

BAB II

LANDASAN TEORI

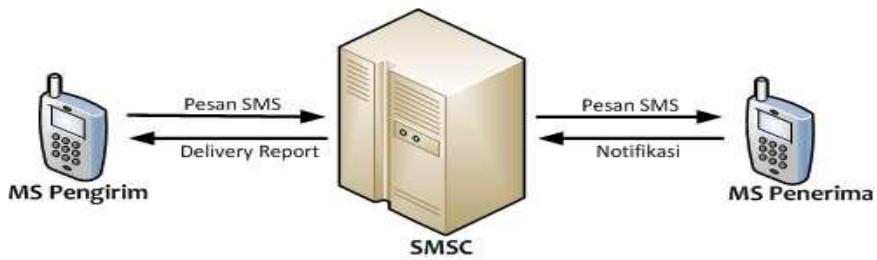
2.1 SMS (*Short Message Service*)

(Pakpahan, 2014) *Short Message Service (SMS)* adalah suatu fasilitas untuk mengirim dan menerima suatu pesan singkat berupa teks melalui perangkat nirkabel, yaitu perangkat komunikasi *telepon selular*, dalam hal ini perangkat nirkabel yang digunakan adalah *telepon selular*. Salah satu kelebihan dari *SMS* adalah biaya yang murah. Selain itu *SMS* merupakan metode *store* dan *forward* sehingga keuntungan yang didapat adalah pada saat *telepon selular* penerima tidak dapat dijangkau, dalam arti tidak aktif atau diluar *service area*, penerima tetap dapat menerima *SMS* apabila *telepon selular* tersebut sudah aktif kembali. *SMS* menyediakan mekanisme untuk mengirimkan pesan singkat dari dan menuju media-media *wireless* dengan menggunakan sebuah *Short Messaging Service Center (SMSC)*, yang bertindak sebagai sistem yang berfungsi menyimpan dan mengirimkan kembali pesan-pesan singkat.

Dikutip dari (*Jurnal Informatika Darmajaya, Vol.13, No.2, Desember 20013* Bobby Ghani dan Nurfiana) Aplikasi mobile dengan memanfaatkan layanan *SMS* (*Short Message Service*) merupakan suatu teknologi layanan penyampaian pesan dengan singkat. Kegemaran pengguna dalam ber-SMS yang lebih murah, praktis dan terdapat pada semua jenis tipe ponsel.

2.2.1 Cara Kerja *SMS*

Mekanisme dari sistem *SMS* ialah *store and forward*, dimana sistem dapat melakukan pengiriman *short message* dari satu terminal ke terminal lainnya. Hal ini dapat dilakukan berkat adanya sebuah entitas dalam sistem *SMS* yang bernama *Short Message Service Center (SMSC)*, disebut juga *Message Center (MC)*. *SMSC* merupakan sebuah perangkat yang melakukan tugas *store and forward traffic short message* seperti pada gambar 2.1 tersebut :



Gambar 2.1 Mekanisme pengiriman SMS (*Store and Forward*)

Pengiriman pesan *SMS* secara *store and forward* yaitu pengirim *SMS* memasukkan pesan *SMS* dan nomor tujuan dan kemudian mengirimkannya (*store*) ke *server SMS (SMS Center)* yang kemudian bertanggung jawab mengirimkan pesan *SMS* tersebut (*forward*) ke nomor *telepon* tujuan.

Hal ini berarti bahwa pengirim dan penerima *SMS* tidak berada dalam status berhubungan (*connected*) satu sama lain, ketika akan saling bertukar pesan *SMS*. Pesan yang dikirim oleh pengirim ke *SMSC* yang kemudian menunggu untuk dapat meneruskan pesan tersebut ke penerima. Ketika status penerima dalam keadaan aktif pesan segera dikirim oleh *SMSC* ke nomor tujuan beserta isi pesan pengirim. Pengirim akan menerima *delivery report* bahwa pesan telah terkirim.

