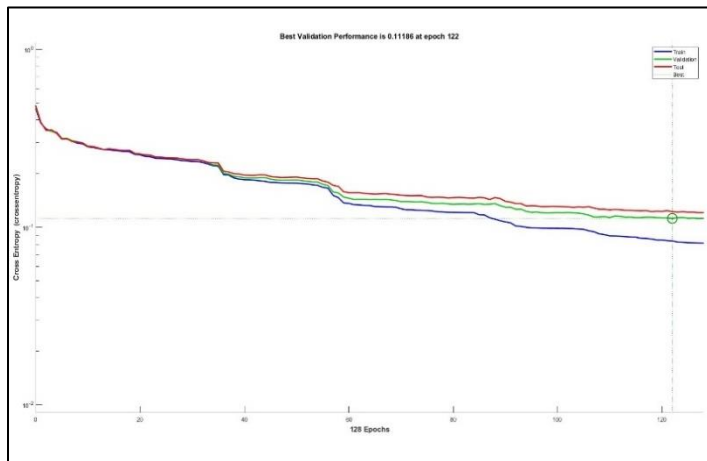


(b2)



(b3)

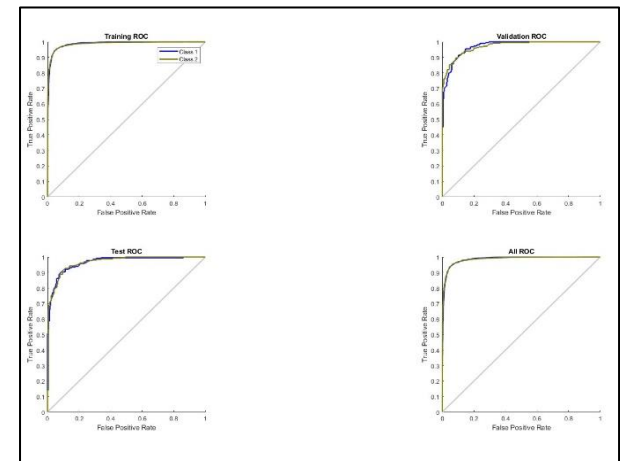
Gambar 4. 26 (a1) (b1) Performance Run 1 dan 2 Training OSS HN 10, (a2) (b2) Confusion Matrix Run 1 dan 2 Training OSS HN 10, (a3) (b3) ROC Run 1 dan 2 Training OSS HN 10



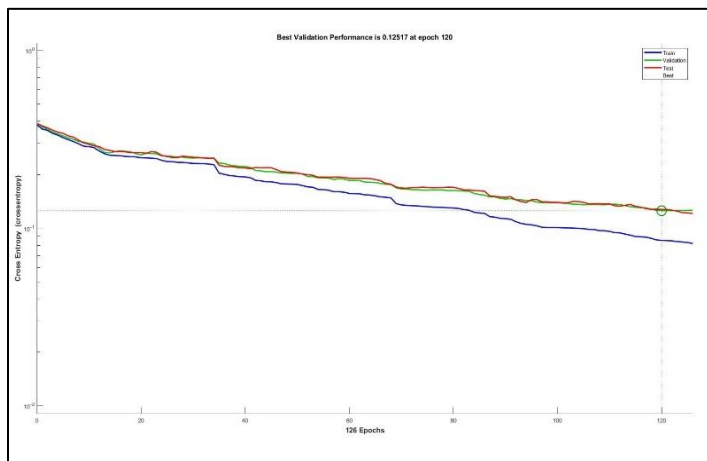
(c1)



(c2)



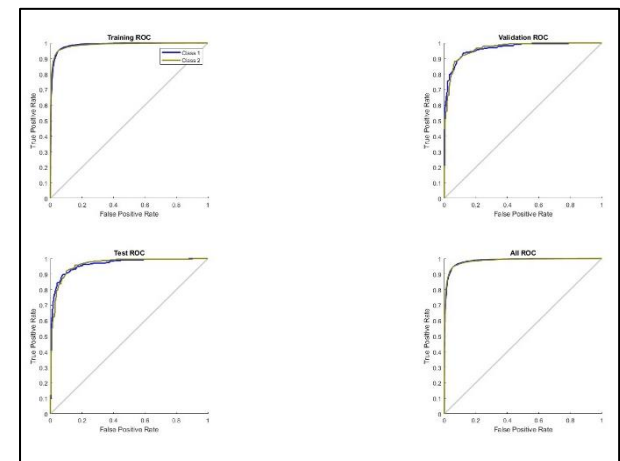
(c3)



(d1)

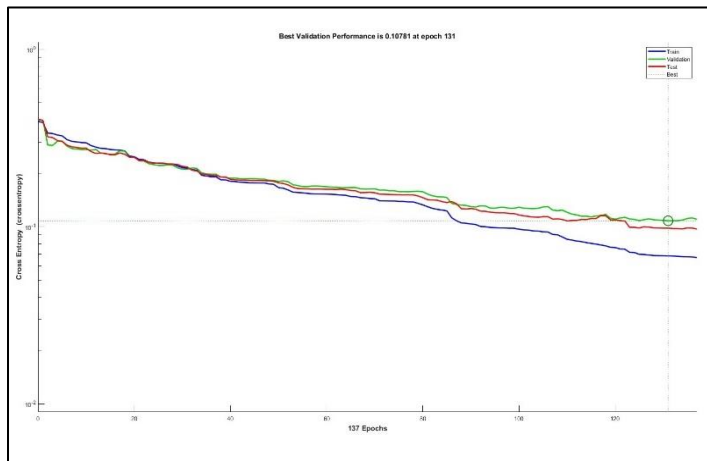


(d2)



(d3)

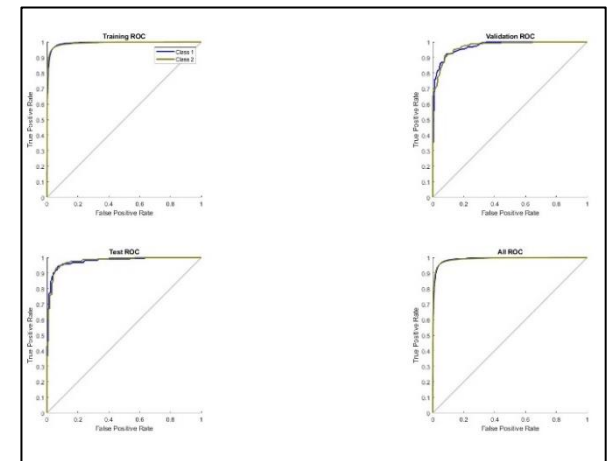
Gambar 4.26 (c1) (d1) Performance Run 3 dan 4 Training OSS HN 10, (c2) (d2) Confusion Matrix Run 3 dan 4 Training OSS HN 10, (c3) (d3) ROC Run 3 dan 4 Training OSS HN 10



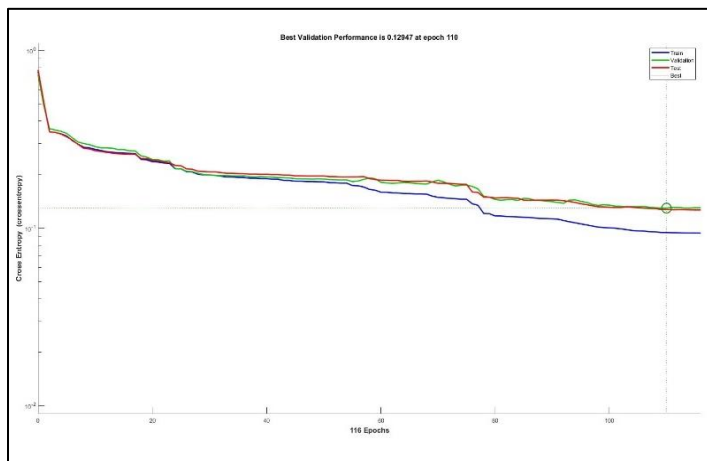
(e1)



(e2)



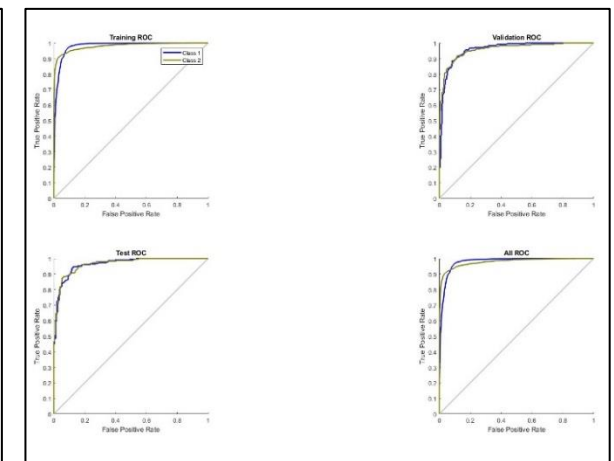
(e3)



(f1)

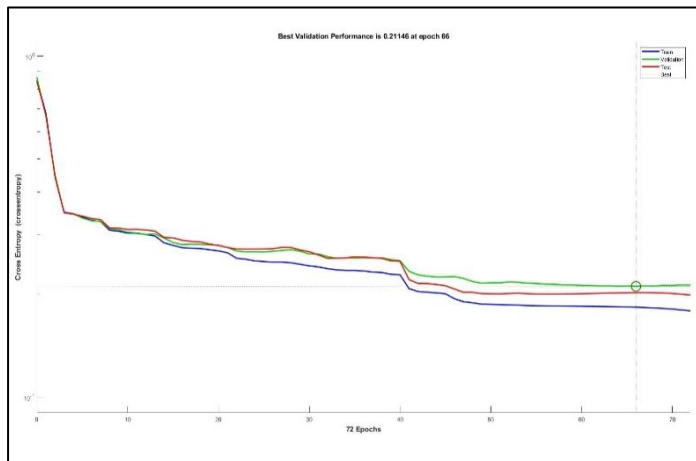


(f2)



(f3)

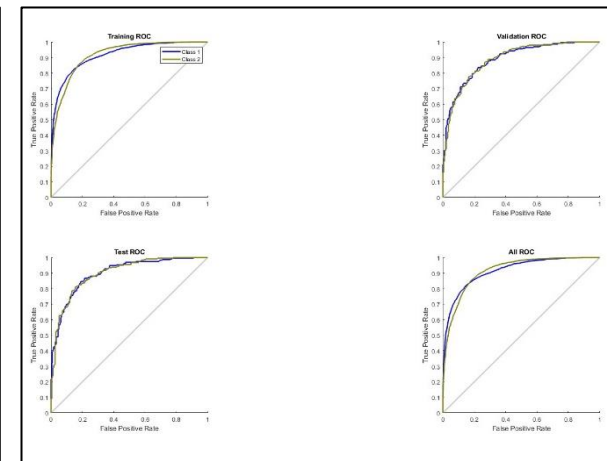
Gambar 4.26 (e1) (f1) Performance Run 5 dan 6 Training OSS HN 10, (e2) (f2) Confusion Matrix Run 5 dan 6 Training OSS HN 10, (e3) (f3) ROC Run 5 dan 6 Training OSS HN 10



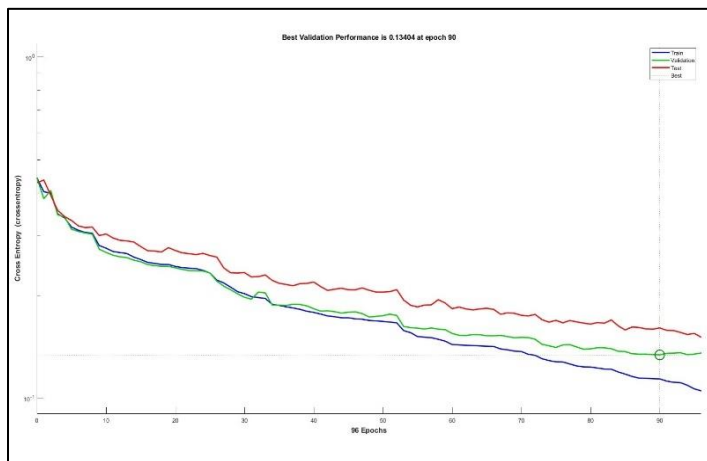
(g1)



(g2)



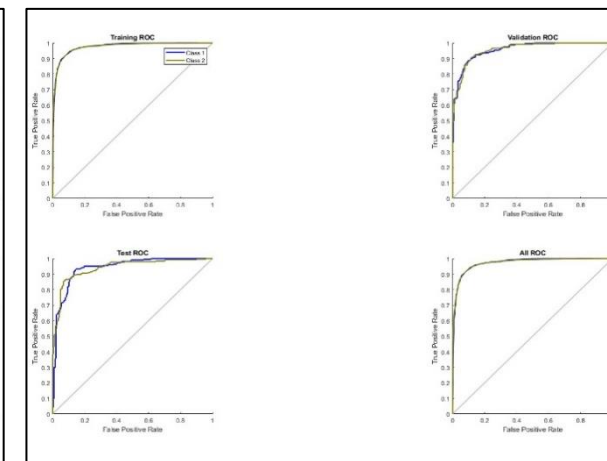
(g3)



(h1)

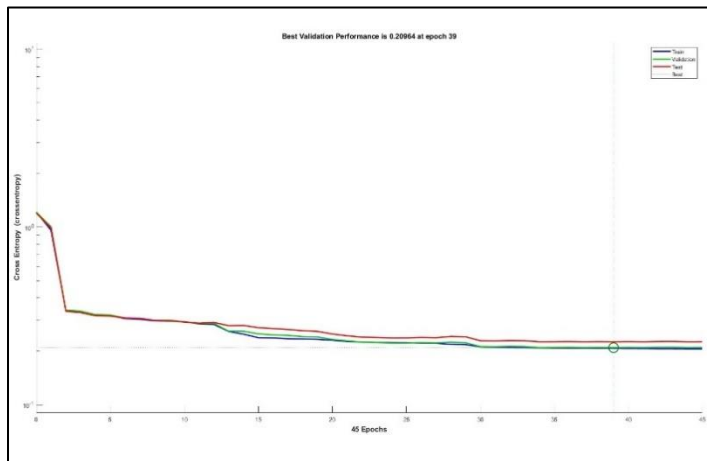


(h2)



(h3)

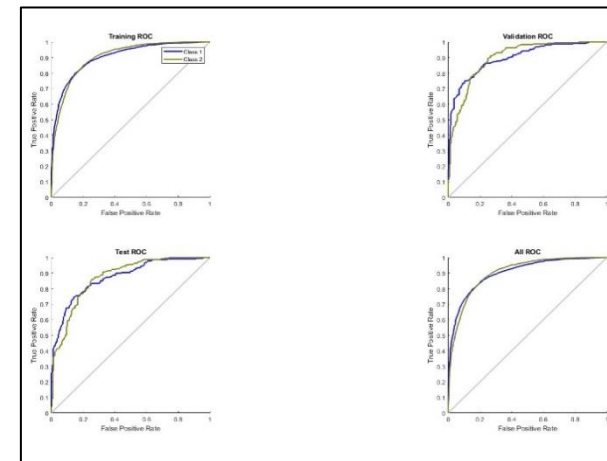
Gambar 4.26 (g1) (h1) Performance Run 7 dan 8 Training OSS HN 10, (g2) (h2) Confusion Matrix Run 7 dan 8 Training OSS HN 10, (g3) (h3) ROC Run 7 dan 8 Training OSS HN 10



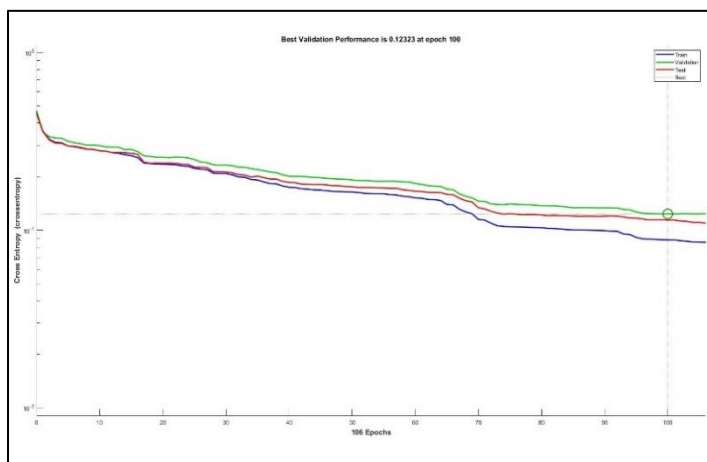
(i1)



(i2)



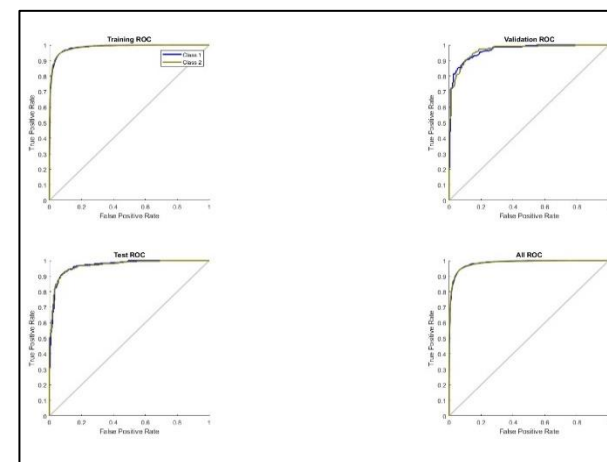
(i3)



(j1)



(j2)



(j3)

Gambar 4.26 (i1) (j1) Performance Run 9 dan 10 Training OSS HN 5, (i2) (j2) Confusion Matrix Run 9 dan 10 Training OSS HN 10, (i3) (j3) ROC Run 9 dan 10 Training OSS HN 10

Berdasarkan visualisasi hasil *training* di atas, dapat diketahui bahwa grafik *performance* terdiri dari 3 grafik, yaitu grafik biru merepresentasikan hasil *training*, grafik hijau merepresentasikan hasil validasi, dan grafik merah merepresentasikan hasil *testing*. Gambar 4.26 menampilkan sepuluh grafik *performane training* OSS *hidden neuron* 10 dimana grafik mengalami penurunan *error* signifikan pada *range epoch* 0-3. Penurunan performa *error* melandai terjadi pada *epoch* 3 hingga *epoch* maksimalnya terlihat pada semua hasil *run*. Semakin rendah grafik menunjukkan bahwa nilai *error* semakin kecil.

Hasil *training* yang menunjukkan akurasi dari sistem ditunjukkan oleh *confusion matrix* dimana warna hijau merepresentasikan citra yang berhasil diklasifikasikan ke dalam kelas secara benar dan warna merah adalah citra yang gagal diklasifikasikan dalam kelasnya. Perbandingan data yang digunakan untuk *10-run cross validation* yaitu 90% dari total data (8.591) digunakan sebagai data *training*, 5% dari total data (477) digunakan sebagai data validasi, dan 5% lainnya (477) digunakan sebagai data *testing*. Dari *confusion matrix* tersebut, diketahui bahwa rata-rata akurasi *training*, validasi, dan *testing* di atas 90%. Nilai rata-rata *error training* adalah $\leq 10\%$ (859) dan rata-rata nilai *error testing* adalah $\leq 10\%$ (47). Hal ini menunjukkan bahwa model memiliki performa yang baik.

Visualisasi grafik lainnya adalah grafik *Receiver Operating Characteristic* (ROC). Fungsi dari grafik ROC ini adalah untuk mengetahui hasil evaluasi performa *machine learning*. Pada grafik ROC, terdapat dua warna grafik yaitu biru yang menunjukkan klasifikasi citra kelas 1 dan grafik hijau menunjukkan klasifikasi citra kelas 2. Kurva validasi dan *testing* mendekati angka 1 dan semakin menjauhi garis *threshold* kecuali pada *run* 7 dan 9. Hasil validasi dan *testing* *run* 7 dan 9, kurva berada pada angka 0,7 sehingga jauh dari angka 1. Dengan demikian, dapat disimpulkan bahwa performa *machine learning* (*Multilayer Perceptron* model OSS *hidden neuron* 10) menunjukkan performa *perfect*.

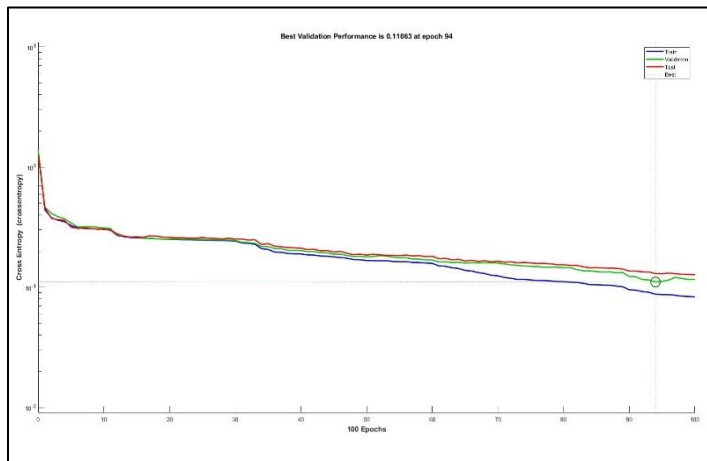
d. *Hidden Neuron 15*

Hasil akurasi *training* citra *body fluids* menggunakan model *One Step Secant Backpropagation* (OSS) *hidden neuron 15* dapat dilihat pada Tabel 4.29.

Tabel 4. 29 Hasil Akurasi *Training* Menggunakan Model OSS *Hidden Neuron 15*

<i>Run</i>	<i>Epoch</i>		MLP-TrainOSS Hidden Neuron 15			<i>Elapsed Time</i>
	<i>All</i>	<i>Best</i>	<i>Acc Training</i>	<i>Acc Validation</i>	<i>Acc Testing</i>	
1	100	94	93,7%	92,5%	90,4%	00:00:17
2	101	95	94,0%	89,5%	91,2%	00:00:17
3	88	82	92,6%	91,4%	89,3%	00:00:15
4	70	64	88,4%	84,9%	87,8%	00:00:12
5	26	19	75,9%	76,7%	73,8%	00:00:05
6	82	76	93,7%	90,1%	87,0%	00:00:14
7	109	103	95,1%	90,1%	89,9%	00:00:18
8	128	122	96,3%	94,3%	90,1%	00:00:21
9	90	84	89,6%	85,3%	86,6%	00:00:15
10	125	119	95,3%	91,8%	91,2%	00:00:21
Average			91,5%	88,7%	87,7%	00:00:16
Standar Deviasi			0,06001	0,0513424	0,051625	

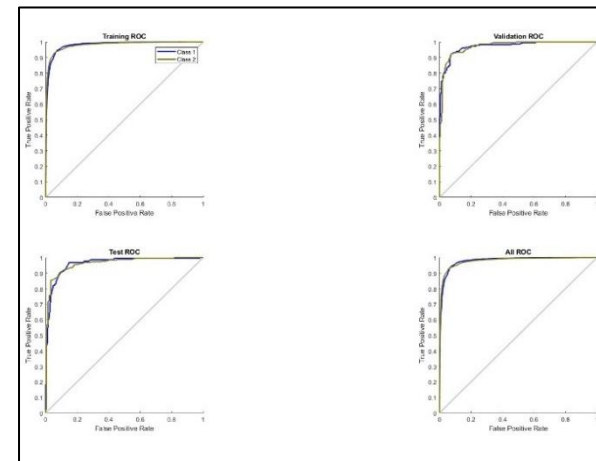
Berdasarkan Tabel 4.29 diketahui bahwa rata-rata akurasi *training* menggunakan model OSS dengan *hidden neuron 15* adalah 75-96%. Hasil akurasi *training* terbaik terdapat pada *run 8* dengan akurasi *training* sebesar 96,3%, akurasi validasi 94,3%, dan akurasi *testing* 90,1%. Akurasi *training* terbaik ini diperoleh pada *epoch* ke 122 dari total 128 *epoch* dengan waktu *training* selama 21 detik. Sementara itu, hasil akurasi *training* terendah terdapat pada *run 5* dengan akurasi *training* 75,9%, akurasi validasi 76,7%, dan akurasi *testing* 73,8%. Dari pengujian diatas, diperoleh hasil bahwa rata-rata akurasi *training* adalah 91,5%, akurasi validasi 88,7%, dan akurasi *testing* 87,7%. Rata-rata waktu *training* menggunakan model OSS *hidden neuron 15* adalah 16 detik. Visualisasi hasil *training* dapat dilihat pada Gambar 4.27.



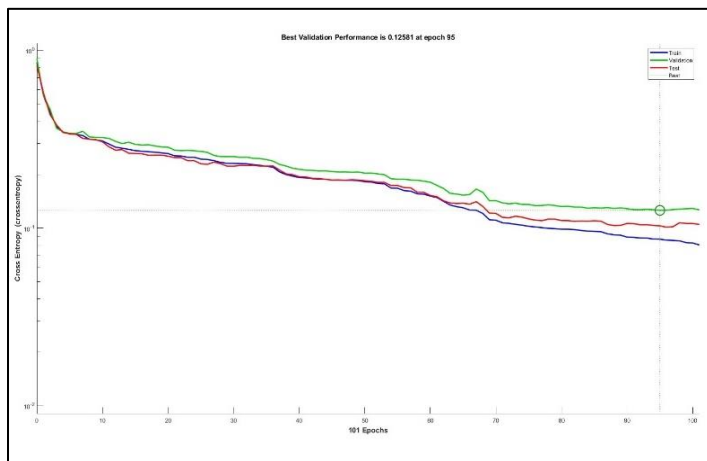
(a1)



(a2)



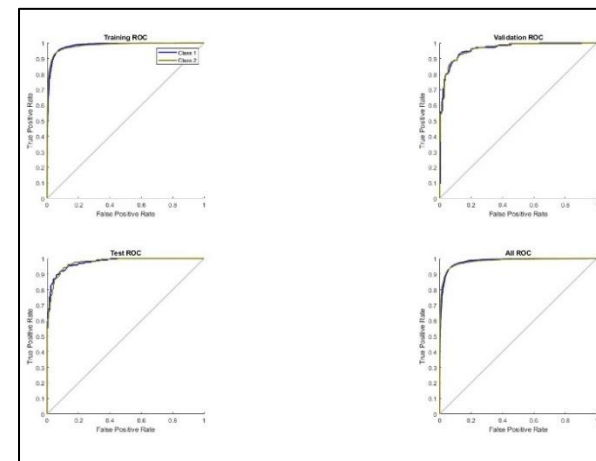
(a3)



(b1)

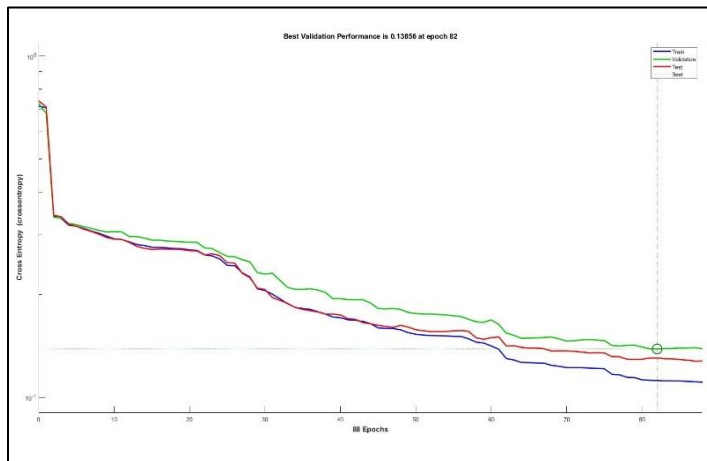


(b2)



(b3)

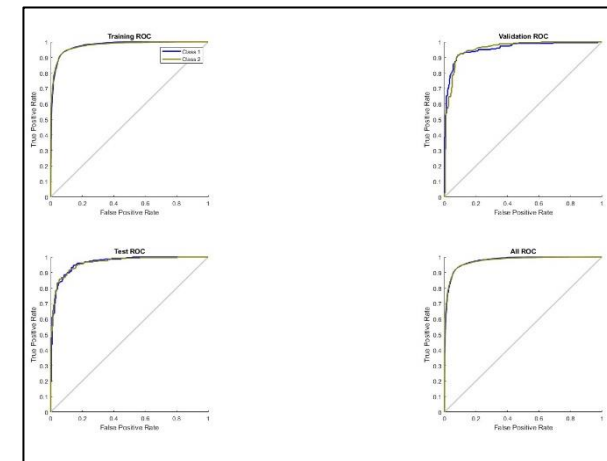
Gambar 4. 27 (a1) (b1) Performance Run 1 dan 2 Training OSS HN 15, (a2) (b2) Confusion Matrix Run 1 dan 2 Training OSS HN 15, (a3) (b3) ROC Run 1 dan 2 Training OSS HN 15



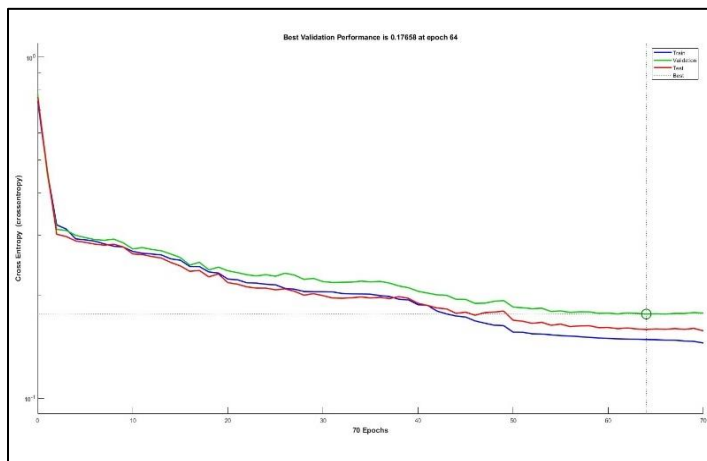
(c1)



(c2)



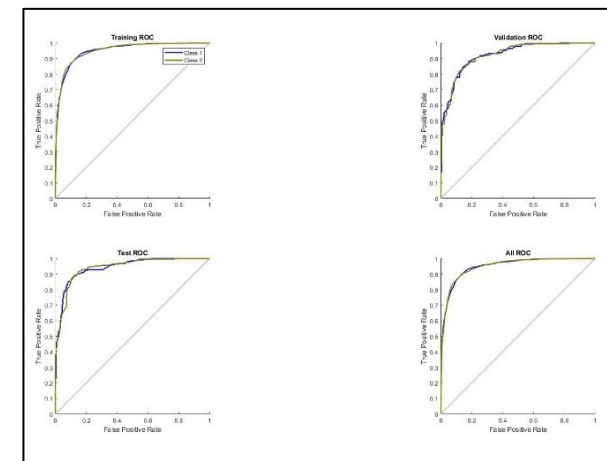
(c3)



(d1)

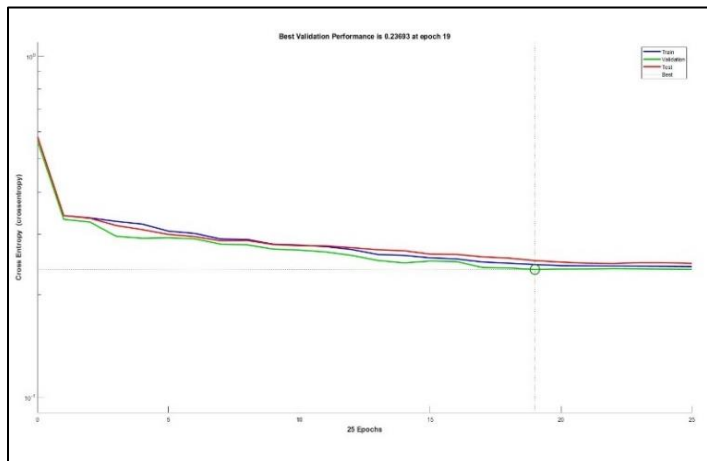


(d2)



(d3)

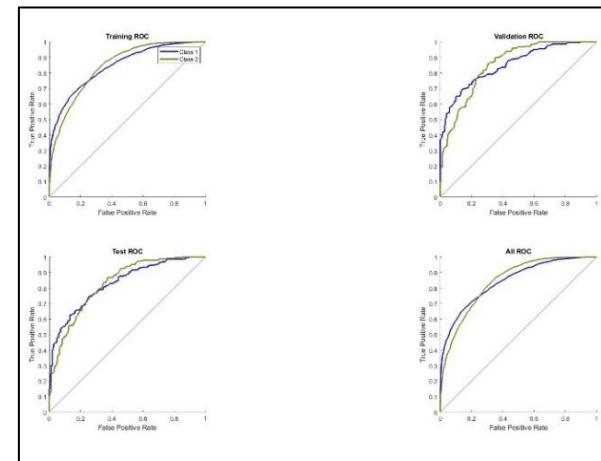
Gambar 4.27 (c1) (d1) Performance Run 3 dan 4 Training OSS HN 15, (c2) (d2) Confusion Matrix Run 3 dan 4 Training OSS HN 15, (c3) (d3) ROC Run 3 dan 4 Training OSS HN 15



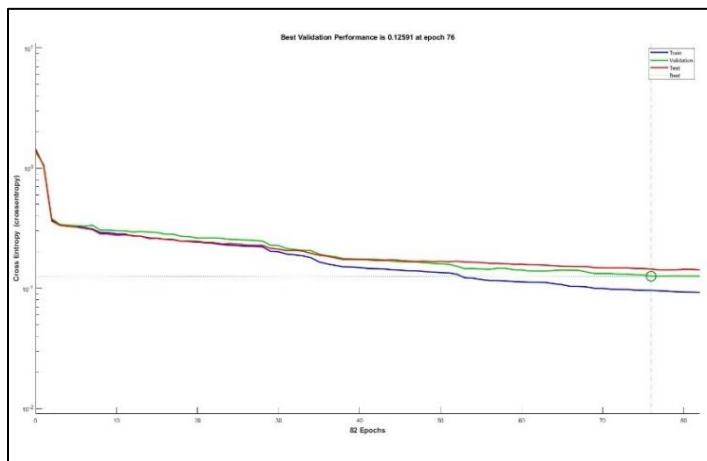
(e1)



(e2)



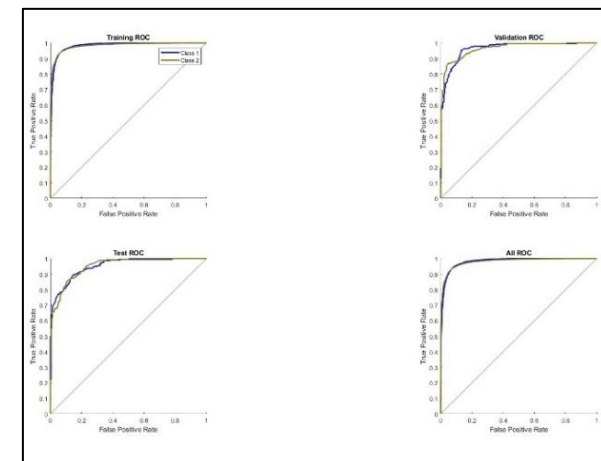
(e3)



