

BAB III METODOLOGI PENELITIAN

3.1. Metode Pengembangan Sistem

Dalam penelitian ini, metode pengembangan sistem dilakukan dengan menggunakan metode *waterfall* yang merupakan salah satu model dari *Software Development Life Cycle* (SDLC). Tahapan-tahapan mengenai metode *waterfall* dapat dilihat pada gambar 2.2. Berikut penjelasan masing-masing tahapan dari metode *Waterfall* yang telah peneliti terapkan dalam penelitian ini.

3.1.1 *Analysis*

Proses *analysis* dibutuhkan untuk mengidentifikasi kebutuhan *user*, dalam hal ini divisi *marketing* NIIT Indonesia, guna mendefinisikan kebutuhan akan aspek fungsional dan non-fungsional dari sistem CRM yang akan dibuat. Untuk dapat mendefinisikan kebutuhan-kebutuhan tersebut dilakukanlah proses *requirement gathering* dan *requirement analysis*. *Requirement* atau kebutuhan merupakan sesuatu yang harus dimiliki atau bisa dilakukan oleh sebuah sistem.

Secara garis besar *requirement* dibagi menjadi 3 yaitu:

1. *Business Requirement*

Melalui *business requirement* kita dapat mengidentifikasi *requirement* sebuah sistem dari segi bisnis/perusahaan. *Business requirement* pada umumnya menjawab pertanyaan, mengapa proyek dilaksanakan. Beberapa hal yang perlu diperhatikan dalam hal ini adalah tujuan proyek dilaksanakan, cakupan proyek, anggaran biaya, sumber daya serta batasan waktu pelaksanaan.

2. *User Requirement*

Berbeda dengan *business requirement*, *user requirement* mengidentifikasi *requirement* sebuah sistem dari segi *user*. Pada umumnya *user requirement* menjawab pertanyaan apa yang bisa *user* lakukan dengan sistem yang dibuat. Hal-hal yang harus diperhatikan dalam *user requirement* adalah

karakteristik *user*, tujuan *user*, aktifitas *user* serta lingkungan *user* dalam beraktifitas.

3. *System Requirementment*

Pada umumnya *system requirement* menjawab pertanyaan bagaimana sistem akan dibangun dan bagaimana nantinya sistem bekerja.

Requirement gathering merupakan proses pengumpulan data guna mendefinisikan *requirement* dari sistem yang akan dibuat. Pada *requirement gathering* proses pengumpulan data bisa dilakukan dengan berbagai cara seperti menyebar kuisioner, melakukan observasi atau pengamatan, wawancara, ataupun mempelajari dokumentasi dari aktivitas *user* yang bersangkutan. Pada penelitian ini peneliti menggunakan metode wawancara dan studi literatur untuk melakukan *requirement gathering*.

Data hasil dari *requirement gathering* selanjutnya dianalisa melalui *requirement analysis*. Di sinilah nantinya kebutuhan akan aspek fungsional dan non-fungsional ditentukan. Kebutuhan fungsional meliputi fitur serta atribut yang harus ada dalam sistem, persyaratan antarmuka pengguna serta persyaratan *database* yang dibangun. Dalam metode *Waterfall*, biasanya hasil dari proses *analysis* akan kebutuhan fungsional digambarkan dengan sebuah *Usecase*. Kebutuhan non-fungsional berfokus pada persyaratan dan batasan-batasan yang mengacu pada sifat serta perilaku dari sistem seperti kualitas, performa, skalabilitas serta *maintenablity*. Proses *analysis* akan peneliti bahas lebih lengkap pada bab 3.3.

3.1.2 *Design*

Proses *design* mendefinisikan rencana atau solusi yang akan diambil berdasarkan data hasil dari proses *analysis*. Proses *design* menghasilkan *blueprint* berupa dokumentasi rencana atau solusi sebagai pedoman *developer* serta *designer* dalam membuat sistem. Dokumentasi tersebut meliputi *design* sistem yang akan dibuat, *design* arsitektur, dan juga *design* antar muka. Dokumentasi-dokumentasi tersebut ditampilkan dalam diagram-diagram UML. Diagram UML yang

digunakan dalam penelitian ini diantaranya yaitu *usecase diagram* dan *activity diagram*. Hasil dari proses *design* akan peneliti bahas lebih lengkap pada bab 3.4.

3.1.3 Implementation

Implementation merupakan proses pengaplikasian *design* ke dalam bentuk arsitektur nyata. *Implementation* dilakukan melalui proses *coding*, dimana setiap komponen pembentuk sistem dibuat dan disatukan sehingga menjadi satu kesatuan *executable program*. Di sini peneliti menggunakan Microsoft Visual Studio 2015 sebagai IDE dan ASP.NET MVC5 dengan bahasa pemrograman C#. Untuk implementasi *database*, peneliti menggunakan Microsoft SQL Server 2014.

3.1.4 Testing

Testing merupakan proses pengujian untuk menilai kualitas dari sistem yang dibuat. Pada proses ini dilakukan validasi dan verifikasi terhadap *executable program* hasil dari proses *implementation*, untuk mengetahui apakah *executable program* tersebut sesuai dengan *requirement* awal yang didokumentasikan dalam proses *design*. Pada penelitian ini proses *testing* yang digunakan adalah *Black Box Testing*.

3.2. Metode Pengumpulan Data

Seperti yang sudah disebutkan sebelumnya dalam bab 3.1., pada penelitian ini peneliti menggunakan metode wawancara dan studi literatur untuk melakukan proses pengumpulan data atau *requirement gathering*.

3.2.1 Wawancara

Wawancara merupakan proses menggali informasi yang dilakukan dengan proses tanya jawab bersama narasumber. Pada penelitian ini wawancara dilakukan bersama *bussiness development executive* NIIT Indonesia sdr. Cinky Priyanto, S.T. Melalui wawancara, peneliti dapat mengetahui proses bisnis yang sekarang dijalankan serta mengidentifikasi kelemahan dari proses bisnis tersebut. Dari sinilah nantinya data hasil dari wawancara akan diolah dan dianalisis untuk menentukan solusi yang tepat dalam mengatasi masalah yang ditemukan.

3.2.2 Studi Literatur

Studi literatur dilakukan untuk membandingkan permasalahan yang ditemukan dari hasil wawancara dengan permasalahan sejenis yang sudah pernah diselesaikan sebelumnya oleh pihak lain. Studi literatur diperlukan untuk mendapatkan referensi informasi tambahan yang nantinya bisa berguna dalam proses pembentukan solusi.

3.3. Analisis Data

Untuk mempermudah proses analisis data, peneliti membagi data hasil dari *requirement gathering* kedalam 3 bagian yaitu *user analysis*, *task analysis* dan *architecture analysis*.

3.3.1. User Analysis

User analysis berfokus pada karakteristik *user* yang nantinya akan berhubungan dengan sistem yang dibuat. Dari hasil *requirement gathering* didapati bahwa nantinya terdapat 2 *user*, 1 *user* aktif yang akan berhubungan langsung dengan sistem, yaitu *operational marketing* NIIT Indonesia dan 1 *user* pasif yang tidak berhubungan langsung dengan sistem, yaitu *manager* divisi *marketing* NIIT Indonesia. *Manager* divisi *marketing* adalah pihak yang bertanggungjawab untuk memastikan bahwa setiap tahapan *customer development* berjalan dengan baik.

Operational marketing adalah pihak yang diberi tanggungjawab untuk melaksanakan proses *customer development* dari awal sampai akhir. Aktifitas-aktifitas *operational marketing* dalam proses *customer development* yaitu melakukan pendataan calon target konsumen, melakukan pendataan terhadap program yang akan dipasarkan serta melakukan pengajuan penawaran kerjasama kepada calon target. Karakteristik dari kedua *user* di atas peneliti sajikan dengan

Tabel 3.1. Tabel karakteristik *user*.

