

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Telah ada beberapa penelitian yang dilakukan terkait dengan penerapan *E-Customer Relationship Management*, diantaranya adalah :

1. Menurut penelitian yang dilakukan oleh Evri Ekadiansyah, Harris Kurniawan (2015), Perancangan E-CRM (*Customer Relationship Management*) Pada PT. Sonya Fm Medan. Perkembangan internet dewasa ini mengalami kemajuan yang sangat pesat. Hal ini disebabkan karena internet mampu menutupi kelemahan media lain, seperti keterbatasan ruang dan waktu. CRM berfungsi untuk menghubungkan para pengguna website. Tujuannya untuk mempererat hubungan antara pihak radio, pihak luar, maupun penggemar radio yang menggunakan system informasi berbasis internet yang dapat di akses kapanpun dimana saja berada. Sehingga akan terjadi interaktif antar pengguna website dan memberikan efek timbal balik antara pengguna, dengan pihak radio demi meningkatkan perkembangan radio saat ini juga dimasa yang akan datang.
2. Menurut penelitian yang dilakukan oleh Rahmad Hidayat (2014), Sistem Informasi Ekspedisi Barang Dengan Metode E-CRM Untuk Meningkatkan Pelayanan Pelanggan. Pesatnya perkembangan Teknologi Informasi yang terjadi sekarang ini khususnya di negara kita Indonesia, Teknologi tidak lagi menjadi barang yang aneh, bahkan sangat diperlukan untuk mendukung

kinerja dari suatu organisasi. Salah satu metode yang dapat digunakan untuk melakukan improvisasi dan peningkatan pelayanan maupun promosi kepada pelanggan adalah dengan CRM yaitu Customer Relationship Management, PT. Zima Translogistic perusahaan yang bergerak dibidang ekspedisi dengan banyak pelanggan sangat membutuhkan proses sistem yang sangat cepat dan tepat sehingga Pelanggan akan mendapatkan kepuasan dalam pelayanan administrasi dan informasi yang akurat, dari permasalahan tersebut dibuatkan sistem e-CRM berbasis dekstop untuk melakukan transaksi pelanggan dan website untuk mendapatkan hasil informasi berkaitan transaksi pelanggan.

3. Berdasarkan penelitian yang dilakukan oleh Abdul Rahman Ariga (2018) dengan judul “Implementasi *Customer Relationship Management (CRM)* Pelayanan Pelanggan (*Corporate*) Pada PT. Telkom Witel Sumsel”. Tujuan penelitian yang dilakukan adalah menjaga hubungan baik dengan pelanggan. Karena dari hasil penelitian mempertahankan pelanggan yang sudah ada jauh lebih mudah dibandingkan dengan mendapatkan pelanggan baru. Persaingan bisnis yang sangat ketat akan sangat memerlukan langkah-langkah strategis untuk menghadapinya. Kehadiran *Customer Relationship Management (CRM)* akan mampu mengatasi masalah yang ada salah satu langkah yang juga dapat ditempuh untuk menghadapi masalah tersebut adalah dengan memanfaatkan teknologi yang ada yaitu dengan pemanfaatan *Customer Relationship Management (CRM)* berbasis web.

4. Berdasarkan penelitian yang dilakukan oleh Sandy Kosasi (2015), dengan judul “Perancangan Sistem Electronic Customer Relationship Management Untuk Mempertahankan Loyalitas Pelanggan”. Tujuan dari penelitian menghasilkan sistem E-CRM yang dapat membantu pihak manajemen distributor sparepart sepeda motor mengetahui dan mengelola semua pengetahuan dan kebutuhan pelanggan secara lebih personal sehingga dapat memberikan pelayanan yang lebih cepat dan berkualitas dalam meningkatkan kepuasan dan upaya mempertahankan loyalitas pelanggan. Hasil penelitian memperlihatkan sistem E-CRM dapat memberikan kemudahan dalam mengelola semua data pelanggannya dan menarik calon pelanggan baru melalui fasilitas yang komunikatif dan interaktif secara online.
5. Berdasarkan penelitian yang dilakukan oleh Hendra Darmawan (2018), dengan judul “Analisis dan Perancangan Electronic Customer Relationship Management pada Cetta Mom and Baby Spa Pontianak”. Tujuan dari penelitian ini untuk merancang aplikasi E-CRM pada perusahaan Cetta Mom and Baby Spa Pontianak dengan memberikan layanan informasi secara online dalam hal reservasi jasa treatment yang up-date. Metode penelitian yang digunakan dalam penelitian ini adalah metode penelitian dan pengembangan atau yang biasa dikenal dengan Research and Development. Hasil penelitian memperlihatkan adanya website E-CRM lebih mempermudah konsumen untuk melakukan transaksi dan berkomunikasi dengan pihak Perusahaan serta mempertahankan konsumen lamanya dan