(f1)

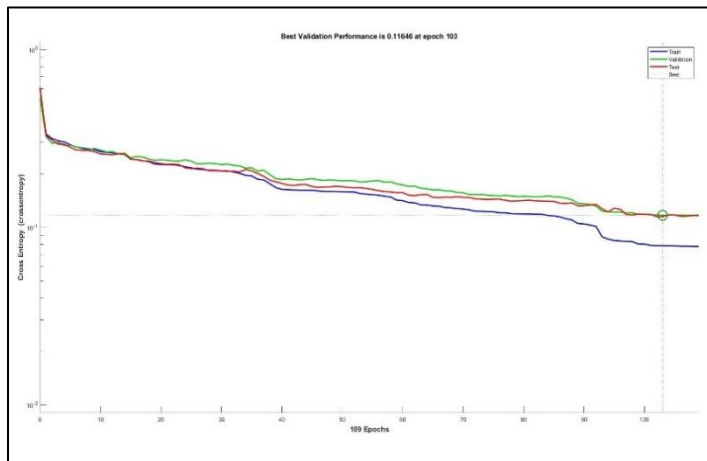


(f2)



(f3)

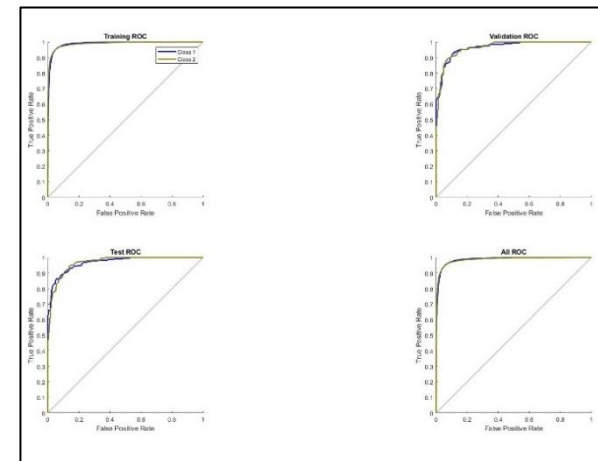
Gambar 4.27 (e1) (f1) Performance Run 5 dan 6 Training OSS HN 15, (e2) (f2) Confusion Matrix Run 5 dan 6 Training OSS HN 15, (e3) (f3) ROC Run 5 dan 6 Training OSS HN 15



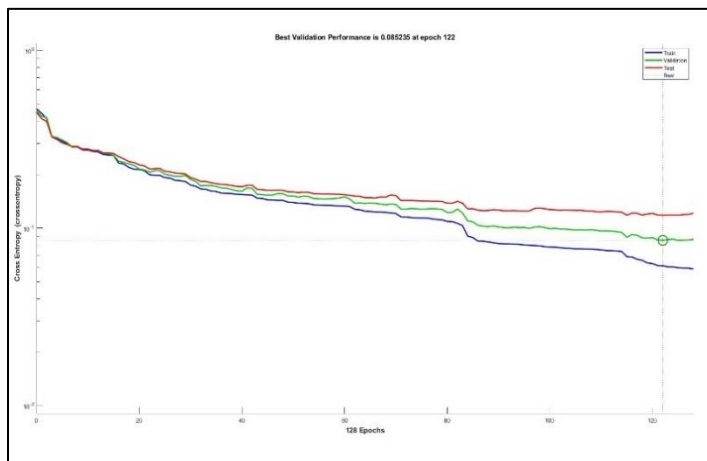
(g1)



(g2)



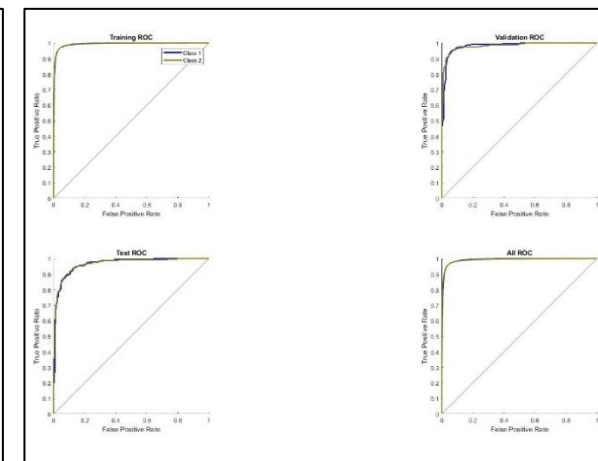
(g3)



(h1)

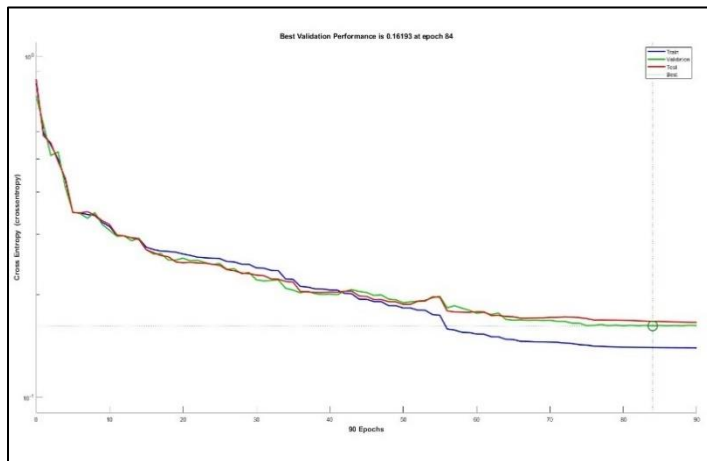


(h2)



(h3)

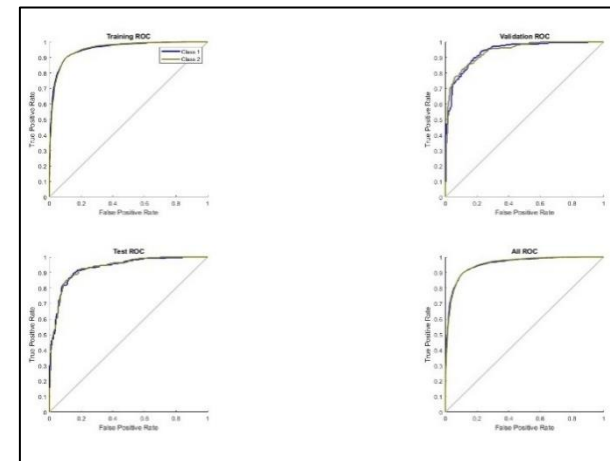
Gambar 4.27 (g1) (h1) Performance Run 7 dan 8 Training OSS HN 15, (g2) (h2) Confusion Matrix Run 7 dan 8 Training OSS HN 15, (g3) (h3) ROC Run 7 dan 8 Training OSS HN 15



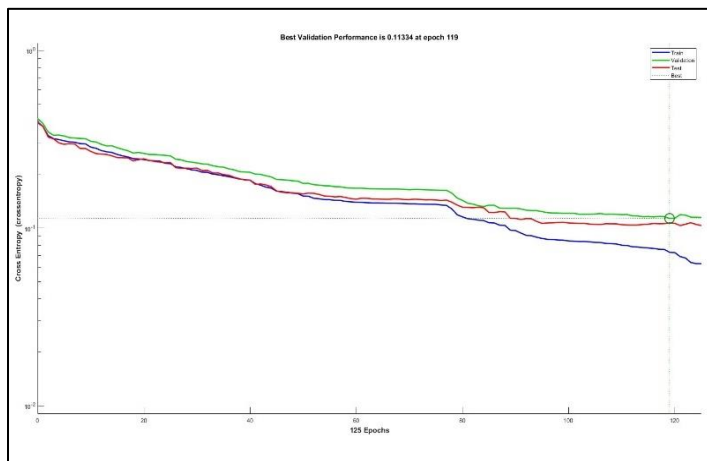
(i1)



(i2)



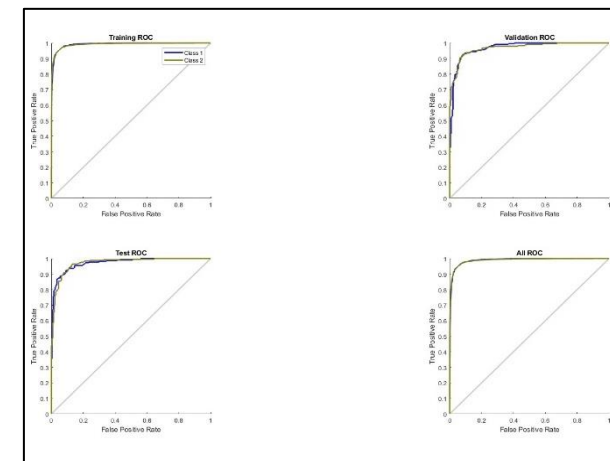
(i3)



(j1)



(j2)



(j3)

Gambar 4.27 (i1) (j1) Performance Run 9 dan 10 Training OSS HN 15, (i2) (j2) Confusion Matrix Run 9 dan 10 Training OSS HN 15, (i3) (j3) ROC Run 9 dan 10 Training OSS HN 15

Berdasarkan visualisasi hasil training di atas, dapat diketahui bahwa grafik *performance* terdiri dari 3 grafik, yaitu grafik biru merepresentasikan hasil *training*, grafik hijau merepresentasikan hasil validasi, dan grafik merah merepresentasikan hasil *testing*. Gambar 4.27 menampilkan sepuluh grafik *performane training* OSS *hidden neuron* 15 dimana grafik mengalami penurunan *error* pada *range epoch* 0-5 kecuali *run* 5 dan 10. Penurunan *error* melandai terjadi mulai *epoch* 5 hingga maksimalnya. *Run* 5 mengalami penurunan *error* signifikan pada *range epoch* 0-2 kemudian penurunan melandai hingga *epoch* maksimalnya. Sementara itu, *run* 10 tidak mengalami penurunan *error* signifikan dari awal hingga *epoch* maksimalnya.

Hasil *training* yang menunjukkan akurasi dari sistem ditunjukkan oleh *confusion matrix* dimana warna hijau merepresentasikan citra yang berhasil diklasifikasikan ke dalam kelas secara benar dan warna merah adalah citra yang gagal diklasifikasikan dalam kelasnya. Perbandingan data yang digunakan untuk *10-run cross validation* yaitu 90% dari total data (8.591) digunakan sebagai data *training*, 5% dari total data (477) digunakan sebagai data validasi, dan 5% lainnya (477) digunakan sebagai data *testing*. Dari *confusion matrix* tersebut, diketahui bahwa rata-rata akurasi *training*, validasi, dan *testing* di atas 90%. Nilai rata-rata *error training* adalah $\leq 10\%$ (859) dan rata-rata nilai *error testing* adalah $\leq 10\%$ (47) kecuali *run* 5. Meskipun demikian, model tetap dikatakan memiliki performa yang baik.

Visualisasi grafik lainnya adalah grafik *Receiver Operating Characteristic* (ROC). Fungsi dari grafik ROC ini adalah untuk mengetahui hasil evaluasi performa *machine learning*. Pada grafik ROC, terdapat dua warna grafik yaitu biru yang menunjukkan klasifikasi citra kelas 1 dan grafik hijau menunjukkan klasifikasi citra kelas 2. Kurva validasi dan *testing* berada pada angka 0,8 dan menjauhi garis *threshold* kecuali pada *run* 5 yang berada pada angka 0,7. Dengan demikian, dapat disimpulkan bahwa performa *machine learning* (*Multilayer Perceptron* model OSS *hidden neuron* 15) menunjukkan performa *perfect*.

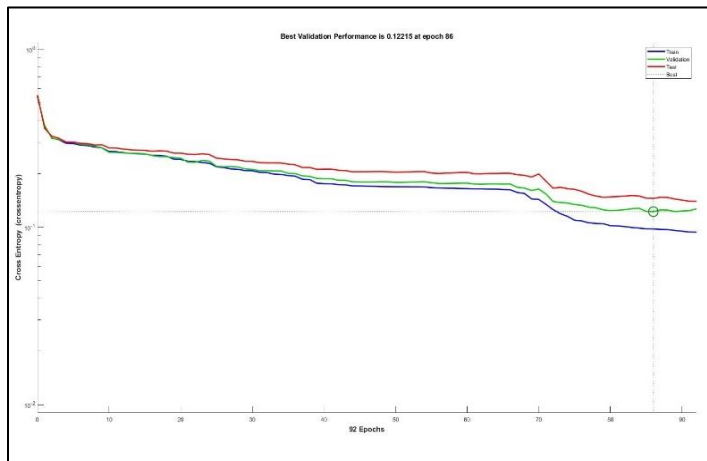
e. *Hidden Neuron 20*

Hasil akurasi *training* citra *body fluids* menggunakan model *One Step Secant Backpropagation* (OSS) *hidden neuron 20* dapat dilihat pada Tabel 4.30.

Tabel 4. 30 Hasil Akurasi *Training* Menggunakan Model OSS *Hidden Neuron 20*

<i>Run</i>	<i>Epoch</i>		MLP-TrainOSS Hidden Neuron 20			<i>Elapsed Time</i>
	<i>All</i>	<i>Best</i>	<i>Acc Training</i>	<i>Acc Validation</i>	<i>Acc Testing</i>	
1	92	86	92,2%	90,6%	86,0%	00:00:17
2	81	75	94,2%	92,2%	90,1%	00:00:15
3	112	106	94,4%	90,6%	90,8%	00:00:20
4	54	48	87,1%	83,6%	85,5%	00:00:11
5	100	94	93,9%	91,6%	89,1%	00:00:18
6	137	131	96,9%	93,1%	92,2%	00:00:24
7	93	87	93,3%	88,5%	88,9%	00:00:17
8	46	40	84,8%	82,8%	83,0%	00:00:09
9	39	33	79,5%	82,6%	79,5%	00:00:08
10	107	101	92,2%	88,5%	90,4%	00:00:19
Average			90,9%	88,4%	87,6%	00:00:16
Standar Deviasi			0,05363	0,0400706	0,039881	

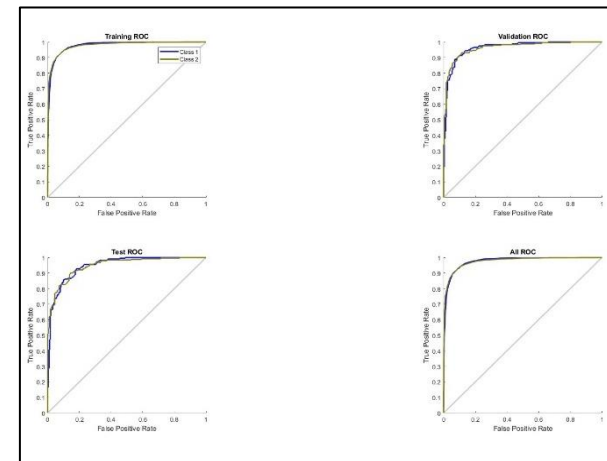
Berdasarkan Tabel 4.30 diketahui bahwa rata-rata akurasi *training* menggunakan model OSS dengan *hidden neuron 20* adalah 79-96%. Hasil akurasi *training* terbaik terdapat pada *run 6* dengan akurasi *training* sebesar 96,9%, akurasi validasi 93,1%, dan akurasi *testing* 92,2%. Akurasi *training* terbaik ini diperoleh pada *epoch* ke 131 dari total 137 *epoch* dengan waktu *training* selama 24 detik. Sementara itu, hasil akurasi *training* terendah terdapat pada *run 9* dengan akurasi *training* 79,5%, akurasi validasi 82,6%, dan akurasi *testing* 79,5%. Dari pengujian diatas, diperoleh hasil bahwa rata-rata akurasi *training* adalah 90,9%, akurasi validasi 88,4%, dan akurasi *testing* 87,6%. Rata-rata waktu *training* menggunakan model OSS *hidden neuron 20* adalah 16 detik. Visualisasi hasil *training* dapat dilihat pada Gambar 4.28.



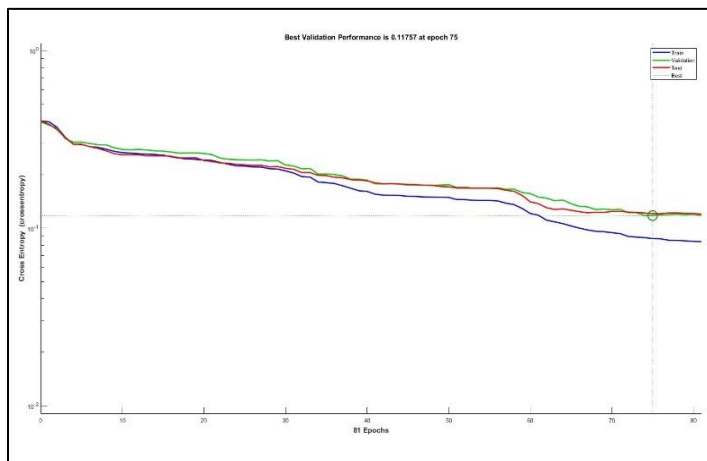
(a1)



(a2)



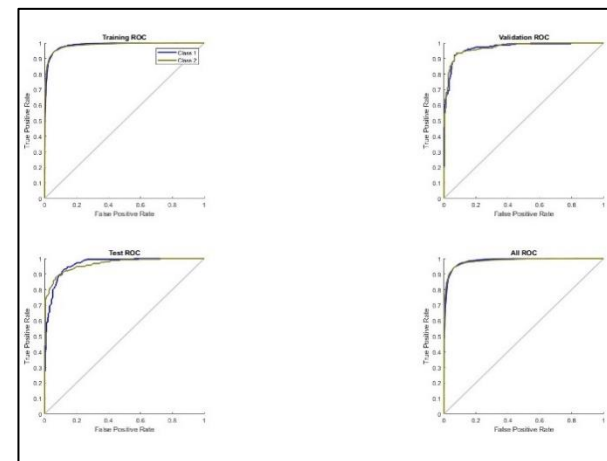
(a3)



(b1)

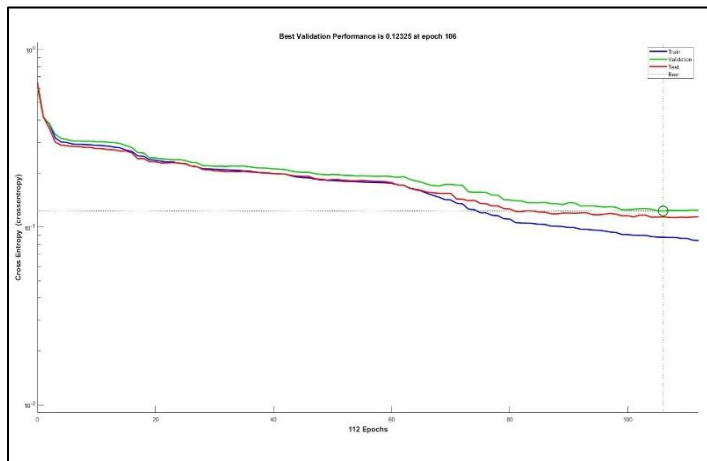


(b2)



(b3)

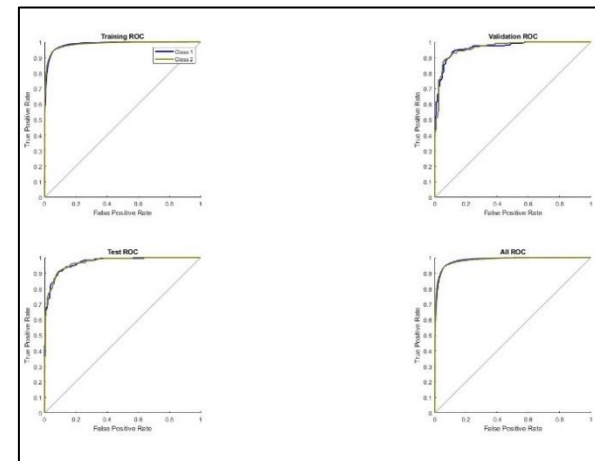
Gambar 4. 28 (a1) (b1) Performance Run 1 dan 2 Training OSS HN 20, (a2) (b2) Confusion Matrix Run 1 dan 2 Training OSS HN 20, (a3) (b3) ROC Run 1 dan 2 Training OSS HN 20



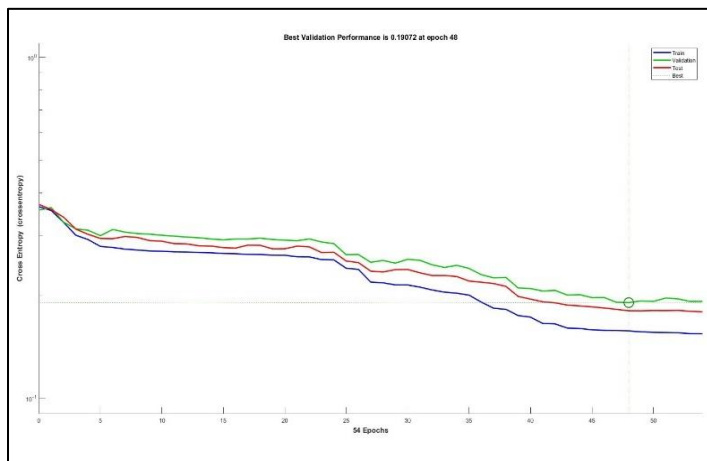
(c1)



(c2)



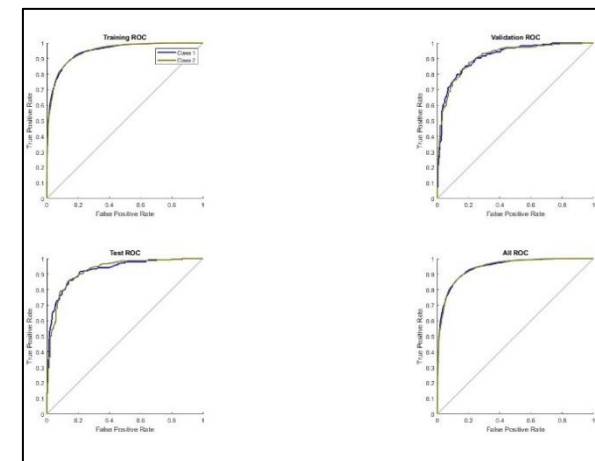
(c3)



(d1)

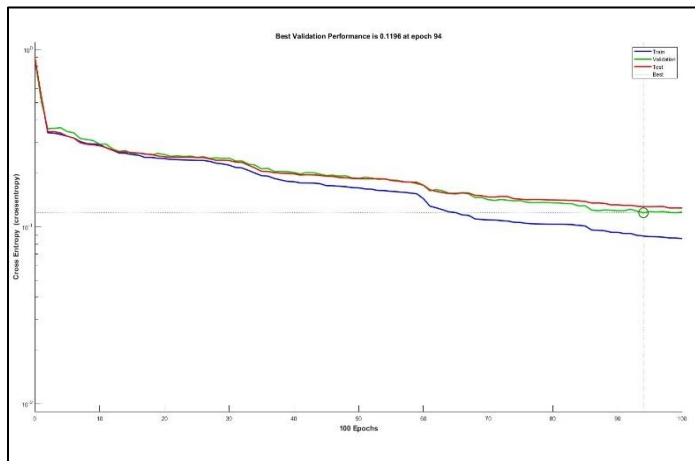


(d2)



(d3)

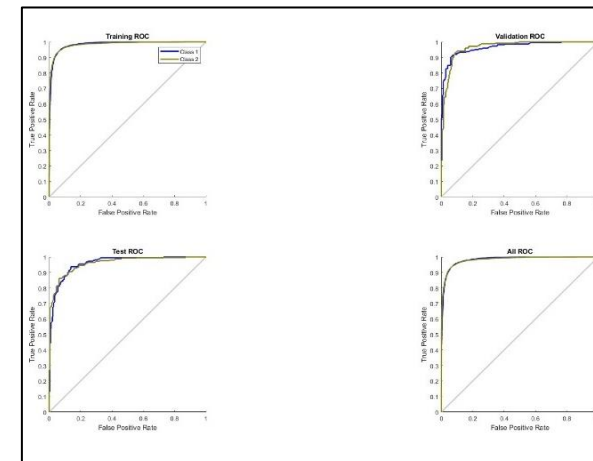
Gambar 4.28 (c1) (d1) Performance Run 3 dan 4 Training OSS HN 20, (c2) (d2) Confusion Matrix Run 3 dan 4 Training OSS HN 20, (c3) (d3) ROC Run 3 dan 4 Training OSS HN 20



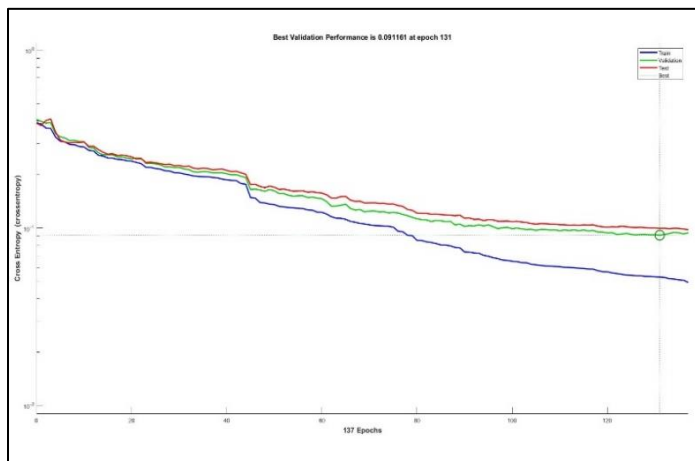
(e1)



(e2)



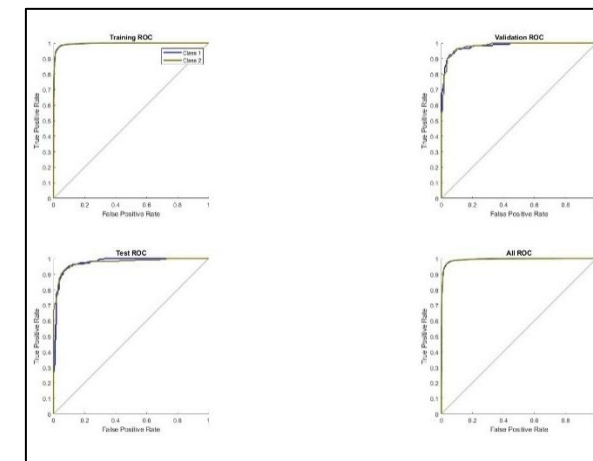
(e3)



(f1)

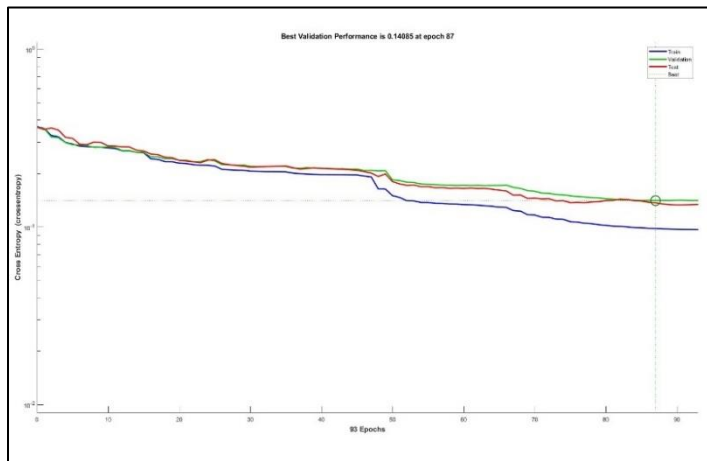


(f2)



(f3)

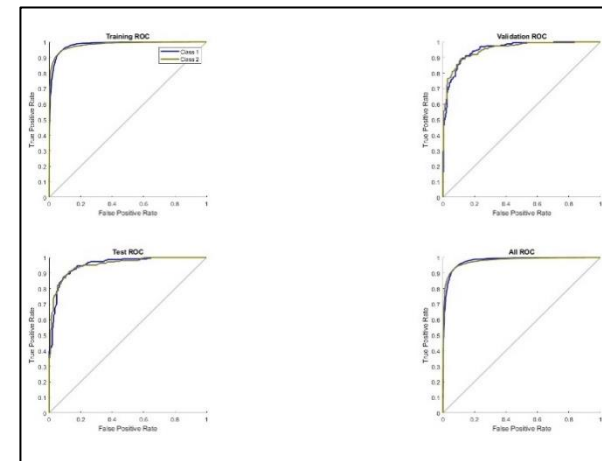
Gambar 4.28 (e1) (f1) Performance Run 5 dan 6 Training OSS HN 20, (e2) (f2) Confusion Matrix Run 5 dan 6 Training OSS HN 20, (e3) (f3) ROC Run 5 dan 6 Training OSS HN 20



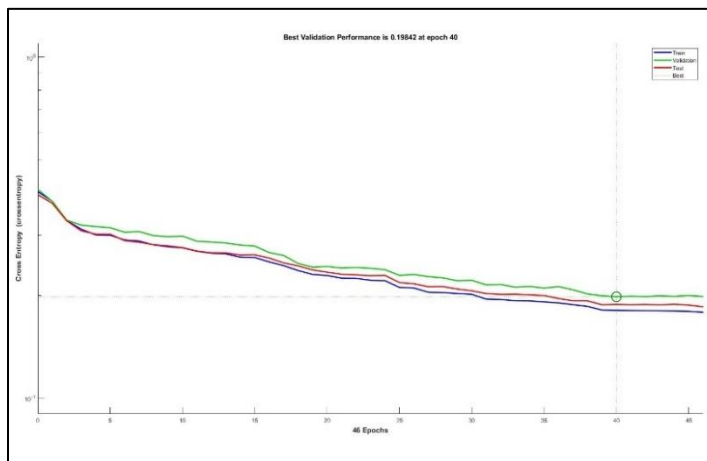
(g1)



(g2)



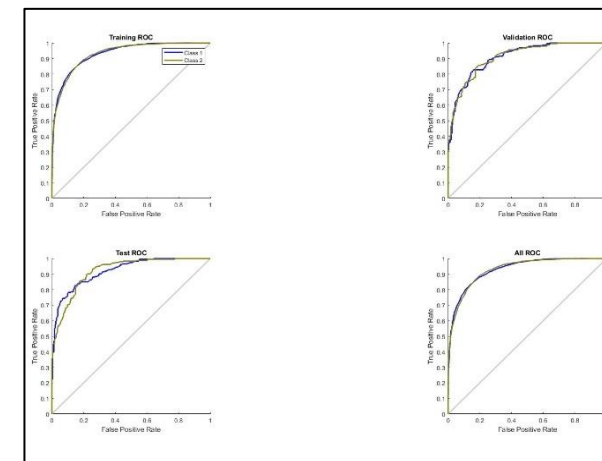
(g3)



(h1)

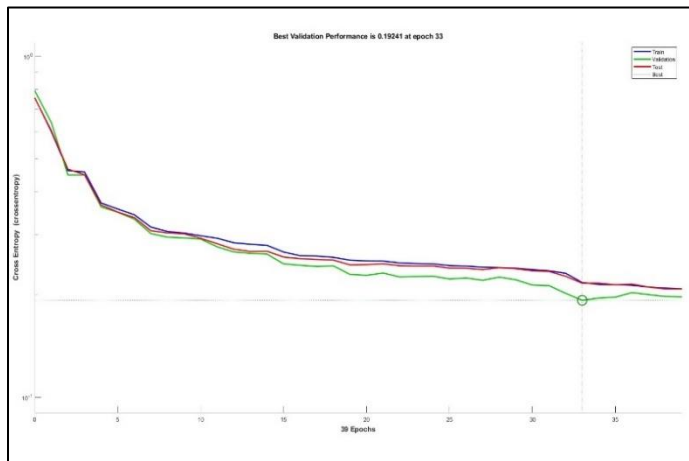


(h2)



(h3)

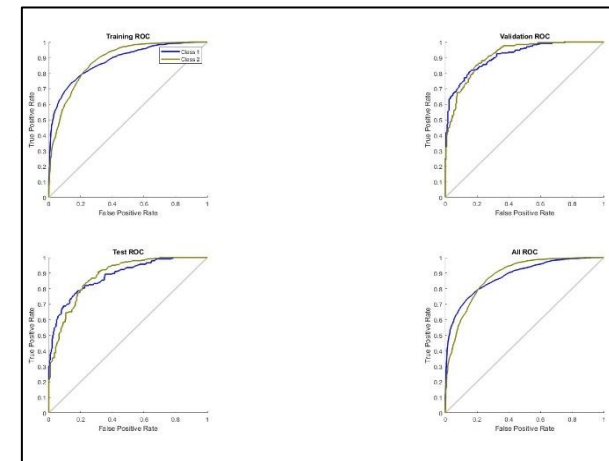
Gambar 4.28 (g1) (h1) Performance Run 7 dan 8 Training OSS HN 20, (g2) (h2) Confusion Matrix Run 7 dan 8 Training OSS HN 20, (g3) (h3) ROC Run 7 dan 8 Training OSS HN 20



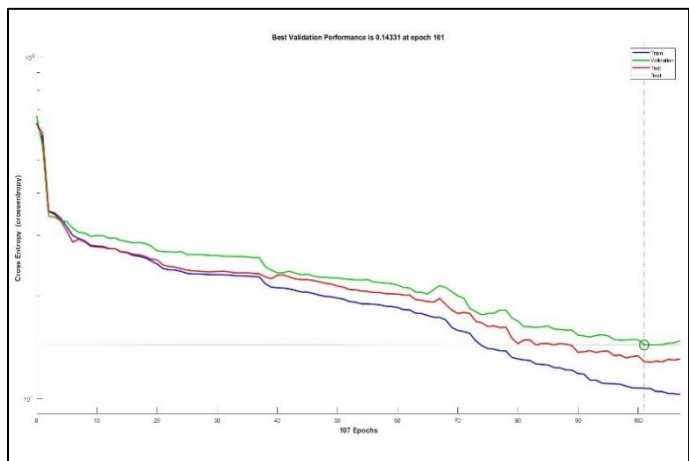
(i1)



(i2)



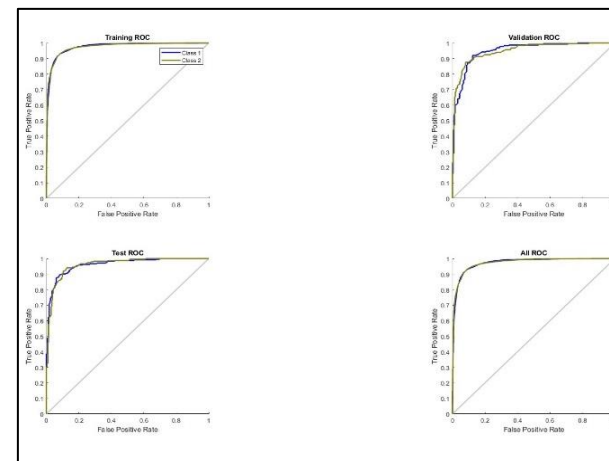
(i3)



(j1)



(j2)



(j3)

Gambar 4.28 (i1) (j1) Performance Run 9 dan 10 Training OSS HN 20, (i2) (j2) Confusion Matrix Run 9 dan 10 Training OSS HN 20, (i3) (j3) ROC Run 9 dan 10 Training OSS HN 20

Berdasarkan visualisasi hasil *training* di atas, dapat diketahui bahwa grafik *performance* terdiri dari 3 grafik, yaitu grafik biru merepresentasikan hasil *training*, grafik hijau merepresentasikan hasil validasi, dan grafik merah merepresentasikan hasil *testing*. Gambar 4.28 menampilkan sepuluh grafik *performane training* OSS *hidden neuron* 20 dimana grafik mengalami penurunan signifikan pada *range epoch* 0-5 kecuali *run* 6 sampai 8. Penurunan *error* melandai mulai *epoch* 5 hingga *epoch* maksimal terlihat pada *run* 1 sampai 5, 9, dan 10. Grafik *run* 6 hingga 8 tidak mengalami penurunan *error* signifikan dari awal hingga *epoch* maksimalnya.