Dikutip dari (*Jurnal Informatika Darmajaya, Vol.7, No.1, Juni 2007* Rangga F Dan Chairani) Cara Kerja SMS Secara sederhana pengiriman SMS hingga ke nomor tujuan adalah sebagai berikut:

1. Pesan SMS dikirim ke SMS Center (SMSC) milik operator asal SMS.
2. Setelah pesan tersebut diproses secara internal, SMS Center mengirimkan permintaan informasi penerima pesan ke Home Location Register (HLR).
3. Kemudian SMS Center mengirim pesan SMS tadi ke Mobile Switching Center (MSC)
4. Di MSC inilah informasi tentang kondisi jalur penerima pesan didapat dan dikumpulkan, yang datanya diambil dari lisitor Location Register (VLR) Dalam proses inilah terkadang memerlukan autentifikasi, apakah SMS bisa diterima nomor tujuan atau tidak

5. Setelah itu MSC meneruskan pesan ke server operator
6. Selelah SMS dikirim, MSC mengembalikan informasi keberhasilan pengiriman ke SMS Center
7. Akhirnya SMS Center melaporkan status pengiriman SMS kembali ke pengirim pesan.

2.2.2 Jenis-Jenis Aplikasi SMS

Jenis-jenis aplikasi *SMS* memiliki tiga macam, yaitu:

1. *SMS Premium*

jenis aplikasi *SMS* dua arah dengan menggunakan 4(empat) digit nomor unik seperti 9688. Contoh *SMS Quiz* dan *SMS Polling*.

2. *SMS Broadcast*

jenis aplikasi satu arah berupa pengiriman pesan ke banyak nomor sekaligus. Contoh *SMS promo* dan *SMS kampanye*.

3 . *SMS Gateway*

jenis aplikasi *SMS* dua arah, dengan keunikan bahwa semua tarif yang diberlakukan adalah tarif *SMS* normal sesuai dengan apa yang diberlakukan oleh operator. Karena sifatnya yang dua arah, maka jenis *SMS* ini sangat cocok dijadikan sebagai *SMS Center* organisasi, perusahaan atau instansi.

2.2.3 SMS Gateway

SMS Gateway adalah sebuah perangkat yang menawarkan layanan transit *SMS*, mentransformasikan pesan ke *jaringan selular* dari media lain, atau sebaliknya, sehingga memungkinkan pengiriman atau penerimaan pesan *SMS* dengan atau tanpa menggunakan ponsel. Sebagaimana penjelasan diatas, *SMS Gateway* dapat terhubung ke media lain seperti perangkat *SMSC* dan *server* milik *Content Provider* melalui link *IP* untuk memproses suatu layanan *SMS*. Sebuah sistem *SMS Gateway*, umumnya terdiri komponen *Hardware Server Komputer* yang dilengkapi dengan perangkat jaringan dan *Software Aplikasi* yang digunakan untuk pengolahan pesan. Dan untuk sebuah sistem yang besar umumnya menggunakan Database untuk penyimpanan data.

2.3 Manajemen Hubungan Pelanggan CRM

(Didip Diandra, 2016) adalah suatu jenis manajemen yang secara khusus membahas teori mengenai penanganan hubungan antara perusahaan dengan pelanggannya dengan tujuan meningkatkan nilai perusahaan di mata para pelanggannya.

Pengertian lain mengatakan bahwa ia adalah sebuah sistem informasi yang terintegrasi yang digunakan untuk merencanakan, menjadwalkan, dan mengendalikan aktivitas-aktivitas prapenjualan dan pasca penjualan dalam sebuah organisasi. CRM melingkupi semua aspek yang berhubungan dengan calon pelanggan dan pelanggan saat ini, termasuk di dalamnya adalah pusat panggilan (*call center*), tenaga penjualan (*sales force*), pemasaran, dukungan teknis (*technical support*) dan layanan lapangan (*field service*).