juga untuk memperoleh konsumen baru dengan memasarkan barang kepada masyarakat luas.

II.2. Landasan Teori

II.2.1. Implementasi

Implementasi adalah pelaksanaan atau penerapan. Dalam setiap perumusan suatu kebijakan apakah menyangkut program maupun kegiatan-kegiatan selalu diiringi dengan suatu tindakan pelaksana atau implementasi. Karena betapa pun baiknya suatu kebijakan tanpa implementasi, maka tidak akan banyak berarti. Kata implementasi bermuara pada aktivitas, adanya aksi, tindakan, atau mekanisme suatu sistem. Ungkapan mekanisme mengandung arti bahwa implementasi bukan sekadar aktivitas, tetapi suatu kegiatan yang terencana dan dilakukan secara sungguh-sungguh berdasarkan acuan norma tertentu untuk mencapai tujuan kegiatan (Musrifah, 2017 : 5).

II.2.2. *E-Customer Relationship Management*

CRM adalah strategi bisnis yang terdiri dari software dan layanan yang didesain untuk meningkatkan keuntungan (profit), pendapatan (revenue) dan kepuasan pelanggan (customer satisfaction). Caranya adalah dengan membantu berbagai bentuk perusahaan untuk mengidentifikasi pelanggannya dengan tepat, memperoleh lebih banyak pelanggan dengan lebih cepat, dan mempertahankan kesetiaan pelanggannya. (Hidayat, 2014 : 2).

E-CRM (Electronic Customer Relationship Management) adalah suatu usaha mengelola relasi antara perusahaan dengan pelanggan dalam rangka peningkatan loyalitas pengkonsumsian produk-produk yang diproduksi oleh perusahaan dengan memanfaatkan peralatan elektronik seperti TV, Radio, Telepon dan Internet. (Hidayat, 2014 : 2).

Manfaat yang diperoleh dari penerapan CRM . (Hidayat, 2014 : 2) antara lain :

1. Memfungsikan penjualan dengan mengembangkan profil pelanggan.
2. Layanan penunjang melalui manajemen jaminan.
3. Pelacakan dan pemecahan problem.
4. Menjual produk yang dibutuhkan pelanggan berdasarkan pembeliannya.
5. Menawarkan status pelanggan yang lebih tinggi.
6. Menarik pelanggan baru dengan menawarkan layanan yang bersifat pribadi.
7. Menjaga pelanggan yang sudah ada.
8. Memberikan kemudahan konsumen untuk bertransaksi dengan perusahaan.
9. Mendesain ulang bisnis proses berhadapan muka dengan customer.
10. Meningkatkan profit perusahaan.

II.2.4. *Unified Modeling Language (UML)*

Unified Modeling Language (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan


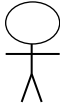
sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.



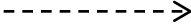
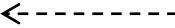
Dalam membangun perancangan sistem dengan alat bantu perancangan *Unified Modeling Language* (UML) ada beberapa tahapan yang akan dilakukan, yaitu sebagai berikut :

1. *Use Case Diagram*

Use Case Diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Adapun simbol *Use Case Diagram* yang dapat di lihat pada Tabel II.1. :

Tabel II.1. Simbol *Use Case diagram*

Gambar	Keterangan
	<p><i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i>.</p>
	<p>Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem.</p>

	Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki control terhadap <i>use case</i> .
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

(Sumber : Urva ; 2015)