Hasil *training* yang menunjukkan akurasi dari sistem ditunjukkan oleh *confusion matrix* dimana warna hijau merepresentasikan citra yang berhasil diklasifikasikan ke dalam kelas secara benar dan warna merah adalah citra yang gagal diklasifikasikan dalam kelasnya. Perbandingan data yang digunakan untuk *10-run cross validation* yaitu 90% dari total data (8.591) digunakan sebagai data *training*, 5% dari total data (477) digunakan sebagai data validasi, dan 5% lainnya (477) digunakan sebagai data *testing*. Dari *confusion matrix* tersebut, diketahui bahwa rata-rata akurasi *training*, validasi, dan *testing* di atas 90%. Nilai rata-rata *error training* adalah $\leq 10\%$ (859) dan rata-rata nilai *error testing* adalah $\leq 10\%$ (47). Hal ini menunjukkan bahwa model memiliki performa yang baik.

Visualisasi grafik lainnya adalah grafik *Receiver Operating Characteristic* (ROC). Fungsi dari grafik ROC ini adalah untuk mengetahui hasil evaluasi performa *machine learning*. Pada grafik ROC, terdapat dua warna grafik yaitu biru yang menunjukkan klasifikasi citra kelas 1 dan grafik hijau menunjukkan klasifikasi citra kelas 2. Kurva validasi dan *testing* terlihat mendekati angka 1 dan menjauhi garis *threshold* kecuali *run* 8 dan 9. Dengan demikian, dapat disimpulkan bahwa performa *machine learning* (*Multilayer Perceptron* model OSS *hidden neuron* 20) memiliki performa perfect.

4.3.2.3 Gradient Descent with Momentum (GDM)

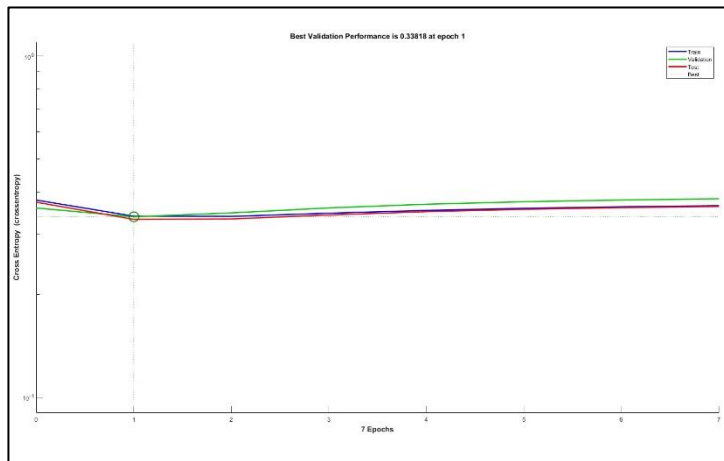
a. Hidden Neuron 1

Hasil akurasi *training* citra *body fluids* menggunakan model *Gradient Descent with Momentum* (GDM) *hidden neuron* 1 dapat dilihat pada Tabel 4.31.

Tabel 4. 31 Hasil Akurasi *Training* Menggunakan Model GDM *Hidden Neuron* 1

<i>Run</i>	<i>Epoch</i>		MLP-TrainGDM Hidden Neuron 1			<i>Elapsed Time</i>
	<i>All</i>	<i>Best</i>	<i>Acc Training</i>	<i>Acc Validation</i>	<i>Acc Testing</i>	
1	7	1	60,5%	61,4%	63,9%	00:00:02
2	1000	1000	52,5%	50,9%	51,8%	00:00:21
3	9	3	54,9%	55,3%	51,2%	00:00:02
4	1000	1000	70,5%	72,5%	71,3%	00:00:20
5	1000	1000	72,8%	71,3%	70,6%	00:00:20
6	1000	1000	69,4%	70,4%	66,2%	00:00:21
7	9	3	59,9%	54,7%	59,1%	00:00:02
8	1000	1000	71,5%	71,7%	71,5%	00:00:21
9	1000	1000	68,2%	66,2%	67,7%	00:00:21
10	1000	1000	68,4%	68,3%	67,1%	00:00:21
Average			64,9%	64,3%	64,0%	00:00:15
Standar Deviasi			0,07294	0,080894	0,07586	

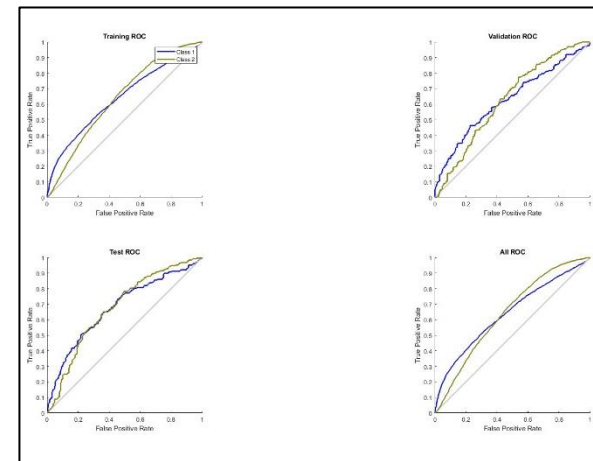
Berdasarkan Tabel 4.31 diketahui bahwa rata-rata akurasi *training* menggunakan model GDM dengan *hidden neuron* 1 adalah 52-72%. Hasil akurasi *training* terbaik terdapat pada *run* 5 dengan akurasi *training* sebesar 72,8%, akurasi validasi 71,3%, dan akurasi *testing* 70,6%. Akurasi *training* terbaik ini diperoleh pada *epoch* ke 1000 dari total 1000 *epoch* dengan waktu *training* selama 20 detik. Sementara itu, hasil akurasi *training* terendah terdapat pada *run* 2 dengan akurasi *training* 52,5%, akurasi validasi 50,9%, dan akurasi *testing* 51,8%. Dari pengujian diatas, diperoleh hasil bahwa rata-rata akurasi *training* adalah 64,9%, akurasi validasi 64,3%, dan akurasi *testing* 64,0%. Rata-rata waktu *training* menggunakan model GDM *hidden neuron* 1 adalah 15 detik. Visualisasi hasil *training* dapat dilihat pada Gambar 4.29.



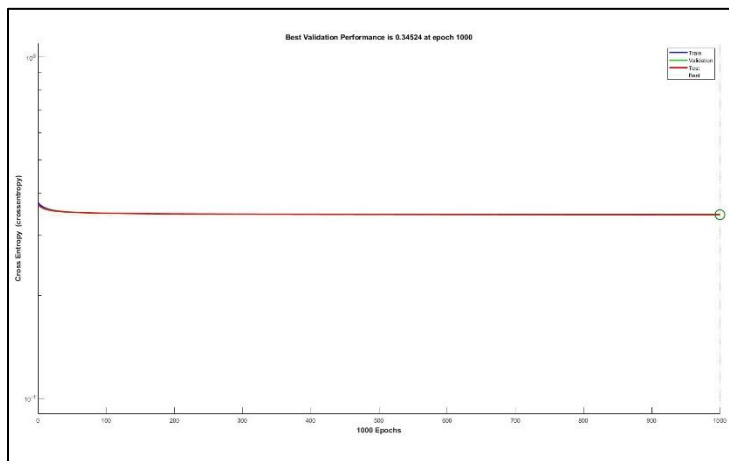
(a1)



(a2)



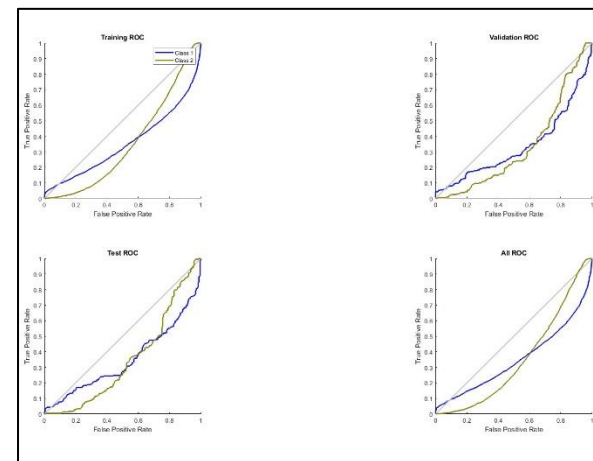
(a3)



(b1)

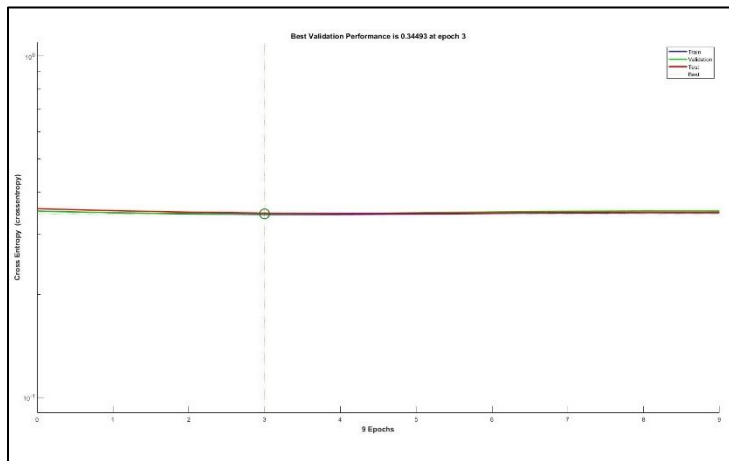


(b2)



(b3)

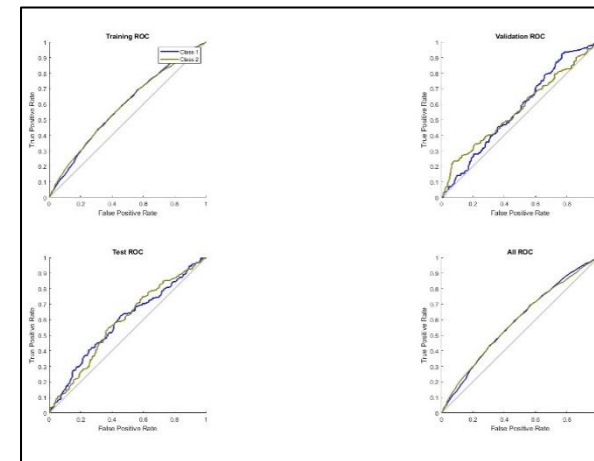
Gambar 4. 29 (a1) (b1) Performance Run 1 dan 2 Training GDM HN 1, (a2) (b2) Confusion Matrix Run 1 dan 2 Training GDM HN 1, (a3) (b3) ROC Run 1 dan 2 Training GDM HN 1



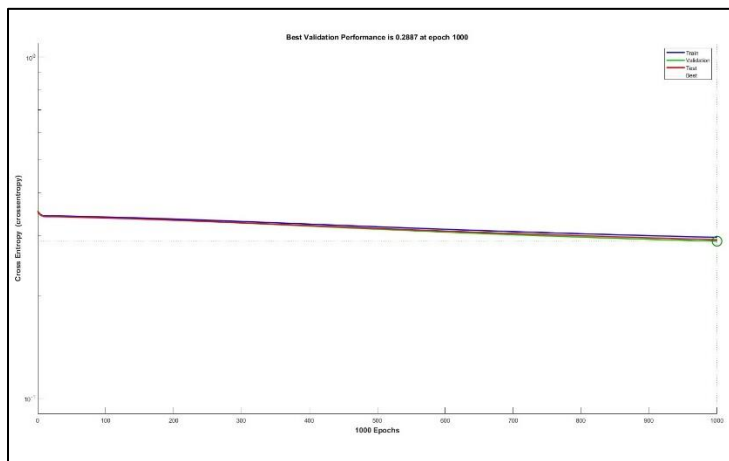
(c1)



(c2)



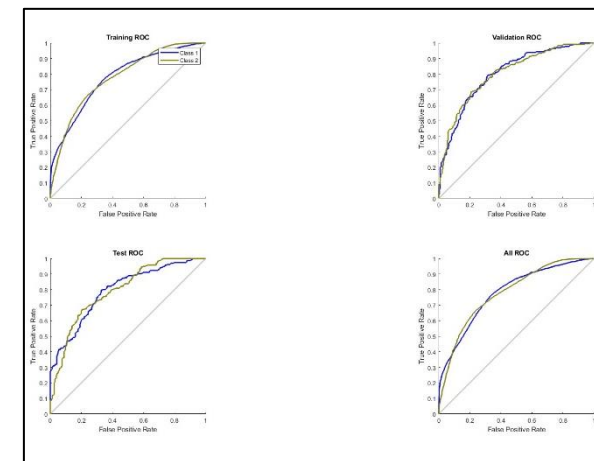
(c3)



(d1)

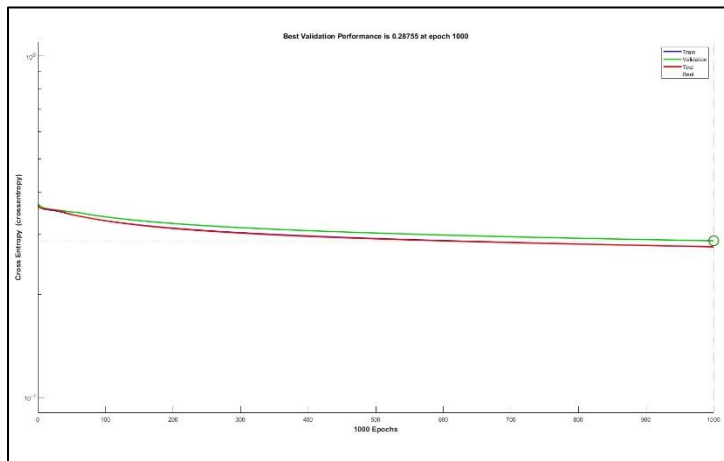


(d2)



(d3)

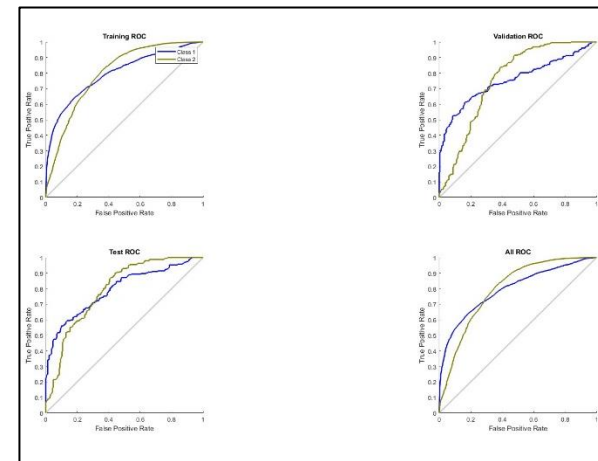
Gambar 4.29 (c1) (d1) Performance Run 3 dan 4 Training GDM HN 1, (c2) (d2) Confusion Matrix Run 3 dan 4 Training GDM HN 1, (c3) (d3) ROC Run 3 dan 4 Training GDM HN 1



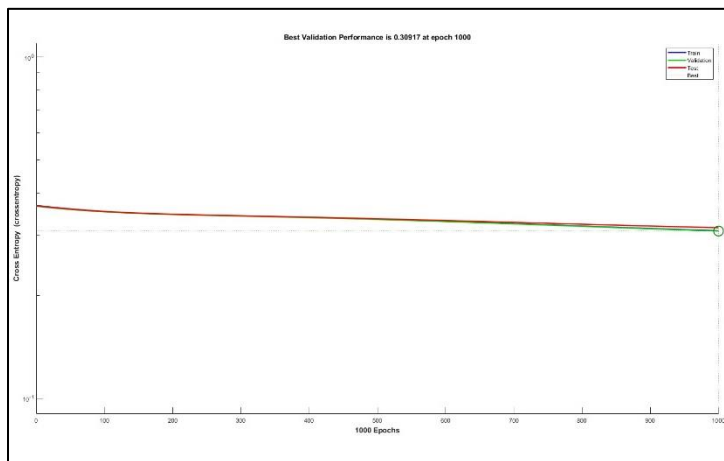
(e1)



(e2)



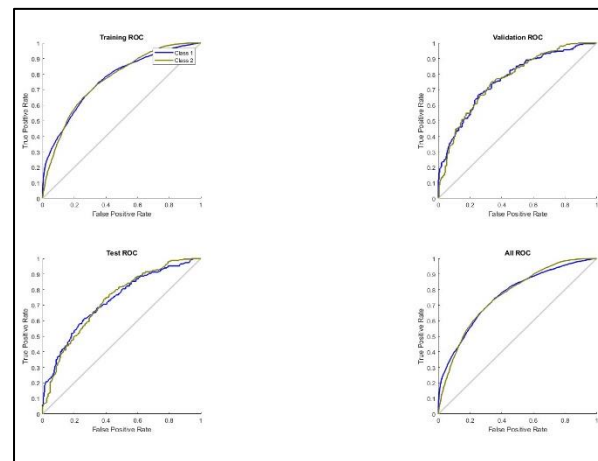
(e3)



(f1)

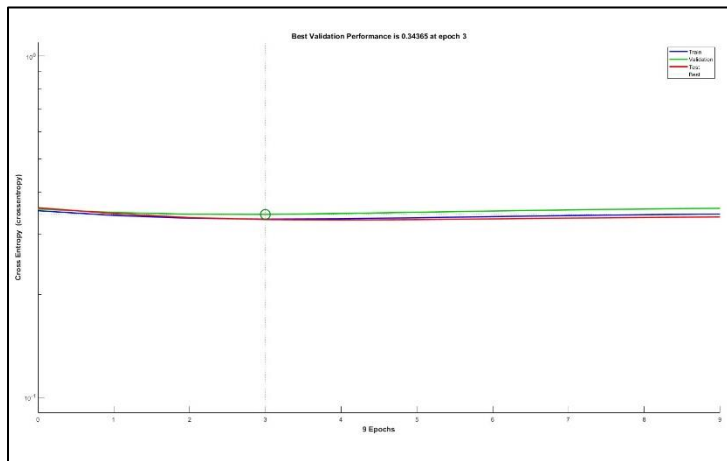


(f2)



(f3)

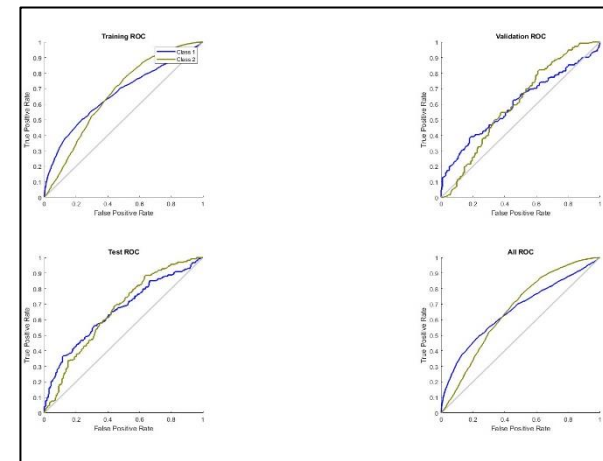
Gambar 4.29 (e1) (f1) Performance Run 5 dan 6 Training GDM HN 1, (e2) (f2) Confusion Matrix Run 5 dan 6 Training GDM HN 1, (e3) (f3) ROC Run 5 dan 6 Training GDM HN 1



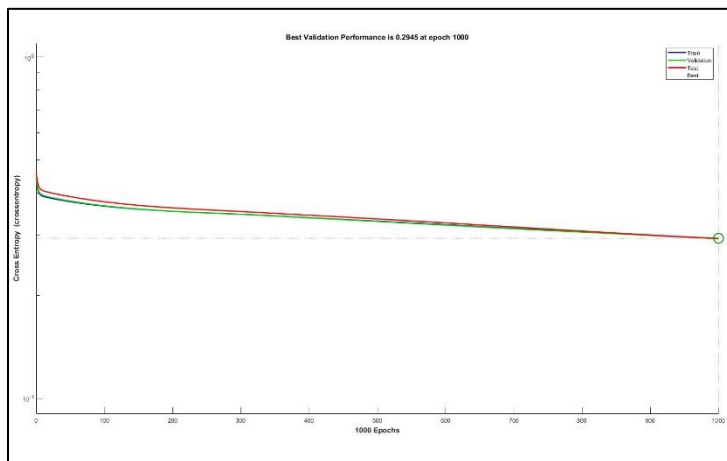
(g1)



(g2)



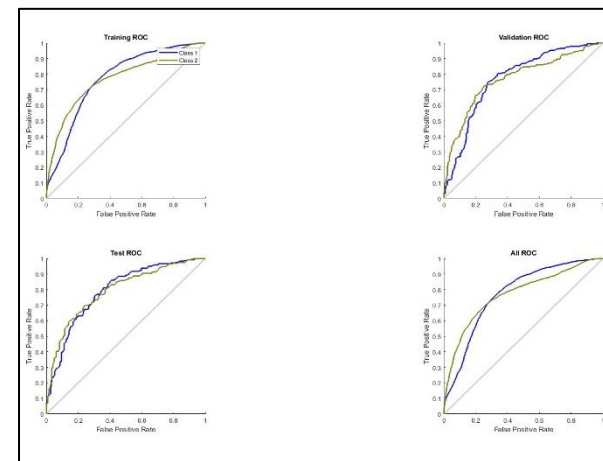
(g3)



(h1)

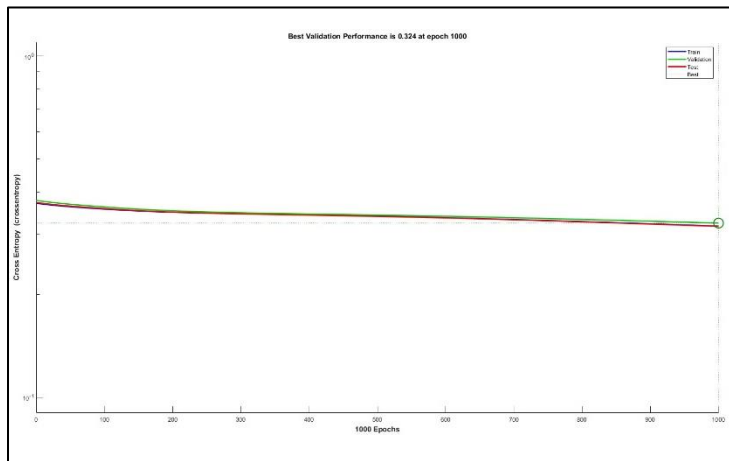


(h2)



(h3)

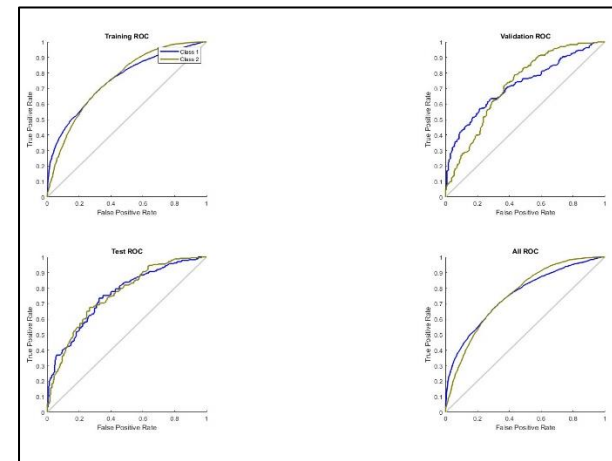
Gambar 4.29 (g1) (h1) Performance Run 7 dan 8 Training GDM HN 1, (g2) (h2) Confusion Matrix Run 7 dan 8 Training GDM HN 1, (g3) (h3) ROC Run 7 dan 8 Training GDM HN 1



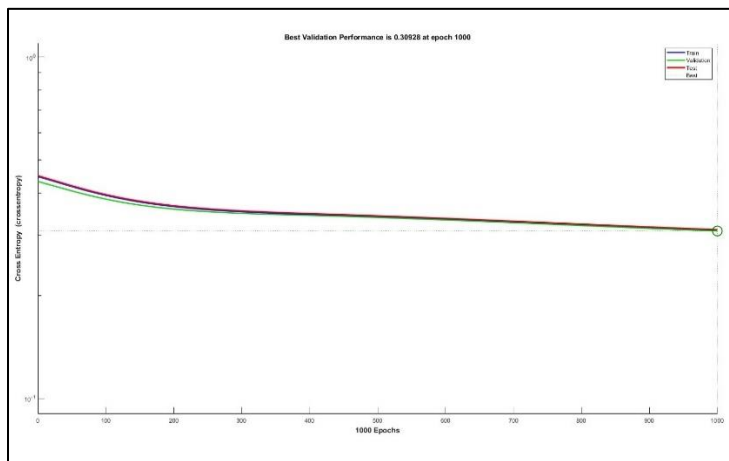
(i1)



(i2)



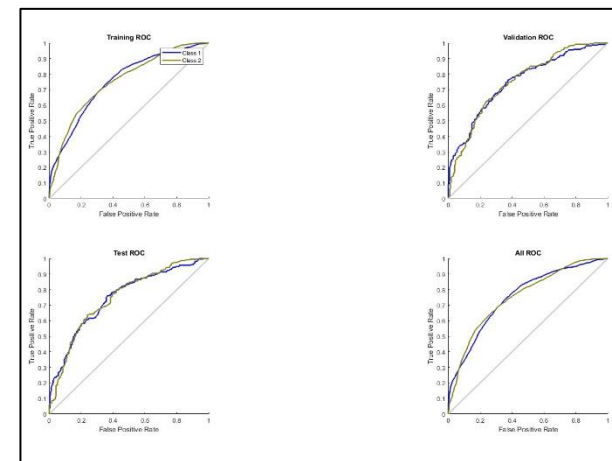
(i3)



(j1)



(j2)



(j3)

Gambar 4.29 (i1) (j1) Performance Run 9 dan 10 Training GDM HN 1, (i2) (j2) Confusion Matrix Run 9 dan 10 Training GDM HN 1, (i3) (j3) ROC Run 9 dan 10 Training GDM HN 1

Berdasarkan visualisasi hasil training di atas, dapat diketahui bahwa grafik *performance* terdiri dari 3 grafik, yaitu grafik biru merepresentasikan hasil *training*, grafik hijau merepresentasikan hasil validasi, dan grafik merah merepresentasikan hasil *testing*. Gambar 4.29 menampilkan sepuluh grafik *performane training GDM hidden neuron 1* dimana grafik tidak mengalami penurunan *error* signifikan dari awal hingga *epoch* maksimalnya.

Hasil *training* yang menunjukkan akurasi dari sistem ditunjukkan oleh *confusion matrix* dimana warna hijau merepresentasikan citra yang berhasil diklasifikasikan ke dalam kelas secara benar dan warna merah adalah citra yang gagal diklasifikasikan dalam kelasnya. Perbandingan data yang digunakan untuk *10-run cross validation* yaitu 90% dari total data (8.591) digunakan sebagai data *training*, 5% dari total data (477) digunakan sebagai data validasi, dan 5% lainnya (477) digunakan sebagai data *testing*. Dari *confusion matrix* tersebut, diketahui bahwa rata-rata akurasi *training*, validasi, dan *testing* adalah 50-70%. Nilai rata-rata *error training* adalah $\leq 30\%$ (2.577) dan rata-rata nilai *error testing* adalah $\leq 30\%$ (143). Hal ini menunjukkan bahwa model memiliki performa tidak cukup baik.

Visualisasi grafik lainnya adalah grafik *Receiver Operating Characteristic* (ROC). Fungsi dari grafik ROC ini adalah untuk mengetahui hasil evaluasi performa *machine learning*. Pada grafik ROC, terdapat dua warna grafik yaitu biru yang menunjukkan klasifikasi citra kelas 1 dan grafik hijau menunjukkan klasifikasi citra kelas 2. Kurva validasi dan *testing* berada pada angka 0,7 kebawah dan menjauhi angka 1. Grafik terlihat mendekati garis *threshold* bahkan kurva *run 2* berada di bawah garis *threshold*. Dengan demikian, dapat disimpulkan bahwa performa *machine learning* (*Multilayer Perceptron* model *GDM hidden neuron 1*) menunjukkan performa tidak perfect.

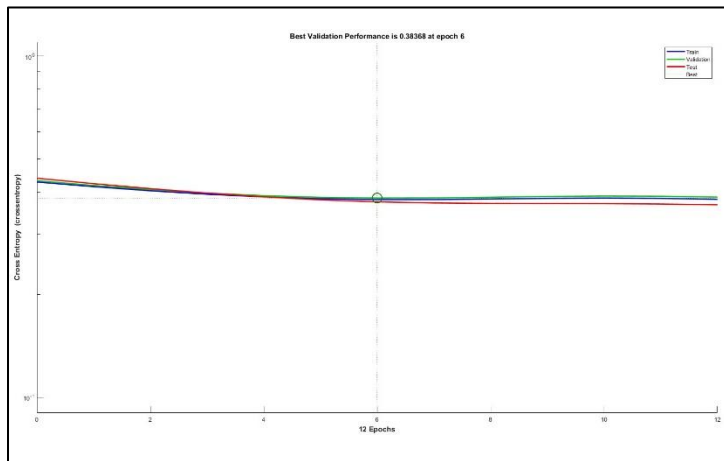
b. *Hidden Neuron 5*

Hasil akurasi *training* citra *body fluids* menggunakan model *Gradient Descent with Momentum* (GDM) *hidden neuron 5* dapat dilihat pada Tabel 4.32.

Tabel 4. 32 Hasil Akurasi *Training* Menggunakan Model GDM *Hidden Neuron 5*

<i>Run</i>	<i>Epoch</i>		MLP-TrainGDM <i>Hidden Neuron 5</i>			<i>Elapsed Time</i>
	<i>All</i>	<i>Best</i>	<i>Acc Training</i>	<i>Acc Validation</i>	<i>Acc Testing</i>	
1	12	6	43,5%	44,2%	45,1%	00:00:02
2	22	16	60,6%	63,5%	58,7%	00:00:03
3	1000	1000	72,3%	71,5%	73,0%	00:00:40
4	23	17	60,9%	65,6%	61,8%	00:00:03
5	9	3	61,8%	62,9%	63,1%	00:00:02
6	1000	1000	74,7%	75,3%	74,2%	00:00:39
7	1000	1000	70,3%	64,6%	69,6%	00:00:39
8	9	3	56,5%	57,0%	55,8%	00:00:02
9	1000	1000	74,2%	72,7%	72,5%	00:00:40
10	11	5	37,6%	39,2%	38,2%	00:00:02
Average			61,2%	61,7%	61,2%	00:00:17
Standar Deviasi			0,126947	0,11855	0,12175	

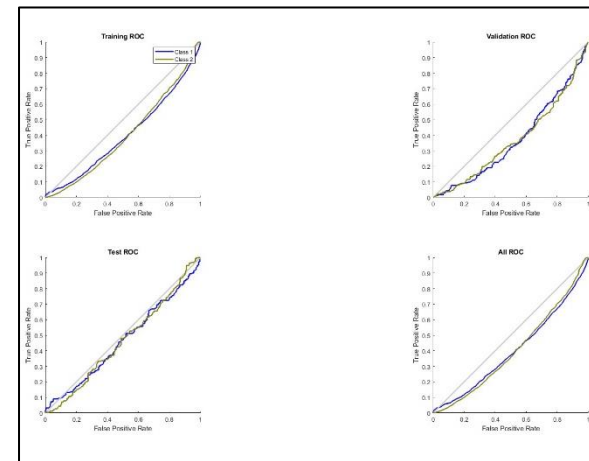
Berdasarkan Tabel 4.32 diketahui bahwa rata-rata akurasi *training* menggunakan model GDM dengan *hidden neuron 5* adalah 37-74%. Hasil akurasi *training* terbaik terdapat pada *run 6* dengan akurasi *training* sebesar 74,7%, akurasi validasi 75,3%, dan akurasi *testing* 74,2%. Akurasi *training* terbaik ini diperoleh pada *epoch* ke 1000 dari total 1000 *epoch* dengan waktu *training* selama 39 detik. Sementara itu, hasil akurasi *training* terendah terdapat pada *run 10* dengan akurasi *training* 37,6%, akurasi validasi 39,2%, dan akurasi *testing* 38,2%. Dari pengujian diatas, diperoleh hasil bahwa rata-rata akurasi *training* adalah 61,2%, akurasi validasi 61,7%, dan akurasi *testing* 61,2%. Rata-rata waktu *training* menggunakan model GDM *hidden neuron 5* adalah 17 detik. Visualisasi hasil *training* dapat dilihat pada Gambar 4.30.