2.3.1 Fungsi-Fungsi Dalam CRM

Sebuah sistem CRM harus bisa menjalankan fungsi:

1. Mengidentifikasi faktor-faktor yang penting bagi pelanggan.
2. Mengusung falsafah *customer-oriented* (*customer centric*)
3. Mengadopsi pengukuran berdasarkan sudut pandang pelanggan
4. Membangun proses ujung ke ujung dalam melayani pelanggan
5. Menyediakan dukungan pelanggan yang sempurna
6. Menangani keluhan/komplaining pelanggan
7. Mencatat dan mengikuti semua aspek dalam penjualan
8. Membuat informasi holistik tentang informasi layanan dan penjualan dari pelanggan

2.3.2 Mengimplementasikan CRM

Customer relationship management adalah strategi tingkat korporasi, yang berfokus pada pembangunan dan pemeliharaan hubungan dengan pelanggan. Beberapa paket perangkat lunak telah tersedia dengan pendekatan yang berbeda-beda terhadap CRM. Bagaimanapun, CRM bukanlah teknologi itu sendiri, tapi ia adalah pendekatan holistik terhadap falsafah organisasi, yang menekankan hubungan yang erat dengan pelanggan. CRM mengurus filosofi organisasi pada semua tingkatan, termasuk kebijakan dan proses, customer

service, pelatihan pegawai, pemasaran, dana manajemen sistem dan informasi. Sistem CRM mengintegrasikan pemasaran, penjualan, dan customer service dari ujung ke ujung.

2.4 Iklan

(Enung Nuraeni, 2015) adalah suatu pesan tentang barang/jasa (produk) yang dibuat oleh produser/pemrakasa yang disampaikan lewat media (cetak, audio, elektronik) yang di tujuhan kepada masyarakat. Tujuan iklan adalah agar masyarakat tertarik untuk membeli atau menggunakan barang atau jasa tersebut.

2.4.1 Pengertian Iklan

1. Iklan adalah berita pesanan untuk mendorong dan membujuk orang agar tertarik pada barang atau jasa yang ditawarkan.
2. Iklan adalah kegiatan memberitahukan suatu hal, barang, atau jasa melalui media masa.
3. Media yang digunakan antara lain televisi, radio, Koran, majalah, internet, dan sebagainya.
4. Iklan berfungsi untuk menawarkan barang atau jasa kepada orang banyak melalui berbagai media.

2.4.2 Macam-macam Iklan

Berdasarkan isinya, iklan dikelompokan menjadi iklan pengumuman, penerangan, reklame, dan iklan baris.

1. Iklan Pengumuman

Iklan pengumuman adalah iklan yang isinya menyampaikan pengumuman mengenai suatu kegiatan, peristiwa, ataupun yang lainnya.

2. Iklan Penerangan

Iklan penerangan adalah iklan yang isinya memberikan penerangan tentang suatu hal

3. Iklan Reklame

1. Iklan reklame adalah iklan yang isinya mempromosikan barang atau jasa yang hendak dijual.

2. Iklan reklame bersifat mencari keuntungan dan disampaikan secara berulang-ulang melalui media massa.
3. Iklan reklame selalu memuji atau menyanjung sesuatu yang ditawarkan. Misalnya, dengan megatakan tiada banding, satu-satunya, tiada duanya, dan sebagainya.

2.5 Perangkat Lunak Yang Digunakan

Untuk membangun *website* diperlukan beberapa perangkat lunak yang digunakan. Beberapa perangkat lunak yang digunakan adalah sebagai berikut:

2.5.1 Aplikasi Web

(Sukamoto dan Shalahudin, 2013), aplikasi web adalah jenis aplikasi yang diakses melalui *browser*, misalnya *Internet Explorer* dan *Mozilla Firefox*. Aplikasi web juga merupakan suatu perangkat lunak komputer yang dikodekan dalam bahasa pemrograman yang mendukung perangkat lunak berbasis web seperti HTML, JavaScript, CSS, Ruby, Python, Php, Java dan bahasa pemrograman lainnya..