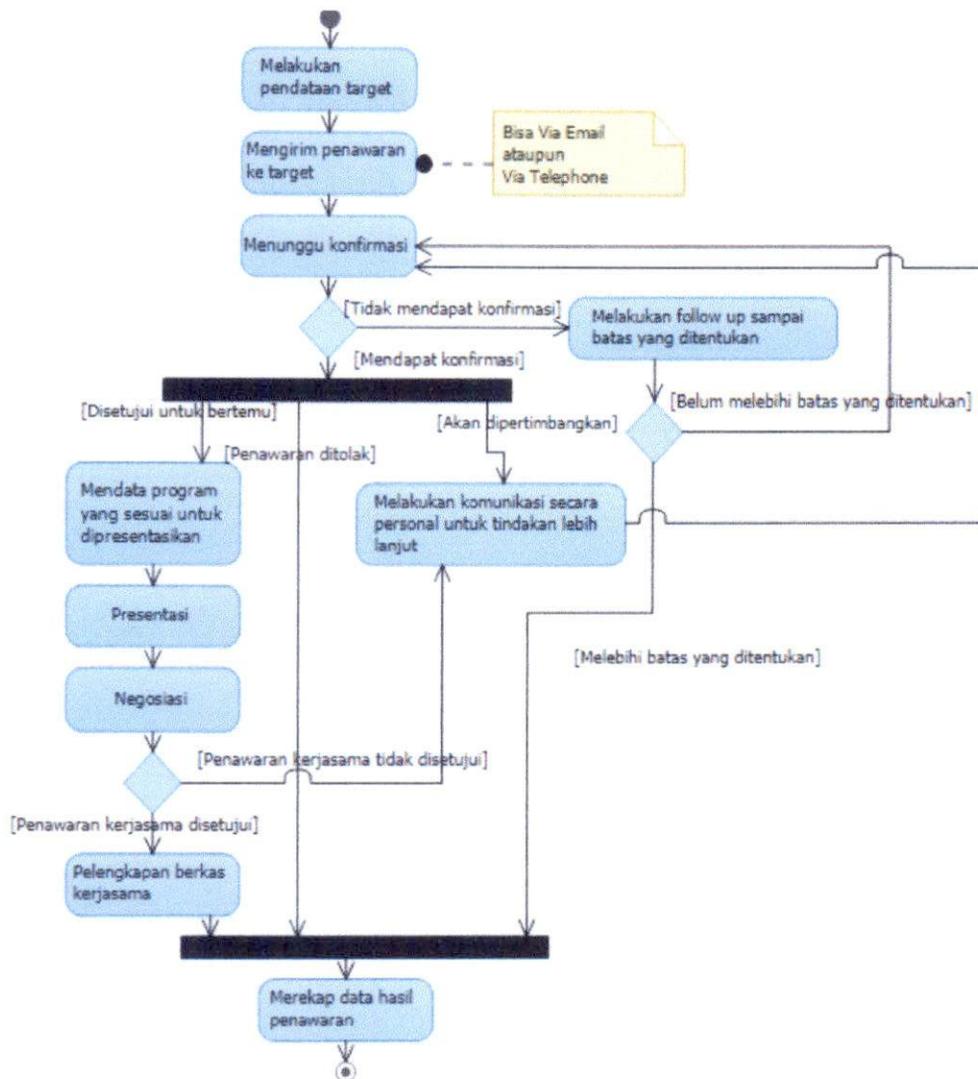
Karakteristik <i>User</i>	Divisi <i>Marketing</i> NIIT Indonesia	
	<i>Manager</i>	<i>Operational Marketing</i>
Usia	25-30 tahun	25-30 tahun
Jenis kelamin	Laki-laki	Perempuan
Keterbatasan fisik	Normal	Normal
Latar belakang pendidikan	S1	S1
Pengalaman bekerja dengan komputer / TI	Sudah terbiasa bekerja dengan menggunakan komputer/TI	Sudah terbiasa bekerja dengan menggunakan komputer/TI
Kendala saat ini	Butuh data <i>realtime</i> tentang kondisi penawaran pada tiap tahapan <i>customer development</i> yang sedang berlangsung	<ul style="list-style-type: none"> • Susah dalam melakukan pendataan terhadap proses <i>customer development</i> karena banyaknya tahapan yang harus dilalui • Susah dalam memetakan data antar tiap tahapan • Rawan terjadi <i>human error</i> seperti lupa input data dan lupa untuk melakukan <i>follow up</i>.
<i>Goal</i>	Menginginkan laporan yang <i>up-todate</i> dan terstruktur mengenai kondisi penawaran pada	<ul style="list-style-type: none"> • Menginginkan kemudahan dalam manajemen data konsumen

	tiap tahapan proses <i>customer development</i>	<ul style="list-style-type: none"> • Menginginkan sistem yang mampu meminimalisir terjadinya <i>human error</i>
--	--	--

Melihat dari karakteristik *user* seperti yang telah disebutkan di atas, peneliti menyimpulkan bahwa terdapat kemungkinan ketimpangan antara alat yang digunakan *user* dalam melakukan manajemen data sekarang ini, dengan kemampuan dan *goal* yang dimiliki oleh *user*. Keterbatasan *operational marketing* dengan banyaknya kemungkinan melakukan *human error* saat bekerja menjadi salah satu point utama yang harus diselesaikan. Untuk dapat meminimalisir *human error* dalam hal pemetaan data, dibutuhkan sebuah sistem yang terintegrasi dan terotomatisasi yang dapat menghubungkan data satu dengan data lainnya. Hal yang tak kalah penting adalah, dibutuhkannya sistem *reminder* agar *user* tidak lupa untuk merespon suati kondisi. *User* yang dalam hal ini adalah orang yang sudah terbiasa menggunakan komputer / TI dalam bekerja, membutuhkan alat manajemen data yang lebih terstruktur serta mampu meminimalisir terjadinya *human error* ketimbang menggunakan alat yang ada sekarang ini. Hal ini memungkinkan bahwa solusi yang nantinya diberikan menggunakan arsitektur teknologi yang lebih canggih.

3.3.2. Task Analysis

Task Analysis berfokus pada bagaimana proses bisnis berjalan dalam tahapan *customer development* NIIT Indonesia. Dalam praktiknya selama ini, proses *customer development* dilakukan dengan tahapan seperti pada *activity*



Gambar 3.1. Activity diagram proses customer development.

Berikut penjelasan dari *activity diagram* proses *customer development* pada gambar di atas

1. *Operational marketing* melakukan pendataan siapa yang akan menjadi calon target. Proses pendataan dilakukan dengan menggunakan Microsoft Excel. Data-data penting yang dihimpun meliputi nama instansi yang menjadi target, alamat instansi, email, nomor handphone, nama departemen, *key contact person*, *key contact email*, *key contact phone* serta jumlah mahasiswa dalam departemen tersebut.
2. Selanjutnya sebagai langkah awal penawaran kerjasama, dari data tersebut di atas *operational marketing* akan menghubungi pihak instansi untuk

mengajukan surat permohonan kunjungan. Proses ini, bisa dilakukan dengan dua cara yaitu mengirim surat permohonan melalui *email* ataupun dengan menghubungi pihak terkait melalui jaringan telepon. Tujuan yang ingin dicapai dalam tahapan ini adalah untuk mendapatkan kesempatan melakukan presentasi secara langsung dengan pihak terkait mengenai program yang akan ditawarkan.