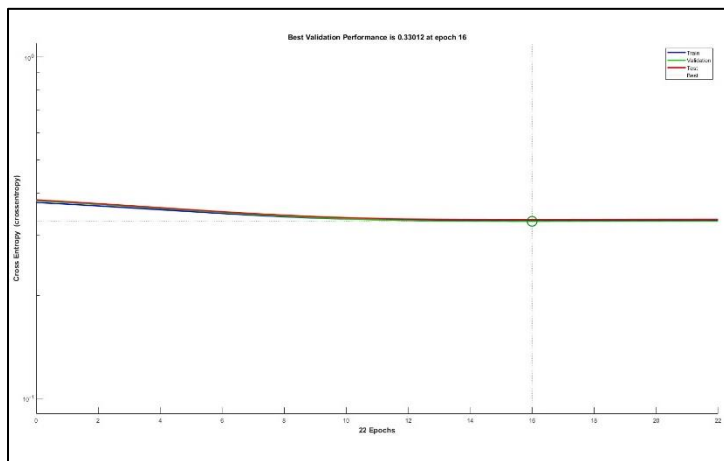
(a1)



(a2)



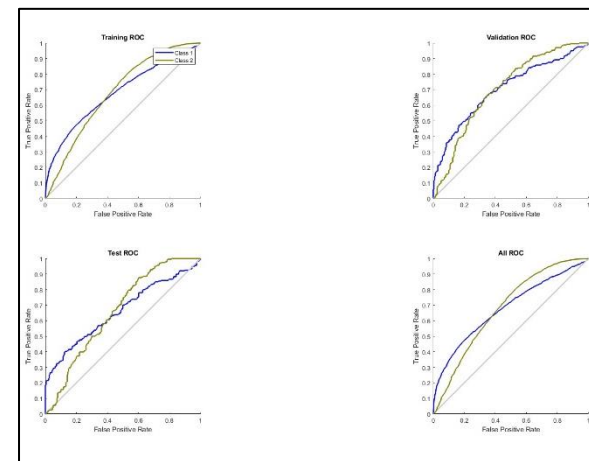
(a3)



(b1)

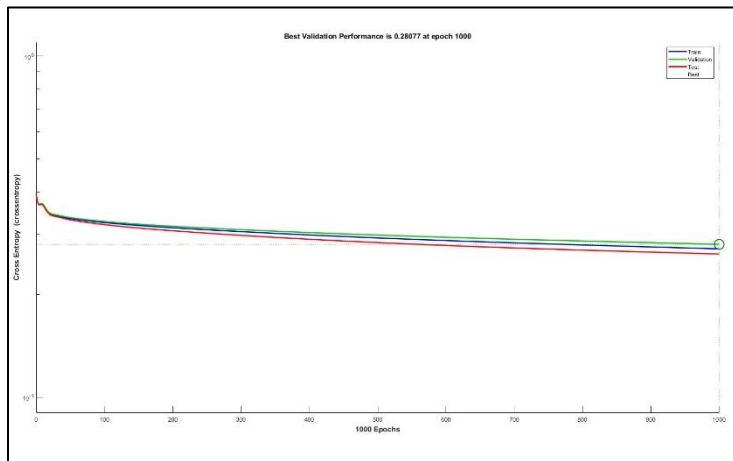


(b2)



(b3)

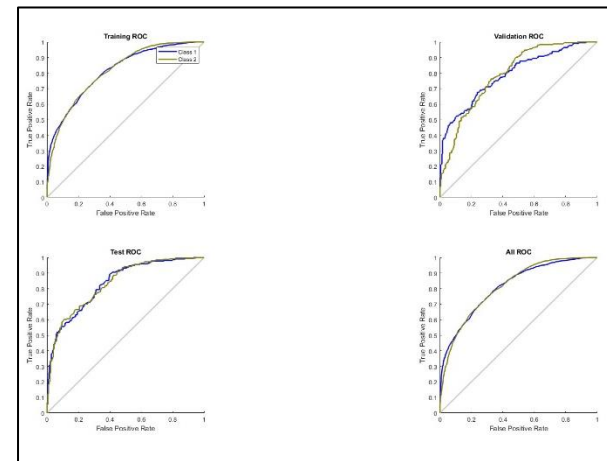
Gambar 4. 30 (a1) (b1) Performance Run 1 dan 2 Training GDM HN 5, (a2) (b2) Confusion Matrix Run 1 dan 2 Training GDM HN 5, (a3) (b3) ROC Run 1 dan 2 Training GDM HN 5



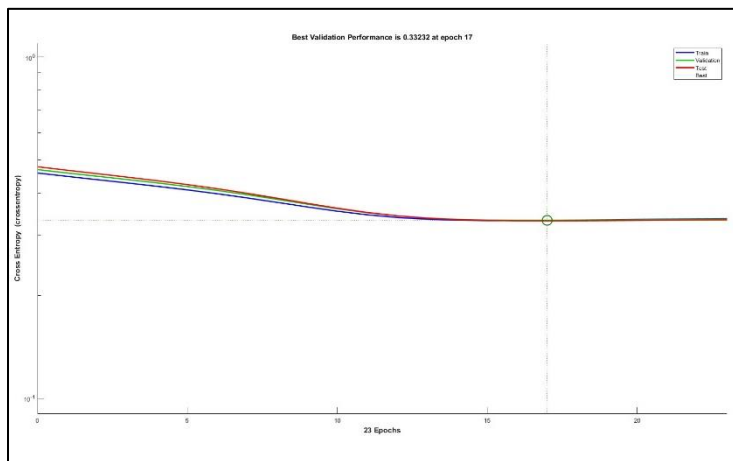
(c1)



(c2)



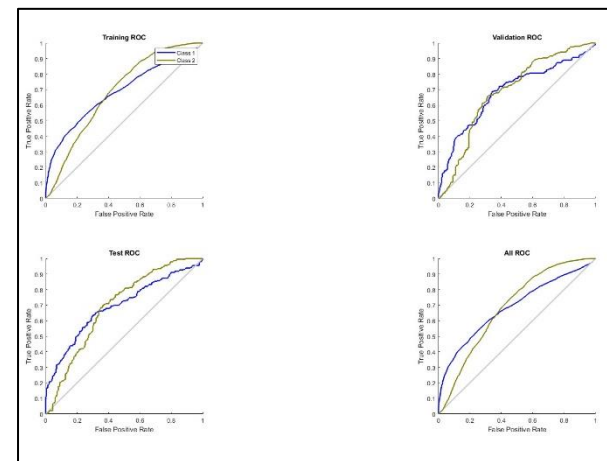
(c3)



(d1)

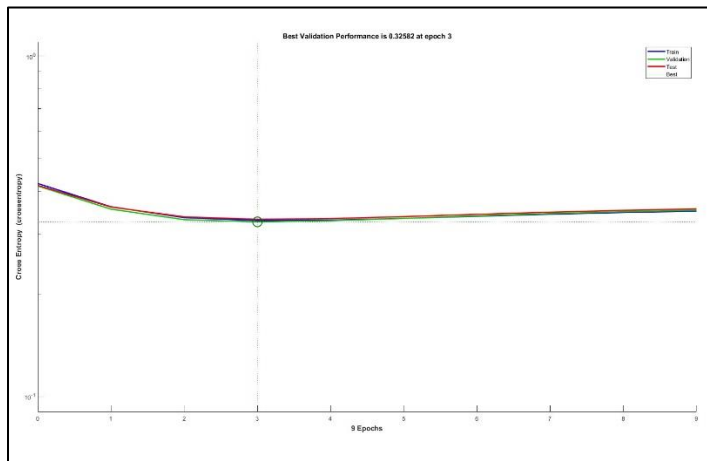


(d2)



(d3)

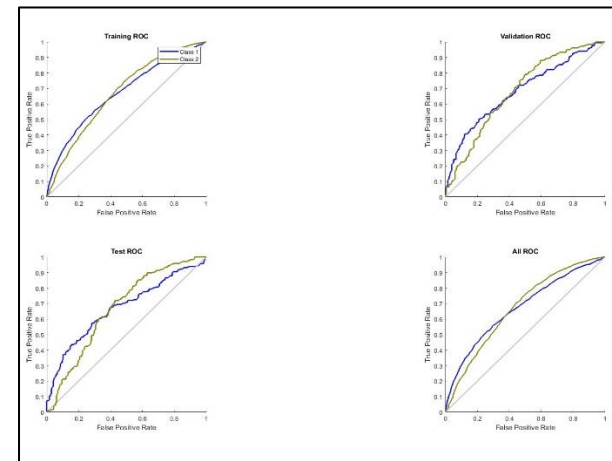
Gambar 4.30 (c1) (d1) Performance Run 3 dan 4 Training GDM HN 5, (c2) (d2) Confusion Matrix Run 3 dan 4 Training GDM HN 5, (c3) (d3) ROC Run 3 dan 4 Training GDM HN 5



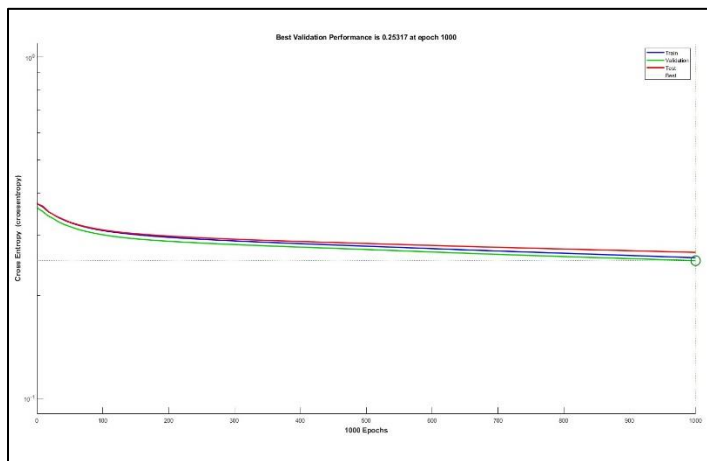
(e1)



(e2)



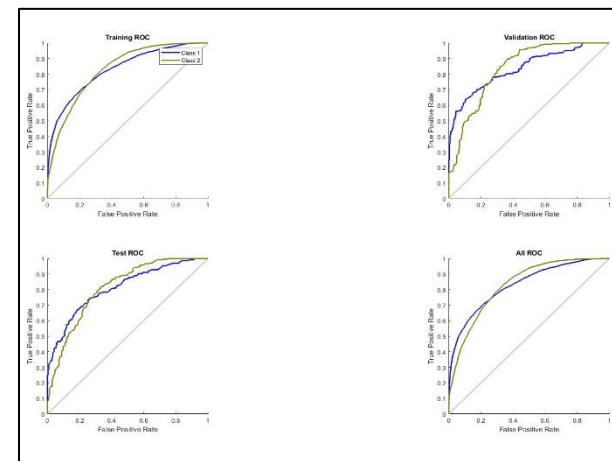
(e3)



(f1)

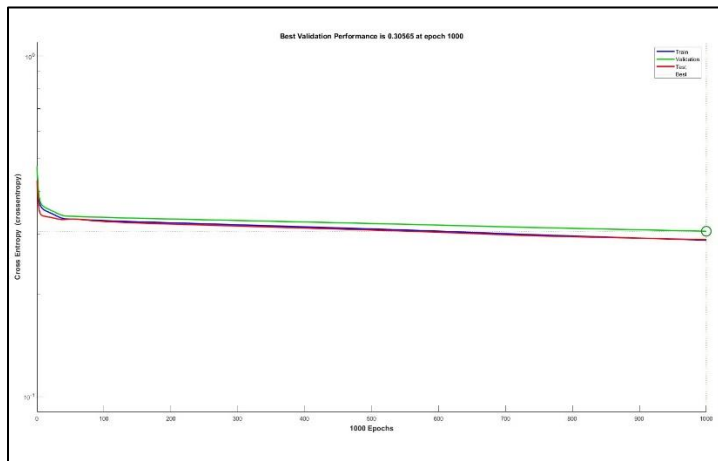


(f2)



(f3)

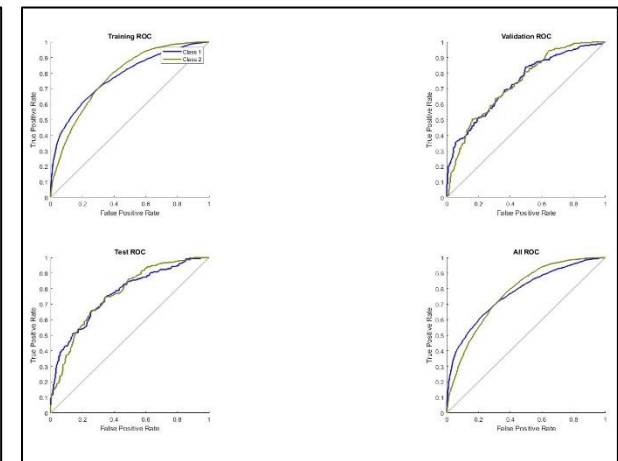
Gambar 4.30 (e1) (f1) Performance Run 5 dan 6 Training GDM HN 5, (e2) (f2) Confusion Matrix Run 5 dan 6 Training GDM HN 5, (e3) (f3) ROC Run 5 dan 6 Training GDM HN 5



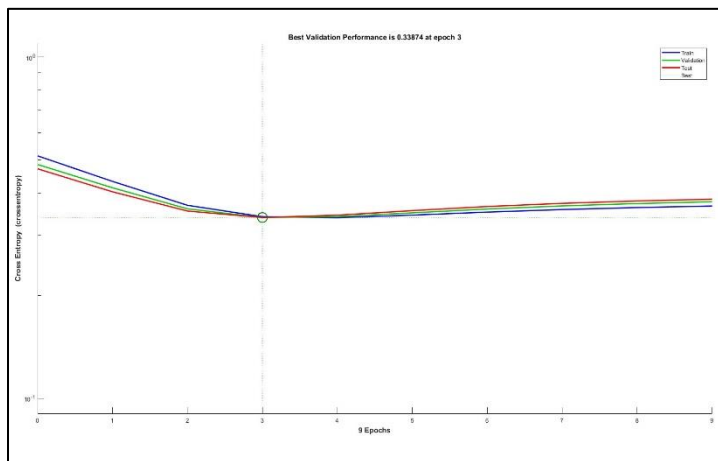
(g1)



(g2)



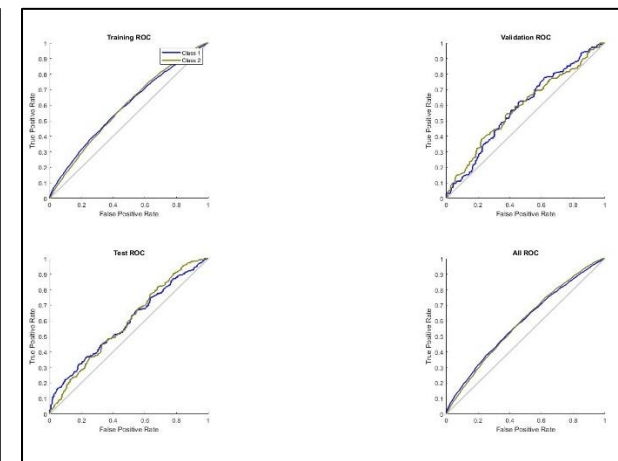
(g3)



(h1)

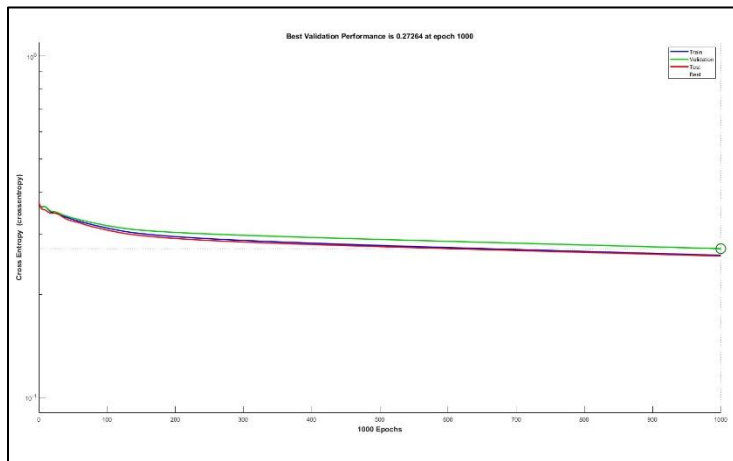


(h2)



(h3)

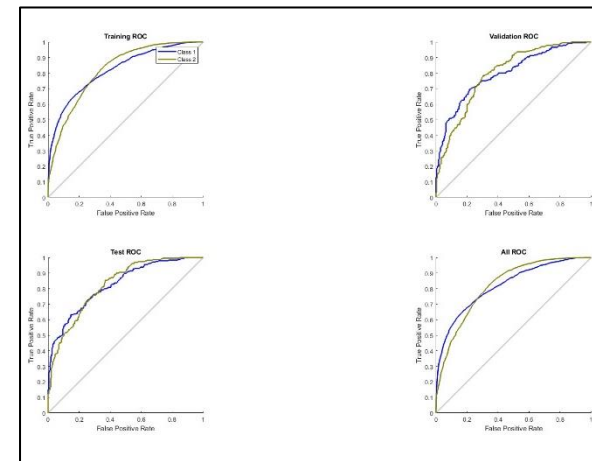
Gambar 4.30 (g1) (h1) Performance Run 7 dan 8 Training GDM HN 5, (g2) (h2) Confusion Matrix Run 7 dan 8 Training GDM HN 5, (g3) (h3) ROC Run 7 dan 8 Training GDM HN 5



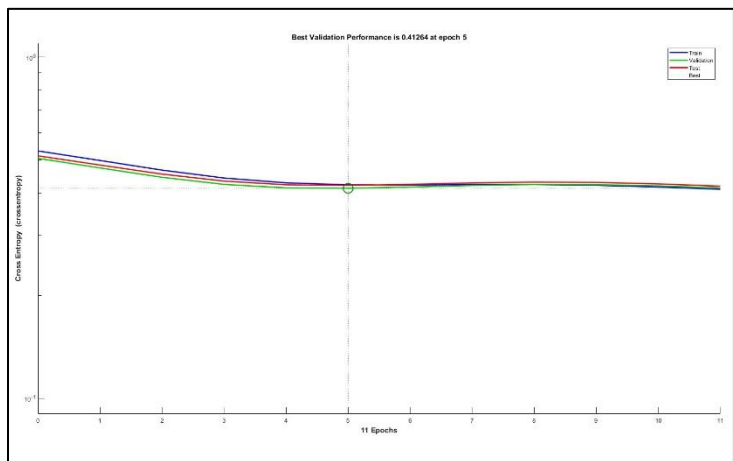
(i1)



(i2)



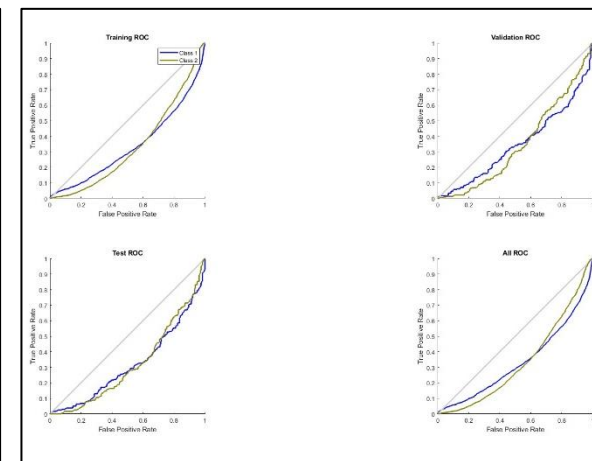
(i3)



(j1)



(j2)



(j3)

Gambar 4.30 (i1) (j1) Performance Run 9 dan 10 Training GDM HN 5, (i2) (j2) Confusion Matrix Run 9 dan 10 Training GDM HN 5, (i3) (j3) ROC Run 9 dan 10 Training GDM HN 5

Berdasarkan visualisasi hasil *training* di atas, dapat diketahui bahwa grafik *performance* terdiri dari 3 grafik, yaitu grafik biru merepresentasikan hasil *training*, grafik hijau merepresentasikan hasil validasi, dan grafik merah merepresentasikan hasil *testing*. Gambar 4.30 menampilkan sepuluh grafik *performane training* GDM *hidden neuron* 5 dimana grafik mengalami penurunan *error* melandai dari *epoch* 0-2 pada *run* 5 dan 8, penurunan melandai dari *epoch* 0-10 terjadi pada *run* 4 dan 7. Grafik hasil *run* 1 sampai 3, 6, 9, dan 10 tidak mengalami penurunan signifikan dari awal hingga *epoch* maksimalnya. Dari uraian di atas, dapat disimpulkan bahwa semakin rendah nilai entropi dari grafik (nilai *error*) maka semakin baik performa dari model.

Hasil *training* yang menunjukkan akurasi dari sistem ditunjukkan oleh *confusion matrix* dimana warna hijau merepresentasikan citra yang berhasil diklasifikasikan ke dalam kelas secara benar dan warna merah adalah citra yang gagal diklasifikasikan dalam kelasnya. Perbandingan data yang digunakan untuk *10-run cross validation* yaitu 90% dari total data (8.591) digunakan sebagai data *training*, 5% dari total data (477) digunakan sebagai data validasi, dan 5% lainnya (477) digunakan sebagai data *testing*. Dari *confusion matrix* tersebut, diketahui bahwa rata-rata akurasi *training* adalah 37-74%. Nilai rata-rata *error training* $\geq 20\%$ (1.718) serta rata-rata *error validasi* dan *testing* $\geq 20\%$ (95). Hal ini menunjukkan bahwa model memiliki performa yang baik.

Visualisasi grafik lainnya adalah grafik *Receiver Operating Characteristic* (ROC). Fungsi dari grafik ROC ini adalah untuk mengetahui hasil evaluasi performa *machine learning*. Pada grafik ROC, terdapat dua warna grafik yaitu biru yang menunjukkan klasifikasi citra kelas 1 dan grafik hijau menunjukkan klasifikasi citra kelas 2. Kurva validasi dan *testing* terlihat menjauhi angka 1 dan mendekati garis *threshold*. Bahkan kurva *run* 1 dan 10 berada di bawah garis *threshold*. Dengan demikian, dapat disimpulkan bahwa performa *machine learning* (*Multilayer Perceptron* model GDM *hidden neuron* 5) dinilai tidak *perfect*.

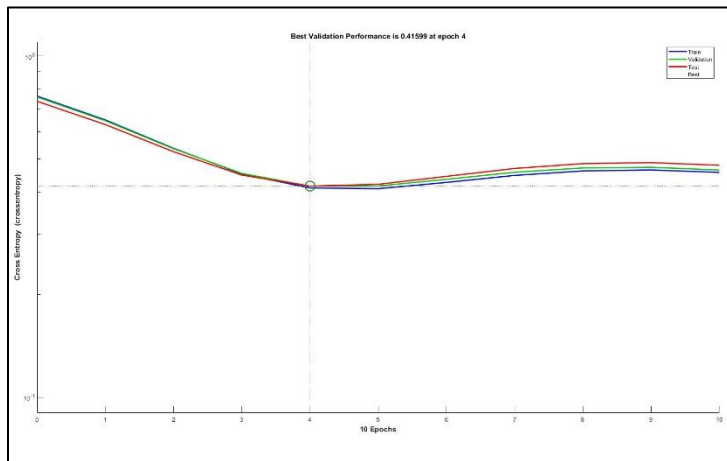
c. *Hidden Neuron 10*

Hasil akurasi *training* citra *body fluids* menggunakan model *Gradient Descent with Momentum* (GDM) *hidden neuron 10* dapat dilihat pada Tabel 4.33.

Tabel 4. 33 Hasil Akurasi *Training* Menggunakan Model GDM *Hidden Neuron 10*

<i>Run</i>	<i>Epoch</i>		MLP-TrainGDM Hidden Neuron 10			<i>Elapsed Time</i>
	<i>All</i>	<i>Best</i>	<i>Acc Training</i>	<i>Acc Validation</i>	<i>Acc Testing</i>	
1	10	4	42,4%	42,6%	39,0%	00:00:02
2	7	1	58,5%	58,5%	61,0%	00:00:02
3	8	2	61,4%	61,6%	58,9%	00:00:02
4	7	1	40,4%	40,7%	38,6%	00:00:02
5	7	1	50,7%	50,3%	48,2%	00:00:02
6	1000	1000	76,7%	78,4%	75,7%	00:00:43
7	9	3	37,4%	36,5%	38,8%	00:00:02
8	7	1	36,6%	37,9%	39,2%	00:00:02
9	8	2	40,2%	39,8%	45,1%	00:00:03
10	10	4	61,5%	65,6%	60,2%	00:00:02
Average			50,6%	51,2%	50,5%	00:00:06
Standar Deviasi			0,134526	0,142072	0,128358	

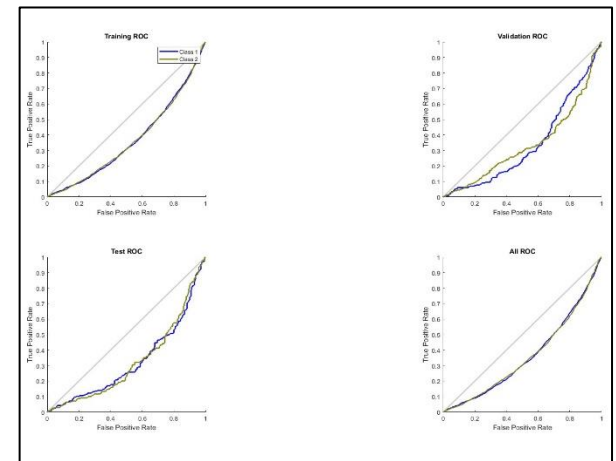
Berdasarkan Tabel 4.33 diketahui bahwa rata-rata akurasi *training* menggunakan model GDM dengan *hidden neuron 10* adalah 36-76%. Hasil akurasi *training* terbaik terdapat pada *run 6* dengan akurasi *training* sebesar 76,7%, akurasi validasi 78,4%, dan akurasi *testing* 75,7%. Akurasi *training* terbaik ini diperoleh pada *epoch* ke 1000 dari total 1000 *epoch* dengan waktu *training* selama 43 detik. Sementara itu, hasil akurasi *training* terendah terdapat pada *run 8* dengan akurasi *training* 36,6%, akurasi validasi 37,9%, dan akurasi *testing* 39,2%. Dari pengujian diatas, diperoleh hasil bahwa rata-rata akurasi *training* adalah 50,6%, akurasi validasi 51,2%, dan akurasi *testing* 50,5%. Rata-rata waktu *training* menggunakan model GDM *hidden neuron 10* adalah 6 detik. Visualisasi hasil *training* dapat dilihat pada Gambar 4.31.



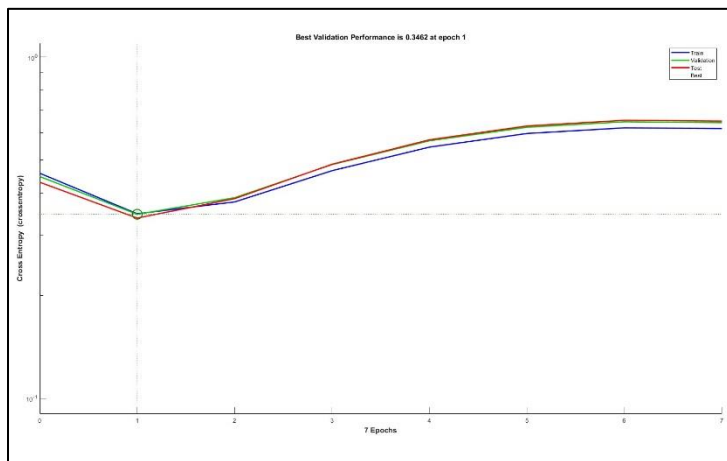
(a1)



(a2)



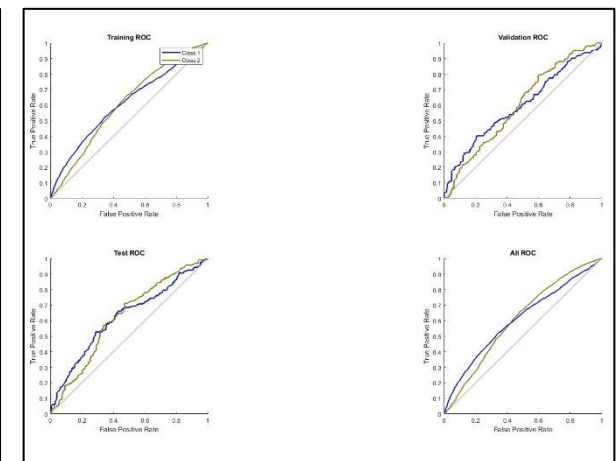
(a3)



(b1)

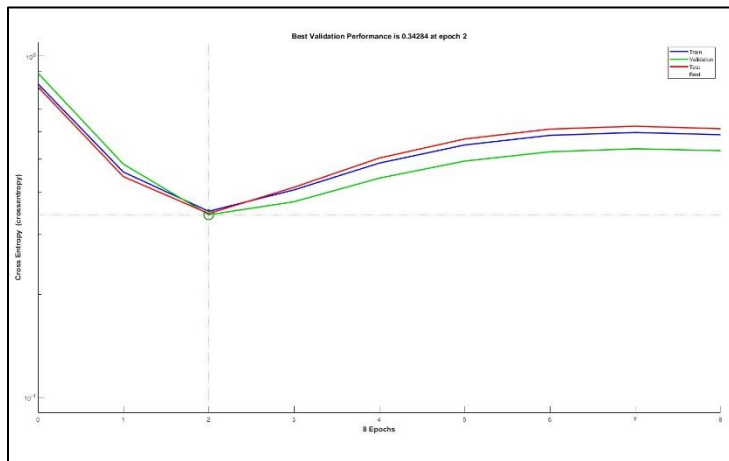


(b2)



(b3)

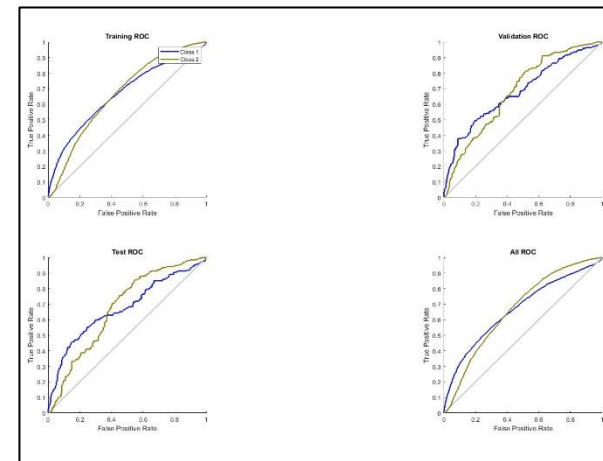
Gambar 4. 31 (a1) (b1) Performance Run 1 dan 2 Training GDM HN 10, (a2) (b2) Confusion Matrix Run 1 dan 2 Training GDM HN 10, (a3) (b3) ROC Run 1 dan 2 Training GDM HN 10



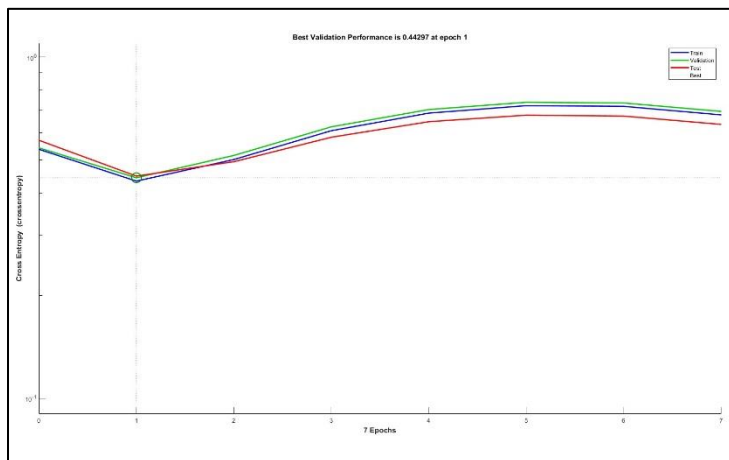
(c1)



(c2)



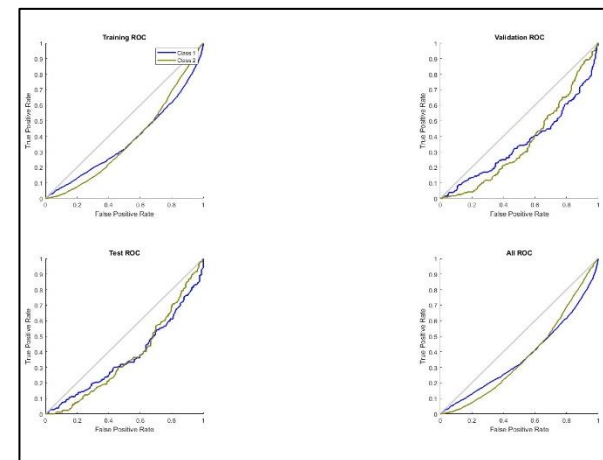
(c3)



(d1)

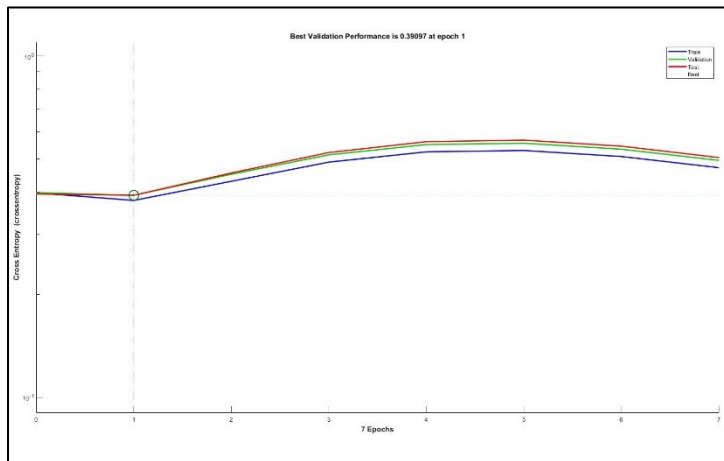


(d2)



(d3)

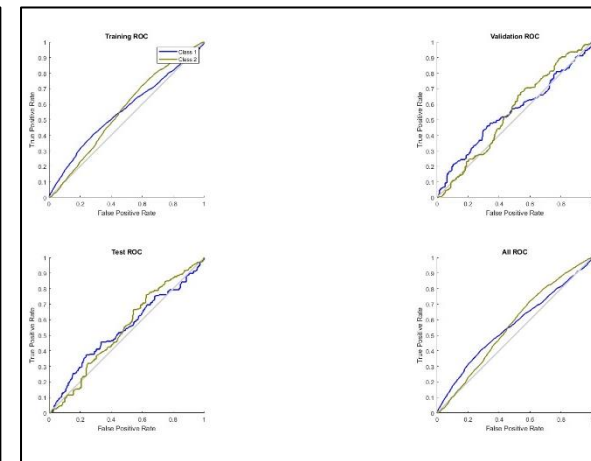
Gambar 4.31 (c1) (d1) Performance Run 3 dan 4 Training GDM HN 10, (c2) (d2) Confusion Matrix Run 3 dan 4 Training GDM HN 10, (c3) (d3) ROC Run 3 dan 4 Training GDM HN 10



