BAB II

TINJAUAN PUSTAKA

II.1. Rancang Bangun

Model perancangan sesungguhnya adalah modal objek yang mendeskripsikan realisasi fisik *use case* dengan cara berfokus pada bagaimana spesifikasi-spesifikasi kebutuhan fungsional dan non-fungsional, bersama dengan batasan-batasan lain yang berhubungan dengan lingkungan implemenatasi, memiliki efek langsung pada pertimbangan-pertimbangan pada aktivitas-aktivitas yang dilakukan pada tahap implementasi. Model perancangan sesungguhnya secara langsung bertindak sebagai abstraksi implementasi sistem atau pun perangkat lunak dan dengan sendirinya model perancangan suatu saat nanti akan menjadi asupan bagi aktivitas-aktivitas selanjutnya yang kelak akan terdefinisi pada tahap implementasi (Adi Nugroho ; 2010 : 212).

II.2. Aplikasi

Program aplikasi adalah program siap pakai atau program yang dirancang untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain. Aplikasi juga diartikan sebagai penggunaanatau penerapan suatu konsep yang menjadi pokok pembahasan atau sebagai program komputer yang dibuat untuk menolong manusia dalam melaksanakan tugas tertentu. Aplikasi software yang dirancang untuk penggunaan praktisi khusus, klasifikasi luas ini dapat dibagi menjadi 2 (dua) yaitu :

1. Aplikasi *software* spesialis, program dengan dokumentasi tergabung yang dirancang untuk menjalankan tugas tertentu.
2. Aplikasi paket, suatu program dengan dokumentasi tergabung yang dirancang untuk jenis masalah tertentu (Rahmatillah ; 2011: 3).

II.3. Keamanan Data

Keamanan sistem komputer mencakup empat aspek yang hasur dipertimbangkan yaitu *privacy* atau *confidentiality* adalah aspek-aspek yang perlu diperhatikan dalam usaha menjaga informasi dari orang yang tidak berhak mengakses. Aspek yang kedua yaitu *integrity* adalah usaha untuk menjaga data atau sistem sehingga tidak bisa di ubah oleh yang orang-orang yang tidak berhak mengubahnya. Aspek *authentication* yaitu usaha atau metode untuk mengetahui keaslian dari informasi, misalnya apakah informasi yang dikirim dibuka oleh orang yang benar atau layanan dari *server* yang diberikan benar berasal dari *server* yang dimaksud. Sedangkan aspek *avallbility* berhubungan dengan ketersediaan sistem dan data ketika dibutuhkan.

Keempat aspek tersebut menjadi dasar untuk melakukan pengamanan terhadap sistem atau data. Keamanan komputera dalah sebuah proses, yang harus dijalankan untuk mengamankan sistem dan dalam penerapannya harus dilakukan dengan menyeluruh. Bagian-bagian keamanan yang ada pada pusat data mengacu pada empat aspek dasar keamanan yang disebutkan sebelumnya (Teguh Wahyono ; 2010 : 115).

II.4. Data SMS

Short Message Service (SMS) adalah layanan global dengan sistem komunikasi nirkabel yang mentransmisikan pesan teks antara dua atau lebih handphone dan sistem eksternal seperti surat elektronik, pager dan pesan suara.

Layanan dari titik ke titik SMS menyediakan mekanisme pengiriman pesan pendek ke dan dari perangkat nirkabel seperti handphone. Layanan ini menggunakan SMSC (*Short Message Service Centre*) yang berfungsi sebagai penyimpan sementara dan penerus pesan yang dikirim. Oleh karena itu sebuah jaringan nirkabel harus memiliki kemampuan untuk mencari tujuan dan mengirim pesan antara SMSC dan perangkat tanpa kabel tersebut (Nurhanudin D ; 2012 : 2).

Ada beberapa karakteristik pesan SMS yang penting, yakni :

1. Pesan SMS yang sampai atau tidak sama sekali, akan diberikan informasi (*report*) status pengiriman pesan SMS.
2. Berbeda dengan fungsi panggilan, sekalipun saat mengirimkan SMS handphone tujuan tidak aktif, bukan berarti pengiriman SMS akan gagal. Namun SMS akan ditampung dulu oleh SMSC sebelum diteruskan ke handphone tujuan setelah aktif.
3. Lebar pita yang digunakan rendah (Nurhanudin D ; 2012 : 3).

II.5. Kriptografi

Kriptografi merupakan metode untuk mengamankan data, baik itu data teks maupun data gambar. Metode ini dilakukan dengan penyandian atau pengacakan data asli, sehingga pihak lain yang tidak mempunyai hak akses atas

data tersebut tidak dapat memperoleh informasi yang ada di dalamnya. Ilmu kriptografi sebenarnya telah lama digunakan, sejak jaman sebelum mengenal metode pengiriman data menggunakan komputer. Seiring dengan perkembangan teknologi telekomunikasi, maka semakin berkembang pula ilmu kriptografi baik jenis maupun fungsinya.

Secara umum ada dua tipe algoritma kriptografi berdasarkan kuncinya yaitu algoritma simetris dan algoritma asimetris. Algoritma simetris adalah algoritma yang memiliki kunci enkripsi dan dekripsi yang sama, sedangkan untuk algoritma asimetris terdiri atas dua buah kunci yaitu kunci publik untuk melakukan enkripsi sedangkan kunci pribadi untuk melakukan dekripsi. Dalam algoritma kunci asimetris, kunci yang didistribusikan adalah kunci publik yang tidak diperlukan kerahasiaanya sedangkan kunci pribadi tetap disimpan atau tidak didistribusikan. Setiap orang yang memiliki kunci publik dapat melakukan proses enkripsi tetapi hasil dari enkripsi tersebut hanya bisa dibaca oleh orang yang memiliki kunci pribadi (M. Taufik T amam ; Vol. IV No 1, Juni 2012 : 8).