3. Pada proses selanjutnya *operational marketing* akan menunggu konfirmasi dari target apakah permohonan kunjungan tersebut disetujui atau tidak. Jika ternyata dalam jangka waktu satu minggu belum ada konfirmasi, *operational marketing* akan menghubungi target kembali untuk meminta konfirmasi.
4. Dari poin 2 dan 3 selanjutnya didapat data mengenai target-target yang menyetujui permohonan kunjungan, menolak, serta akan mempertimbangkan permohonan tersebut.
5. Jika permohonan disetujui maka *operational marketing* akan melakukan pendataan tentang program yang sesuai dengan target tersebut. Data program ini nantinya akan dipakai sebagai bahan presentasi. Jika permohonan ditolak, maka target tersebut akan dipertimbangkan apakah akan menjadi target untuk pengajuan permohonan di periode selanjutnya atau tidak. Jika permohonan akan dipertimbangkan ulang, maka *operational marketing* akan melakukan komunikasi secara personal untuk melakukan negosiasi sampai didapat konfirmasi apakah permohonan disetujui atau tidak.
6. Langkah ke 6 adalah proses presentasi. *Operational marketing* akan mendatangi target secara langsung, untuk mempresentasikan program-program yang ditawarkan.
7. Setelah proses presentasi dilaksanakan, maka langkah selanjutnya adalah melakukan proses negosiasi. Proses negosiasi sendiri merupakan proses yang fleksibel, dimana proses ini bisa selesai sesaat setelah presentasi dilaksanakan, akan tetapi bisa juga memakan waktu yang cukup lama. Untuk proses negosiasi yang cukup lama biasanya pihak *operational*

marketing akan tetap melakukan komunikasi secara personal dan bahkan menjadwalkan beberapa pertemuan tambahan.

8. Dari proses negosiasi seperti yang tercantum pada point 5 selanjutnya didapat target yang menyetujui dan tidak menyetujui untuk melakukan kerjasama dengan pihak NIIT Indonesia.
9. Langkah terakhir adalah melengkapi pemberkasan kerjasama. Selanjutnya hasil dari semua tahapan akan direkap untuk dijadikan data laporan *customer development*.

Dari setiap kemungkinan-kemungkinan yang ada pada tahapan *customer development*, dilakukan segmentasi target dengan memberi pelabelan. Label-label tersebut diberikan untuk mempermudah proses manajemen data. Label-label tersebut yaitu *hot*, *warm* dan *cold*. Label *hot* diberikan untuk target yang memiliki kemungkinan paling besar untuk menyetujui penawaran kerjasama. Label *warm* diberikan untuk target yang masih memerlukan pertimbangan dalam menyetujui penawaran kerjasama. Sedangkan label *cold* diberikan untuk target dengan kemungkinan paling kecil untuk menyetujui penawaran kerjasama.

Melihat tahapan-tahapan proses *customer development* yang kompleks seperti yang disebutkan dalam *task analysis* di atas, peneliti memiliki beberapa kesimpulan yaitu:

1. Banyaknya tahapan *customer development* sebanding dengan banyaknya data yang harus direkap. Hal ini membutuhkan ketelitian yang tinggi agar tidak terjadi *human error*.
2. Banyaknya tahapan yang harus dilalui membutuhkan pemetaan data yang tepat.
3. Proses pendataan dengan melakukan segmentasi target berdasarkan kemungkinan-kemungkinan yang terjadi pada proses *customer development*, sangat diperlukan untuk mempermudah proses manajemen data.
4. Fleksibilitas waktu dalam proses menunggu konfirmasi dan negosiasi membutuhkan daya ingat yang tinggi agar tidak lupa dalam melakukan

3.3.3. *Architecture Analysis*

Architecture analysis menggambarkan kerangka komponen teknologi yang digunakan dalam menunjang setiap aktivitas *user*. Dalam praktiknya selama ini, *operational marketing* hanya menggunakan perangkat *personal computer* yang sudah terinstal Microsoft Excel dalam mengelola data *customer development* seperti terlihat pada gambar 3.2.



Gambar 3.2. Arsitektur saat ini.

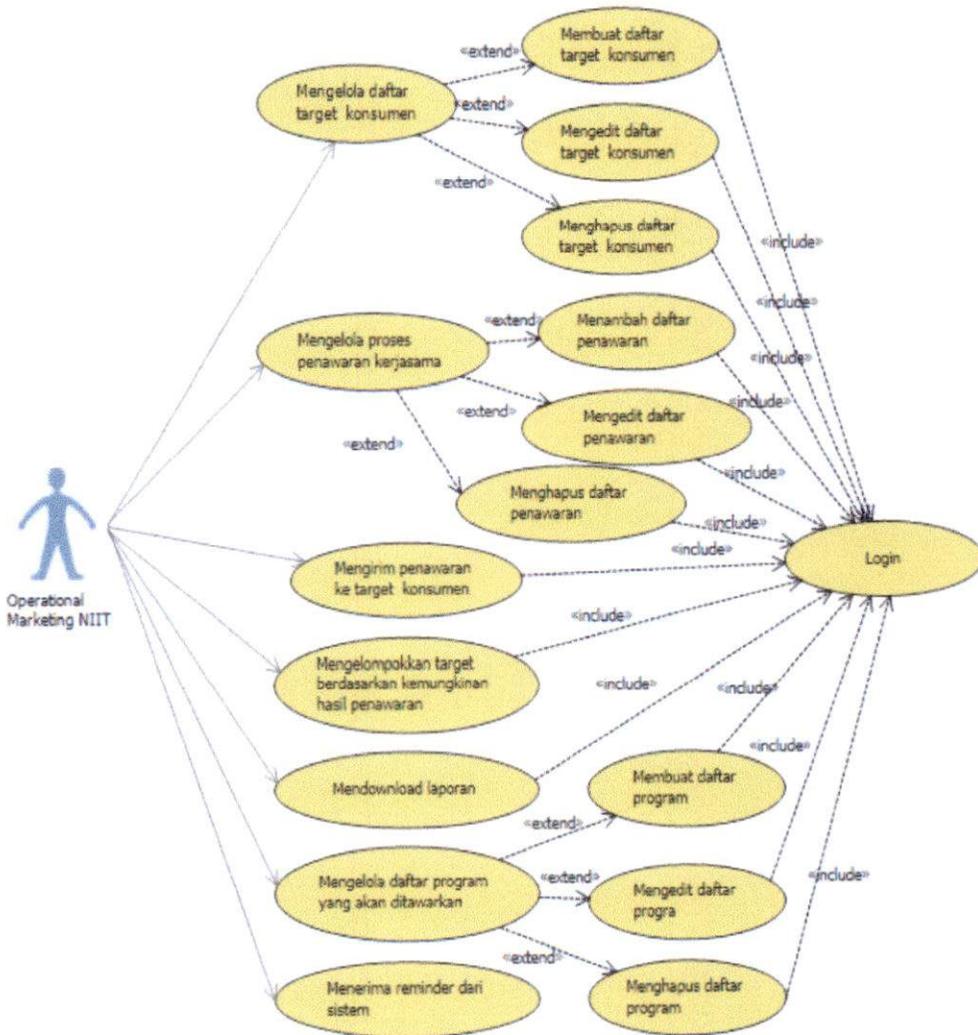
Dari *architecture analysis* di atas, peneliti mempunyai kesimpulan bahwa melihat dari segi aktifitas *customer development* yang kompleks, karakteristik *user* dengan kemampuan yang mumpuni dalam bekerja menggunakan komputer/TI, serta *goal* yang ingin dicapai *user*, pemilihan Microsoft Excel sebagai *tools* utama dalam bekerja kuranglah tepat. Untuk proses manajemen data dalam skala besar dan kompleks, Microsoft Excel mempunyai beberapa kelemahan diantaranya adalah ketidakmampuan Microsoft Excel untuk terintegrasi dengan file Microsoft Excel lainnya. Dalam hal ini, integrasi dibutuhkan untuk mempermudah proses pemetaan data. Microsoft Excel juga tidak mendukung kemampuan yang bisa memberikan peringatan dini, agar pihak *operational* divisi *marketing* NIIT Indonesia dapat tahu kapan harus bertindak terhadap suatu kondisi atau kemungkinan dalam *customer development*. Selain itu Microsoft Excel juga tidak mendukung kemampuan *automatic backup data*, yang dibutuhkan sebagai media pencadangan data jika sewaktu-waktu terjadi kehilangan data.