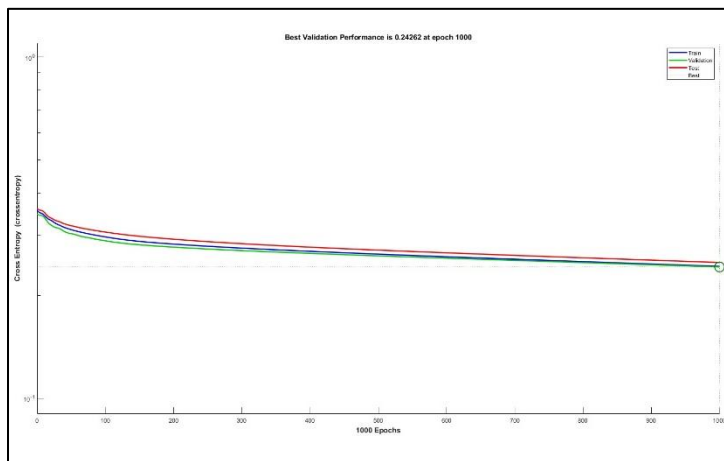
(e1)



(e2)



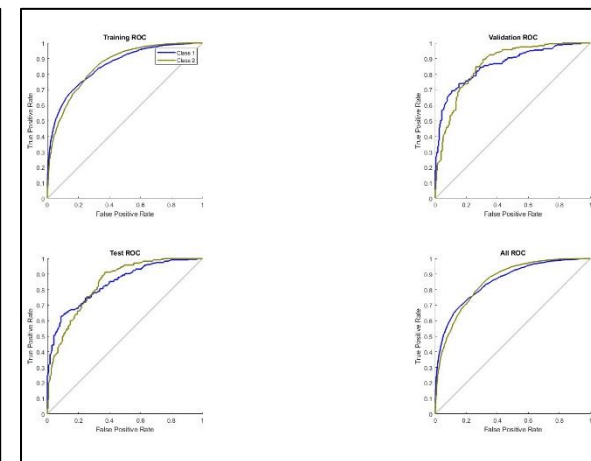
(e3)



(f1)

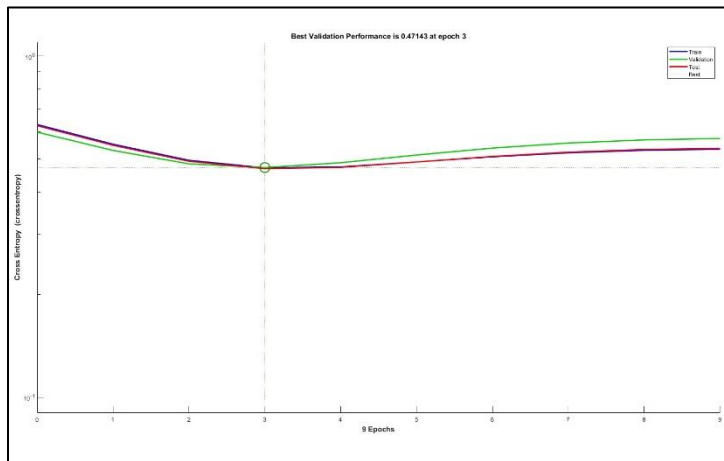


(f2)



(f3)

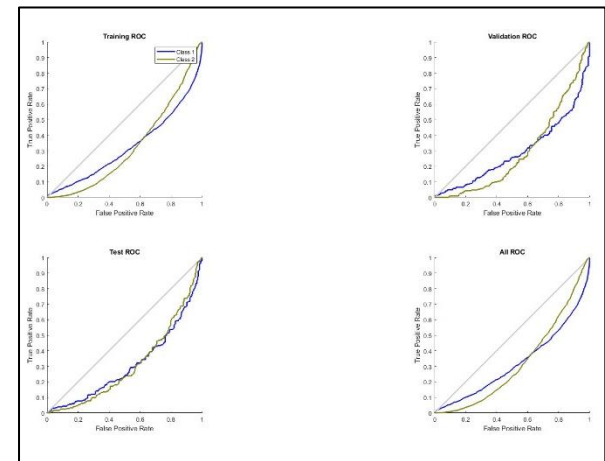
Gambar 4.31 (e1) (f1) Performance Run 5 dan 6 Training GDM HN 10, (e2) (f2) Confusion Matrix Run 5 dan 6 Training GDM HN 10, (e3) (f3) ROC Run 5 dan 6 Training GDM HN 10



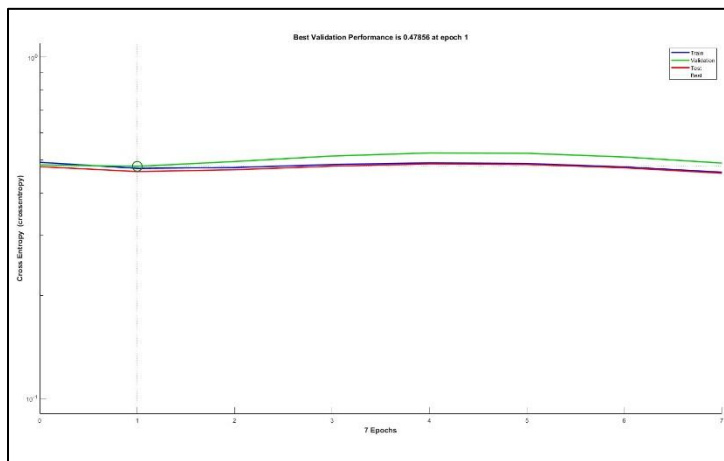
(g1)



(g2)



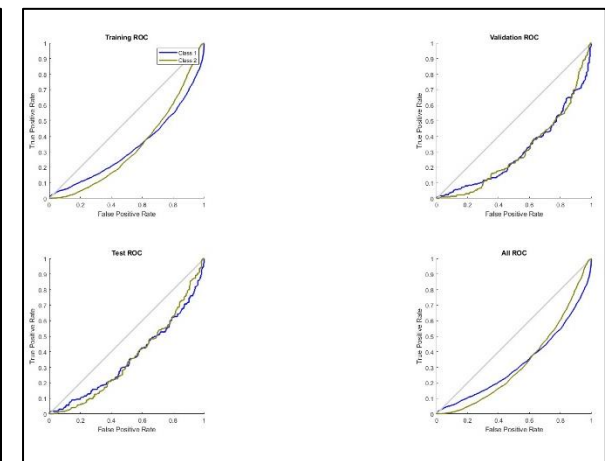
(g3)



(h1)

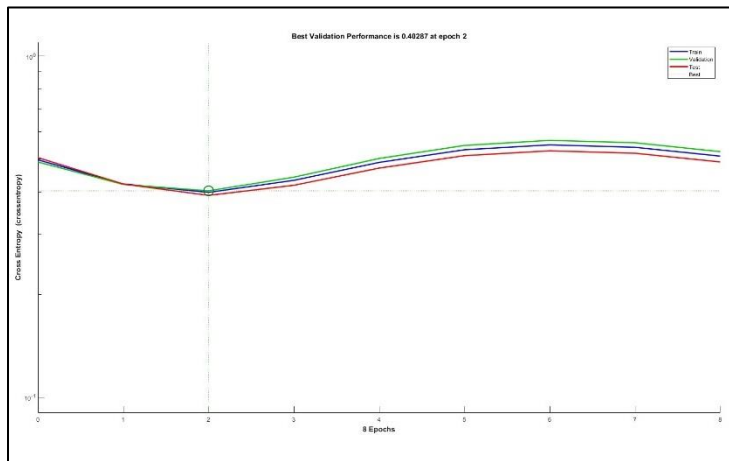


(h2)



(h3)

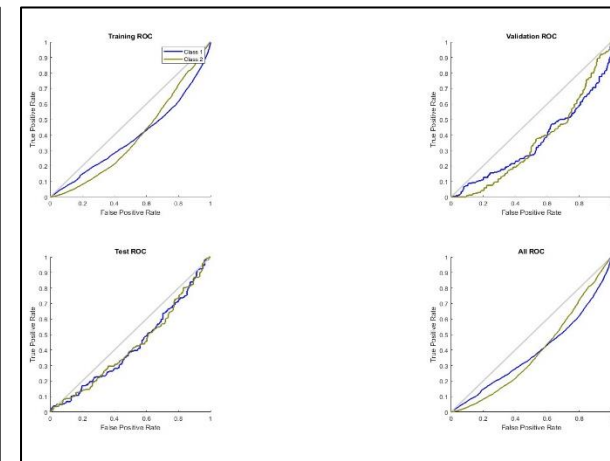
Gambar 4.31 (g1) (h1) Performance Run 7 dan 8 Training GDM HN 10, (g2) (h2) Confusion Matrix Run 7 dan 8 Training GDM HN 10, (g3) (h3) ROC Run 7 dan 8 Training GDM HN 10



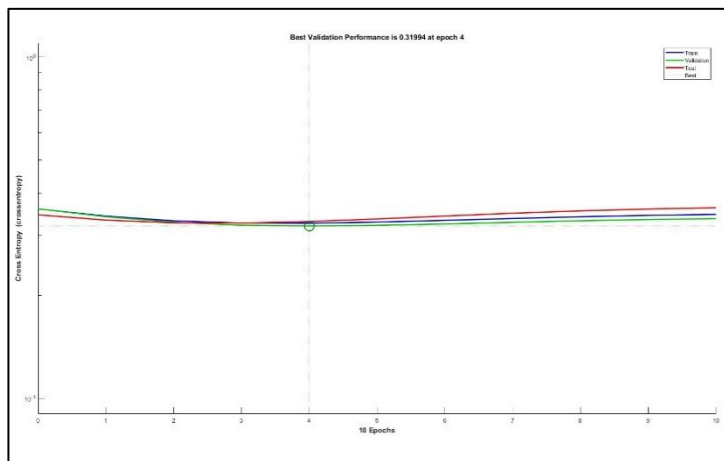
(i1)



(i2)



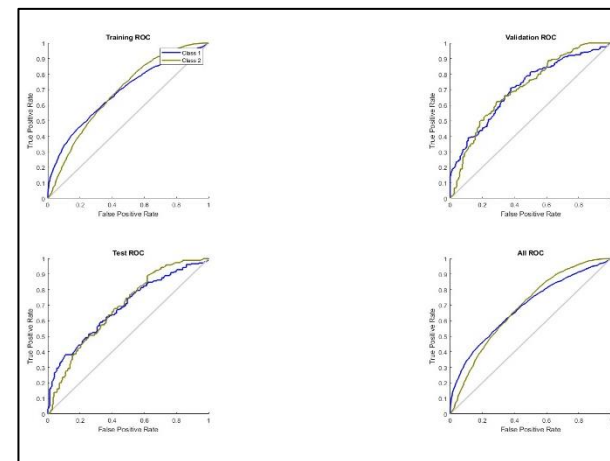
(i3)



(j1)



(j2)



(j3)

Gambar 4.31 (i1) (j1) Performance Run 9 dan 10 Training GDM HN 10, (i2) (j2) Confusion Matrix Run 9 dan 10 Training GDM HN 10, (i3) (j3) ROC Run 9 dan 10 Training GDM HN 10

Berdasarkan visualisasi hasil training di atas, dapat diketahui bahwa grafik *performance* terdiri dari 3 grafik, yaitu grafik biru merepresentasikan hasil *training*, grafik hijau merepresentasikan hasil validasi, dan grafik merah merepresentasikan hasil *testing*. Gambar 4.31 menampilkan sepuluh grafik *performane training* GDM *hidden neuron* 10 dimana grafik mengalami penurunan *error* signifikan pada *range epoch* 0-1 yaitu pada *run* 2, 4, dan 5. Penurunan signifikan *range epoch* 0-2 terjadi pada *run* 1, 3, 7, dan 9. Grafik performa *error* *run* 6, 8, dan 10 tidak mengalami penurunan signifikan dari awal hingga *epoch* maksimalnya. Dari uraian di atas, dapat disimpulkan bahwa semakin rendah nilai entropi dari grafik (nilai *error*) maka semakin baik performa dari model.

Hasil *training* yang menunjukkan akurasi dari sistem ditunjukkan oleh *confusion matrix* dimana warna hijau merepresentasikan citra yang berhasil diklasifikasikan ke dalam kelas secara benar dan warna merah adalah citra yang gagal diklasifikasikan dalam kelasnya. Perbandingan data yang digunakan untuk *10-run cross validation* yaitu 90% dari total data (8.591) digunakan sebagai data *training*, 5% dari total data (477) digunakan sebagai data validasi, dan 5% lainnya (477) digunakan sebagai data *testing*. Dari *confusion matrix* tersebut, diketahui bahwa nilai rata-rata akurasi *training* adalah 37-76%. Nilai rata-rata *error training* $\geq 20\%$ (1.718) serta rata-rata *error* validasi dan *testing* $\geq 20\%$ (95). Hal ini menunjukkan bahwa model memiliki performa yang tidak cukup baik.

Visualisasi grafik lainnya adalah grafik *Receiver Operating Characteristic* (ROC). Fungsi dari grafik ROC ini adalah untuk mengetahui hasil evaluasi performa *machine learning*. Pada grafik ROC, terdapat dua warna grafik yaitu biru yang menunjukkan klasifikasi citra kelas 1 dan grafik hijau menunjukkan klasifikasi citra kelas 2. Kurva validasi dan *testing* terlihat jauh dari angka 1 dan mendekati garis *threshold*. Bahkan, kurva hasil validasi dan *testing* *run* 1, 4, 7, 8, dan 9 berada di bawah garis *threshold*. Dengan demikian, dapat disimpulkan bahwa performa *machine learning* (*Multilayer Perceptron* model GDM *hidden neuron* 10) tidak *perfect*.

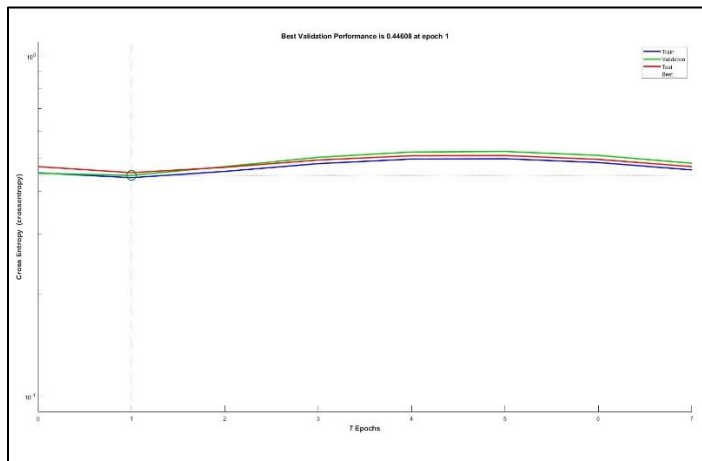
d. *Hidden Neuron 15*

Hasil akurasi *training* citra *body fluids* menggunakan model *Gradient Descent with Momentum* (GDM) *hidden neuron 15* dapat dilihat pada Tabel 4.34.

Tabel 4. 34 Hasil Akurasi *Training* Menggunakan Model GDM *Hidden Neuron 15*

<i>Run</i>	<i>Epoch</i>		MLP-TrainGDM Hidden Neuron 15			<i>Elapsed Time</i>
	<i>All</i>	<i>Best</i>	<i>Acc Training</i>	<i>Acc Validation</i>	<i>Acc Testing</i>	
1	7	1	39,1%	35,6%	39,2%	00:00:02
2	8	2	61,4%	63,3%	61,4%	00:00:02
3	7	1	59,8%	62,5%	61,2%	00:00:02
4	7	1	52,3%	48,0%	49,3%	00:00:02
5	7	1	43,6%	42,3%	44,2%	00:00:02
6	7	1	47,8%	45,7%	47,4%	00:00:02
7	7	1	41,7%	43,0%	38,2%	00:00:02
8	7	1	50,0%	50,1%	45,1%	00:00:02
9	1000	1000	78,4%	78,8%	76,7%	00:00:46
10	1000	1000	75,6%	75,9%	72,7%	00:00:46
Average			55,0%	54,5%	53,5%	00:00:11
Standar Deviasi			0,136614	0,147845	0,136587	

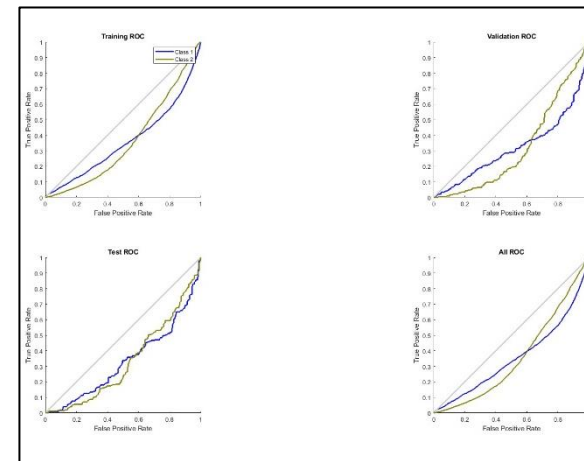
Berdasarkan Tabel 4.34 diketahui bahwa rata-rata akurasi *training* menggunakan model GDM dengan *hidden neuron 15* adalah 39-78%. Hasil akurasi *training* terbaik terdapat pada *run 9* dengan akurasi *training* sebesar 78,4%, akurasi validasi 78,8%, dan akurasi *testing* 76,7%. Akurasi *training* terbaik ini diperoleh pada *epoch* ke 1000 dari total 1000 *epoch* dengan waktu *training* selama 46 detik. Sementara itu, hasil akurasi *training* terendah terdapat pada *run 1* dengan akurasi *training* 39,1%, akurasi validasi 35,6%, dan akurasi *testing* 39,2%. Dari pengujian diatas, diperoleh hasil bahwa rata-rata akurasi *training* adalah 55,0%, akurasi validasi 54,5%, dan akurasi *testing* 53,5%. Rata-rata waktu *training* menggunakan model GDM *hidden neuron 15* adalah 11 detik. Visualisasi hasil *training* dapat dilihat pada Gambar 4.32.



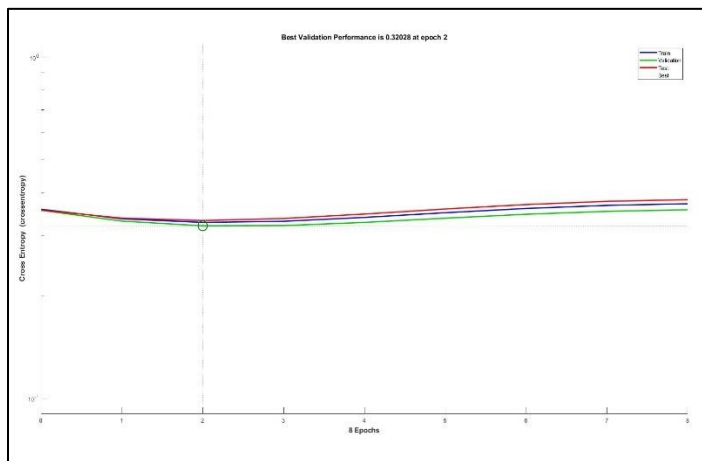
(a1)



(a2)



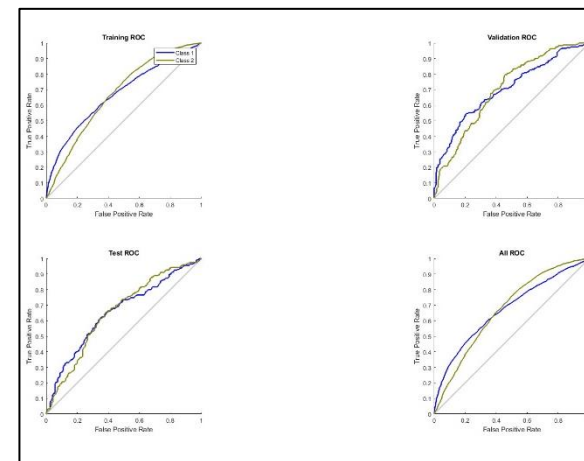
(a3)



(b1)

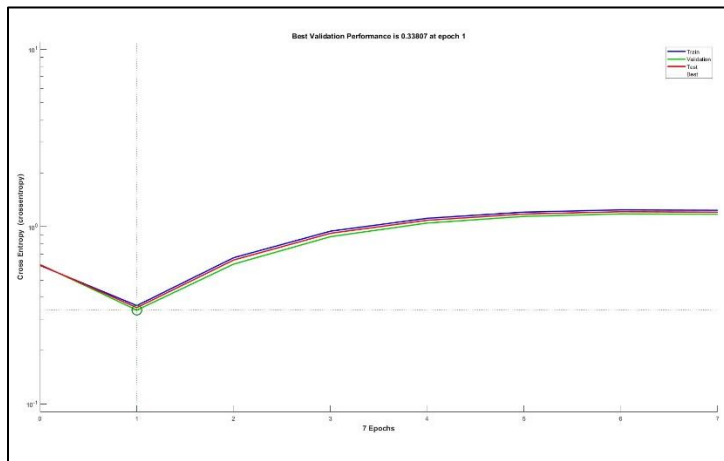


(b2)



(b3)

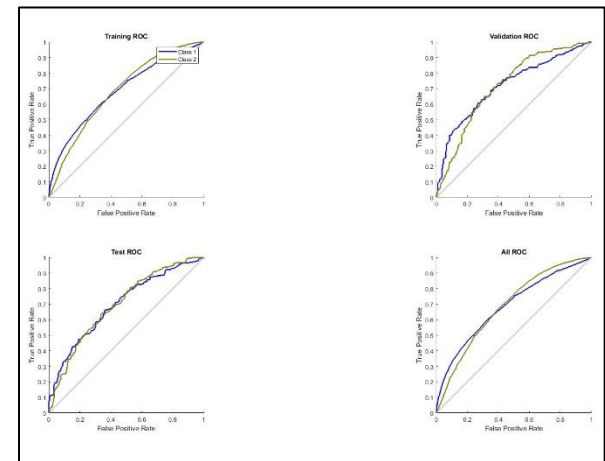
Gambar 4. 32 (a1) (b1) Performance Run 1 dan 2 Training GDM HN 15, (a2) (b2) Confusion Matrix Run 1 dan 2 Training GDM HN 15, (a3) (b3) ROC Run 1 dan 2 Training GDM HN 15



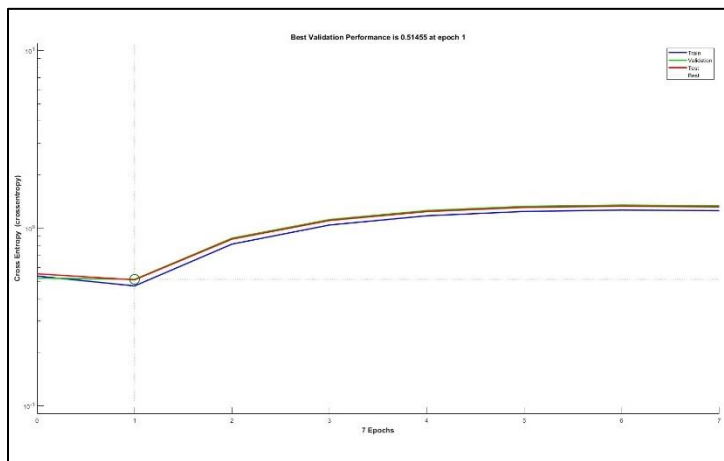
(c1)



(c2)



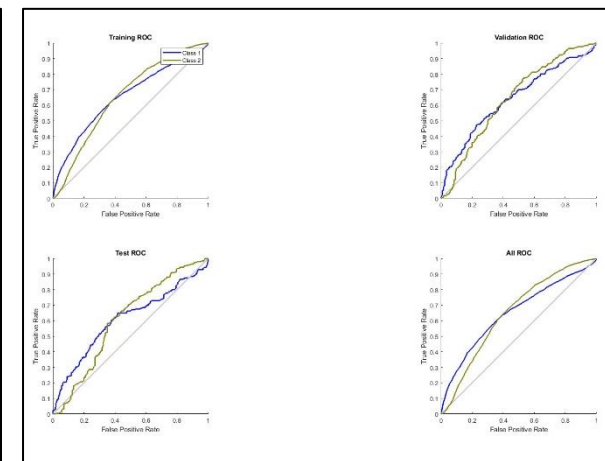
(c3)



(d1)

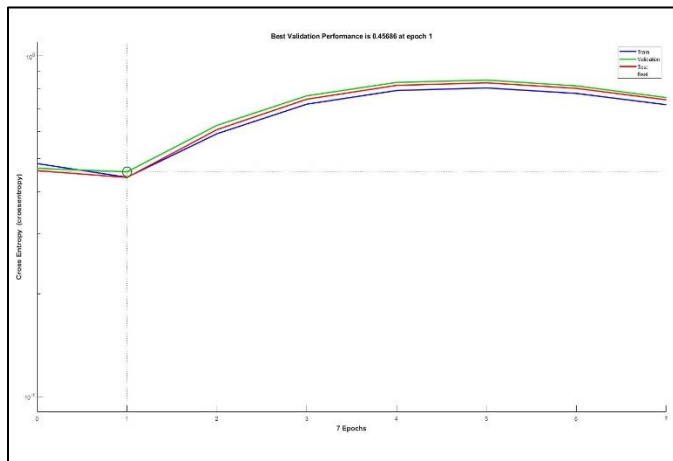


(d2)



(d3)

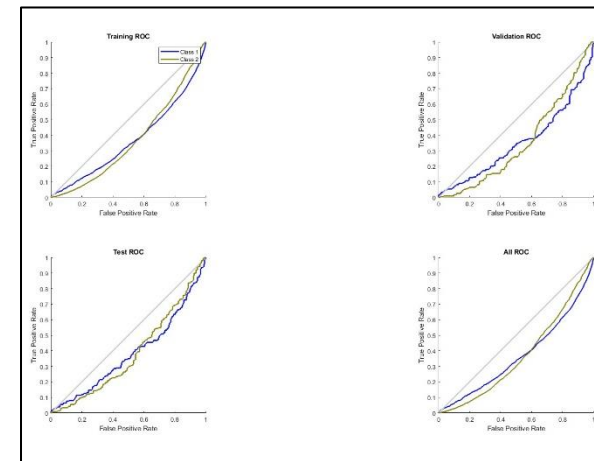
Gambar 4.32 (c1) (d1) Performance Run 3 dan 4 Training GDM HN 15, (c2) (d2) Confusion Matrix Run 3 dan 4 Training GDM HN 15, (c3) (d3) ROC Run 3 dan 4 Training GDM HN 15



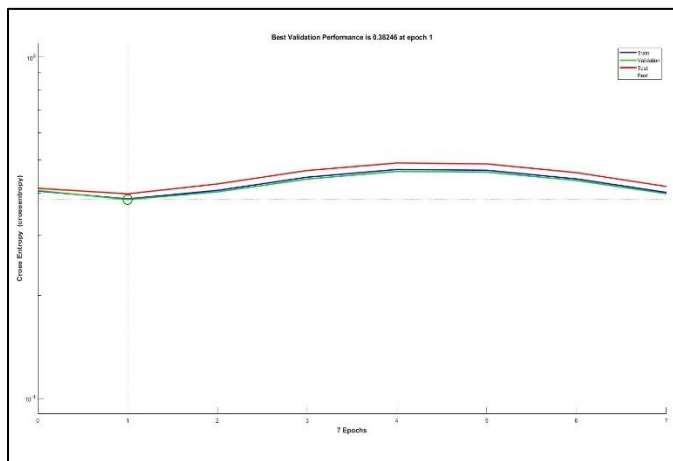
(e1)



(e2)



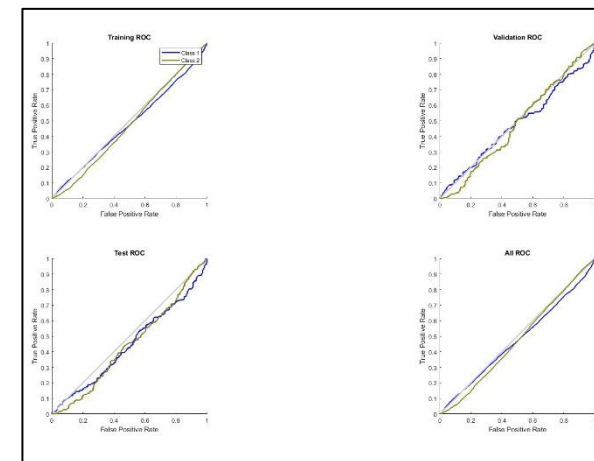
(e3)



(f1)

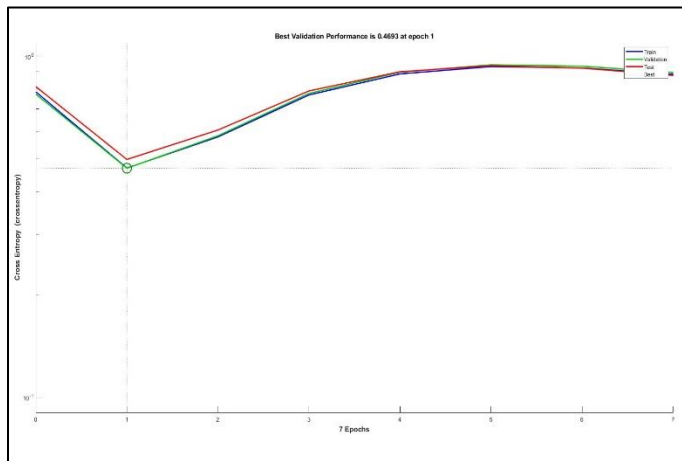


(f2)



(f3)

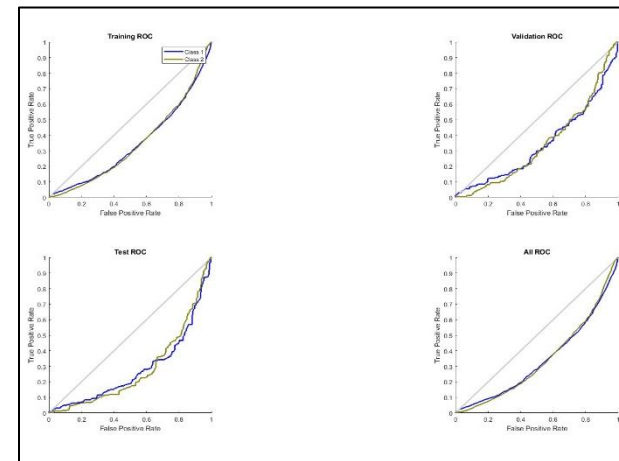
Gambar 4.32 (e1) (f1) Performance Run 5 dan 6 Training GDM HN 15, (e2) (f2) Confusion Matrix Run 5 dan 6 Training GDM HN 15, (e3) (f3) ROC Run 5 dan 6 Training GDM HN 15



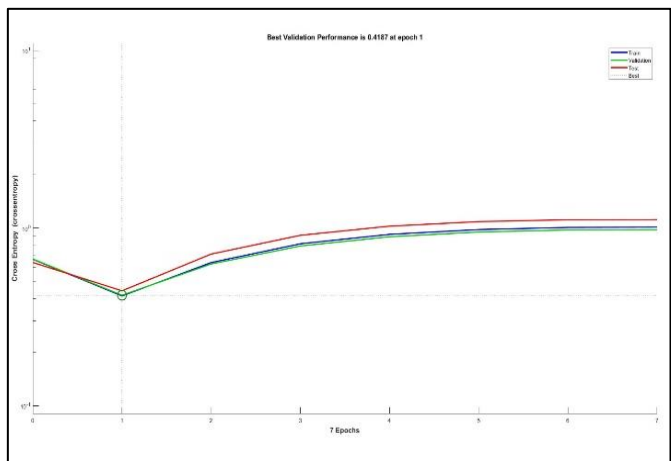
(g1)



(g2)



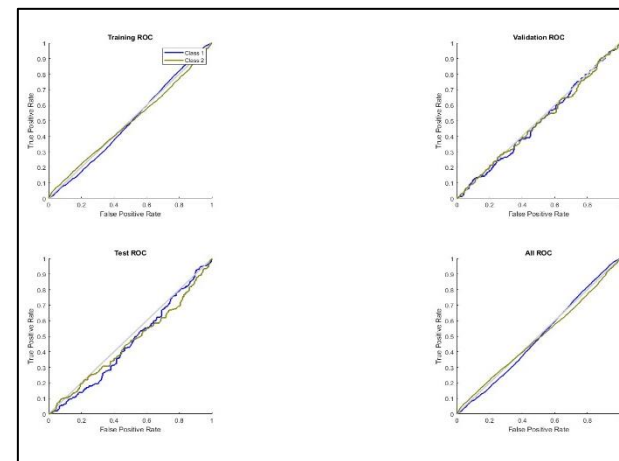
(g3)



(h1)

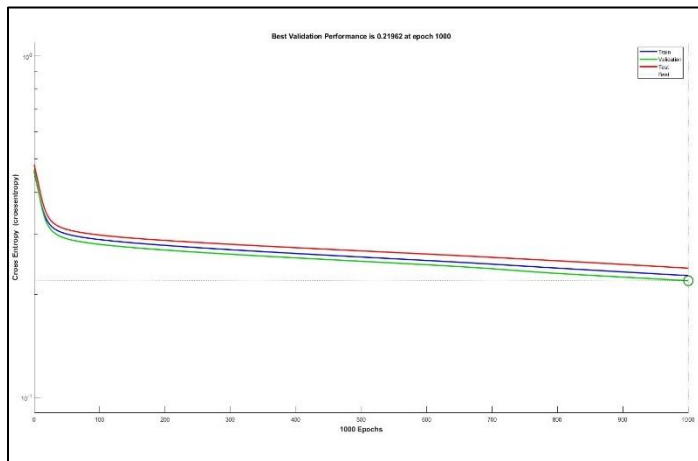


(h2)



(h3)

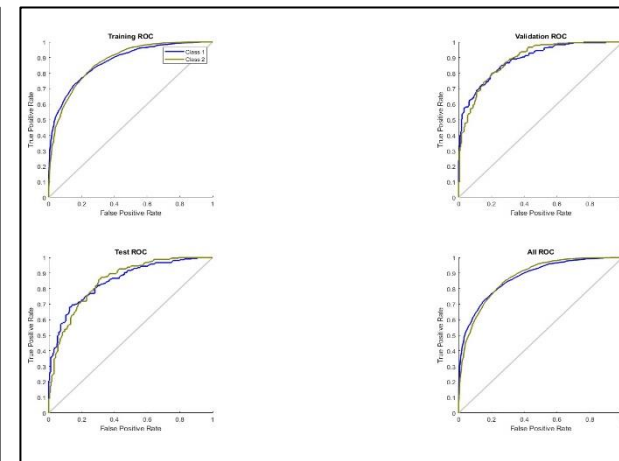
Gambar 4.32 (g1) (h1) Performance Run 7 dan 8 Training GDM HN 15, (g2) (h2) Confusion Matrix Run 7 dan 8 Training GDM HN 15, (g3) (h3) ROC Run 7 dan 8 Training GDM HN 15



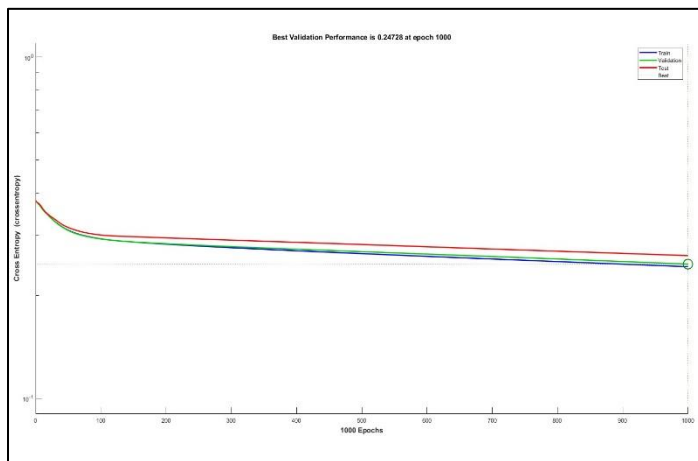
(i1)



(i2)



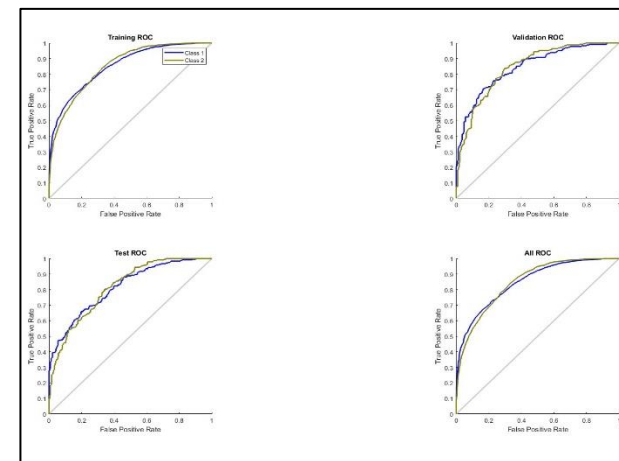
(i3)



(j1)



(j2)



(j3)

Gambar 4.32 (i1) (j1) Performance Run 9 dan 10 Training GDM HN 15, (i2) (j2) Confusion Matrix Run 9 dan 10 Training GDM HN 15, (i3) (j3) ROC Run 9 dan 10 Training GDM HN 15

Berdasarkan visualisasi hasil *training* di atas, dapat diketahui bahwa grafik *performance* terdiri dari 3 grafik, yaitu grafik biru merepresentasikan hasil *training*, grafik hijau merepresentasikan hasil validasi, dan grafik merah merepresentasikan hasil *testing*. Gambar 4.32 menampilkan sepuluh grafik *performane training* GDM hidden neuron 15 dimana grafik mengalami penurunan signifikan pada *epoch* 0-1 kecuali *run* 1, 2 dan 10. Namun, grafik mengalami kenaikan *error* mulai *epoch* 1 hingga *epoch* maksimal sedangkan *run* 1, 2, dan 10 tidak mengalami penurunan atau kenaikan signifikan dari *epoch* awal hingga *epoch* maksimalnya. Dari uraian di atas, dapat disimpulkan bahwa semakin rendah nilai entropi dari grafik (nilai *error*) maka semakin baik performa dari model.

