

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1. Tinjauan Pustaka

Sistem *Customer Relationship Management* atau lebih sering disingkat CRM telah banyak diimplementasikan saat ini, khususnya di dunia bisnis. Adelia (2011) dalam "Implementasi *Customer Relationship Management* (CRM) pada Sistem Reservasi Hotel berbasis *Website* dan Desktop" telah membuat sebuah sistem CRM berbasis aplikasi dekstop dan *website* dengan studi kasus Hotel Trio di Bandung. Aplikasi desktop yang dibuat memudahkan pihak hotel untuk dapat mengelola data master yang terdiri dari data tamu, data kamar, data fasilitas, dan data makanan. Data master tersebut berguna untuk pengelolaan hotel dalam melakukan pemesanan kamar, *check-in*, *check-out*, dan pembayaran. *Website* yang dibuat sangat memudahkan masyarakat untuk mendapatkan informasi mengenai hotel Trio dan untuk melakukan pemesanan kamar secara *online*. Sistem ini dibangun dengan menggunakan Bahasa pemrograman VB.

Ovi Dyantina, Mira Afrina, dan Ali Ibrahim (2012) dalam penelitiannya berjudul "Penerapan *Customer Relationship Management* (CRM) Berbasis Web (Studi Kasus Pada Sistem Informasi Pemasaran di Toko YEN-YEN)" membuat sebuah sistem CRM berbasis *website* untuk membantu memasarkan produk oleh-oleh toko YEN-YEN. Dikatakan bahwa sistem CRM yang dibuat memudahkan pelanggan dalam menyampaikan saran, kritik dan pertanyaan serta membantu pihak Toko YEN-YEN dalam pengelolaan dan penyimpanan data produk, data pelanggan dan data promosi.

Dalam penelitian lain berjudul "Aplikasi *Customer Relationship Management* pada *H₂O Boutique*" oleh Yulia, Shinta Wijaya dan Oviliani Y. Yuliana (2013), dikatakan bahwa aplikasi CRM merupakan strategi yang muncul sebagai jawaban atas semakin ketatnya persaingan dan tingginya tuntutan konsumen, terhadap tingkat layanan perusahaan yang dapat membantu pemilik perusahaan agar lebih dekat dan mengerti selera tiap pelanggannya. Hasil yang

dicapai adalah sebuah sistem informasi yang dapat mengelola data pelanggan, promosi produk, transaksi penjualan dan pembelian, serta beberapa layanan bagi pelanggan untuk membina hubungan baik dengan pelanggan.

Berdasarkan ketiga contoh penerapan sistem CRM di atas, secara garis besar dapat dilihat bahwa CRM memungkinkan pelaku usaha melakukan manajemen terhadap data konsumen, untuk tetap menjaga hubungan baik dengan konsumen tersebut. Adapun luaran yang diharapkan dengan terjalinnya hubungan baik tersebut adalah terciptanya pelayanan yang sesuai dengan kebutuhan konsumen. Melalui hal tersebut diharapkan nantinya perusahaan akan dapat menambah jumlah konsumen baru serta mempertahankan loyalitas konsumen lama. Sejalan dengan beberapa penelitian di atas, pada penelitian “Pengembangan Sistem *Customer Relationship Management* Berbasis *Web Application* dengan menggunakan ASP.NET MVC 5 Studi Kasus Divisi *Marketing* NIIT Indonesia” ini, tujuan penerapan sistem CRM tidak lepas dari pembentukan *good relationship* dengan konsumen.

Beberapa cara dapat digunakan untuk merealisasikan tujuan dari dibuatnya sistem CRM. Berdasarkan analisis dan pengujian yang sudah dilakukan oleh beberapa penelitian di atas, terdapat beberapa kelebihan dan kekurangan. Dalam penelitian yang dilakukan oleh Adelia (2011) sistem CRM yang mereka buat mempunyai fitur *SMS gateway* yang memungkinkan penyampaian informasi berupa pesan singkat melalui *website* kepada tamu. Dilihat dari sudut pandang estetika, rasanya pemilihan *SMS gateway* untuk era sekarang ini kurang begitu menarik. *SMS gateway* tidak memungkinkan untuk menyertakan informasi secara visual berupa file, gambar ataupun video. Padahal informasi yang menyertakan visualisasi akan lebih mudah dimengerti dan lebih menarik ketimbang informasi yang hanya berupa text saja.