3.4. **Rancangan Sistem**

Rancangan sistem digunakan untuk mendefinisikan kebutuhan fungsional, kebutuhan non-fungsional serta kebutuhan arsitektur dari sistem yang akan dibuat.

3.4.1. Kebutuhan Fungsional dan Non-Fungsional

Dari ketiga aspek analisis kebutuhan yang telah dibahas sebelumnya, peneliti menyimpulkan kebutuhan fungsional dari sistem yang akan dibuat, seperti terlihat dalam *usecase diagram* pada gambar 3.3.



Gambar 3.3. *Usecase diagram* kebutuhan fungsional.

Berikut penjelasan lebih detail tentang kebutuhan fungsional seperti terlihat pada gambar di atas.

1. Sistem harus mampu mengelola data target konsumen serta memiliki kemampuan integrasi antara data satu dan data lainnya.
2. Sistem harus mampu mengelola daftar program yang akan ditawarkan

3. Sistem harus mampu mengirim penawaran secara langsung sehingga mempermudah proses pengajuan penawaran.
4. Sistem harus mampu mengelola proses penawaran pada tiap tahapan serta mengatur jadwal untuk bertemu.
5. Sistem harus mampu mengelompokkan target berdasarkan kemungkinan-kemungkinan hasil penawaran.
6. Sistem harus mampu memberikan *reminder* sebagai peringatan untuk meminimalisir kesalahan akibat lupa untuk merespon tindakan tertentu.
7. Sistem harus mampu mengunduh laporan mengenai kondisi penawaran pada tiap tahapan pengajuan penawaran yang sedang berlangsung.
8. Agar sistem lebih aman, maka diperlukan autentikasi dengan menambahkan fitur *login*.

Untuk kebutuhan non-fungsional sendiri terdapat beberapa aspek yang harus dipenuhi yaitu:

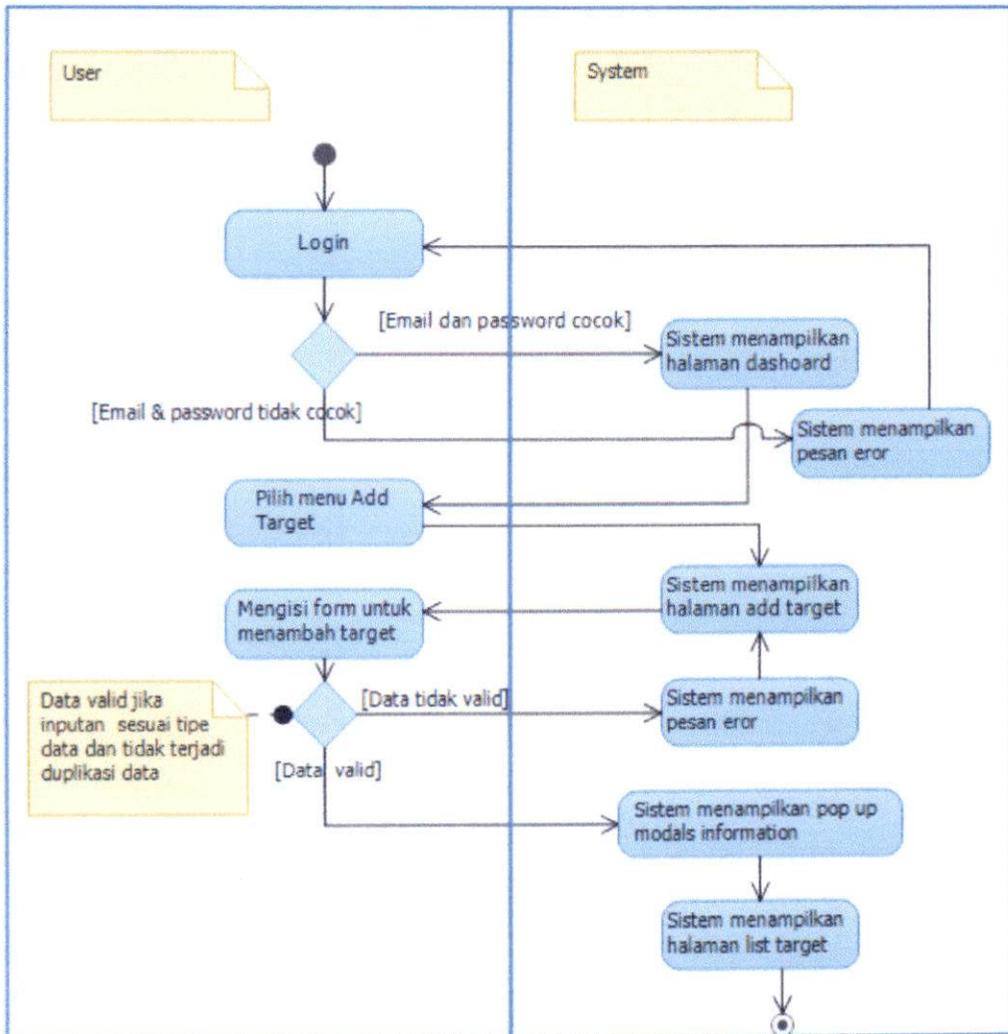
1. Sistem harus bisa diakses kapan saja dan dari mana saja. Melihat dari segi arsitektur hal ini bisa dipenuhi dengan memanfaatkan teknologi server.
2. Teks dalam sistem yang dibuat dan ditampilkan dalam *end-user* harus memakai bahasa inggris.

Dalam proses *customer development* yang sudah ada sekarang, peneliti menyimpulkan bahwa setiap tahapannya sudah tepat dan tidak perlu menambahkan atau mengurangi tahapan-tahapan tersebut. Akan tetapi proses yang kompleks dan rumit akan digantikan dengan sistem yang otomatis untuk mengurangi kemungkinan terjadinya *human error*. Proses-proses yang rawan terjadinya *human error* diantaranya yaitu:

1. Proses input data

Proses input data rawan akan kesalahan seperti halnya duplikasi data. Dalam pengelolaan data yang besar dan kompleks, beberapa data mempunyai relasi dengan data lainnya. Dengan sistem yang ada sekarang kemungkinan kesalahan dalam pemetaan data cukuplah besar. *Activity diagram* dalam

gambar 3.4. menunjukkan proses input data target dengan sistem validasi yang lebih bisa meminimalisir terjadinya *human error*.



Gambar 3.4. Activity diagram proses input data target.

Berikut penjelasan dari *activity diagram* di atas.