2.5.2 Star Unified Modelling

(Suardika, 2013), *Star UML* adalah platform pemodelan perangkat lunak yang mendukung UML (*Unified Modeling Language*). *Star UML* yang berbasiskan pada UML versi 1.4, menyediakan sebelas jenis Diagram yang berbeda, dan mendukung notasi UML 2.0. *Star UML* juga secara aktif mendukung konsep UML profile. *Star UML* unggul dalam hal kustomisasi lingkungan kerja pengguna, dan memiliki ekstensibilitas tinggi pada fungsionalitasnya. *Star UML* mengklaim diri sebagai salah satu alat pemodelan perangkat lunak termuka yang menjamin dapat memaksimalkan produktivitas dan kualitas proyek perangkat lunak anda.

2.5.3 Php

(Achmad Solichin, 2016), menyatakan bahwa PHP adalah skrip yang dijalankan di *server*. Jadi, konsepnya berbeda dengan *JavaScript* yang dijalankan pada sisi klien. Keuntungan penggunaan PHP, kode yang

menyusun program tidak perlu dibagikan ke pemakai yang berarti bahwa kerahasiaan kode dapat dilindungi. Hal menarik yang didukung oleh PHP tetapi tidak mungkin dilakukan oleh *JavaScript* adalah kenyataan bahwa PHP bisa digunakan untuk mengakses berbagai macam *database*, seperti *Access*, *Oracle*, *MySQL*, dan lain-lain. Seperti halnya *JavaScript*, kode PHP dapat disisipkan pada kode HTML. Selain itu PHP juga bisa digunakan untuk menghasilkan kode-kode HTML. Skrip PHP diawali dengan <?php dan diakhiri dengan ?>. Di dalam pasangan tanda tersebut terdapat pernyataan-pernyataan PHP. Seperti halnya pada *JavaScript*, antar pernyataan harus dipisahkan oleh tanda titik-koma (;).

2.5.4 *Html*

(Komputindo, 2016), menyatakan bahwa HTML adalah bahasa dengan tanda-tanda khusus yang digunakan di awal era web untuk menyajikan informasi. Kode HTML di awali dengan <html> dan diakhiri dengan </html>. Namun sebelum <html> diharuskan untuk diberikan deklarasi <!DOCTYPE>.

Fungsi deklarasi ini adalah sebagai *Public Text Identifier*. Beberapa hal penting dalam kode HTML :

1. Tanda <> men yatakan sebuah tag.
2. Pada umumnya tag berpasangan. Contoh, <html> dengan </html>, <head> dengan </head> dan <body> dengan </body>.
3. Tag yang tidak berpasangan antara lain
 dan <hr/>
4. Pada tag yang berpasangan, tag yang berkedudukan sebagai tag penutup mempunyai bentuk berupa </>.

2.5.5 *Java script*

(Aloysius Sigit, 2013), menyatakan bahwa Javascript adalah bahasa skrip (bahasa yang kodennya ditulis menggunakan teks biasa) yang ditempelkan pada dokumen HTML dan diproses pada sisi klien. Dengan adanya bahasa ini, kemampuan dokumen HTML menjadi semakin luas. Sebagai contoh, dengan menggunakan JavaScript dimungkinkan untuk memvalidasi masukan-masukan pda formulir sebelum formulir dikirm ke *server*. Selain itu, dengan

menggunakan JavaScript juga dimungkinkan untuk mengimplementasikan tugas yang bersifat interaktif tanpa berhubungan dengan *server*. Beberapa contoh yang bisa dilakukan melalui JavaScript :

1. Memanampilkan jam lokal pada halaman web.
2. Mengatur warna latar belakang halaman web.
3. Mengganti gambar ketika pemakai menempatkan penunjuk *mouse* ke suatu gambar.
4. Memvalidasi keabsahan data yang dimasukkan oleh pemakai.