Hasil *training* yang menunjukkan akurasi dari sistem ditunjukkan oleh *confusion matrix* dimana warna hijau merepresentasikan citra yang berhasil diklasifikasikan ke dalam kelas secara benar dan warna merah adalah citra yang gagal diklasifikasikan dalam kelasnya. Perbandingan data yang digunakan untuk *10-run cross validation* yaitu 90% dari total data (8.591) digunakan sebagai data *training*, 5% dari total data (477) digunakan sebagai data validasi, dan 5% lainnya (477) digunakan sebagai data *testing*. Dari *confusion matrix* tersebut, diketahui bahwa nilai rata-rata *error* hasil *training* $\geq 20\%$ (1.718) serta rata-rata *error* validasi dan *testing* $\geq 20\%$ (95). Hal tersebut menunjukkan performa model tidak cukup baik.

Visualisasi grafik lainnya adalah grafik *Receiver Operating Characteristic* (ROC). Fungsi dari grafik ROC ini adalah untuk mengetahui hasil evaluasi performa *machine learning*. Pada grafik ROC, terdapat dua warna grafik yaitu biru yang menunjukkan klasifikasi citra kelas 1 dan grafik hijau menunjukkan klasifikasi citra kelas 2. Kurva validasi dan *testing* berada pada angka 0,6 ke bawah dan mendekati garis *threshold*. Bahkan kurva validasi dan *testing* *run* 1, 5, 6, 7, dan 8 berada di bawah garis *threshold*. Dengan demikian, dapat disimpulkan bahwa performa *machine learning* (*Multilayer Perceptron* model GDM *hidden neuron* 15) menunjukkan performa tidak *perfect*.

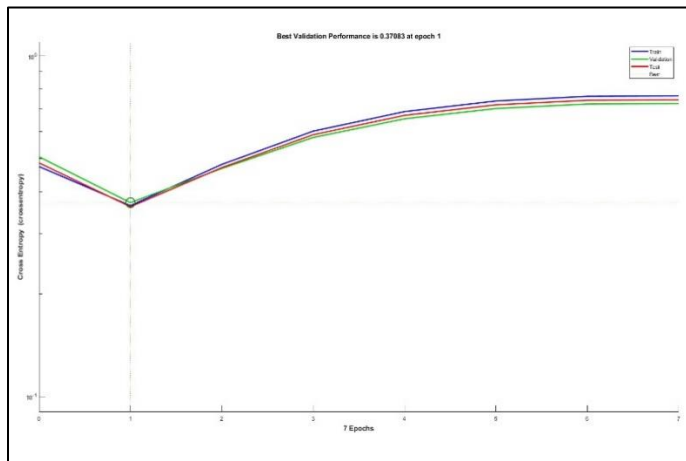
e. *Hidden Neuron 20*

Hasil akurasi *training* citra *body fluids* menggunakan model *Gradient Descent with Momentum* (GDM) *hidden neuron 20* dapat dilihat pada Tabel 4.35.

Tabel 4. 35 Hasil Akurasi *Training* Menggunakan Model GDM *Hidden Neuron 20*

<i>Run</i>	<i>Epoch</i>		MLP-TrainGDM <i>Hidden Neuron 20</i>			<i>Elapsed Time</i>
	<i>All</i>	<i>Best</i>	<i>Acc Training</i>	<i>Acc Validation</i>	<i>Acc Testing</i>	
1	7	1	56,1%	54,9%	57,7%	00:00:02
2	1000	1000	81,8%	79,0%	78,0%	00:00:54
3	8	2	61,1%	61,2%	59,5%	00:00:02
4	7	1	61,6%	58,7%	60,2%	00:00:02
5	7	1	49,8%	52,8%	48,4%	00:00:02
6	7	1	48,5%	50,9%	47,8%	00:00:02
7	7	1	57,5%	54,1%	56,8%	00:00:02
8	7	1	41,6%	43,8%	37,9%	00:00:02
9	8	2	37,6%	36,5%	33,1%	00:00:02
10	7	1	52,3%	52,6%	50,7%	00:00:02
Average			54,8%	54,5%	53,0%	00:00:07
Standar Deviasi			0,12315	0,111606	0,126189	

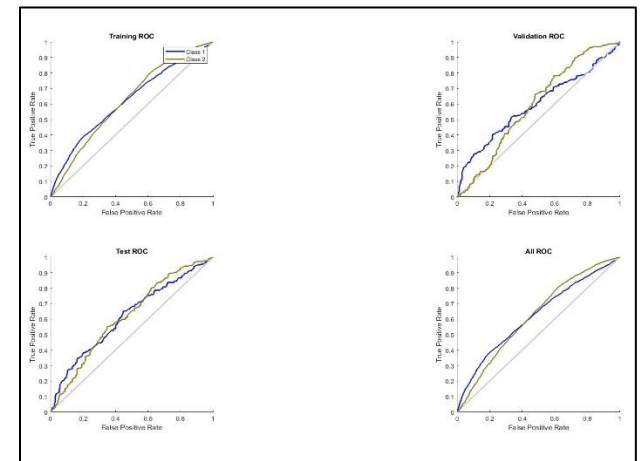
Berdasarkan Tabel 4.35 diketahui bahwa rata-rata akurasi *training* menggunakan model GDM dengan *hidden neuron 20* adalah 37-81%. Hasil akurasi *training* terbaik terdapat pada *run 2* dengan akurasi *training* sebesar 81,8%, akurasi validasi 79,0%, dan akurasi *testing* 78,0%. Akurasi *training* terbaik ini diperoleh pada *epoch* ke 1000 dari total 1000 *epoch* dengan waktu *training* selama 54 detik. Sementara itu, hasil akurasi *training* terendah terdapat pada *run 9* dengan akurasi *training* 37,6%, akurasi validasi 36,5%, dan akurasi *testing* 33,1%. Dari pengujian diatas, diperoleh hasil bahwa rata-rata akurasi *training* adalah 54,8%, akurasi validasi 54,5%, dan akurasi *testing* 53,0%. Rata-rata waktu *training* menggunakan model GDM *hidden neuron 20* adalah 7 detik. Visualisasi hasil *training* dapat dilihat pada Gambar 4.33.



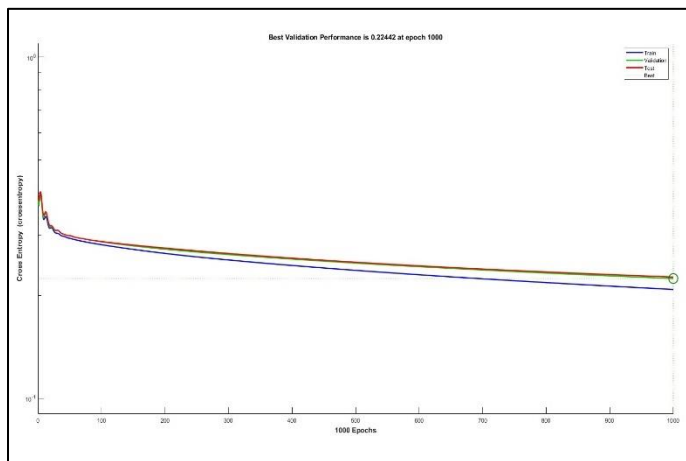
(a1)



(a2)



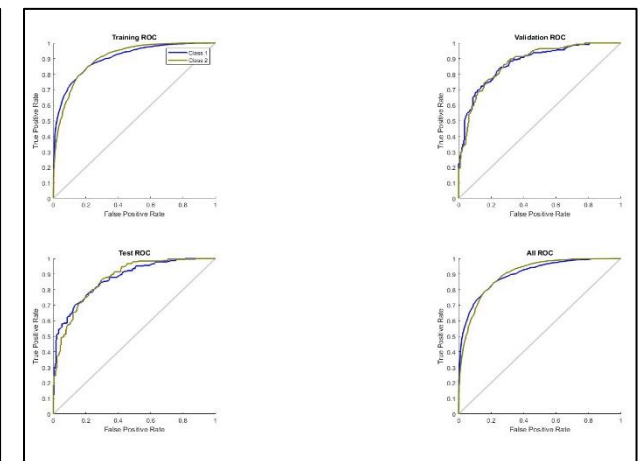
(a3)



(b1)

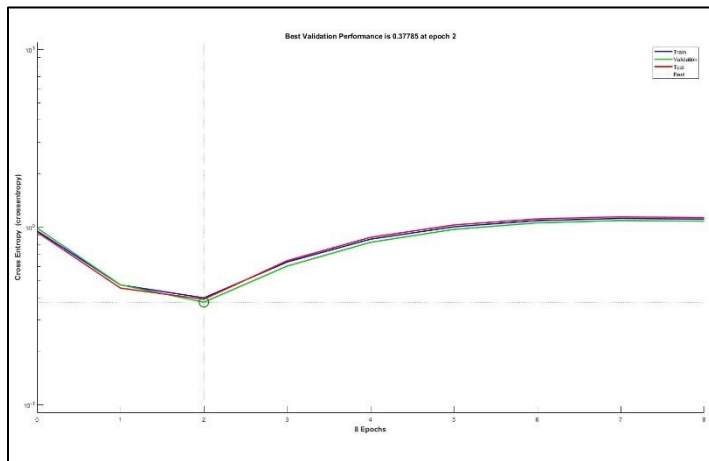


(b2)



(b3)

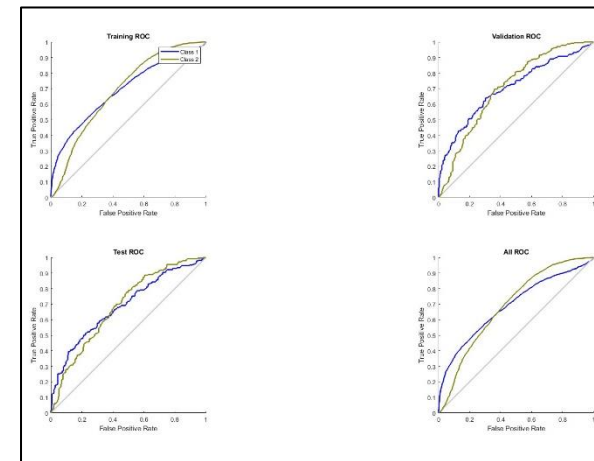
Gambar 4. 33 (a1) (b1) Performance Run 1 dan 2 Training GDM HN 20, (a2) (b2) Confusion Matrix Run 1 dan 2 Training GDM HN 20, (a3) (b3) ROC Run 1 dan 2 Training GDM HN 20



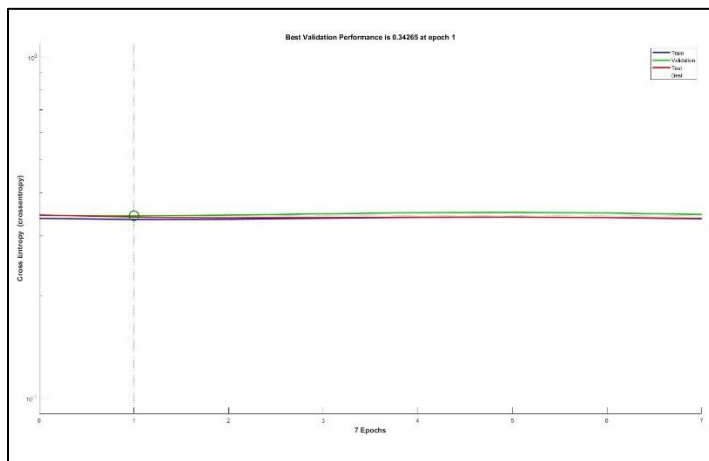
(c1)



(c2)



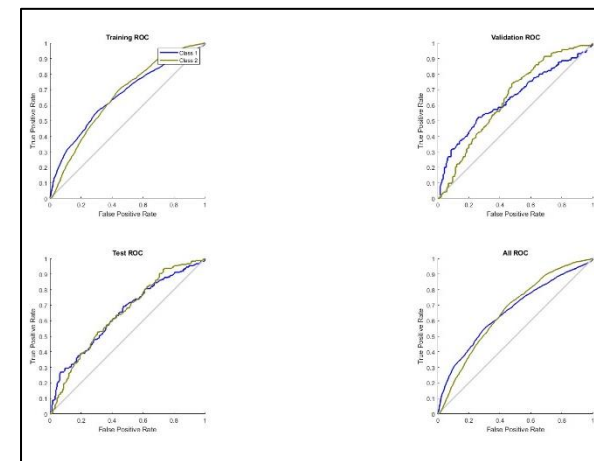
(c3)



(d1)

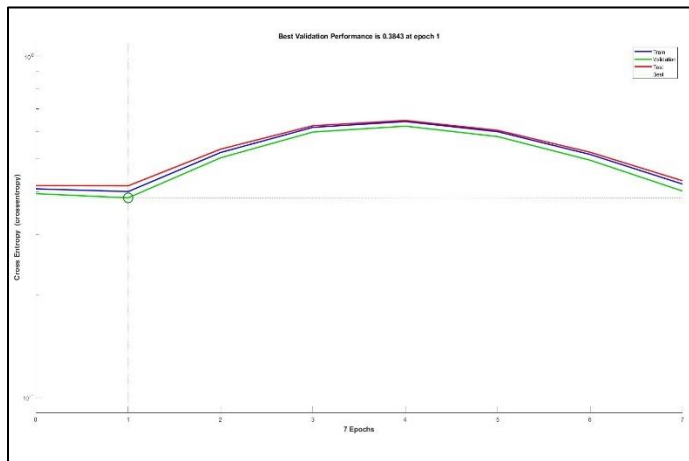


(d2)



(d3)

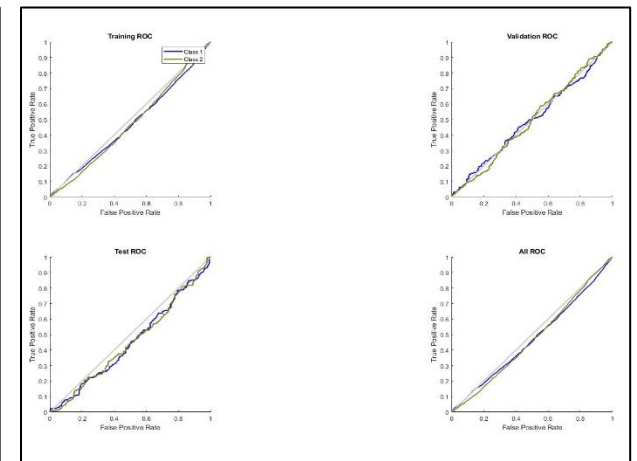
Gambar 4.33 (c1) (d1) Performance Run 3 dan 4 Training GDM HN 20, (c2) (d2) Confusion Matrix Run 3 dan 4 Training GDM HN 20, (c3) (d3) ROC Run 3 dan 4 Training GDM HN 20



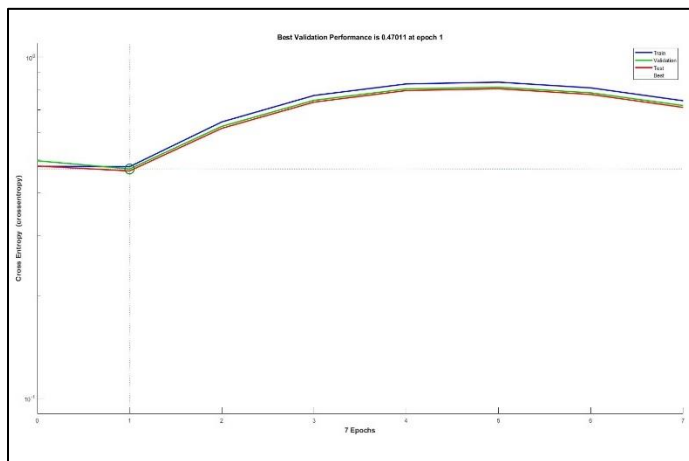
(e1)



(e2)



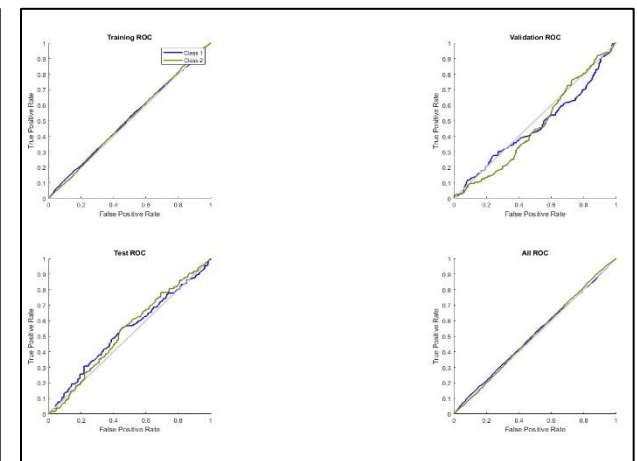
(e3)



(f1)

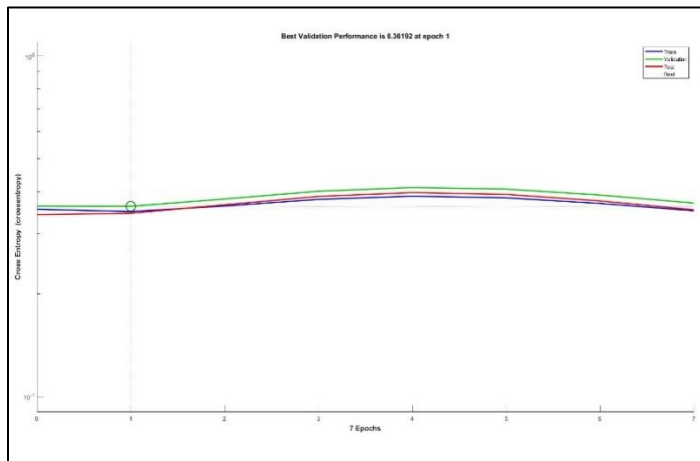


(f2)



(f3)

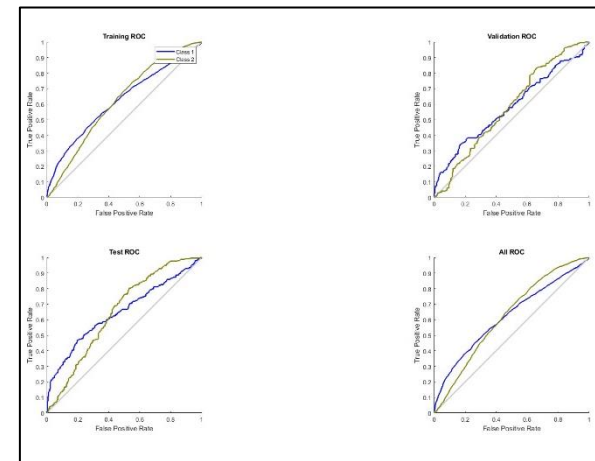
Gambar 4.33 (e1) (f1) Performance Run 5 dan 6 Training GDM HN 20, (e2) (f2) Confusion Matrix Run 5 dan 6 Training GDM HN 20, (e3) (f3) ROC Run 5 dan 6 Training GDM HN 20



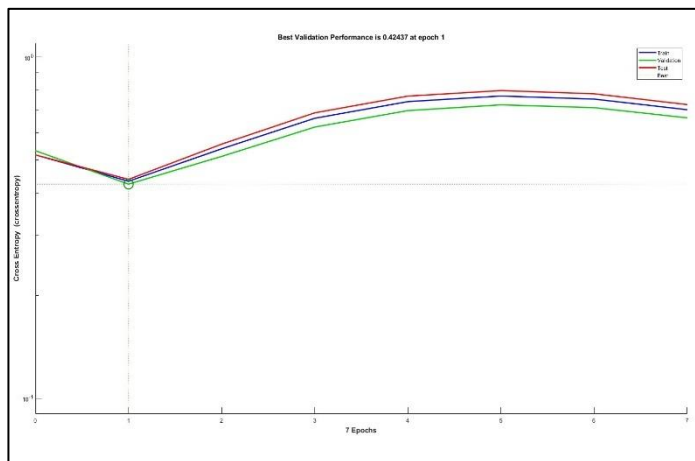
(g1)



(g2)



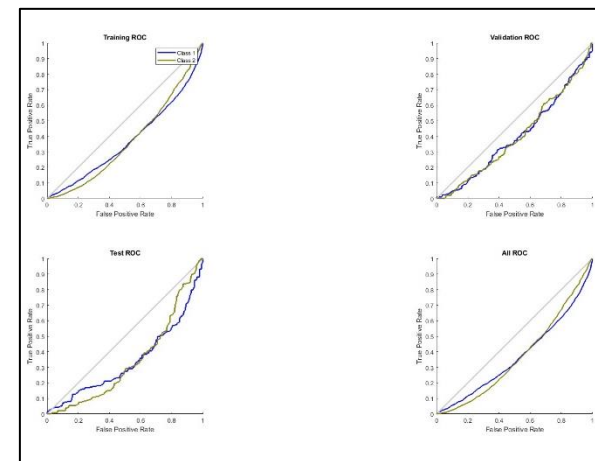
(g3)



(h1)

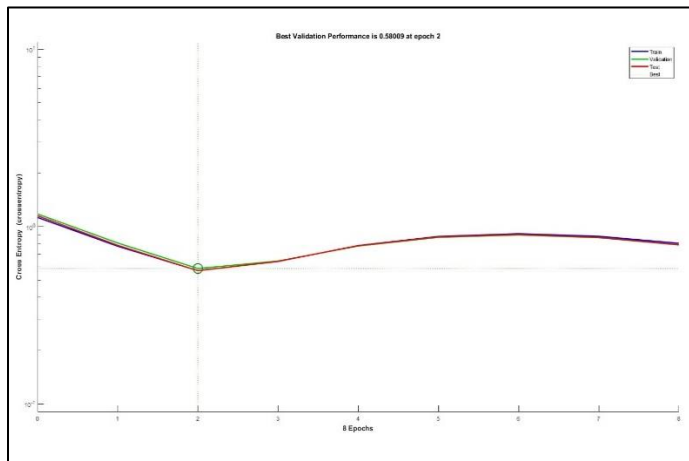


(h2)



(h3)

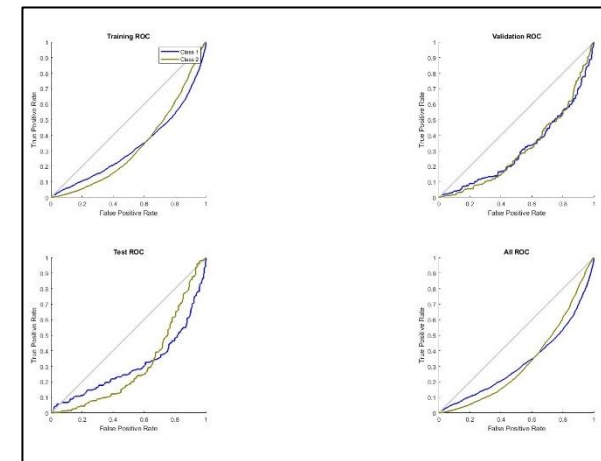
Gambar 4.33 (g1) (h1) Performance Run 7 dan 8 Training GDM HN 20, (g2) (h2) Confusion Matrix Run 7 dan 8 Training GDM HN 20, (g3) (h3) ROC Run 7 dan 8 Training GDM HN 20



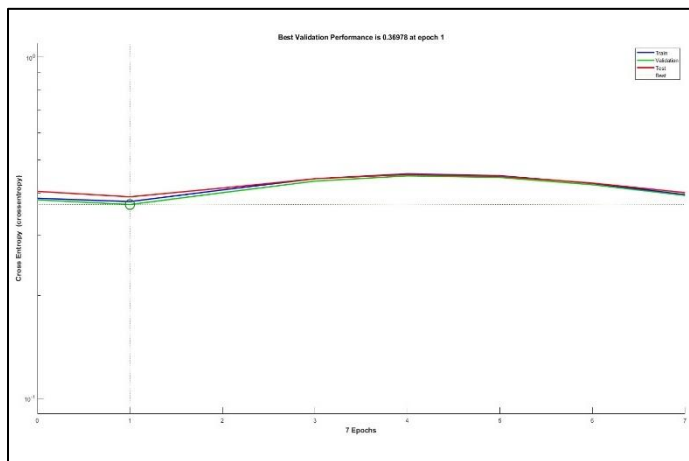
(i1)



(i2)



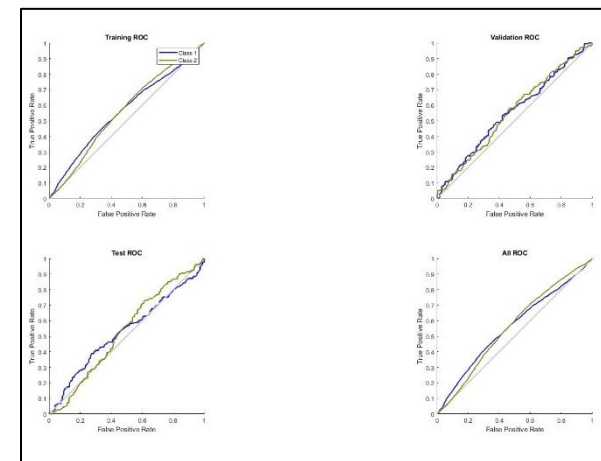
(i3)



(j1)



(j2)



(j3)

Gambar 4.33 (i1) (j1) Performance Run 9 dan 10 Training GDM HN 20, (i2) (j2) Confusion Matrix Run 9 dan 10 Training GDM HN 20, (i3) (j3) ROC Run 9 dan 10 Training GDM HN 20

Berdasarkan visualisasi hasil *training* di atas, dapat diketahui bahwa grafik *performance* terdiri dari 3 grafik, yaitu grafik biru merepresentasikan hasil *training*, grafik hijau merepresentasikan hasil validasi, dan grafik merah merepresentasikan hasil *testing*. Gambar 4.33 menampilkan sepuluh grafik *performane training* GDM hidden neuron 20 dimana grafik mengalami penurunan signifikan pada *epoch* 0-1 kecuali *run* 2, 4, dan 7. Namun, grafik mengalami kenaikan *error* mulai *epoch* 1 hingga *epoch* maksimal sedangkan *run* 2, 4, dan 7 tidak mengalami penurunan atau kenaikan signifikan dari *epoch* awal hingga *epoch* maksimalnya. Dari uraian di atas, dapat disimpulkan bahwa semakin rendah nilai entropi dari grafik (nilai *error*) maka semakin baik performa dari model.

Hasil *training* yang menunjukkan akurasi dari sistem ditunjukkan oleh *confusion matrix* dimana warna hijau merepresentasikan citra yang berhasil diklasifikasikan ke dalam kelas secara benar dan warna merah adalah citra yang gagal diklasifikasikan dalam kelasnya. Perbandingan data yang digunakan untuk *10-run cross validation* yaitu 90% dari total data (8.591) digunakan sebagai data *training*, 5% dari total data (477) digunakan sebagai data validasi, dan 5% lainnya (477) digunakan sebagai data *testing*. Dari *confusion matrix* tersebut, diketahui bahwa nilai *error* hasil *training* $\geq 20\%$ (1,718) serta *error* validasi dan *testing* $\geq 20\%$. (95). Hal tersebut menunjukkan bahwa performa model tidak cukup baik.

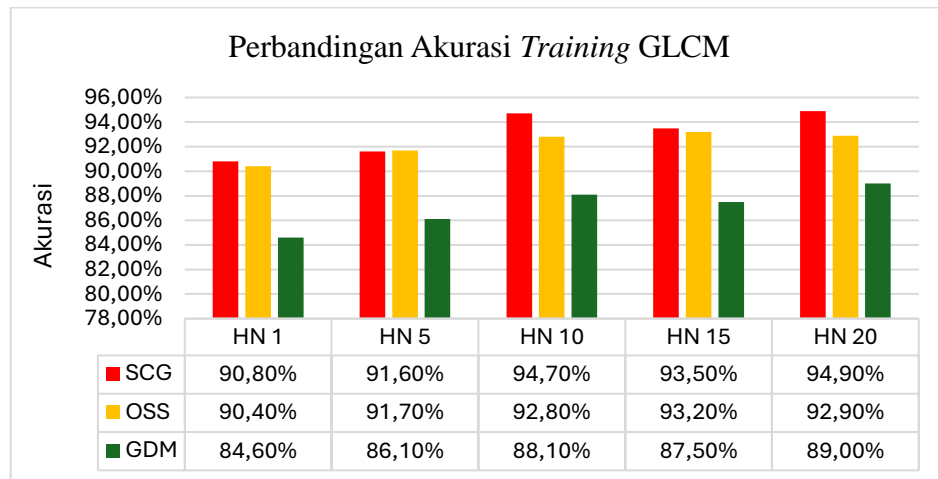
Visualisasi grafik lainnya adalah grafik *Receiver Operating Characteristic* (ROC). Fungsi dari grafik ROC ini adalah untuk mengetahui hasil evaluasi performa *machine learning*. Pada grafik ROC, terdapat dua warna grafik yaitu biru yang menunjukkan klasifikasi citra kelas 1 dan grafik hijau menunjukkan klasifikasi citra kelas 2. Kurva validasi dan *testing* terlihat jauh dari angka 1 kecuali *run* 2. Kurva *run* 1, 3, 4, dan 7 berada di atas garis *threshold* meskipun jauh dari angka 1 sedangkan kurva *run* 5, 6, 8, 9, dan 10 berada pada garis *threshold* bahkan di bawahnya. Dengan demikian, dapat disimpulkan bahwa performa *machine learning* (*Multilayer Perceptron* model GDM hidden neuron 20) menunjukkan performa tidak *perfect*.