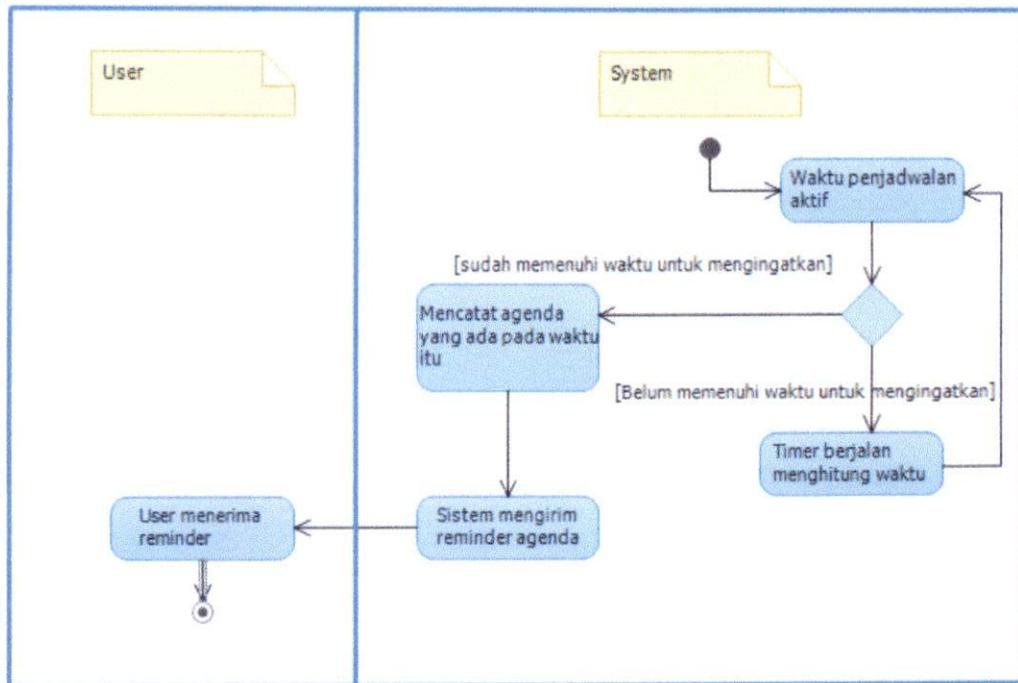
1. *User* harus *login* terlebih dahulu sebelum menggunakan sistem CRM.
2. Pada saat *login*, terdapat proses validasi untuk mengecek apakah data yang dimasukkan sesuai dengan data *user* yang ada, dalam hal ini *email* dan *password* haruslah sesuai satu sama lain.
3. Apabila data yang dimasukkan valid, maka sistem akan membawa *user* masuk ke halaman *dashboard*. Akan tetapi, jika data yang dimasukkan tidak valid, sistem akan menampilkan pesan *error*.

4. Halaman *dashboard* adalah halaman dimana semua informasi tentang proses penawaran terangkum. Pada halaman ini *user* mempunyai banyak pilihan untuk melakukan pengolahan data.
5. Untuk memasukkan data target, *user* harus memilih menu *add target* terlebih dahulu.
6. Setelah menu *add target* dipilih, kemudian akan muncul form isian untuk memasasukkan data target.
7. Setelah *user* selesai mengisi form isian tersebut, *user* bisa menekan tombol *submit* untuk menyimpan data. Ketika tombol *submit* ditekan akan terjadi proses validasi terhadap data yang dimasukkan. Jika data yang dimasukkan valid, maka data akan secara otomatis tersimpan di dalam *database* dan sistem akan menampilkan *success pop up modals*. Akan tetapi jika data yang dimasukkan tidak valid, akan muncul pesan *error*.

2. Proses Konfirmasi

Proses konfirmasi memakan waktu yang fleksibel. Hal ini memerlukan respon yang bervariasi pula. Dengan sifat dasar manusia yang pelupa hal ini menjadi salah satu kemungkinan terjadinya *human error*, disamping itu sistem yang ada sekarang tidak memungkinkan adanya *reminder*.

Gambaran mengenai tahapan *customer development* dengan kemampuan sistem yang terotomatisasi untuk memberikan *reminder* dapat dilihat dalam *activity*



Gambar 3.5. Activity diagram sistem reminder.

Berikut penjelasan mengenai *activity diagram* di atas.

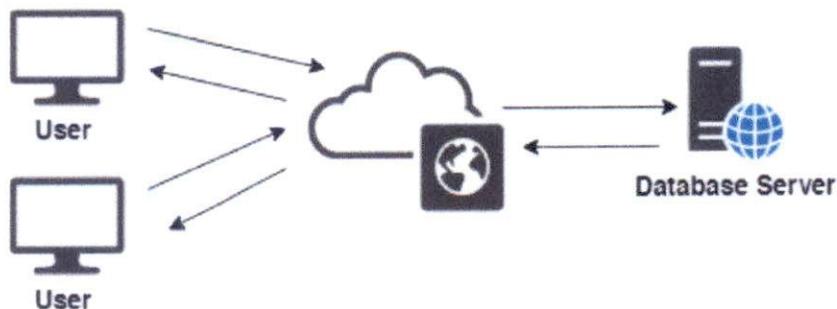
1. Sistem CRM yang dibuat memiliki fitur *reminder*, dengan memanfaatkan *job scheduler*. Ketika sistem diaktifkan, *job scheduler* akan secara otomatis berjalan pada bagian latar belakang sistem.
2. *Job scheduler* secara otomatis membaca waktu untuk selanjutnya memberikan *reminder*. Dalam hal ini *reminder* akan diberikan setiap sehari sekali.
3. *Reminder* akan dikirimkan kepada *user* melalui email. *Reminder* ini berisikan informasi tentang agenda apa yang harus dikerjakan pada hari berikutnya.

3.4.2. Kebutuhan Arsitektur

Pada penelitian ini memang tidak ada permintaan khusus dari *user* terkait arsitektur yang digunakan. Hal ini membuat peneliti lebih fleksibel dalam menentukan arsitektur yang digunakan. Melihat data dari kebutuhan fungsional dan non-fungsional di atas, peneliti memilih untuk membangun sistem dengan memanfaatkan teknologi server. Pemilihan teknologi server dapat memenuhi kriteria fleksibilitas akses karena bisa diakses kapan saja dan dimana saja selama

ada jaringan internet. Pada server pusat peneliti menggunakan Microsoft SQL Server 2014 sebagai *database server* kemudian pada *application server* peneliti menggunakan IIS untuk mengelola *web server*. Sistem CRM yang dibangun berupa *web application* yang dibangun dengan ASP.NET MVC5.

User dapat mengakses sistem CRM menggunakan *browser*. Nantinya *application server* akan memuat antarmuka *website* dan melakukan pengambilan data dari *database server*. Melalui antarmuka inilah *user* dapat mengelola data, baik membuat, menyimpan, menghapus dan memperbaharui data. Rancangan arsitektur yang diusulkan peneliti tampilan dalam gambar 3.6.



Gambar 3.6. Arsitektur sistem yang diusulkan

3.5. Rancangan Antarmuka

Rancangan antarmuka adalah dasar yang menggambarkan tampilan akhir dari sistem CRM yang akan dibuat, dimana nantinya *user* bisa langsung berinteraksi dengannya. Rancangan antarmuka memberikan gambaran setiap bagian pada sistem CRM dan menjelaskan keterkaitan antar bagiannya.

3.5.1. Rancangan Antarmuka Halaman *Login*

Login adalah halaman yang pertama kali tampil ketika *user* mengakses sistem CRM. Di sini terjadi proses autentikasi untuk memastikan hanya orang yang berhak yang bisa mengakses sistem CRM. Rancangan halaman *login* dapat dilihat

A Web Page

http://nitcare.com/login

NIITCRM

Use your email and password to login

Email

Password

[Forgot password?](#)

Gambar 3.7. Rancangan antarmuka halaman *login*.

3.5.2. Rancangan Antarmuka Halaman *Register*

Halaman *register* adalah halaman yang digunakan untuk mendaftarkan *user* baru. Penambahan *user* baru ini hanya bisa dilakukan oleh *user* utama (*parent user*) atau *user* yang telah didaftarkan sebelumnya. Tampilan dari rancangan halaman *register* bisa dilihat pada gambar 3.8.