2.5.6 *Css*

(Prasetyo, 2012), “*Cascading Style Sheet* (CSS) adalah suatu teknologi yang di gunakan untuk memperindah halaman *website* (situs)”. CSS mempunyai 2 bagian utama yaitu selectors dan deklarasi. Yang dimaksud selectors biasanya element HTML yang ingin diubah, sedangkan deklarasi biasanya terdiri dari properti dan nilai. Properti sendiri adalah atribut style yang di ingin diubah, dan setiap properti memiliki nilai. *Cascading Style Sheet* (CSS) merupakan aturan untuk mengatur beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam.

2.5.7 *MySQL*

(Anhar, 2012), mengatakan bahwa MySQL (*My Structured Query Language*) adalah sebuah program pembuat dan pengelola database atau yang sering disebut DBMS (*Database Management System*), sifat dari DBMS ini adalah Open Source dan ini didapatkan gratis pada alamat <http://www.mysql.com>. MySQL awalnya dibuat oleh perusahaan konsultan bernama TcX yang berlokasi di Swedia dan dulunya MySQL berjalan pada Platform Linux, dengan adanya perkembangan dan banyaknya pengguna, serta lisensi dari database ini adalah *Open Source*, maka para ahli pengembang merilisnya dalam versi Windows.

Dikutip dari (*Jurnal Informatika Darmajaya, Vol.7, No.1, Juni 2007* Rangga F Dan Chairani) MySQL adalah suatu structured query language relational database management system (RDBMS) yang memiliki fasilitas multithreaded dan multi-user, MySOL AB membuat MySQL sebagai perangkat lunak yang gratis di bawah lisensi GNU General Public License (GPL), tetapi mereka juga menjualnya dengan lisensi konvensional. (<http://en.wikipedia.org/wiki/MySQL>) Beberapa keunggulan MySQL sebagai database server adalah (http://dev.mysql.com/doc/refman/5.1/en/what_is.html):

1. Open source. MySQL, didistribusikan secara gratis dibawah lisensi GPL (General Public License) sehingga dapat menggunakannya secara cuma-cuma tanpa dipungut biaya dan dapat di-download secara gratis melalui situs resmi MySQL yaitu <http://www.mysql.com>.
2. Cepat, handal, dan mudah digunakan, MySQL dapat memproses banyak MySQL per satuan waktu, mampu menangani database dalam skala besar, dengan jumlah record lebih dari 50 juta dan 60 ribu tabel serta 5 miliar baris. Tersedia berbagai tool yang dapat digunakan untuk administrasi database.
3. Dapat bekerja di client server atau sebagai sistem yang terintegrasi. MySQL dapat melakukan koneksi dengan client menggunakan protokol TCP/IP, Unix Socket (Unix), atau Named Pipes (NT).
4. Portability MySQL dapat berjalan stabil pada berbagai sistem operasi yang di antaranya adalah Window, Linux, FreeBSD, Mac OS X Server, Solaris, Amiga, HP.UX, dan masih banyak lainnya.

2.5.8 Gammu

(Ramadhika A 2012) *Gammu* adalah sebuah aplikasi cross-platform yang digunakan untuk menjembatani / mengomunikasikan antara database SMS Gateway dengan sms devices. Aplikasi Gammu berupa daemon yang berjalan secara background. Setiap saat, gammu memonitor sms devices dan database sms gateway. Saat ada sms masuk ke sms devices, maka gammu langsung memindahkannya ke dalam inbox dalam database sms gateway. Sebaliknya saat Aplikasi Pengirim SMS memasukkan sms ke dalam outbox dalam

database sms gateway, maka gammu mengirimkannya melalui sms devices, dan memindahkan sms ke sentitem dalam database.

Gammu juga merupakan salah satu *tools* untuk mengembangkan aplikasi *sms gateway* yang cukup mudah diimplementasikan dan pastinya gratis. Kelebihan *gammu* dari *tool sms gateway* lainnya adalah :

1. *Gammu* bisa di jalankan di *Windows* maupun *Linux*.
2. Menggunakan database *Mysql*.
3. Baik kabel data *usb* maupun *serial*, semuanya kompatibel di*Gammu*, Dapat membaca, menghapus dan mengirim *sms*.