2. *Class Diagram*

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

Class Diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu

operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti. Adapun simbol-simbol *Multipcity class diagram* yang dapat di lihat pada Tabel II.2. :

Tabel II.2. Multiplicity Class Diagram




Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

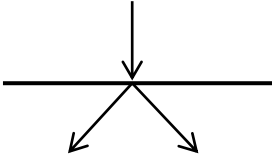
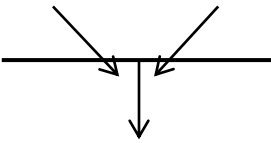
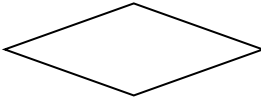

(Sumber : Urva ; 2015)

3. Activity Diagram

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Adapun simbol-simbol *activity diagram* yang dapat di lihat pada Tabel II.3. :

Tabel II.3. Simbol Activity Diagram

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.

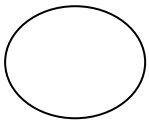
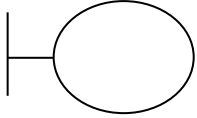
	<p><i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.</p>
	<p><i>Join</i> (penggabungan) atau <i>rake</i>, digunakan untuk menunjukkan adanya dekomposisi.</p>
	<p><i>Decision Points</i>, menggambarkan pilihan untuk pengambilan keputusan, <i>true</i>, <i>false</i>.</p>
	<p><i>Swimlane</i>, pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.</p>

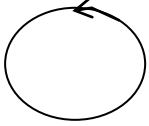
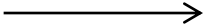
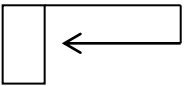


(Sumber : Urva ; 2015)

4. Sequence Diagram

Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Adapun simbol *Sequence Diagram* yang dapat di lihat pada Tabel II.4. : (Urva, 2015 : 2 – 4).

Tabel II.4. Simbol Sequence Diagram

Gambar	Keterangan
	<p><i>EntityClass</i>, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.</p>
	<p><i>Boundary Class</i>, berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>form</i> cetak.</p>

	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Urva ; 2015)

II.2.5. Normalisasi

Normalisasi adalah proses penyusunan table-table yang tidak redundan (double) yang dapat menyebabkan anomali pada saat terjadi manipulasi data seperti tambah, ubah dan hapus. Tujuan dari normalisasi adalah:

- a. Untuk meghilangkan kerangkapan data.
- b. Untuk mengurangi, kompleksitas.
- c. Untuk mempermudah pemodifikasian data. (Herlina Trisnawati : 2016)

Masing – masing level normalisasi mempunyai aturan tersendiri.

1. *First Normal Form*

Suatu tabel dikatakan dalam keadaan *first normal form* (1NF) jika :

- a. Tidak ada perulangan record data dalam tabel.
- b. Setiap sel memiliki satu nilai saja. Artinya tidak ada perulangan group dan array.

c. Data yang diinputkan memiliki tipe data yang sama dengan tipe data kolom dalam tabel.

2. *Second Normal Form*

Suatu tabel dikatakan dalam keadaan *Second Normal Form* (2NF) jika tabel tersebut sudah dalam keadaan *First Normal Form* (1NF) dan jika semua atribut yang bukan kunci tabel, baik *primary key* maupun *foreign key* tergantung pada semua kunci dalam tabel.

3. *Third Normal Form*

Suatu tabel dikatakan dalam keadaan *third normal form* (3NF) jika tabel tersebut sudah dalam keadaan *second normal form* (2NF) dan jika tidak terdapat ketergantungan yang transitif. Artinya, data-data yang mungkin diisi berulang-ulang dapat dibuat sebuah tabel baru.

4. *Boyce-Codd Normal Form* (BCNF)

Tabel dikatakan dalam keadaan *boyce-codd normal form* (BCNF) jika tabel tersebut dalam keadaan *third normal form* (3NF) dan setiap determinan adalah kunci kandidat.

5. *Fourth Normal Form* (4NF)

Suatu tabel dikatakan dalam keadaan *fourth normal form* (4NF) jika tabel tersebut dalam keadaan *boyce-codd normal form* (BCNF) dan jika tidak terdapat ketergantungan nilai ganda.