Dalam dua penelitian lainnya yang dilakukan oleh Ovi Dyantina, Mira Afrina, Ali Ibrahim (2012) dan juga Yulia, Shinta Wijaya dan Oviliani Y. Yuliana (2013), mereka melakukan segmentasi konsumen dalam sistem CRM yang dibuat. Segmentasi konsumen merupakan salah satu strategi yang memberikan dampak

bagus dalam memahami kebutuhan serta karakteristik dari konsumen. Melalui segmentasi ini pihak pengelola CRM bisa membuat strategi bisnis yang tepat sasaran.

Adapun pada penelitian ini, peneliti membangun sistem CRM berbasis *web application* dengan fitur-fitur yang terintegrasi dan terotomatisasi untuk melakukan pengolahan data konsumen pada tiap tahapan *customer development* divisi *marketing* NIIT Indonesia. Sistem CRM yang dibuat juga memiliki kemampuan untuk melakukan segmentasi konsumen berdasarkan fase-fase dalam proses *customer development* divisi *marketing* NIIT Indonesia. Dengan memanfaatkan kemampuan sistem (*machine*) yang konstan sesuai dengan standar algoritma yang dibuat, nantinya fitur yang terotomatisasi ini akan mengurangi porsi kerja *operational marketing* dengan menggantinya dengan sistem yang otomatis. Melalui hal tersebut, diharapkan sistem dapat meminimalisir terjadinya *human error* akibat kesalahan *operational* yang dilakukan divisi *marketing* NIIT Indonesia. Nantinya sistem yang dibuat akan mempunyai fitur *reminder* sebagai media utama, yang menjadi penghubung antara *web application* dan *operational marketing* dalam mengambil tindakan terhadap suatu kondisi. Jika pada penelitian yang dilakukan sebelumnya oleh Adelia (2011) sistem penyampaian informasi menggunakan SMS *gateway*, pada penelitian ini sistem *reminder* akan disampaikan melalui *email*. Dipilihnya *email* karena kemampuan *email* dalam menyampaikan pesan yang lebih interaktif ketimbang SMS *gateway*.

2.2. Landasan Teori

2.2.1. Customer Relationship Management

Customer Relationship Management (CRM) adalah salah satu strategi bisnis untuk meningkatkan keuntungan, penghasilan, dan kepuasan konsumen dengan cara mengatur segmentasi konsumen, cara memberikan perlakuan terhadap konsumen, dan mengimplementasikan proses *customer-centric* (Buttle, F., 2009). Sedangkan menurut Ian Gordon (2002) dalam "*Best Practices: Customer Relationship Management*", CRM adalah serangkaian strategi dan proses yang mencintakan nilai baru dan saling menguntungkan bagi konsumen secara

individual, membangun preferensi untuk organisasi/perusahaan dan meningkatkan hasil bisnis selama hubungan kerja dengan pelanggan.

Ada empat komponen utama strategi dalam CRM (Gordon, 2002) yaitu:

1. *Technology*

Terdiri dari teknologi yang dipakai untuk mendukung penerapan proses CRM.

2. *People*

Keahlian, kemampuan dan sikap dari orang yang mengatur dan mengelola proses CRM.

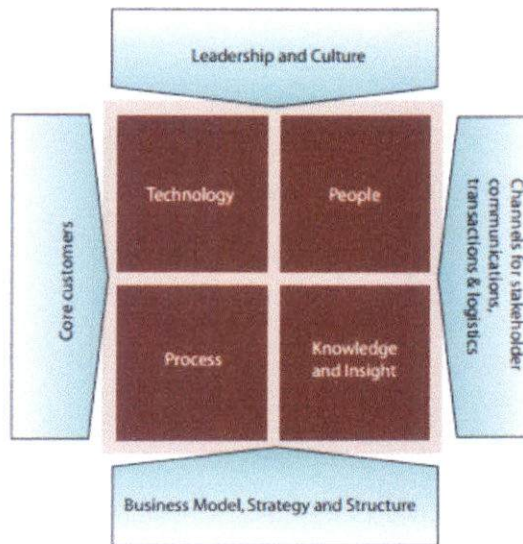
3. *Process*

Merupakan proses yang digunakan perusahaan dalam mengakses dan berinteraksi dengan pelanggan dalam menciptakan nilai baru dan kepuasan.

4. *Knowledge*

Merupakan pendekatan yang digunakan perusahaan untuk menambah nilai pada data konsumen sehingga mereka memperoleh pengetahuan dan pemahaman yang diperlukan untuk memperdalam suatu hubungan.

Penelitian ini sendiri berfokus pada komponen teknologi. Keempat komponen tersebut di atas dipengaruhi oleh faktor-faktor seperti konsumen tetap, kepemimpinan dan budaya dari perusahaan, *channel* yang digunakan oleh perusahaan untuk mendistribusikan *value* serta bisnis model yang diterapkan perusahaan seperti tertera pada Gambar 2.1



Gambar 2.1. Ilustrasi 4 komponen CRM menurut Gordon.