2.6 Metode Perangkat Lunak

Pada sub bab ini akan diuraikan mengenai pembuatan metode perangkat lunak yang digunakan dalam antara lain :

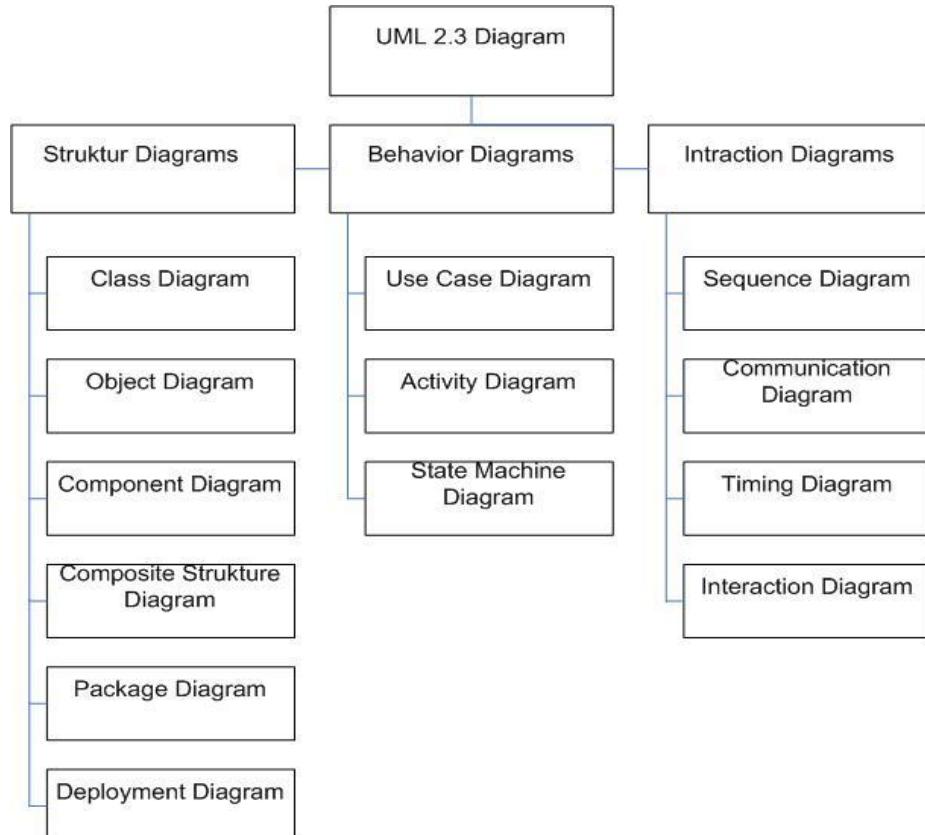
2.6.1 UML (*Unified Modeling Language*)

(Rosa dan Shalahuddin, 2013), *Unified Modelling Language (UML)* adalah salah satu standar bahasa yang banyak di gunakan di dunia industri untuk mendefinisikan requierement, untuk membuat analisus dan desain, serta menggambarkan arsitektur dengan menggunakan diagram dan teks-teks pendukung.

2.6.1.1 Diagram UML

(Rosa dan Shalahuddin, 2013), menyatakan pada *UML 2.3* terdiri dari 13 macam diagram yang di kelompokkan dalam 3 kategori:

1. *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.\|
2. *Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem



Gambar 2.2 Diagram UML

Berikut penjelasan dari jenis-jenis diagram tersebut antara lain:

1. *Class Diagram*

Menggambarkan struktur sistem dari segi pendefenisian kelas-kelas yang akan di buat untuk membangun sistem

2. *Object Diagram*

Menggambarkan struktur sistem dari segi penanaman objek dan jalannya objek dalam sistem.