6. *Fiveth Normal Form* (5NF)

Tabel dikatakan dalam keadaan *Fiveth Normal Form* (5NF) jika tabel tersebut dalam keadaan *fourth normal form* (4NF) dan jika setiap

ketergantungan dalam join ada pada tabel sudah konsekuen dengan kunci kandidat pada tabel tersebut (Herlina Trisnawati : 2016)

II.2.6. Basis Data (*Database*)

Database adalah kumpulan dari semua data yang di perlukan oleh sistem. Dengan menggunakan *database*, beberapa apalikasi berbeda bisa saling terintegrasi (Mulyani, 2016:148).

II.2.6.1. *Database Management System* (DBMS)

Database Management System adalah sistem *software* yang memungkinkan pengguna untuk mendefinisikan, membuat, memelihara, dan kontrol akses ke *database* (Mulyani, 2016:170).

II.2.6.2. MySQL

MySQL adalah sebuah implementasi dari sistem manajemen basisdata relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya; SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Kehandalan suatu sistem basisdata (DBMS) dapat diketahui dari cara kerja pengoptimasi-nya dalam melakukan proses perintah-perintah SQL yang dibuat oleh pengguna maupun program-program aplikasi yang memanfaatkannya. Sebagai peladen basis data, MySQL mendukung operasi basisdata transaksional maupun operasi basisdata non-transaksional. Pada modus operasi non-transaksional, MySQL dapat dikatakan unggul dalam hal unjuk kerja dibandingkan perangkat lunak peladen basisdata kompetitor lainnya. Namun pada modus non-transaksional tidak ada jaminan atas reliabilitas terhadap data yang tersimpan, karenanya modus non-transaksional hanya cocok untuk jenis aplikasi yang tidak membutuhkan reliabilitas data seperti aplikasi blogging berbasis web (wordpress), CMS, dan sejenisnya. Untuk kebutuhan sistem yang ditujukan untuk bisnis sangat disarankan untuk menggunakan modus basisdata transaksional, hanya saja sebagai konsekuensinya unjuk kerja MySQL pada modus transaksional tidak secepat unjuk kerja pada modus non-transaksional.

MySQL pada awalnya diciptakan pada tahun 1979, oleh Michael "Monty" Widenius, seorang programmer komputer asal Swedia. Monty mengembangkan sebuah sistem database sederhana yang dinamakan UNIREG yang menggunakan koneksi low-level ISAM database engine dengan indexing. Pada saat itu Monty bekerja pada perusahaan bernama TcX di Swedia. TcX pada tahun 1994 mulai mengembangkan aplikasi berbasis web, dan berencana menggunakan UNIREG sebagai sistem database. Namun sayangnya, UNIREG dianggap tidak cocok untuk database yang dinamis seperti web. (Herpendi : 2016)

II.2.7. PHP

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah web dan bisa digunakan pada dokumen HTML. PHP dirancang untuk dapat bekerja sama dengan database server dan dibuat sedemikian rupa sehingga pembuatan dokumen HTML yang dapat mengakses database menjadi begitu mudah. Tujuan dari bahasa scripting ini adalah untuk membuat aplikasi dimana aplikasi tersebut yang dibangun oleh PHP pada umumnya akan memberikan hasil pada web browser, tetapi proses secara keseluruhan dijalankan di server. (Saipul Anwar : 2016)

II.2.8. Aplikasi Berbasis Web

Menurut Agusvianto (2017 : 2), Aplikasi berbasis *web* adalah aplikasi yang dijalankan melalui *browser*. Aplikasi aplikasi seperti ini pertama kali dibangun hanya dengan menggunakan bahasa yang disebut dengan HTML (*HyperText Markup Language*) dan protokol yang digunakan dan dinamakan HTTP (*HyperText Transfer Protokol*). Namun, tentu saja hal hal seperti ini memiliki kelemahan. Semua perubahan harus dilakukan pada level aplikasi. Pada perkembangan berikutnya, sejumlah skrip dan objek saat ini, banyak skrip seperti itu antara lain diantaranya yaitu PHP, ASP, *ASP.NET* sedangkan contoh yang berupa objek antara lain adalah *applet (Java)*.