Pengertian lain mengatakan bahwa CRM adalah sebuah sistem informasi terintegrasi yang digunakan untuk merencanakan, menjadwalkan, dan mengendalikan aktivitas-aktivitas prapenjualan dan pascapenjualan dalam sebuah organisasi (Adelia, 2011).

2.2.2. Sistem Manajemen yang Terorganisasi dan Terintegrasi

Sistem informasi manajemen merupakan serangkaian sub-sistem informasi yang menyeluruh dan terkoordinasi, yang secara rasional mampu mentransformasikan data sehingga menjadi informasi dengan berbagai cara, guna meningkatkan produktivitas yang sesuai dengan gaya dan sifat manajer (Ismail, M., 2004).

Seperti yang sudah dijelaskan sebelumnya dalam rumusan masalah, bahwa salah satu masalah yang ingin diselesaikan dalam penelitian ini adalah bagaimana agar data konsumen yang diolah oleh divisi *marketing* NIIT Indonesia dapat terorganisasi dan terintegrasi, sehingga mempermudah proses manajemen data konsumen. Sistem manajemen yang terorganisasi mempunyai maksud bahwa sistem dikoordinasikan secara terpusat untuk menjamin bahwa data yang diproses dapat dioneraskan secara terencana dan terkoordinasi. Semuanya dilakukan untuk

menjamin bahwa informasi melewati dan menuju sub-sistem yang diperlukan, serta menjamin bahwa sistem informasi bekerja secara efisien (Ismail, M., 2004). Masih menurut Mutia Ismail (2004), terintegrasi berarti kegiatan dari masing-masing sub-sistem saling berkaitan satu dengan yang lainnya. Proses integrasi ini dilakukan dengan melewatkan data diantara sub-sistem

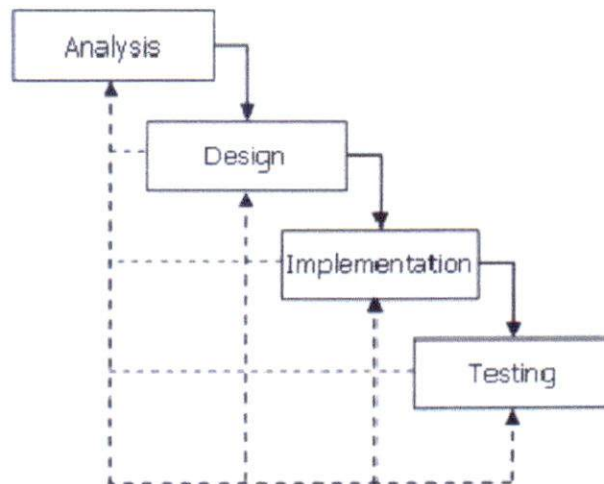
2.2.3. Sistem *Reminder*

Sistem *reminder* adalah sistem yang berfungsi untuk memberi peringatan terkait tindakan yang harus dilakukan pada waktu tertentu sebelum tiba waktu pelaksanaannya (Wulaningrum, Resty, at al., 2016). *Reminder* dibutuhkan agar pelaksana kegiatan dapat bersiap-siap sebelum datangnya waktu pelaksanaan. Dalam penelitian ini sistem *reminder* diimplementasikan melalui *email*.

2.2.4. *Software Development Life Cycle*

Untuk mempermudah proses pengembangan sistem CRM dalam penelitian ini, peneliti memakai metode *Waterfall* yang merupakan salah satu model dari *Software Development Life Cycle* (SDLC). SDLC merupakan suatu kerangka yang menggambarkan beberapa kegiatan yang dilakukan melalui beberapa tahapan dalam pembuatan sebuah *software* (Fatta, 2007). Beberapa proses atau tahapan dalam pengembangan sebuah *software* dalam SDLC diantaranya seperti *analysis*, *design*, *implementation*, dan *testing*. Masing-masing tahapan memungkinkan kita untuk memonitor waktu, biaya, serta kualitas dari *software* yang akan dibuat.