A Web Page

http://nitcare.com/register

NIITCRM

Create new user

Email

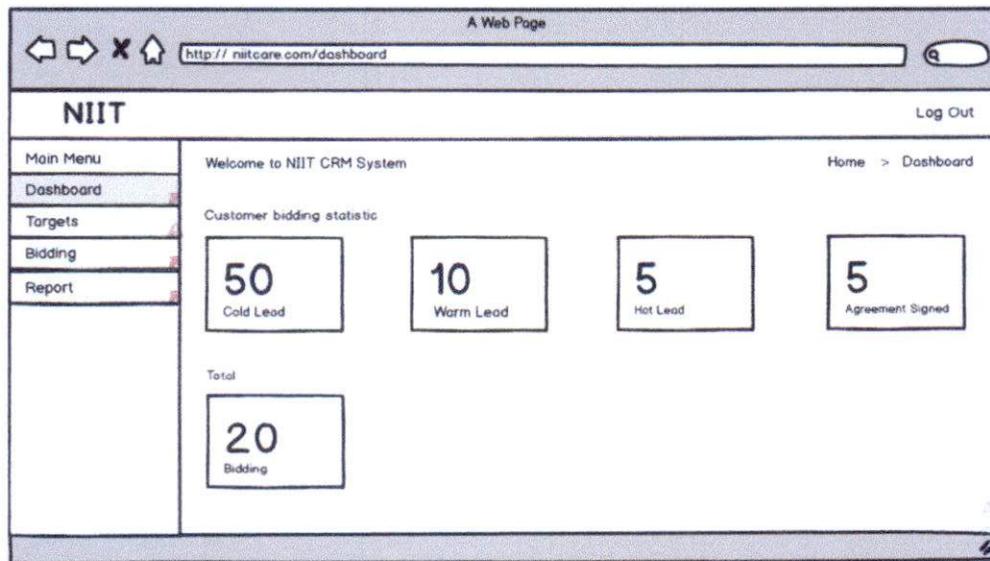
Password

Confirm Password

Gambar 3.8 Rancangan antarmuka halaman *register*.

3.5.3. Rancangan Antarmuka Halaman *Dashboard*

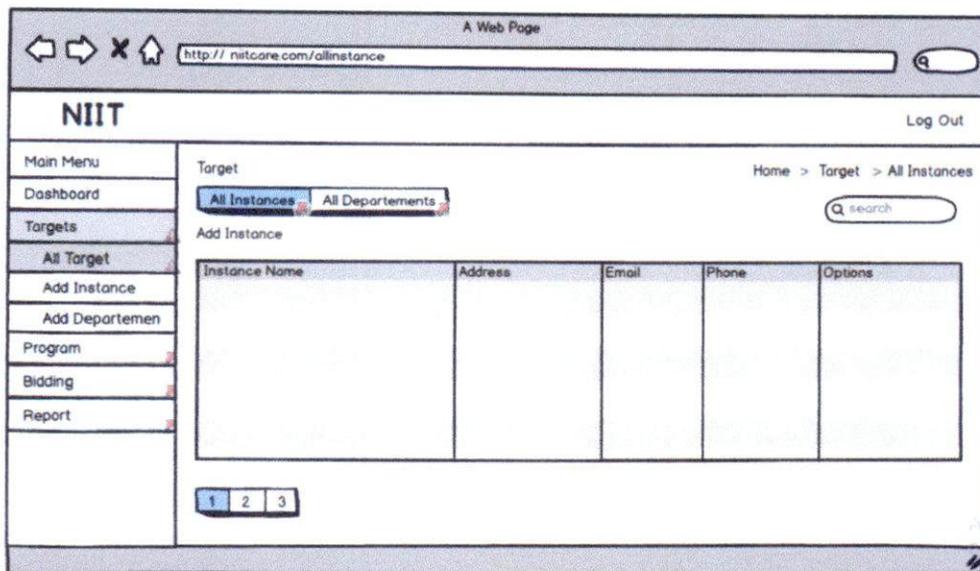
Setelah *user* berhasil *login*, *user* akan langsung diarahkan ke halaman *dashboard*. Halaman *dashboard* berisikan rangkuman mengenai informasi utama tentang seluruh aktifitas *customer development*. Rancangan antarmuka halaman *dashboard* dapat dilihat pada gambar 3.9.



Gambar 3.9. Rancangan antarmuka halaman *dashboard*.

3.5.4. Rancangan Antarmuka Halaman *All Target*

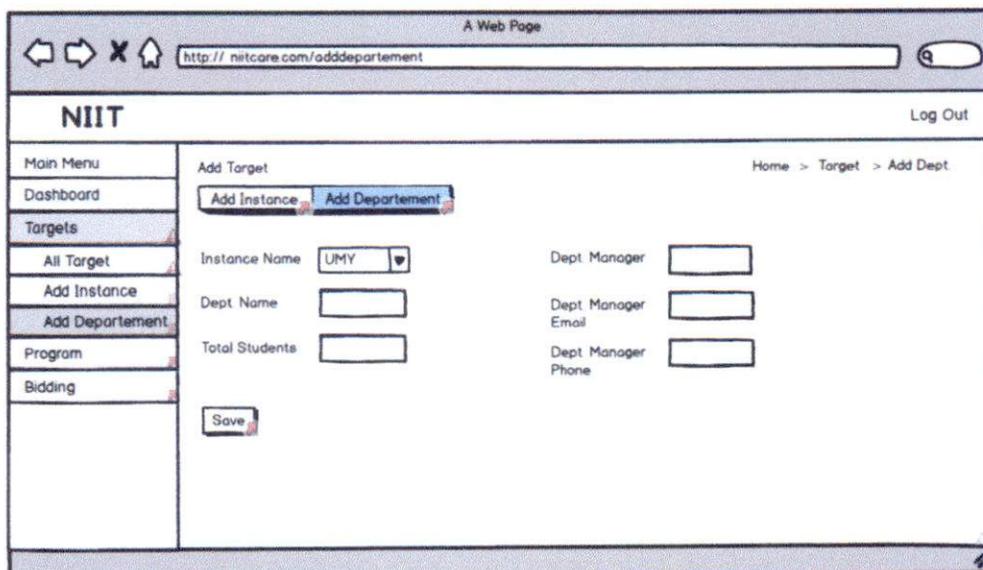
Halaman *all target* adalah halaman yang berisikan informasi mengenai daftar target yang sudah terdata oleh sistem. Target terdiri dari instansi dan instansi terdiri atas departemen-departemen. Rancangan antarmuka halaman *all target*,



Gambar 3.10. Rancangan antarmuka halaman *all target*.

3.5.5. Rancangan Antarmuka Halaman *Add Target*

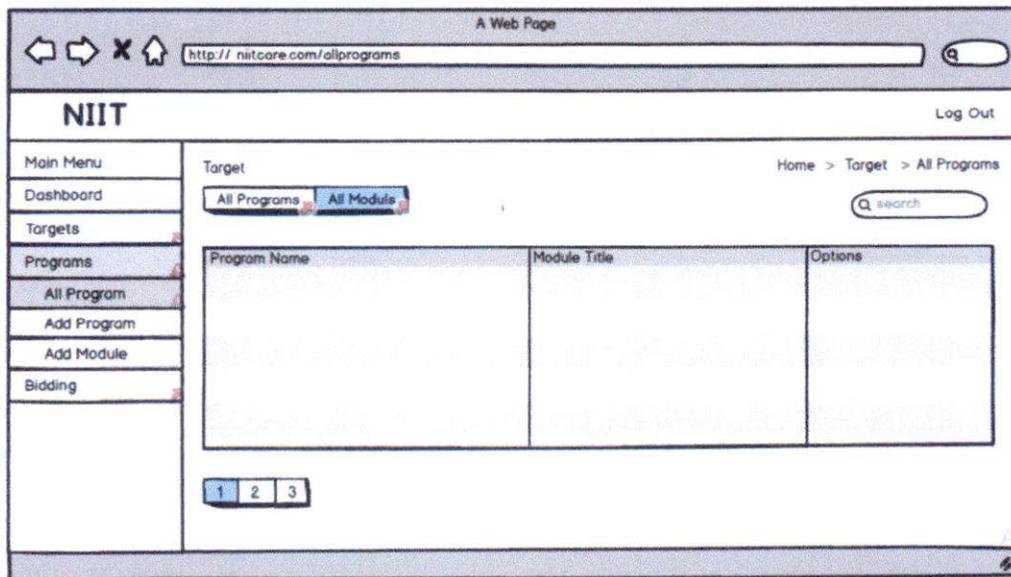
Halaman *add target* adalah halaman yang terdiri atas *add instance* dan *add department*. Halaman ini berisi form isian mengenai data instansi dan departemen yang akan dijadikan sebagai target penawaran. Rancangan antarmuka halaman *add target*, dapat dilihat pada gambar 3.11.