4.4 Hasil Perbandingan Performa Terbaik

4.4.1 *Gray Level Co-occurrence Matrix (GLCM)*

a. Berdasarkan Akurasi Tertinggi

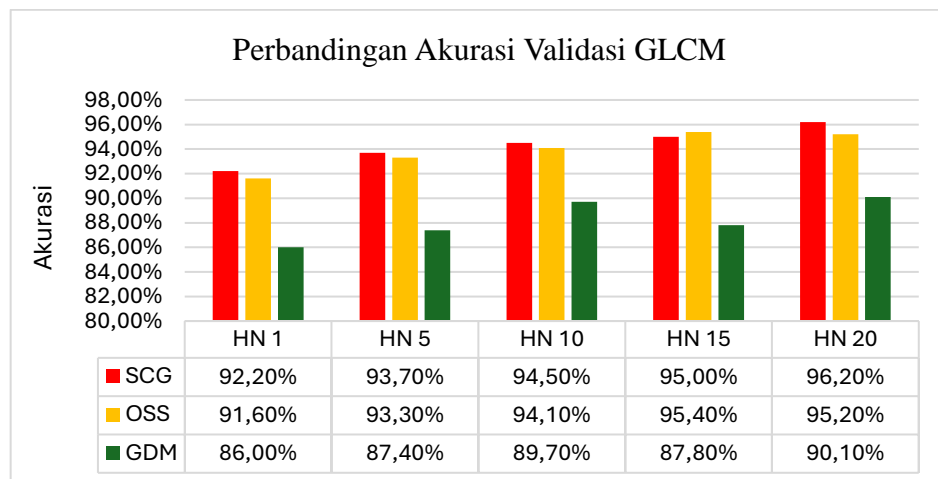
a.1 *Training*



Gambar 4. 34 Perbandingan Akurasi *Training* Tertinggi GLCM

Berdasarkan Gambar 4.34 diketahui bahwa *training* menggunakan metode GLCM menghasilkan akurasi yang baik. Dari lima percobaan variasi jumlah *hidden neuron*, model *Scaled Conjugate Gradient Backpropagation (SCG)* *hidden neuron 20* menghasilkan akurasi tertinggi, yaitu 94,9%.

a.2 *Validasi*

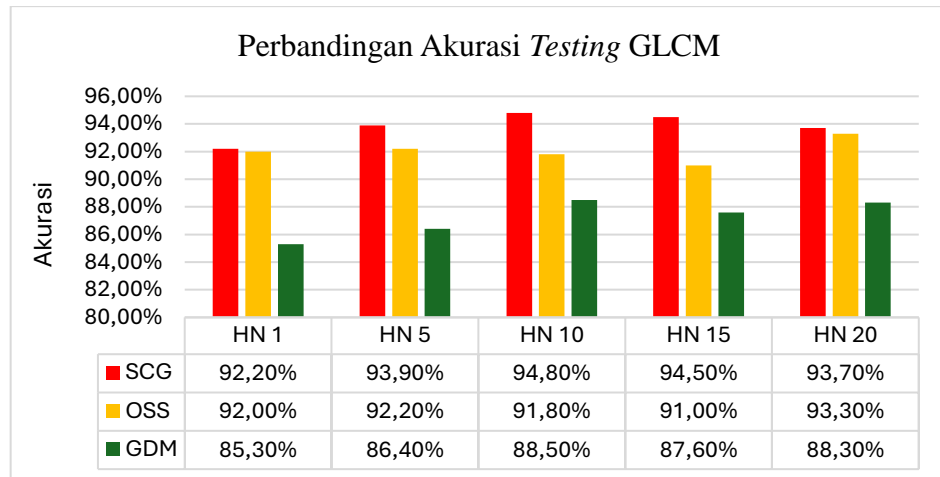


Gambar 4. 35 Perbandingan Akurasi *Validasi* Tertinggi GLCM

Berdasarkan Gambar 4.35 diketahui bahwa *validasi* menggunakan metode GLCM menghasilkan akurasi yang baik. Dari lima percobaan variasi jumlah

hidden neuron, model *Scaled Conjugate Gradient Backpropagation* (SCG) *hidden neuron* 20 menghasilkan akurasi tertinggi, yaitu 96,2%.

a.3 *Testing*

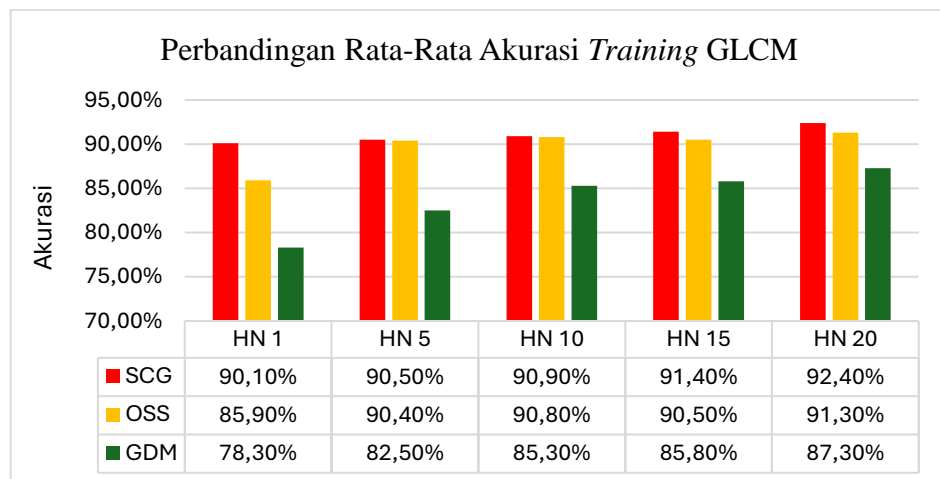


Gambar 4. 36 Perbandingan Akurasi *Testing* Tertinggi GLCM

Berdasarkan Gambar 4.36 diketahui bahwa *testing* menggunakan metode GLCM menghasilkan akurasi yang baik. Dari lima percobaan variasi jumlah *hidden neuron*, model *Scaled Conjugate Gradient Backpropagation* (SCG) *hidden neuron* 10 menghasilkan akurasi tertinggi, yaitu 94,8%.

b. Berdasarkan Nilai Rata-Rata Akurasi

a.1 *Training*

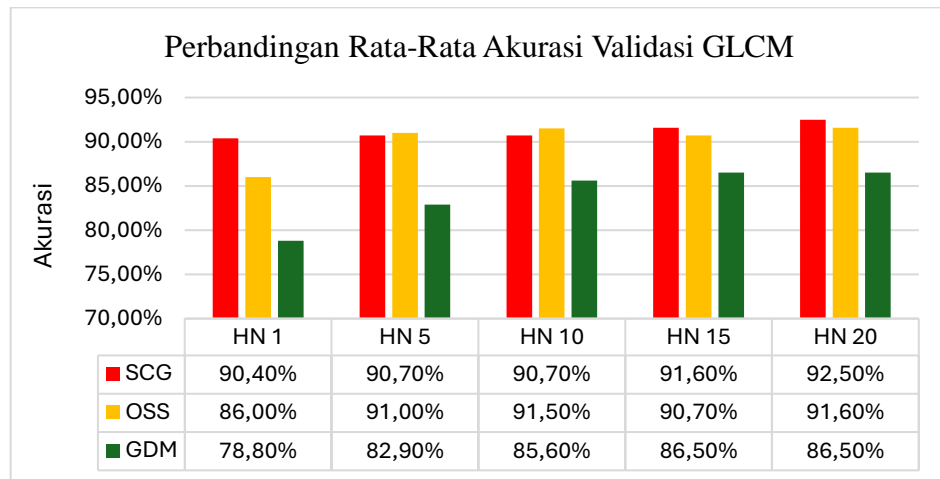


Gambar 4. 37 Perbandingan Akurasi *Training* Tertinggi GLCM

Berdasarkan Gambar 4.37 diketahui bahwa *training* menggunakan metode GLCM menghasilkan akurasi yang baik. Dari lima percobaan variasi jumlah

hidden neuron, nilai rata-rata *training* terbaik terdapat pada model *Scaled Conjugate Gradient Backpropagation (SCG) hidden neuron 20* dengan nilai sebesar 92,4%.

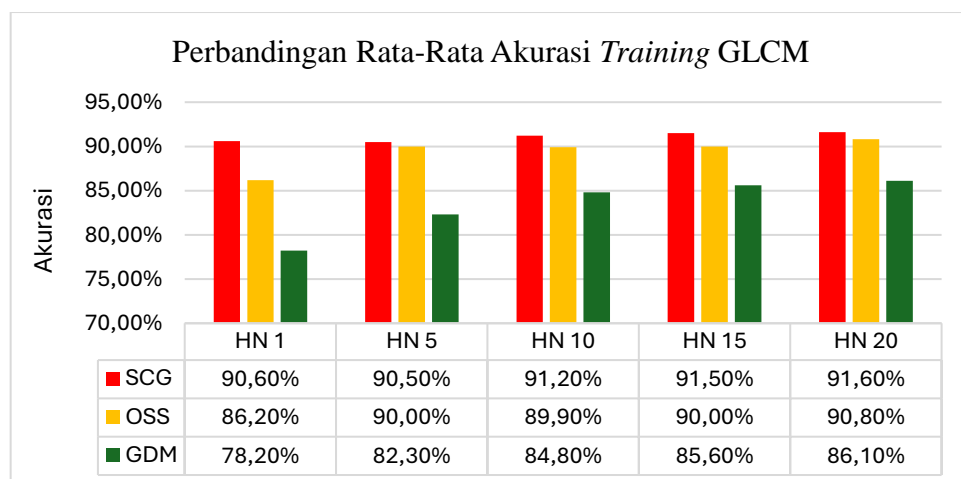
a.2 Validasi



Gambar 4. 38 Perbandingan Akurasi Validasi Tertinggi GLCM

Berdasarkan Gambar 4.38 diketahui bahwa validasi menggunakan metode GLCM menghasilkan akurasi yang baik. Dari lima percobaan variasi jumlah *hidden neuron*, nilai rata-rata validasi terbaik terdapat pada model *Scaled Conjugate Gradient Backpropagation (SCG) hidden neuron 20* dengan nilai sebesar 92,5%.

a.1 Testing



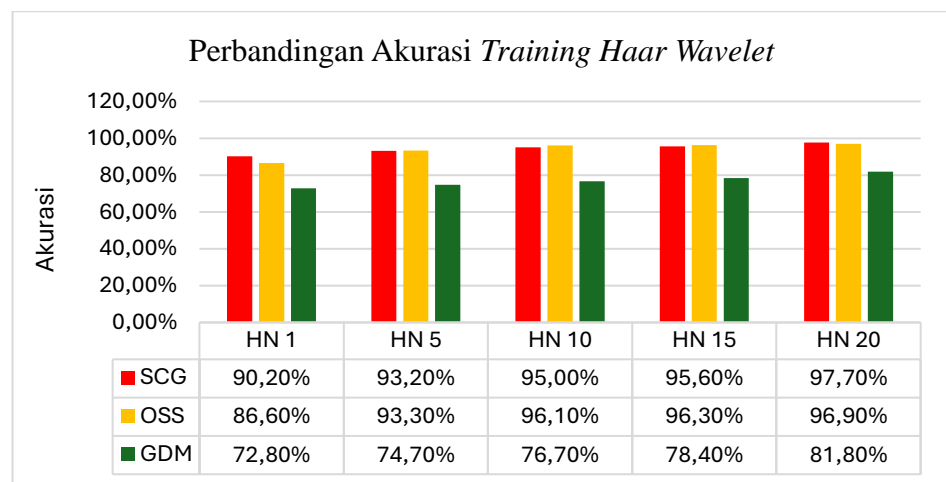
Gambar 4. 39 Perbandingan Akurasi Testing Tertinggi GLCM

Berdasarkan Gambar 4.39 diketahui bahwa *testing* menggunakan metode GLCM menghasilkan akurasi yang baik. Dari lima percobaan variasi jumlah *hidden neuron*, nilai rata-rata *testing* terbaik terdapat pada model *Scaled Conjugate Gradient Backpropagation* (SCG) *hidden neuron* 20 dengan nilai sebesar 91,6%.

4.4.2 Haar Wavelet

a. Berdasarkan Akurasi Tertinggi

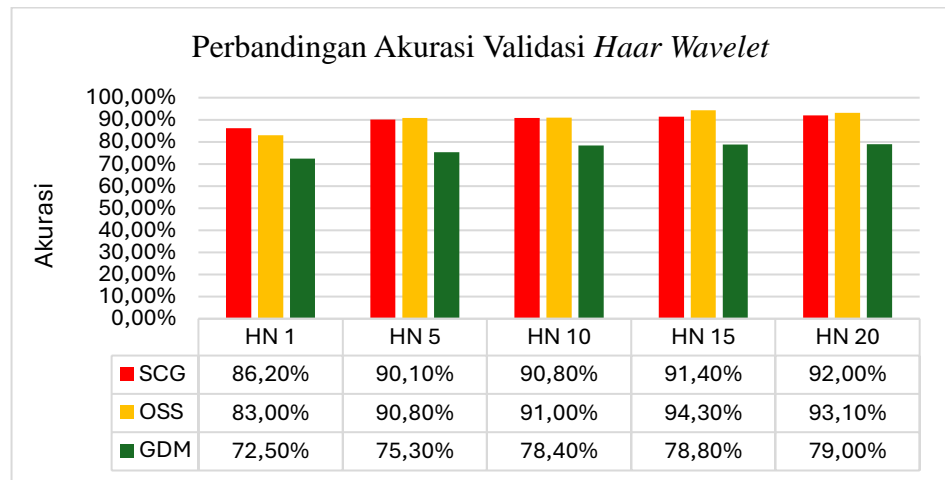
a.1 Training



Gambar 4. 40 Perbandingan Akurasi Training Tertinggi Haar Wavelet

Berdasarkan Gambar 4.40 diketahui bahwa *training* menggunakan metode *Haar Wavelet* menghasilkan akurasi yang baik. Dari lima percobaan variasi jumlah *hidden neuron*, model *Scaled Conjugate Gradient Backpropagation* (SCG) *hidden neuron* 20 menghasilkan akurasi tertinggi, yaitu 97,7%.

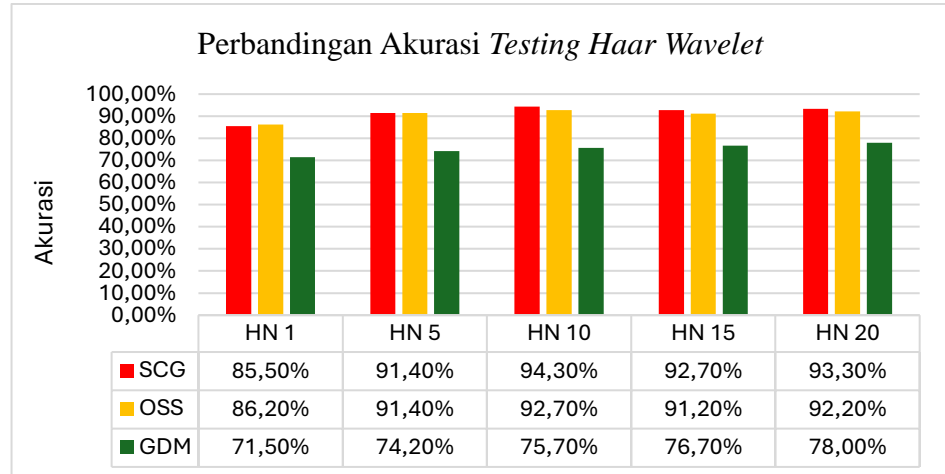
a.2 Validasi



Gambar 4. 41 Perbandingan Akurasi Validasi Tertinggi *Haar Wavelet*

Berdasarkan Gambar 4.41 diketahui bahwa validasi menggunakan metode *Haar Wavelet* menghasilkan akurasi yang baik. Dari lima percobaan variasi jumlah *hidden neuron*, model *One Step Secant Backpropagation* (OSS) *hidden neuron* 15 menghasilkan akurasi tertinggi, yaitu 94,3%.

a.3 Testing

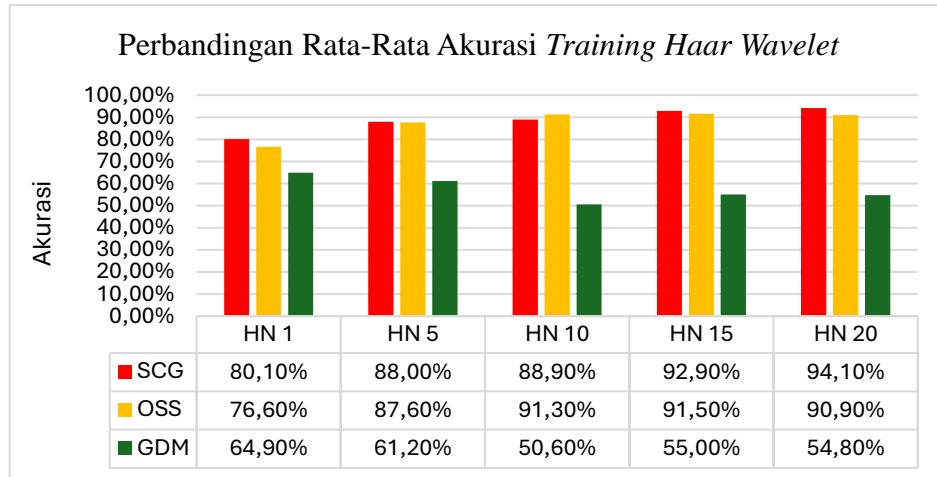


Gambar 4. 42 Perbandingan Akurasi *Testing Tertinggi Haar Wavelet*

Berdasarkan Gambar 4.42 diketahui bahwa *testing* menggunakan metode *Haar Wavelet* menghasilkan akurasi yang baik. Dari lima percobaan variasi jumlah *hidden neuron*, model *Scaled Conjugate Gradient Backpropagation* (SCG) *hidden neuron* 10 menghasilkan akurasi tertinggi, yaitu 94,3%.

b. Berdasarkan Nilai Rata-rata Akurasi

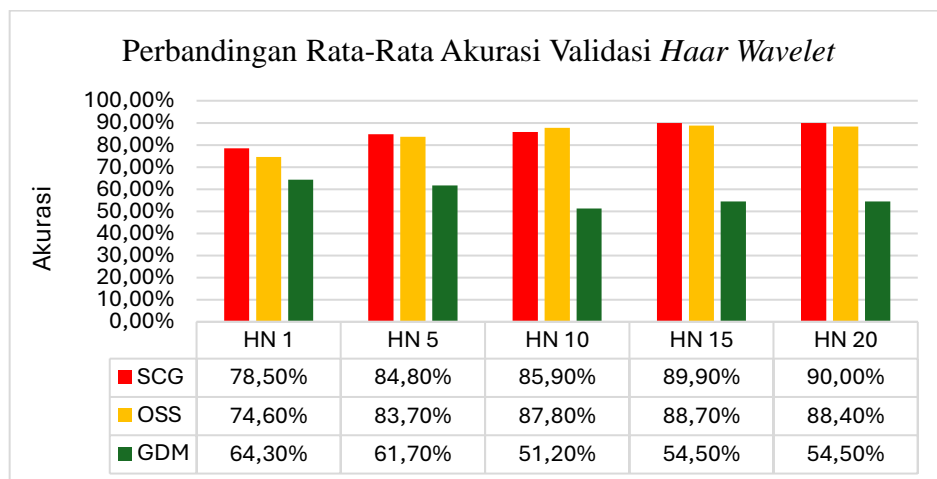
a.1 *Training*



Gambar 4. 43 Perbandingan Akurasi *Training* Tertinggi *Haar Wavelet*

Berdasarkan Gambar 4.43 diketahui bahwa *training* menggunakan metode *Haar Wavelet* menghasilkan akurasi yang baik. Dari lima percobaan variasi jumlah *hidden neuron*, nilai rata-rata *training* terbaik terdapat pada model *Scaled Conjugate Gradient Backpropagation (SCG)* *hidden neuron* 20 dengan nilai sebesar 94,1%.

a.2 *Validasi*

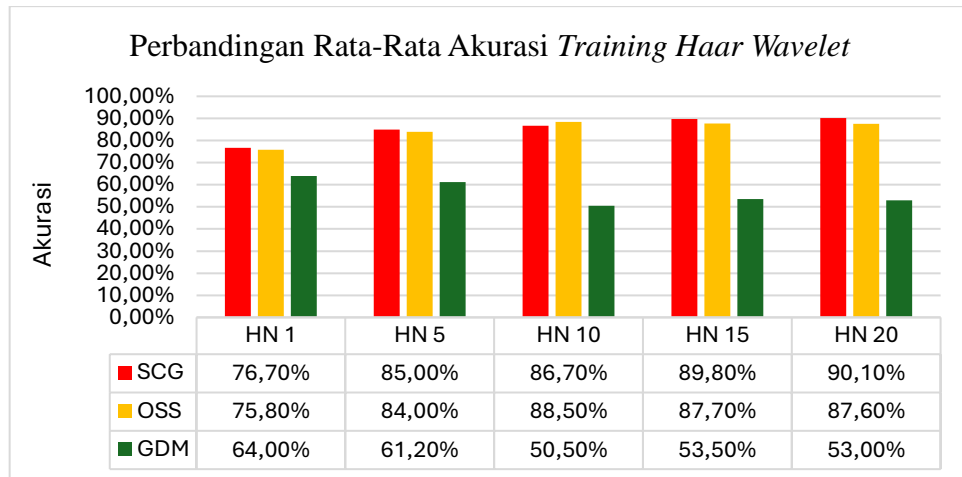


Gambar 4. 44 Perbandingan Akurasi *Validasi* Tertinggi *Haar Wavelet*

Berdasarkan Gambar 4.44 diketahui bahwa *validasi* menggunakan metode *Haar Wavelet* menghasilkan akurasi yang baik. Dari lima percobaan variasi jumlah *hidden neuron*, nilai rata-rata *validasi* terbaik terdapat pada model

Scaled Conjugate Gradient Backpropagation (SCG) hidden neuron 20 dengan nilai sebesar 90,0%.

a.1 *Testing*



Gambar 4. 45 Perbandingan Akurasi *Testing* Tertinggi Haar Wavelet

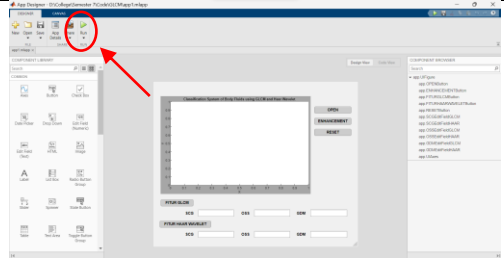
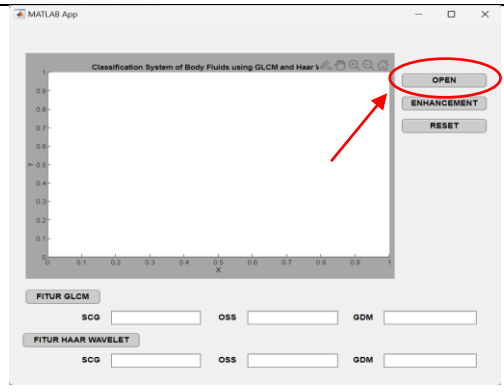
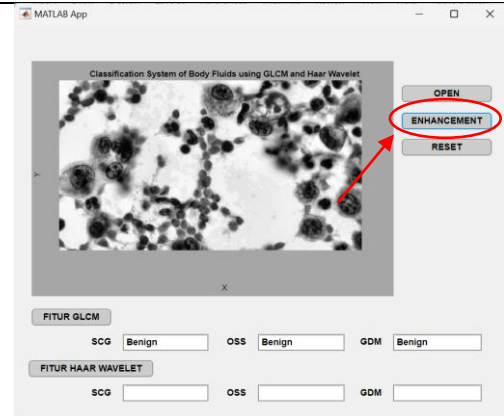
Berdasarkan Gambar 4.45 diketahui bahwa *testing* menggunakan metode Haar Wavelet menghasilkan akurasi yang baik. Dari lima percobaan variasi jumlah *hidden neuron*, nilai rata-rata *testing* terbaik terdapat pada model *Scaled Conjugate Gradient Backpropagation (SCG) hidden neuron 20* dengan nilai sebesar 90,10%.

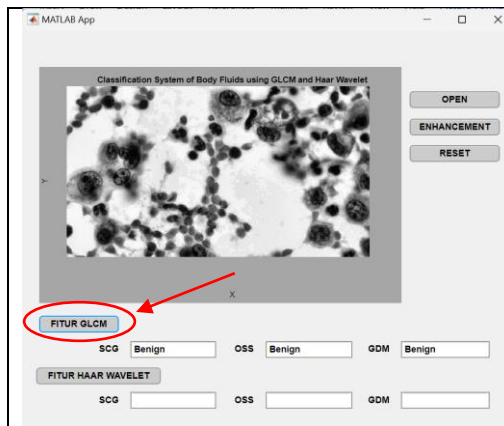
Berdasarkan beberapa tabel perbandingan hasil akurasi di atas, dapat diketahui bahwa hasil *training*, validasi, dan *testing* terbaik menggunakan metode *Gray Level Co-Occurrence Matrix (GLCM)* dihasilkan oleh model *Scaled Conjugate Gradient Backpropagation (SCG)*. Begitu juga dengan hasil *training*, validasi, dan *testing* terbaik menggunakan metode Haar Wavelet dihasilkan oleh model *Scaled Conjugate Gradient Backpropagation (SCG)*.

4.5 Implementasi APPS

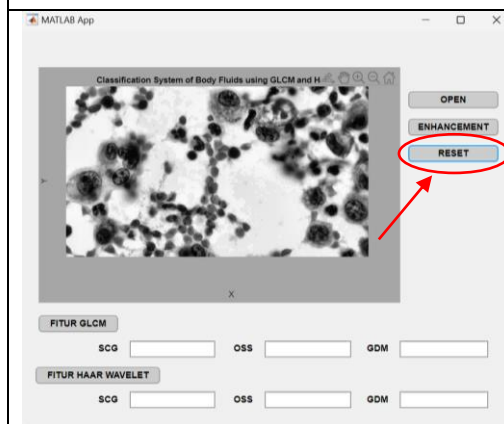
Hasil penelitian tentang sistem klasifikasi citra kanker *Adenocarcinoma body fluids* berbasis *neural network* yang menggunakan metode ekstraksi *Gray Level Co-occurrence Matrix* (GLCM) dan *Haar Wavelet* serta metode klasifikasi *Multilayer Perceptron* (MLP) dapat diimplementasikan pada APPS MATLAB. Tabel 4.36 menunjukkan contoh penggunaan APPS tersebut.

Tabel 4. 36 Contoh Pengimplementasian APPS Matlab

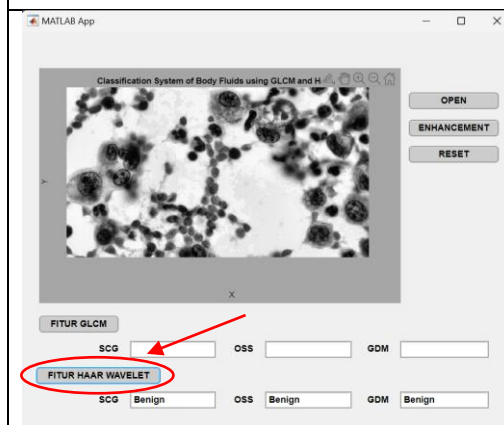
	<p>Buka APPS Matlab yang telah dibuat kemudian klik “RUN”</p>
	<p>Klik <i>button</i> “OPEN” untuk input <i>images</i></p>
	<p>Klik <i>button</i> “ENHANCEMENT” untuk proses <i>enhancing images</i></p>



Klik *button* “FITUR GLCM” untuk mengetahui hasil klasifikasi melalui metode ekstraksi GLCM



Klik *button* “RESET” untuk menghapus hasil klasifikasi sebelumnya



Klik *button* “FITUR HAAR WAVELET” untuk mengetahui hasil klasifikasi melalui metode ekstraksi *Haar Wavelet*

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan penelitian tentang klasifikasi kanker *Adenocarcinoma* melalui citra *body fluids* berbasis *Neural Network* yang telah dilakukan, dapat disimpulkan sebagai berikut.

1. Metode *Gray Level Co-occurrence Matrix* (GLCM) dan *Haar Wavelet* mampu mengekstraksi fitur citra *body fluids* yang terdiri dari dua kelas, yaitu *Malignant* dan *Benign*. Hasil ekstraksi diklasifikasikan menggunakan metode *Multilayer Perceptron* (MLP) dengan tiga model, yaitu *Scaled Conjugate Gradient Backpropagation* (SCG), *One Step Secant Backpropagation* (OSS), dan *Gradient Descent with Momentum* (GDM) dengan *hidden neuron* 1, 5, 10, 15, dan 20 sehingga menghasilkan grafik *performance*, *confusion matrix*, dan grafik *Receiver Operating Characteristic* (ROC).
2. Hasil rata-rata *training* menggunakan metode *Haar Wavelet* menunjukkan tingkat akurasi yang lebih tinggi dibandingkan GLCM dengan nilai akurasi sebesar 94,1% oleh model SCG *hidden neuron* 20 dalam waktu *run* 12 detik sedangkan rata-rata akurasi GLCM adalah 92,4% oleh model SCG *hidden neuron* 20 dalam waktu *run* 3 detik.

5.2 Saran

Setelah dilakukan penelitian di atas, terdapat beberapa saran yang dapat digunakan sebagai masukan untuk penelitian selanjutnya, yaitu

1. Jumlah citra yang digunakan sebagai dataset lebih banyak agar hasil yang dihasilkan lebih akurat.
2. Proses preprocessing dapat dikembangkan atau ditambahkan sehingga dapat menghasilkan variasi citra yang lebih banyak serta menghasilkan akurasi yang lebih baik.
3. Sumber referensi yang digunakan lebih banyak untuk memperkuat dasar teori dalam penelitian.

DAFTAR PUSTAKA

- Anonymus. (4 Februari 2023). Kanker, penyakit berbahaya di Indonesia?. Retrieved from <https://indonesiabaik.id/videografis/kanker-penyakit-berbahaya-di-indonesia>
- Anggraini, M. (2024). Pendeteksi Penyakit Berdasarkan Citra Body Fluid Menggunakan Metode Haar Wavelet dan Gabor Filter Berbasis Neural Network. Skripsi. Fakultas Teknik. Universitas Muhammadiyah Yogyakarta: Yogyakarta.
- Bahaalddin, B., & Ahmed, H. (2020). Breast Mass Classification Based on Hybrid Discrete Cosine Transformation–Haar Wavelet Transformation. *UKH Journal of Science and Engineering*, 4(2), 178–187. <https://doi.org/10.25079/ukhjse.v4n2y2020.pp178-187>
- Baykal, E., Dogan, H., Ekinci, M., Ercin, M. E., & Ersoz, S. (2017). Automated cell nuclei segmentation in pleural effusion cytology using active appearance model. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10425 LNCS, 59–69. https://doi.org/10.1007/978-3-319-64698-5_6
- Faiqoh, N., Ambarwati, L., Widodo, A. W., & Rahman, M. A. (2020). Implementasi Metode Haar Wavelet Untuk Penentuan Mutu Pada Citra Digital Jeruk Keprok (*Citrus Reticulata Blanco*). 4(10), 3730–3738. <http://j-ptiik.ub.ac.id>
- Faria, J. F., Seixas, J. M. de, Souza Filho, J. B. de O. e, Orjuela, A. D., Vieira, A. F. M., Kritski, A. L., Silva, I. M. F., & Trajman, A. (2016). *Pleural Tuberculosis Diagnosis Based On Artificial Neural Networks Models*. July 2014, 1–7. <https://doi.org/10.21528/cbic2011-13.3>
- Fattah, M. S., Haq, D. Z., Candra, D., & Novitasari, R. (2021). Pengolahan Citra Digital untuk Identifikasi Kanker Otak Menggunakan Metode Deep Belief Network (DBN). *Jurnal Informatika Universitas Pamulang*, 6(4), 2622–4615. <http://openjournal.unpam.ac.id/index.php/informatika735>
- Hussain, A., & Khunteta, A. (2020). Semantic Segmentation of Brain Tumor from

- MRI Images and SVM Classification using GLCM Features. *Proceedings of the 2nd International Conference on Inventive Research in Computing Applications, ICIRCA 2020*, 38–43.
<https://doi.org/10.1109/ICIRCA48905.2020.9183385>
- Karthiga, R., & Narasimhan, K. (2018). Automated Diagnosis of Breast Cancer Using Wavelet Based Entropy Features. *Proceedings of the 2nd International Conference on Electronics, Communication and Aerospace Technology, ICECA 2018, Iceca*, 274–279. <https://doi.org/10.1109/ICECA.2018.8474739>
- Kaur, K., & Malhotra, S. (2015). Image Compression using HAAR Wavelet Transform and Discrete Cosine Transform. *International Journal of Computer Applications*, 125(11), 28–31.
<https://doi.org/10.5120/ijca2015906141>
- Margareth, H. (2017). No Title طرق تدريس اللغة العربية. *Экономика Региона*, 32.
- Masood, S., Doja, M. N., & Chandra, P. (2016). Analysis of weight initialization methods for gradient descent with momentum. *International Conference on Soft Computing Techniques and Implementations, ICSCIT 2015*, 2, 131–136.
<https://doi.org/10.1109/ICSCIT.2015.7489618>
- Maulida, N., Paramitha, D. F., & Sukarno, E. a. (2013). Klasifikasi Kanker Paru-Paru Menggunakan Pengolahan Citra. *Jurnal Teknik Pomits*, 2(1), 3–7.
- Melekoodappattu, J. G., Kadan, A. B., & Anoop, V. (2021). Early detection of breast malignancy using wavelet features and optimized classifier. *International Journal of Imaging Systems and Technology*, 31(3), 1551–1563. <https://doi.org/10.1002/ima.22537>
- Mir, A. A., & Sarwar, A. (2021). Artificial intelligence-based techniques for analysis of body cavity fluids: a review. In *Artificial Intelligence Review* (Vol. 54, Issue 6). Springer Netherlands. <https://doi.org/10.1007/s10462-020-09946-y>
- Nayak, S. (2017). Scaled Conjugate Gradient Backpropagation Algorithm for Selection of Industrial Robots. *International Journal of Computer*

Application, 6(7), 92–101. <https://doi.org/10.26808/rs.ca.i7v6.12>

Porcel, J. M., Alemán, C., Bielsa, S., Sarrapio, J., Fernández de Sevilla, T., & Esquerda, A. (2008). A decision tree for differentiating tuberculous from malignant pleural effusions. *Respiratory Medicine*, 102(8), 1159–1164. <https://doi.org/10.1016/j.rmed.2008.03.001>

Potghan, S., Rajamenakshi, R., & Bhise, A. (2018). Multi-Layer Perceptron Based Lung Tumor Classification. *Proceedings of the 2nd International Conference on Electronics, Communication and Aerospace Technology, ICECA 2018, Iceca*, 499–502. <https://doi.org/10.1109/ICECA.2018.8474864>

Raut, A., Mayur, P. A., Kamble, S. D., & Dhondrikar, C. P. (2016). Texture Parameters Extraction of Satellite Image. *IJSTE-International Journal of Science Technology & Engineering* /, 2(11), 13–18. www.ijste.org

Rohman, M. A., Mudjirahardjo, P., & Muslim, M. A. (2021). Implementasi Filter Gray Level Co-Occurance Matriks Terhadap Sistem Klasifikasi Kanker Payudara Dengan Metode Convolutional Neural Network. *Transmisi*, 23(4), 160–168. <https://doi.org/10.14710/transmisi.23.4.160-168>

Sanyal, P., Paul, S., Rana, V., & Kulhari, K. (2021). A machine learning model for screening of body fluid cytology smears. *BioRxiv*, 14, 2021.07.20.453010. <https://www.biorxiv.org/content/10.1101/2021.07.20.453010v2%0Ahttps://www.biorxiv.org/content/10.1101/2021.07.20.453010v2.abstract>

Satria, A., Sani Sembiring, A., Kunci, K., Haar Wavelet Transform, M., & Berwarna, C. (2018). Penerapan Metode Haar Wavelet Transform Dalam Teknik Pemampatan Citra Berwarna. *Jurikom*, 5(3), 296–300. <http://ejurnal.stmik-budidarma.ac.id/index.php/jurikom%7CPage%7C296>

Solikhun, Wahyudi, M., Safii, M., & Zarlis, M. (2020). Backpropagation Network Optimization Using One Step Secant (OSS) Algorithm. *IOP Conference Series: Materials Science and Engineering*, 769(1). <https://doi.org/10.1088/1757-899X/769/1/012037>