3. *Component Diagram*

Menunjukan organisasi dan ketergantungan di antara kumpulan komponen dalam sebuah sistem.

4. *Composite Structure Diagram*

Menggambarkan struktur dari bagian – bagian yang saling terhubung maupun mendeskripsikan struktur pada saat berjalan (*runtime*) dari *instance* yang saling terhubung.

5. Package Diagram

Menyediakan cara mengumpulkan elemen-elemen yang saling terkait dalam diagram uml.

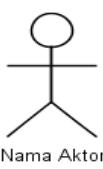
6. Deployment Diagram

Menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi.

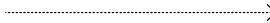
7. Use Case Diagram

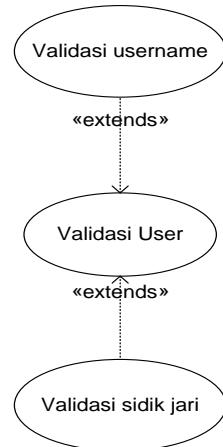
Pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat dapat di lihat pada tabel.

Tabel 2.1 Simbol-simbol *Usecase* Diagram

Simbol	Deskripsi
<i>Use Case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau actor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i> .
<i>Aktor / Actor</i> 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun symbol dari actor adalah gambar orang, tapi actor belum tentu orang; biasanya dinyatakan menggunakan kata benda di awal frase nama actor.

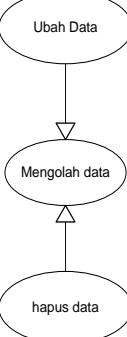
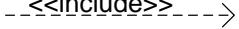
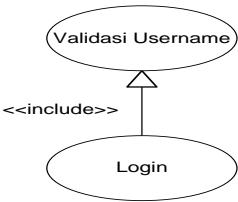
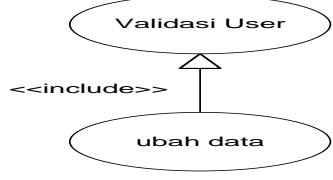
Tabel 2.1 Lanjutan

Asosiasi / Association	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan actor
Ekstensi / extend «extends» 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang sama dengan <i>use case</i> yang ditambahkan, misalnya
Ekstensi / extend «extends» 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang sama dengan <i>use case</i> yang ditambahkan, misalnya



arah panah mengarah pada *use case* yang ditambahkan; biasanya *use case* yang menjadi *extend*-nya merupakan jenis yang sama dengan *use case* yang menjadi induknya.

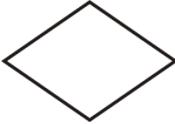
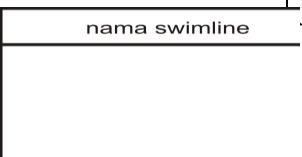
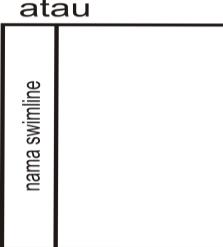
Tabel 2.1 Lanjutan

Generalisasi/ generalization 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi user yang lebih umum dari lainnya, misalnya:</p>  <p>Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum).</p>
Menggunakan /include/uses  	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan pada <i>use case</i> ini.</p> <ol style="list-style-type: none"> 1. <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan, misal pada kasus berikut:  2. <i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, misal pada kasus berikut:  <p>Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>

8. Activity Diagram

Menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak dapat di lihat pada tabel 2.2 berikut:

Tabel 2.2 Simbol-simbol *Activity Diagram*

Simbol	Deskripsi
 Status awal	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
 Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
 Percabangan/decision	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
 Penggabungan/join	Asosiasi penggabungan dimana lebih dari satu aktivitas di gabungkan menjadi satu
 Status akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
 atau	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi
	

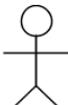
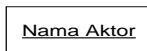
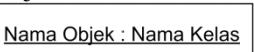
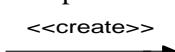
9. State Machine Diagram

Di gunakan untuk menggambarkan perubahan status atau transisi status dari sebuah mesin atau sistem atau objek.