Gambar 3.11. Rancangan antarmuka halaman *add target*.

3.5.6. Rancangan Antarmuka Halaman *All Program*

Halaman *all program* adalah halaman yang berisikan daftar program yang sudah terdata pada sistem. *All program* terdiri dari daftar program, dimana setiap program terdiri atas modul-modul. Berikut rancangan antarmuka halaman *all program* peneliti sajikan dalam gambar 3.12.



Gambar 3.12. Rancangan antarmuka halaman *all program*.

3.5.7. Rancangan Antarmuka Halaman *Add Program*

Halaman ini terdiri dari *add program* dan *add module*. Halaman ini berisi form isian mengenai data program dan modul yang akan dijadikan sebagai program yang akan ditawarkan kepada target. Rancangan antarmuka halaman *add program*,

Gambar 3.13. Rancangan halaman *add program*.

3.5.8. Rancangan Antarmuka Halaman *All Bidding*

Halaman *all bidding* merupakan halaman yang berisikan daftar target yang sudah dikirim penawaran kerjasama. Rancangan antarmuka mengenai halaman ini dapat dilihat pada gambar 3.14.

Instansi Name	Dept. Name	Address	Preferred Program	Status	Info	Starting Date	Date Confirmation

Gambar 3.14. Rancangan antarmuka halaman *all bidding*.

3.5.9. Rancangan Antarmuka Halaman *Send Bidding*

Halaman *send bidding* merupakan halaman yang digunakan untuk mengirim penawaran kerjasama. Halaman ini berisi form isian sebagai data untuk mengajukan penawaran. Fitur *send bidding* memungkinkan *user* untuk memasukkan data *bidding* sekaligus mengirim penawaran kerjasama melalui *email*. Rancangan antarmuka halaman ini dapat dilihat pada gambar 3.15.

Gambar 3.15. Rancangan antarmuka halaman *send bidding*.

3.5.10. Rancangan Antarmuka Halaman *Add Bidding*

Hampir serupa dengan halaman *send bidding*, halaman *add bidding* adalah halaman yang digunakan untuk memasukkan data *bidding*. Akan tetapi pada fitur *add bidding* tidak ada proses pengiriman penawaran kerjasama melalui *email*. Halaman *add bidding* diperuntukkan untuk memasukkan data *bidding* dengan metode penawaran kerjasama melalui telepon. Rancangan mengenai halaman *add bidding* dapat dilihat pada gambar 3.16.

NIIT Log Out

Main Menu

Dashboard

Targets

Bidding

All Bidding

Send Bidding

Add Bidding

Cold Lead

Warm Lead

Hot Lead

Report

Bidding Home > Bidding > Add Bidding

Add New Bidding

Select Intance

Select Departement

Select Program

Select Module

Qualified

Bidding Stage

Status

Date of Current Status

Next Step

Date of Next Step

Save

Gambar 3.16. Rancangan halaman *add bidding*.

3.5.11. Rancangan Antarmuka Halaman *Cold Lead*

Halaman *cold lead* adalah halaman yang berisi informasi mengenai *bidding* dengan label *cold*. Label *cold* digunakan untuk menandai target dengan presentase kemungkinan paling kecil untuk menyetujui penawaran kerjasama. Rancangan halaman *cold lead* dapat dilihat pada gambar 3.17.

NIIT Log Out

Main Menu

Dashboard

Targets

Bidding

All Bidding

Send Bidding

Add Bidding

Cold Lead

Warm Lead

Hot Lead

Report

Bidding Home > Bidding > Cold Lead

search

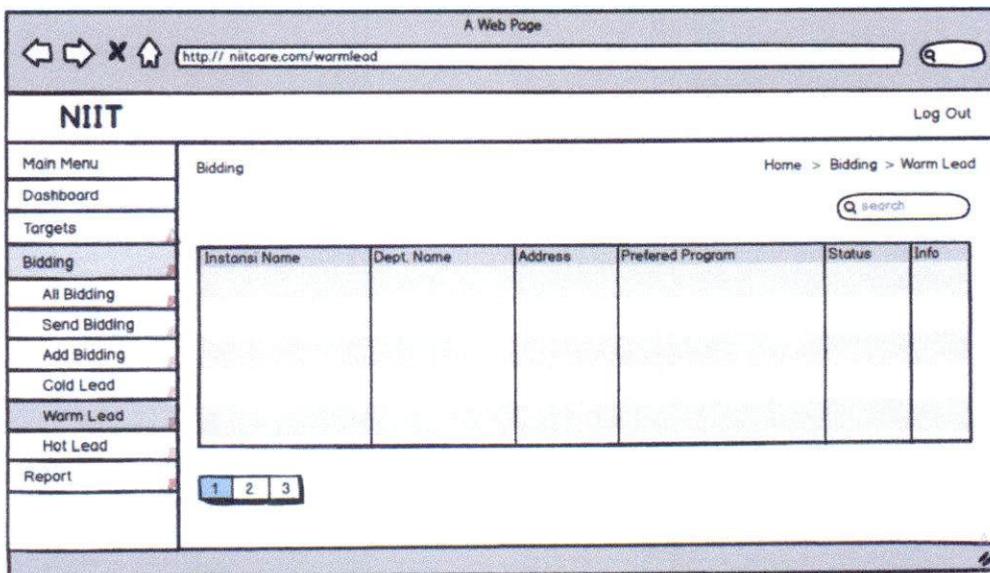
Instansi Name	Dept. Name	Address	Preferred Program	Status	Info

1 2 3

Gambar 3.17. Rancangan halaman *cold lead*

3.5.12. Rancangan Antarmuka Halaman *Warm Lead*

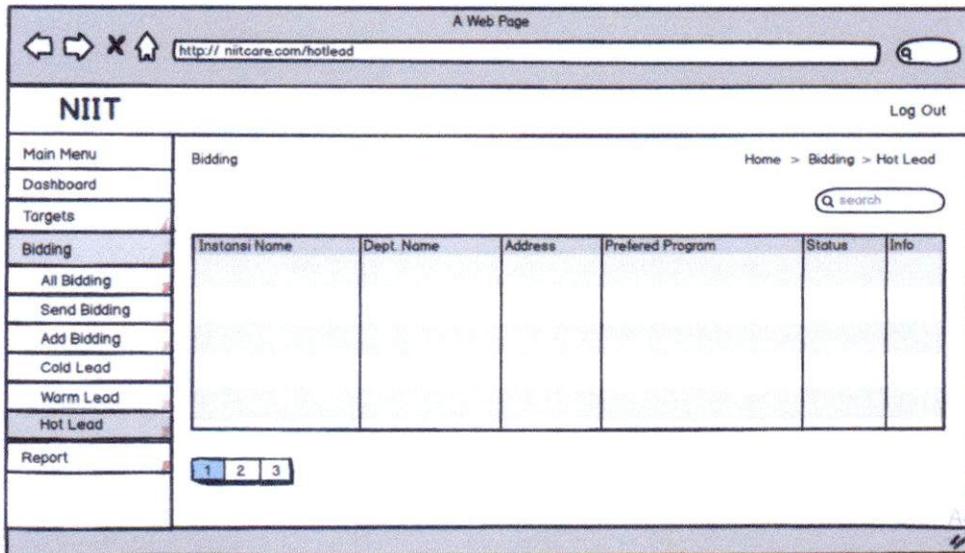
Halaman *warm lead* adalah halaman yang berisi informasi mengenai *bidding* dengan label *warm*. Label *warm* digunakan untuk menandai target yang masih mempertimbangkan untuk menyetujui penawaran kerjasama. Rancangan halaman *warm lead* dapat dilihat pada gambar 3.18.