- Sumijan, & Pradani, A. W. . (2021). *Teori dan Aplikasi Pengolahan Citra Digital Penerapan dalam Bidang Citra Medis*.
<https://play.google.com/books/reader?id=RFEtEAAAQBAJ&hl=id&pg=GBS.PA19>
- Vargason, T. J., Cohn, J., Rios, D., Schultz, O., Cleary, J., Lau, D., Land, W. H., David Schaffer, J., Li, Y., Chou, C. A., Syouri, S. A., Sidhu, J., Nelson, M. F., & Qiao, X. (2020). A clinical decision support system for malignant pleural effusion analysis. *Proceedings of the 2016 Industrial and Systems Engineering Research Conference, ISERC 2016*, 1016–1021.
- Wei, T. T., Zhang, J. F., Cheng, Z., Jiang, L., Li, J. Y., & Zhou, L. (2023). Development and validation of a machine learning model for differential diagnosis of malignant pleural effusion using routine laboratory data. *Therapeutic Advances in Respiratory Disease*, 17, 1–16.
<https://doi.org/10.1177/17534666231208632>
- Win, K. Y., Choomchuay, S., & Hamamoto, K. (2017a). Automated segmentation and isolation of touching cell nuclei in cytopathology smear images of pleural effusion using distance transform watershed method. *Second International Workshop on Pattern Recognition, 10443*, 104430Q.
<https://doi.org/10.1117/12.2280807>
- Win, K. Y., Choomchuay, S., & Hamamoto, K. (2017b). K mean clustering based automated segmentation of overlapping cell nuclei in pleural effusion cytology images. *International Conference on Advanced Technologies for Communications, 2017-October*, 265–269.
<https://doi.org/10.1109/ATC.2017.8167630>
- Win, K. Y., Choomchuay, S., Hamamoto, K., & Raveesunthornkiat, M. (2018). Comparative study on automated cell nuclei segmentation methods for cytology pleural effusion images. *Journal of Healthcare Engineering*, 2018.
<https://doi.org/10.1155/2018/9240389>
- Win, K. Y., Choomchuay, S., Hamamoto, K., Raveesunthornkiat, M.,

- Rangsirattanakul, L., & Pongsawat, S. (2018). Computer aided diagnosis system for detection of cancer cells on cytological pleural effusion images. *BioMed Research International*, 2018. <https://doi.org/10.1155/2018/6456724>
- Wright, C. A., Path, F. R. C., Burg, M. Van Der, Ph, D., Geiger, D., & Sc, M. (2008). Diagnosing Mycobacterial Lymphadenitis in Children Using Fine Needle Aspiration Biopsy : Cytomorphology , ZN Staining and Autofluorescence — Making More of Less. *Diagnostic Cytopathology*, 36(4), 245–251. <https://doi.org/10.1002/dc>
- Yunianto, M., Soeparmi, S., Cari, C., Anwar, F., Septianingsih, D. N., Ardyanto, T. D., & Pradana, R. F. (2021). Klasifikasi Kanker Paru Paru menggunakan Naïve Bayes dengan Variasi Filter dan Ekstraksi Ciri GLCM. *Indonesian Journal of Applied Physics*, 11(2), 256. <https://doi.org/10.13057/ijap.v11i2.53213>
- Zhang, L., Wang, Q., & Qi, J. (2006). Research based on fuzzy algorithm of cancer cells in pleural fluid microscopic images recognition. *Proceedings - 2006 International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IIH-MSP 2006*, 211–214. <https://doi.org/10.1109/IIH-MSP.2006.264982>

LAMPIRAN

Lampiran 1. Function graycomatrix

```
function [GLCMS,SI] = graycomatrix(varargin)
%GRAYCOMATRIX Create gray-level co-occurrence matrix.
%   GLCMS = GRAYCOMATRIX(I) analyzes pairs of horizontally adjacent
pixels
%   in a scaled version of I.  If I is a binary image, it is scaled to 2
%   levels.  If I is an intensity image, it is scaled to 8 levels.  In this
%   case, there are 8 x 8 = 64 possible ordered combinations of values
for
%   each pixel pair.  GRAYCOMATRIX accumulates the total occurrence of
each
%   such combination, producing a 8-by-8 output array, GLCMS.  The row and
%   column subscripts in GLCMS correspond respectively to the first and
%   second (scaled) pixel-pair values.
%
%   GLCMS = GRAYCOMATRIX(I,PARAM1,VALUE1,PARAM2,VALUE2,...) returns one
or
%   more gray-level co-occurrence matrices, depending on the values of
the
%   optional parameter/value pairs.  Parameter names can be abbreviated,
and
%   case does not matter.
%
%   Parameters include:
%
%   'Offset'           A p-by-2 array of offsets specifying the distance
%                       between the pixel-of-interest and its neighbor.  Each
%                       row in the array is a two-element vector,
%                       [ROW_OFFSET COL_OFFSET], that specifies the
%                       relationship, or 'Offset', between a pair of pixels.
%                       ROW_OFFSET is the number of rows between the
%                       pixel-of-interest and its neighbor.  COL_OFFSET is
the
%                       number of columns between the pixel-of-interest and
%                       its neighbor.  For example, if you want the number of
%                       occurrences where the pixel of interest is one pixel
%                       to the left of its neighbor, then
%                       [ROW_OFFSET COL_OFFSET] is [0 1].
%
%                       Because this offset is often expressed as an angle,
%                       the following table lists the offset values that
%                       specify common angles, given the pixel distance D.
%
%                       Angle      OFFSET
%                       -----      -
%                       0           [0 D]
%                       45          [-D D]
%                       90          [-D 0]
%                       135         [-D -D]
%
%                       ROW_OFFSET and COL_OFFSET must be integers.
%
%                       Default: [0 1]
%
%   'NumLevels'       An integer specifying the number of gray levels to
use
%                       when scaling the grayscale values in I.  For example,
%                       if 'NumLevels' is 8, GRAYCOMATRIX scales the values
in
```

```

% I so they are integers between 1 and 8. The number
Of
% gray levels determines the size of the gray-level
% co-occurrence matrix (GLCM).
%
% 'NumLevels' must be an integer. 'NumLevels' must be
2
% if I is logical.
%
% Default: 8 for numeric
%           2 for logical
%
% 'GrayLimits' A two-element vector, [LOW HIGH], that specifies how
% the values in I are scaled into gray levels. If N is
% the number of gray levels (see parameter
'NumLevels')
% to use for scaling, the range [LOW HIGH] is divided
% into N equal width bins and values in a bin get
mapped
% to a single gray level. Grayscale values less than
or
% equal to LOW are scaled to 1. Grayscale values
greater
% than or equal to HIGH are scaled to NumLevels. If
% 'GrayLimits' is set to [], GRAYCOMATRIX uses the
% minimum and maximum grayscale values in I as limits,
% [min(I(:)) max(I(:))].
%
% Default: the LOW and HIGH values specified by the
% class, e.g., [LOW HIGH] is [0 1] if I is double and
% [-32768 32767] if I is int16.
%
% 'Symmetric' A Boolean that creates a GLCM where the ordering of
% values in the pixel pairs is not considered. For
% example, when calculating the number of times the
% value 1 is adjacent to the value 2, GRAYCOMATRIX
% counts both 1,2 and 2,1 pairings, if 'Symmetric' is
% set to true. When 'Symmetric' is set to false,
% GRAYCOMATRIX only counts 1,2 or 2,1, depending on
the
% value of 'offset'. The GLCM created in this way is
% symmetric across its diagonal, and is equivalent to
% the GLCM described by Haralick (1973).
%
% The GLCM produced by the following syntax,
%
% graycomatrix(I, 'offset', [0 1], 'Symmetric', true)
%
% is equivalent to the sum of the two GLCMs produced
by
% these statements.
%
% graycomatrix(I, 'offset', [0 1], 'Symmetric', false)
% graycomatrix(I, 'offset', [0 -1], 'Symmetric',
false)
%
% Default: false
%
%
% [GLCMS,SI] = GRAYCOMATRIX(...) returns the scaled image used to
% calculate GLCM. The values in SI are between 1 and 'NumLevels'.
%
% Class Support

```



```

% -----
% I can be numeric or logical. I must be 2D, real, and nonsparse. SI
is
% a double matrix having the same size as I. GLCMS is an
% 'NumLevels'-by-'NumLevels'-by-P double array where P is the number of
% offsets in OFFSET.
%
% Notes
% -----
% Another name for a gray-level co-occurrence matrix is a gray-level
% spatial dependence matrix.
%
% GRAYCOMATRIX ignores pixels pairs if either of their values is NaN.
It
% also replaces Inf with the value 'NumLevels' and -Inf with the value
1.
%
% GRAYCOMATRIX ignores border pixels, if the corresponding neighbors
% defined by 'Offset' fall outside the image boundaries.
%
% References
% -----
% [1] Haralick, R.M., K. Shanmugan, and I. Dinstein, "Textural Features
for
% Image Classification", IEEE Transactions on Systems, Man, and
% Cybernetics, Vol. SMC-3, 1973, pp. 610-621.
%
% [2] Haralick, R.M., and L.G. Shapiro. Computer and Robot Vision: Vol.
1,
% Addison-Wesley, 1992, p. 459.
%
% Example 1
% -----
% % Calculate the gray-level co-occurrence matrix (GLCM) and return the
% % scaled version of the image, SI, used by GRAYCOMATRIX to generate
the
% % GLCM.
%
% I = [1 1 5 6 8 8;2 3 5 7 0 2; 0 2 3 5 6 7];
% [GLCMS,SI] = graycomatrix(I,'NumLevels',9,'G',[])
%
% Example 2
% -----
% % Calculate the gray-level co-occurrence matrix for a grayscale
image.
%
% I = imread('circuit.tif');
% GLCMS = graycomatrix(I,'Offset',[2 0])
%
% Example 3
% -----
% % Calculate gray-level co-occurrences matrices for a grayscale image
% % using four different offsets.
%
% I = imread('cell.tif');
% offsets = [0 1;-1 1;-1 0;-1 -1];
% [GLCMS,SI] = graycomatrix(I,'Of',offsets);
%
% Example 4
% -----
% % Calculate the symmetric gray-level co-occurrence matrix (the
Haralick
% % definition) for a grayscale image.

```

```

%
%   I = imread('circuit.tif');
%   GLCMS = graycomatrix(I,'Offset',[2 0],'Symmetric', true)
%
%   See also GRAYCOPROPS.
%
%   Copyright 1993-2018 The MathWorks, Inc.
[I, Offset, NL, GL, makeSymmetric] = ParseInputs(varargin{:});

% Scale I so that it contains integers between 1 and NL.
if GL(2) == GL(1)
    SI = ones(size(I));
else
    slope = NL / (GL(2) - GL(1));
    intercept = 1 - (slope*(GL(1)));
    SI = floor(imlincomb(slope,I,intercept,'double'));
end

% Clip values if user had a value that is outside of the range, e.g.,
% double image = [0 .5 2;0 1 1]; 2 is outside of [0,1]. The order of the
% following lines matters in the event that NL = 0.
SI(SI > NL) = NL;
SI(SI < 1) = 1;

numOffsets = size(Offset,1);

if NL ~= 0

    % Create vectors of row and column subscripts for every pixel and its
    % neighbor.
    s = size(I);
    [r,c] = meshgrid(1:s(1),1:s(2));
    r = r(:);
    c = c(:);

    % Compute GLCMS
    GLCMS = zeros(NL,NL,numOffsets);
    for k = 1 : numOffsets
        GLCMS(:,:,k) = computeGLCM(r,c,Offset(k,:),SI,NL);

        if makeSymmetric
            % Reflect glcm across the diagonal
            glcmTranspose = GLCMS(:,:,k).';
            GLCMS(:,:,k) = GLCMS(:,:,k) + glcmTranspose;
        end
    end
else
    GLCMS = zeros(0,0,numOffsets);
end

%-----
%-----
function oneGLCM = computeGLCM(r,c,offset,si,nl)
% computes GLCM given one Offset

r2 = r + offset(1);
c2 = c + offset(2);

[nRow nCol] = size(si);
% Determine locations where subscripts outside the image boundary
outsideBounds = find(c2 < 1 | c2 > nCol | r2 < 1 | r2 > nRow);

```

```

% Create vector containing si(r1,c1)
v1 = shiftdim(si,1);
v1 = v1(:);
v1(outsideBounds) = [];

% Create vector containing si(r2,c2). Not using sub2ind for performance
% reasons
r2(outsideBounds) = [];
c2(outsideBounds) = [];
Index = r2 + (c2 - 1)*nRow;
v2 = si(Index);

% Ensure that v2 is a column vector.
v2 = v2(:);

% Remove pixel and its neighbor if their value is NaN.
bad = isnan(v1) | isnan(v2);
if any(bad)
    warning(message('images:graycomatrix:scaledImageContainsNan'));
end

Ind = [v1 v2];
Ind(bad,:) = [];

if isempty(Ind)
    oneGLCM = zeros(nl);
else
    % Tabulate the occurrences of pixel pairs having v1 and v2.
    oneGLCM = accumarray(Ind, 1, [nl nl]);
end

%-----
-----
function [I, offset, nl, gl, sym] = ParseInputs(varargin)

narginchk(1,9);

% Check I
I = varargin{1};
validateattributes(I,{'logical','numeric'},{'2d','real','nonsparse'}, ...
    mfilename,'I',1);

% Assign Defaults
offset = [0 1];
if islogical(I)
    nl = 2;
else
    nl = 8;
end
gl = getrangefromclass(I);
sym = false;

% Parse Input Arguments
if nargin ~= 1

    paramStrings = {'Offset','NumLevels','GrayLimits','Symmetric'};

    for k = 2:2:nargin

        param = lower(varargin{k});
        inputStr = validatestring(param, paramStrings, mfilename, 'PARAM',
k);

```

```

    idx = k + 1; %Advance index to the VALUE portion of the input.
    if idx > nargin
        error(message('images:graycomatrix:missingParameterValue',
inputStr));
    end

    switch (inputStr)

        case 'Offset'

            offset = varargin{idx};
            validateattributes(offset,{'logical','numeric'},...
                {'2d','nonempty','integer','real'},...
                mfilename, 'OFFSET', idx);
            if size(offset,2) ~= 2
                error(message('images:graycomatrix:invalidOffsetSize'));
            end
            offset = double(offset);

        case 'NumLevels'

            nl = varargin{idx};
            validateattributes(nl,{'logical','numeric'},...
{'real','integer','nonnegative','nonempty','nonsparse'},...
                mfilename, 'NL', idx);
            if numel(nl) > 1
                error(message('images:graycomatrix:invalidNumLevels'));
            elseif islogical(I) && nl ~= 2
                error(message('images:graycomatrix:invalidNumLevelsForBinary'));
            end
            nl = double(nl);

        case 'GrayLimits'

            gl = varargin{idx};
            % step 1: checking for classes
            validateattributes(gl,{'logical','numeric'}, {}, mfilename, 'GL',
idx);
            if isempty(gl)
                gl = [min(I(:)) max(I(:))];
            end

            % step 2: checking for attributes
            validateattributes(gl,{'logical','numeric'}, {'vector','real'}, mfilename,
'GL', idx);

            if numel(gl) ~= 2
                error(message('images:graycomatrix:invalidGrayLimitsSize'));
            end
            gl = double(gl);

        case 'Symmetric'
            sym = varargin{idx};
            validateattributes(sym,{'logical'}, {'scalar'}, mfilename,
'Symmetric', idx);

    end
end
end

```

Lampiran 2. Function graycoprops

```
function stats = graycoprops(varargin)
%GRAYCOPROPS Properties of gray-level co-occurrence matrix.
%   STATS = GRAYCOPROPS(GLCM,PROPERTIES) normalizes the gray-level
%   co-occurrence matrix (GLCM) so that the sum of its elements is one.
Each
%   element in the normalized GLCM, (r,c), is the joint probability
occurrence
%   of pixel pairs with a defined spatial relationship having gray level
%   values r and c in the image. GRAYCOPROPS uses the normalized GLCM to
%   calculate PROPERTIES.
%
%   GLCM can be an m x n x p array of valid gray-level co-occurrence
%   matrices. Each gray-level co-occurrence matrix is normalized so that
its
%   sum is one.
%
%   PROPERTIES can be a comma-separated list of strings or char vectors,
a
%   cell array containing strings or char vectors, the string "all", char
%   vector 'all', or a space separated string or char vector. They can be
%   abbreviated, and case does not matter.
%
%   Properties include:
%
%   'Contrast'      the intensity contrast between a pixel and its
neighbor
%                   over the whole image. Range = [0 (size(GLCM,1)-1)^2].
%                   Contrast is 0 for a constant image.
%
%   'Correlation'   statistical measure of how correlated a pixel is to
its
%                   neighbor over the whole image. Range = [-1 1].
%                   Correlation is 1 or -1 for a perfectly positively or
%                   negatively correlated image. Correlation is NaN for a
%                   constant image.
%
%   'Energy'        summation of squared elements in the GLCM. Range = [0
1].
%                   Energy is 1 for a constant image.
%
%   'Homogeneity'   closeness of the distribution of elements in the GLCM
to
%                   the GLCM diagonal. Range = [0 1]. Homogeneity is 1
for
%                   a diagonal GLCM.
%
%   If PROPERTIES is the string "all" or char vector 'all', then all of
the
%   above properties are calculated. This is the default behavior. Please
%   refer to the Documentation for more information on these properties.
%
%   STATS is a structure with fields that are specified by PROPERTIES.
Each
%   field contains a 1 x p array, where p is the number of gray-level
%   co-occurrence matrices in GLCM. For example, if GLCM is an 8 x 8 x 3
array
%   and PROPERTIES is 'Energy', then STATS is a structure containing the
%   field 'Energy'. This field contains a 1 x 3 array.
%
%   Notes
```

```

% -----
% Energy is also known as uniformity, uniformity of energy, and angular
second
% moment.
%
% Contrast is also known as variance and inertia.
%
% Class Support
% -----
% GLCM can be logical or numeric, and it must contain real, non-
negative, finite,
% integers. STATS is a structure.
%
% Examples
% -----
% GLCM = [0 1 2 3;1 1 2 3;1 0 2 0;0 0 0 3];
% stats = graycoprops(GLCM)
%
% I = imread('circuit.tif');
% GLCM2 = graycomatrix(I,'Offset',[2 0;0 2]);
% stats = graycoprops(GLCM2,{'contrast','homogeneity'})
%
% See also GRAYCOMATRIX.

% Copyright 2003-2016 The MathWorks, Inc.

allStats = {'Contrast','Correlation','Energy','Homogeneity'};

[glcm, requestedStats] = ParseInputs(allStats, varargin{:});

% Initialize output stats structure.
numStats = length(requestedStats);
numGLCM = size(glcm,3);
empties = repmat({zeros(1,numGLCM)},[numStats 1]);
stats = cell2struct(empties,requestedStats,1);

for p = 1 : numGLCM

    if numGLCM ~= 1 %N-D indexing not allowed for sparse.
        tGLCM = normalizeGLCM(glcm(:,:,p));
    else
        tGLCM = normalizeGLCM(glcm);
    end

    % Get row and column subscripts of GLCM. These subscripts correspond
to the
    % pixel values in the GLCM.
    s = size(tGLCM);
    [c,r] = meshgrid(1:s(1),1:s(2));
    r = r(:);
    c = c(:);

    % Calculate fields of output stats structure.
    for k = 1:numStats
        name = requestedStats{k};
        switch name
            case 'Contrast'
                stats.(name)(p) = calculateContrast(tGLCM,r,c);

            case 'Correlation'
                stats.(name)(p) = calculateCorrelation(tGLCM,r,c);

            case 'Energy'

```

```

        stats.(name)(p) = calculateEnergy(tGLCM);

        case 'Homogeneity'
            stats.(name)(p) = calculateHomogeneity(tGLCM,r,c);
        end
    end
end

end

%-----
-----
function glcm = normalizeGLCM(glcm)

% Normalize glcm so that sum(glcm(:)) is one.
if any(glcm(:))
    glcm = glcm ./ sum(glcm(:));
end

%-----
-----
function C = calculateContrast(glcm,r,c)
% Reference: Haralick RM, Shapiro LG. Computer and Robot Vision: Vol. 1,
% Addison-Wesley, 1992, p. 460.
k = 2;
l = 1;
term1 = abs(r - c).^k;
term2 = glcm.^l;

term = term1 .* term2(:);
C = sum(term);

%-----
-----
function Corr = calculateCorrelation(glcm,r,c)
% References:
% Haralick RM, Shapiro LG. Computer and Robot Vision: Vol. 1, Addison-
Wesley,
% 1992, p. 460.
% Bevk M, Kononenko I. A Statistical Approach to Texture Description of
Medical
% Images: A Preliminary Study., The Nineteenth International Conference
of
% Machine Learning, Sydney, 2002.
% http://www.cse.unsw.edu.au/~icml2002/workshops/MLCV02/MLCV02-Bevk.pdf,
p.3.

% Correlation is defined as the covariance(r,c) / S(r)*S(c) where S is
the
% standard deviation.

% Calculate the mean and standard deviation of a pixel value in the row
% direction direction. e.g., for glcm = [0 0;1 0] mr is 2 and Sr is 0.
mr = meanIndex(r,glcm);
Sr = stdIndex(r,glcm,mr);

% mean and standard deviation of pixel value in the column direction,
e.g.,
% for glcm = [0 0;1 0] mc is 1 and Sc is 0.
mc = meanIndex(c,glcm);
Sc = stdIndex(c,glcm,mc);

term1 = (r - mr) .* (c - mc) .* glcm(:);

```

```

term2 = sum(term1);

Corr = term2 / (Sr * Sc);

%-----
%-----
function S = stdIndex(index,glcm,m)

term1 = (index - m).^2 .* glcm(:);
S = sqrt(sum(term1));

%-----
%-----
function M = meanIndex(index,glcm)

M = index .* glcm(:);
M = sum(M);

%-----
%-----
function E = calculateEnergy(glcm)
% Reference: Haralick RM, Shapiro LG. Computer and Robot Vision: Vol. 1,
% Addison-Wesley, 1992, p. 460.

foo = glcm.^2;
E = sum(foo(:));

%-----
%-----
function H = calculateHomogeneity(glcm,r,c)
% Reference: Haralick RM, Shapiro LG. Computer and Robot Vision: Vol. 1,
% Addison-Wesley, 1992, p. 460.

term1 = (1 + abs(r - c));
term = glcm(:) ./ term1;
H = sum(term);

%-----
%-----
function [glcm,reqStats] = ParseInputs(allstats,varargin)

numstats = length(allstats);
narginchk(1,numstats+1);

reqStats = '';
glcm = varargin{1};

% The 'nonnan' and 'finite' attributes are not added to
% validateattributes because the
% 'integer' attribute takes care of these requirements.
validateattributes(glcm,{'logical','numeric'},{'real','nonnegative','inte
ger'}, ...
                 mfilename,'GLCM',1);

if ndims(glcm) > 3
    error(message('images:graycoprops:invalidSizeForGLCM'))
end

% Cast GLCM to double to avoid truncation by data type. Note that GLCM is
% not an
% image.
if ~isa(glcm,'double')
    glcm = double(glcm);

```



```

end

list = varargin(2:end);

if isempty(list)
    % GRAYCOPROPS (GLCM) or GRAYCOPROPS (GLCM,PROPERTIES) where PROPERTIES is
    empty.
    reqStats = allstats;
else
    if iscell(list{1}) || numel(list) == 1
        % GRAYCOPROPS (GLCM,{...})
        list = list{1};
    end

    if ischar(list)
        %GRAYCOPROPS (GLCM,SPACE-SEPARATED STRING)
        C = textscan(list, '%s');
        list = C{1};
    end

    anyprop = allstats;
    anyprop{end+1} = 'all';

    for k = 1 : length(list)
        match = validatestring(list{k}, anyprop, mfilename, 'PROPERTIES',
k+1);
        if strcmp(match,'all')
            reqStats = allstats;
            break;
        end
        reqStats{k} = match;
    end

end

% Make sure that reqStats are in alphabetical order.
reqStats = sort(reqStats);

if isempty(reqStats)
    error(message('images:graycoprops:internalError'))
end

```

Lampiran 3. Function wavedec2

```

function [c,s] = wavedec2(x,n,IN3,IN4)
%WAVEDEC2 Multilevel 2-D wavelet decomposition.
% [C,S] = WAVEDEC2(X,N,'wname') returns the wavelet
% decomposition of the matrix X at level N, using the
% wavelet named in string 'wname' (see WFILTERS).
% Outputs are the decomposition vector C and the
% corresponding bookkeeping matrix S.
% N must be a strictly positive integer (see WMAXLEV).
%
% Instead of giving the wavelet name, you can give the
% filters.
% For [C,S] = WAVEDEC2(X,N,Lo_D,Hi_D),
% Lo_D is the decomposition low-pass filter and
% Hi_D is the decomposition high-pass filter.
%
% The output wavelet 2-D decomposition structure [C,S]
% contains the wavelet decomposition vector C and the
% corresponding bookeeping matrix S.

```

```

% Vector C is organized as:
%   C = [ A(N)   | H(N)   | V(N)   | D(N) | ...
%   H(N-1) | V(N-1) | D(N-1) | ... | H(1) | V(1) | D(1) ].
%   where A, H, V, D, are row vectors such that:
%   A = approximation coefficients,
%   H = hori. detail coefficients,
%   V = vert. detail coefficients,
%   D = diag. detail coefficients,
%   each vector is the vector column-wise storage of a matrix.
%   Matrix S is such that:
%   S(1,:) = size of app. coef.(N)
%   S(i,:) = size of det. coef.(N-i+2) for i = 2,...,N+1
%   and S(N+2,:) = size(X).
%
% NOTE: When X represents an indexed image, then X as well
% as the arrays CA, CH, CV, CD are m-by-n matrices.
% When X represents a truecolor image, then they become
% m-by-n-by-3 arrays. These arrays consist of three m-by-n matrices
% (representing the red, green, and blue color planes) concatenated
% along the third dimension.
% So the sizes of the vector C and the matrix S depend on the
% type of analyzed image.
% For more information on image formats, see the reference pages of
% IMAGE and IMFINFO functions.
%
% See also DWT2, WAVEINFO, WAVEREC2, WFILTERS, WMAXLEV.
%
% M. Misiti, Y. Misiti, G. Oppenheim, J.M. Poggi 12-Mar-96.
% Last Revision: 31-Jul-2008.
% Copyright 1995-2018 The MathWorks, Inc.

% Convert wname from string to char if necessary
if isStringScalar(IN3)
    IN3 = convertStringsToChars(IN3);
end

if nargin==3
    [Lo_D,Hi_D] = wfilters(IN3,'d');
else
    Lo_D = IN3;   Hi_D = IN4;
end

% Initialization.
c = [];
sx = size(x);
s = zeros(n+2,length(sx));
if isempty(x) , return; end

s(end,:) = size(x);
for i=1:n
    [x,h,v,d] = dwt2(x,Lo_D,Hi_D); % decomposition
    c = [h(:)' v(:)' d(:)' c];      % store details
    s(n+2-i,:) = size(x);          % store size
end

% Last approximation.
c = [x(:)' c];
s(1,:) = size(x);

```

Lampiran 4. Function detcoef2

```
function varargout = detcoef2(o,c,s,n)
%DETcoef2 Extract 2-D detail coefficients.
% D = DETCOEF2(O,C,S,N) extracts from the wavelet
% decomposition structure [C,S], the horizontal, vertical
% or diagonal detail coefficients for O = 'h'
% (or 'v' or 'd', respectively), at level N. N must
% be an integer such that 1 <= N <= size(S,1)-2.
% See WAVEDEC2 for more information on C and S.
%
% [H,V,D] = DETCOEF2('all',C,S,N) returns the horizontal H,
% vertical V, and diagonal D detail coefficients at level N.
%
% D = DETCOEF2('compact',C,S,N) returns the detail
% coefficients at level N, stored row-wise.
%
% DETCOEF2('a',C,S,N) is equivalent to DETCOEF2('all',C,S,N).
% DETCOEF2('c',C,S,N) is equivalent to DETCOEF2('compact',C,S,N).
%
% NOTE: If C and S are obtained from an indexed image analysis
% (respectively a true color image analysis) then D is an
% m-by-n matrix (respectively an m-by-n-by-3 array).
% For more information on image formats, see the reference
% pages of IMAGE and IMFINFO functions.
%
% See also APPCOEF2, WAVEDEC2.
%
% M. Misiti, Y. Misiti, G. Oppenheim, J.M. Poggi 12-Mar-96.
% Last Revision: 06-Oct-2007.
% Copyright 1995-2018 The MathWorks, Inc.

% Check arguments.
if nargin > 0
    o = convertStringsToChars(o);
end

nmax = size(s,1)-2;
if (n<1) || (n>nmax) || (n~=fix(n))
    error(message('Wavelet:FunctionArgVal:Invalid_ArgVal'));
end
if length(s(1,:))<3 , dimFactor = 1; else dimFactor = 3; end

k = size(s,1)-n;
first = dimFactor*(s(1,1)*s(1,2) + 3*sum(s(2:k-1,1).*s(2:k-1,2)))+1;
add = dimFactor*s(k,1)*s(k,2);
o = lower(o);

switch o
    case 'h'
    case 'v' , first = first+add;
    case 'd' , first = first+2*add;
    case {'a','all','c','compact'}
        otherwise
            error(message('Wavelet:FunctionArgVal:Invalid_ArgVal'));
end

switch o
    case {'h','v','d'}
        last = first+add-1;
        varargout{1} = reshape(c(first:last),s(k,:));
```

```

case {'c','compact'}
    last = first+3*add-1;
    varargout{1} = c(first:last);

case {'a','all'}
    last = first+add-1;
    varargout{1} = reshape(c(first:last),s(k,:));
    first = first+add; last = first+add-1;
    varargout{2} = reshape(c(first:last),s(k,:));
    first = first+add; last = first+add-1;
    varargout{3} = reshape(c(first:last),s(k,:));
end

```

Lampiran 5. Function appcoef2

```

function a = appcoef2(c,s,varargin)
%APPCOEF2 Extract 2-D approximation coefficients.
%   APPCOEF2 computes the approximation coefficients of a
%   two-dimensional signal.
%
%   A = APPCOEF2(C,S,'wname',N) computes the approximation
%   coefficients at level N using the wavelet decomposition
%   structure [C,S] (see WAVEDEC2).
%   'wname' is a character vector containing the wavelet name.
%   Level N must be an integer such that 0 <= N <= size(S,1)-2.
%
%   A = APPCOEF2(C,S,'wname') extracts the approximation
%   coefficients at the last level size(S,1)-2.
%
%   Instead of giving the wavelet name, you can give the filters.
%   For A = APPCOEF2(C,S,Lo_R,Hi_R) or
%   A = APPCOEF2(C,S,Lo_R,Hi_R,N),
%   Lo_R is the reconstruction low-pass filter and
%   Hi_R is the reconstruction high-pass filter.
%
%   NOTE: If C and S are obtained from an indexed image analysis
%   (respectively a truecolor image analysis) then A is an
%   m-by-n matrix (respectively an m-by-n-by-3 array).
%   For more information on image formats, see the reference
%   pages of IMAGE and IMFINFO functions.
%
%   See also DETCOEF2, WAVEDEC2.
%
%   M. Misiti, Y. Misiti, G. Oppenheim, J.M. Poggi 12-Mar-96.
%   Last Revision: 06-Feb-2011.
%   Copyright 1995-2018 The MathWorks, Inc.

% Check arguments.
if nargin > 2
    [varargin{:}] = convertStringsToChars(varargin{:});
end

narginchk(2,5)
rmax = size(s,1);
nmax = rmax-2;
if ischar(varargin{1})
    [Lo_R,Hi_R] = wfilters(varargin{1},'r'); next = 2;
else
    Lo_R = varargin{1}; Hi_R = varargin{2}; next = 3;
end
if nargin>=(2+next) , n = varargin{next}; else n = nmax; end
if (n<0) || (n>nmax) || (n~=fix(n))

```

```

    error(message('Wavelet:FunctionArgVal:Invalid_LevVal'));
end

% Initialization.
nl = s(1,1);
nc = s(1,2);
if length(s(1,:))<3 , dimFactor = 1; else dimFactor = 3; end
a = zeros(nl,nc,dimFactor);
a(:) = c(1:nl*nc*dimFactor);

% Iterated reconstruction.
rm = rmax+1;
for p=nmax:-1:n+1
    [h,v,d] = detcoef2('all',c,s,p);
    a = idwt2(a,h,v,d,Lo_R,Hi_R,s(rm-p,:));
end

% If true color image.
% if n==0 && ndims(a)>2
%     a(a<0) = 0;
%     a = uint8(a);
% end

```

Lampiran 6. Function wcodemat

```

function y = wcodemat(x,nb,opt,absol)
%WCODEMAT Extended pseudocolor matrix scaling.
% Y = WCODEMAT(X,NBCODES,OPT,ABSOL) returns a coded version
% of input matrix X if ABSOL=0, or ABS(X) if ABSOL is
% nonzero, using the first NBCODES integers.
% Coding can be done row-wise (OPT='row' or 'r'), columnwise
% (OPT='col' or 'c'), or globally (OPT='mat' or 'm').
% Coding uses a regular grid between the minimum and
% the maximum values of each row (column or matrix,
% respectively).
%
% Y = WCODEMAT(X,NBCODES,OPT) is equivalent to
% Y = WCODEMAT(X,NBCODES,OPT,1).
% Y = WCODEMAT(X,NBCODES) is equivalent to
% Y = WCODEMAT(X,NBCODES,'mat',1).
% Y = WCODEMAT(X) is equivalent to
% Y = WCODEMAT(X,16,'mat',1).

% M. Misiti, Y. Misiti, G. Oppenheim, J.M. Poggi 01-May-96.
% Last Revision: 06-Feb-2011.
% Copyright 1995-2018 The MathWorks, Inc.

% Check arguments.
switch nargin
    case 1 , absol = 1; opt = 'm'; nb = 16;
    case 2 , absol = 1; opt = 'm';
    case 3 , absol = 1;
end
% Handle string input
if isStringScalar(opt)
    opt = convertStringsToChars(opt);
end
opt = lower(opt(1));

trans = false;
if isequal(opt(1),'r')
    trans = true;

```

```

    opt    = 'c';
    x      = x';
end
if absol , x = abs(x); end
switch opt
    case 'm'
        y = ones(size(x));
        x = x - min(x(:));
        maxx = max(x(:));
        if maxx<eps , return; end
        x = nb*(x/maxx);
        y(:) = 1 + fix(x);

    case 'c'
        t1 = size(x,1);
        minx = min(x);
        x = (x - minx(ones(1,t1),:));
        maxx = max(x);
        echel = maxx(ones(1,t1),:);
        indexs = echel<eps;
        x = nb*(x./echel);
        y = 1 + fix(x);
        y(indexs) = 1;

    otherwise
        error(message('Wavelet:FunctionArgVal:Unknown_Opt'));
end
y(y>nb) = nb;
if trans, y = y'; end

```