10. Sequence Diagram

Menggambarkan kelakuan objek pada usecase dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek dapat di lihat pada tabel 2.3 dibawah ini:

Tabel 2.3 Simbol – simbol *Squence diagram*

Simbol	Deskripsi
Aktor  Nama Aktor atau  Nama Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
Garis hidup / <i>lifeline</i> 	Menyatakan kehidupan suatu objek.
Objek  Nama Objek : Nama Kelas	Menyatakan objek yang berinteraksi pesan
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan.
Pesan tipe create 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.

Tabel 2.3 Lanjutan	
Pesan tipe call 1: nama_metode() →	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri, <p>arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.</p>
Pesan tipe send 1:masukan →	Menyatakan bahwa suatu objek mengirimkan data/masukan/ informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
Pesan tipe return 1:keluaran →	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
Pesan tipe destroy <<destroy>> →	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy.

11. *Communication Diagram*

Menggambarkan interaksi antar objek /bagian dalam bentuk urutan pengiriman pesan.

12. *Timing Diagram*

Fokus dalam penggambaran terkait batasan waktu

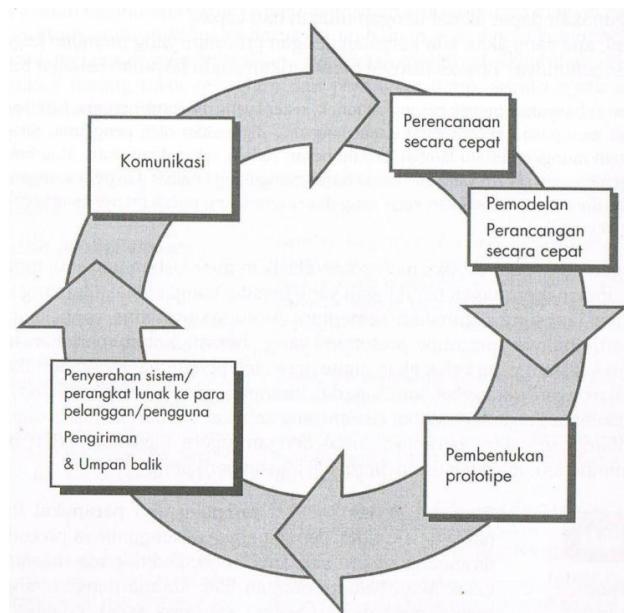
13. *Interaction Overview Diagram*

Menggambarkan sekumpulan urutan aktivitas.

2.6.2 Metode *Prototype Model*

(Pressman 2012), menyatakan bahwa *prototype* merupakan metode yang efektif dalam merancang perangkat lunak. *Prototype* dimulai dengan mengumpulkan kebutuhan yang akan dirancang. Pengembang mendefinisikan *object* keseluruhan dari perangkat lunak, mengidentifikasi segala aktifitas yang diketahui dan kemudian melakukan “perancangan kilat”. Perancangan kilat berfokus pada penyajian dari aspek-aspek perangkat lunak tersebut yang akan nampak bagi pelanggan atau pemakai (contohnya pendekatan *input* dan format *output*).

Perancangan kilat membawa kepada kontruksi sebuah *prototype*, *prototype* tersebut dievaluasi oleh pelanggan dan dipakai untuk menyaring kebutuhan pengembangan perangkat lunak. Adapun tahapan metode pengembangan *prototype* adalah sebagai berikut :



Gambar 2.3 Metode *Prototype*

Prototype juga dapat didefinisikan sebagai proses pengembangan suatu *prototype* secara cepat untuk digunakan terlebih dahulu dan ditingkatkan terus menerus sampai didapatkan sistem yang utuh. *Prototype* merupakan proses yang digunakan untuk membantu pengembang perangkat lunak dalam membentuk *prototype* dari perangkat lunak yang harus dibuat.