Gambar 3.18. Rancangan halaman *warm lead*.

3.5.13. Rancangan Antarmuka Halaman *Hot Lead*

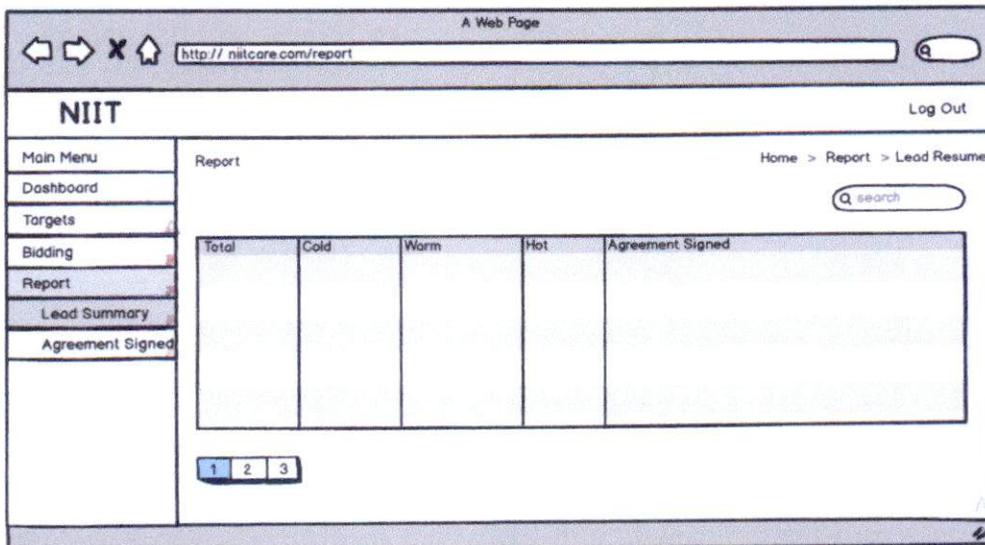
Halaman *hot lead* adalah halaman yang berisi informasi mengenai *bidding* dengan label *hot*. Label *hot* digunakan untuk menandai target dengan presentase kemungkinan paling besar untuk menyetujui penawaran kerjasama. Rancangan halaman *hot lead* dapat dilihat pada gambar 3.19.



Gambar 3.19. Rancangan halaman *hot lead*.

3.5.14. Rancangan Antarmuka Halaman *Lead Summary*

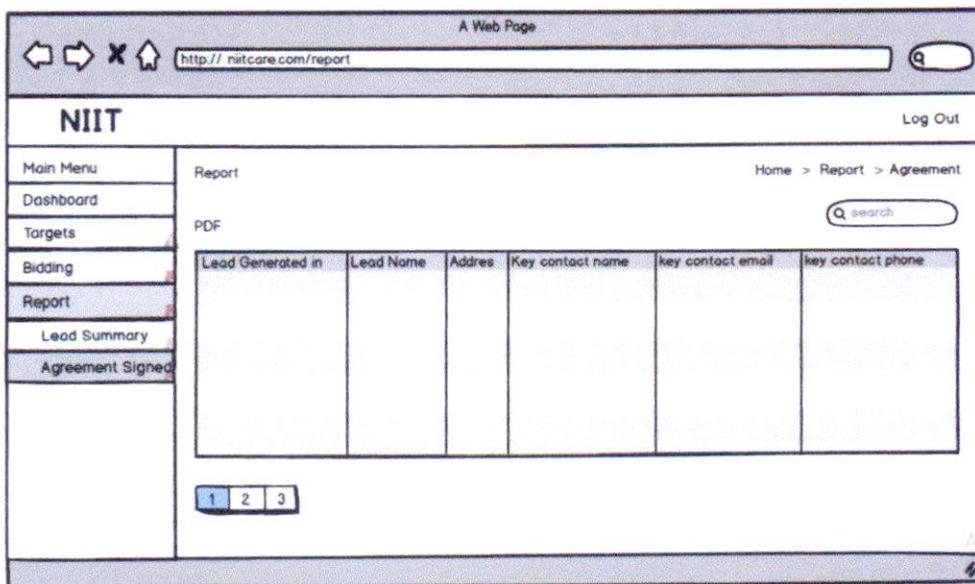
Halaman *lead summary* adalah halaman yang berisi rangkuman mengenai keseluruhan hasil dari proses *bidding*. Pada halaman ini kita bisa mengetahui informasi tentang berapa banyak jumlah *bidding* berdasarkan hasil pelabelan. Selain itu kita juga bisa melihat berapa banyak *bidding* yang disetujui. Rancangan halaman *lead summary* dapat dilihat pada gambar 3.20.



Gambar 3.20. Rancangan halaman *lead summary*.

3.5.15. Rancangan Antarmuka Halaman *Lead Agreement*

Halaman *lead agreement* adalah halaman yang berisi informasi mengenai *bidding* yang disetujui. Pada halaman ini, *user* bisa mendownload file *lead agreement* untuk digunakan sebagai laporan akhirdari proses *bidding*. Rancangan halaman *lead agreement* dapat dilihat pada gambar 3.21.



Gambar 3.21. Rancangan halaman *lead agreement*.

3.6. Peralatan Penelitian

Peralatan penelitian merupakan peralatan yang digunakan selama penelitian dilangsungkan, dengan mempertimbangkan data yang telah diperoleh dari proses analisis. Dalam penelitian ini peralatan dibedakan menjadi dua yaitu peralatan pengembang dan peralatan *user*. Peralatan pengembang adalah peralatan yang digunakan dalam proses pembuatan sistem CRM, sedangkan peralatan *user* adalah peralatan dengan minimum spesifikasi yang bisa dipakai oleh *user* untuk mengakses sistem CRM yang akan dibuat.

3.6.1. Peralatan Pengembang

Proses pengembangan sistem CRM untuk divisi *marketing* NIIT Indonesia menggunakan beberapa perangkat keras dan perangkat lunak dengan spesifikasi

1. Perangkat Keras

Spesifikasi dari perangkat keras yang digunakan dalam penelitian ini, peneliti sajikan dalam tabel berikut ini.

Tabel 3.2. Daftar spesifikasi perangkat keras yang dipakai.

Prosesor	Intel® Core™ i3 CPU M 370 @ 2.40GHz
Memori	2.00 GB
Sistem Model	Acer Aspire 4741G

2. Perangkat Lunak

Spesifikasi dari perangkat lunak yang peneliti gunakan dalam penelitian ini, peneliti sajikan dalam tabel 3.3.

Tabel 3.3. Daftar spesifikasi perangkat lunak yang dipakai.

Sistem Operasi	Windows 10 Enterprise
IDE Pengembang	Microsoft Visual Studio 2015
Manajemen Database	Microsoft SQL Server 2014
Sistem <i>Design</i>	Microsoft Visual Studio 2015, Balsamiq
Browser	Chrome Version 59.0.3071.115

3.6.2. Peralatan *User*

Adapun peralatan *user* yang dapat dipakai untuk mengakses sistem CRM dengan nyaman, harus memenuhi minimum spesifikasi seperti dalam tabel 3.4.

Tabel 3.4. Daftar minimum spesifikasi peralatan *user*.

Prosesor	Intel® Core™ i3 CPU M 370 @ 2.40GHz
Memori	2.00 GB
Browser	Chrome Version 59.0.3071.115