Metode *Waterfall* merupakan sebuah metode pengembangan aplikasi *sequential*, dimana suatu tahapan harus terselesaikan terlebih dahulu sebelum kita berpindah ke tahapan berikutnya (Balaji dan Murugaiyan, 2012). Gambaran mengenai tahapan atau fase-fase dalam metode *Waterfall* secara lebih rinci bisa



Gambar 2.2. *Waterfall phases.*

2.2.5. *Unified Modelling Language (UML)*

Unified Modelling Language (UML) merupakan sebuah standar bahasa yang digunakan untuk memvisualisasikan, merancang dan mendokumentasikan sistem piranti lunak (Sulistyorini, 2009). Melalui pemodelan menggunakan UML diharapkan nantinya sebuah sistem yang kompleks, dapat dipahami secara menyeluruh. UML menggambarkan informasi secara grafis tentang sebuah sistem yang terdiri dari komponen-komponen pembentuk sistem seperti kelas-kelas, obyek dan juga aksi serta koneksi yang ada di dalamnya. Dalam sebuah tim pengembang UML bisa digunakan sebagai sebuah alat komunikasi yang efektif karena setiap informasi yang detail tentang sistem yang akan dibuat sudah terdokumentasikan secara rinci.

Dalam praktiknya terdapat 9 diagram yang bisa digunakan untuk melakukan pemodelan dengan UML. Akan tetapi pada penelitian ini, peneliti hanya menggunakan dua diagram. Diagram-diagram tersebut yaitu:

1. *Use Case Diagram*

Use case diagram merupakan sebuah pemodelan yang menggambarkan hubungan antara actor dan aksi yang dapat dilakukan. Secara lebih sederhana diagram ini memvisualisasikan tentang apa yang bisa pengguna lakukan dengan sistem yang ada. Dengan *use case diagram* kita bisa

mengorganisasikan perilaku dari suatu sistem yang dibutuhkan serta diharapkan pengguna.

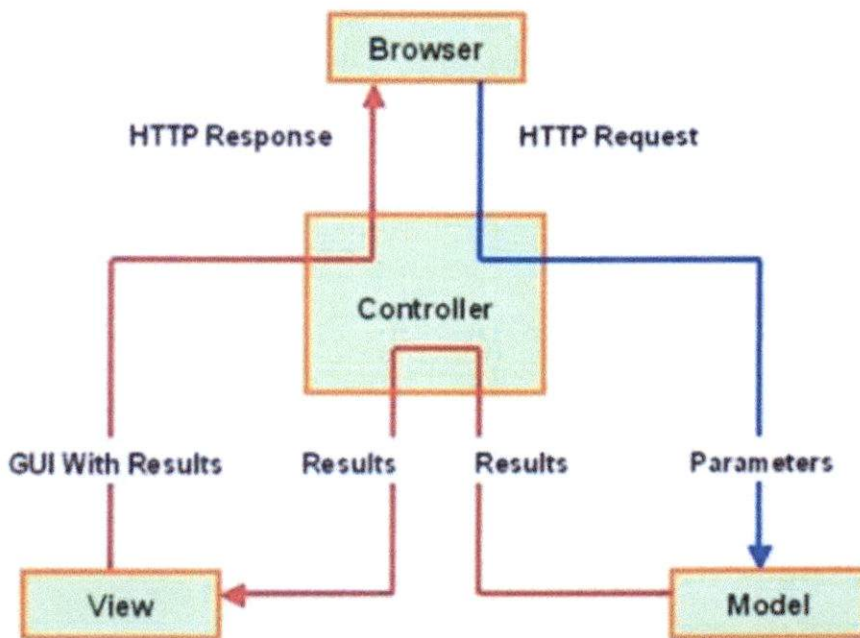
2. *Activity Diagram*

Activity diagram merupakan pemodelan yang menggambarkan aliran dari suatu aktifitas satu ke aktifitas lainnya. Diagram ini bersifat dinamis dan penting untuk menggambarkan bagaimana fungsi-fungsi dari sistem bekerja.

2.2.6. ASP.NET MVC 5

ASP.NET (*Active Server Pages .NET*) merupakan teknologi dari Microsoft yang dikhususkan untuk pengembangan aplikasi berbasis web dinamis pada *platform .NET Framework*. ASP.NET mendukung konsep MVC yang memberikan kemudahan dalam pengembangan aplikasi berbasis web. Pada penelitian ini peneliti menggunakan ASP.NET MVC 5.

Konsep MVC merupakan sebuah konsep yang memisahkan proses pengembangan aplikasi berbasis web berdasarkan 3 komponen utama yang membangun sebuah aplikasi seperti manipulasi data (*Model*), antarmuka pengguna (*View*), dan bagian yang menjadi kontrol dalam sebuah aplikasi web (*Controller*), (Pertiwi, D. H., 2011). Dengan arsitektur MVC ini, suatu aplikasi dapat dikembangkan sesuai dengan kemampuan *developer*, yaitu *programmer* menangani bagian *Model* dan *Controller* sedangkan *designer* menangani bagian *View*. Untuk lebih jelasnya hubungan antara ketiga komponen utama MVC dapat dilihat pada



Gambar 2.3. ASP.NET MVC life cycle.

2.2.7. Entity Framework

Menurut situs resminya ([https://msdn.microsoft.com/en-us/library/aa937723\(v=vs.113\).aspx](https://msdn.microsoft.com/en-us/library/aa937723(v=vs.113).aspx)), Entity Framework merupakan sebuah teknologi pemetaan objek yang memungkinkan *.NET developer* untuk memetakan data menggunakan *domain* yang spesifik. Entity Framework mendukung beberapa skema yang memudahkan *.NET developer* dalam mengolah data yaitu *Code First*, *Model First* dan *Database First*.

1. *Code First*

Code First mempunyai pengertian bahwa proses *development* dilakukan dengan menuliskan kode program terlebih dahulu, termasuk dengan membangun rancangan database melalui kode program tersebut. Dengan teknik ini, kode program yang ditulis akan secara otomatis membuat sebuah database ketika program dieksekusi.

2. *Model First*

Hampir sama dengan *Code First* pada *Model First* proses *development* dilakukan dengan menuliskan kode program terlebih dahulu, akan tetapi kode program yang ditulis pertama kali adalah kode program untuk

komponen model. Komponen model adalah komponen pembentuk database dari aplikasi yang dibuat.

3. *Database First*

Dalam *Database First*, proses development dilakukan dengan memanfaatkan database yang sudah dibangun sebelumnya dalam SQL Server. Database yang telah dibuat akan diimport melalui Visual Studio dengan bantuan ADO.Net Entity Framework.

Penelitian ini sendiri menggunakan proses *developmen Code First*.

2.2.8. Visual Studio

Menurut situs resminya (www.visualstudio.com), Visual Studio merupakan *Integrated Development Environment* (IDE) yang diproduksi oleh Microsoft sebagai alat pengembangan yang digunakan untuk membangun aplikasi. Visual Studio berjalan di atas platform .NET Framework yang mendukung berbagai macam bahasa pemrograman yaitu VB, C# (*CSharp*), dan C++. Banyak sekali fitur yang memanjakan *programmer* yang disediakan oleh Visual Studio, seperti *unit testing*, *package manager console*, *syntax checking*, *integrated compilation*, *debugging*, dan banyak fitur lain yang akan meningkatkan produktifitas *programmer* dalam membuat aplikasi.

2.2.9. Microsoft SQL Server 2014

Microsoft SQL Server 2014 merupakan produk dari Microsoft yang berfungsi sebagai *Relational Database Management System* (RDBMS). Database sendiri merupakan sekumpulan data yang saling terhubung secara logika. Microsoft SQL server 2014 bekerja dengan menggunakan bahasa kueri Transact-SQL yang merupakan implementasi dari SQL standar ANSI/ISO yang digunakan oleh Microsoft dan Sybase. Microsoft SQL Server dapat berkomunikasi lewat jaringan dengan protokol TDS (*Tabular Data Stream*) (Toledo, Chusman, 2007). Dengan menggunakan Microsoft SQL server 2014 inilah nantinya manajemen data konsumen NIIT Indonesia akan dikelola.

2.2.10. *Black Box Testing*

Pengujian atau *testing* merupakan bagian penting dan tak terpisahkan dalam peroses pengembangan aplikasi. Pengujian perlu dilakukan untuk memastikan bahwa aplikasi yang kita buat sesuai dengan rancangan awal dan dapat memenuhi kebutuhan. Dengan pengujian inilah seorang pengembang dapat menentukan kualitas dari aplikasi yang dibuatnya. Metode pengujian yang populer diantaranya adalah *White Box Testing* dan *Black Box Testing*. Adapun pada penelitian ini, proses pengujian dilakukan dengan menggunakan metode *Black Box Testing*.

Black Box Testing atau disebut juga *Functional Testing* adalah metode pengujian yang berfokus pada bagaimana aspek fungsionalitas dari aplikasi dapat bekerja. Berbeda dengan metode *White Box Testing*, metode ini tidak menyentuh sama sekali kode program yang telah dibuat. *Black Box Testing* melakukan pengujian dengan menitikberatkan pada output yang dihasilkan sebagai respons terhadap input dan kondisi eksekusi yang dipilih (Nidhara dan Dondethi, 2012).

Black Box Testing cenderung dilakukan untuk menemukan hal-hal berikut (Mustaqobal, Firdaus, dan Rahmadi, 2015):

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka (*interface errors*).
3. Kesalahan pada struktur data dan akses basis data.
4. Kesalahan performansi (*performance errors*).
5. Kesalahan inisialisasi dan terminasi