II.6. Algoritma Vigenere Cipher

Vigenere Cipher adalah salah satu jenis kriptografi kalsik yang pada dasarnya melakukan subtitusi cipher abjad majemuk *(polyalphabetic substitution).* Metode ini pertama kali dipublikasikan oleh seorang diplomat (sekaligus sesorang kriptologis) Prancis, Blaise de Vigenere pada abad ke-16, tepatnya pada tahun 1586. Sebenarnya Giovan Batista Belaso telah menggambarkan algoritma ini pertama kali pada tahun 1553 seperti ditulis di dalam buku La Cifra del Sig.

metode Vigenere Cipher ini berhasil dipecahkan oleh matematikawan Inggris Charles Babbage dan Kasiski pada pertengahan abad 19.

Vigenere Cipher ini digunakan oleh tentara konfiderasi pada perang sipil Amerika. Perang sipil akhirnya berhasil dihentikan setelah Vigenere Cipher berhasil dipecahkan. Di metode kriptografi klasik Caesar Cipher, setiap huruf alpahabet akan disubtitusi sepanjang 3 huruf sesudah huruf teresbut. Contoh, huruf A akan diganti dengan hurf D, B akan diganti dengan huruf E, Y akan diganti dengan huruf B dengan metode Caesar Cipher. Vigenere Cipher ini menerapkan prinsip Caesar Cipher dalam metode enkripsinya. Untuk memudahkan dalam proses enkripsi, maka dapat digunakan alat bantu berupa bujur sangkar Vigenere.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Gambar II.1. Bujur Sangkar Vigenere Cipher
Sumber : Andi Kurniawan Dwi P : 2012

Baris pada gambar II.1. menyatakan huruf plainteks yang akan dienkripsi dan kolom menyatakan huruf kunci enkripsi. Perpotongan antara baris dan kolom menyatakan huruf yang sudah terenkripsi atau diistilahkan dengan cipherteks.Jika panjang kunci lebih pendek daripada panjang plainteks, maka kunci akan diulang secra periodik (Andi Kurniawan Dwi P ; 2012).

II.7. Android

Android adalah sistem operasi disematkan pada gadget, baik itu handphone, tablet, juga sekarang sudah merambah ke kamera digital dan jam tangan. Saat ini gadget berbasis android, baik itu tablet atau handphone, begitu digandrungi. Selain harganya yang semakin terjangkau, juga banyak varian spesifikasi yang bisa dipilih sesuai kebutuhan. Perkembangan android sangat cepat. Di awal tahun 2002 ada 200 juta pengguna aktif android, dan google play mampu menampung 400.000 aplikasi yang siap digunakan, dan total mencapai 10 triliun kali aplikasi yang sudah di download lewat android market, pertumbuhan yang luar biasa. Jumlah ini diyakini akan terus bertambah seiring waktu dan perkembangan teknologi (Agus Wahadyo ; 2013 : 3).

II.7.1. Arsitektur Android

Secara garis besar Arsitektur Android dapat dijelaskan di gambarkan sebgai berikut :

1. *Applications* dan *Widgets*

Application dan *Widgets* ini adalah *layer* dimana kita berhubungan dengan aplikasi saja, dimana biasanya kita *download* aplikasi kemudian kita lakukan

instalasi dan jalankan aplikasi tersebut. Di layer terdapat aplikasi inti termasuk *client email,* program SMS, kalender, peta, *browser,* kontak, dan lain-lain. Semua aplikasi ditulis menggunakan bahsa pemrograman Java.

2. *Applications Frameworks*

Android adalah *Open Development Platform* yaitu Android menawarkan kepada pengembang atau member kemampuan kepada pengembang untuk membagun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi *resources,* menjalankan *service background,* mengatur alarm, dan menambahkan status *notification,* dan sebagainya. Pengembang memiliki akses penuh menuju API *framework* seperti dirancang supaya kita denggan mudah dapat menggunakan kembali komponen yang sudah digunakan (*reuse*).

3. *Libraries*

Libraries adalah layer dimana fitur-fitur Android berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya.

4. Android *Run Time*

Layer yang membuat aplikasi Android dapat dijalankan dimana prosesnya menggunakan implementasi *Linux. Dalvik Virtual Machine* (DVM) merupakan mesin yang membentuk dasar kerangka aplikasi Android. Di dalam Android *Run Time* dibagi menjadi dua bagian yaitu :

a. *Core Libraries* adalah aplikasi Android dibangun dalam bahasa java sementara *Dalvik* sebagai virtual mesinnya bukan *Virtual Machine Java,*

sehingga diperlukan sebuah *libraries* yang berfungsi untuk menterjemahkan bahsa java yang ditanda tangani oleh *Core Libraries.*

b. *Dalvik Virtual Machine* adalah vitual mesin berbasis register yang dioptimmalkan untuk menjalankan fungsi-fungsi secara efisien, dimana merupakan pengembangan yang mampu membuat *linux karnel* untuk melakukan *threading* dan manajemen tingkat rendah.

5. *Linux Karnel*

Linux karnel adalah inti dari *operating system* dari Android itu berada. Berisi file-file system yang mengatur system *processing, memory, resource, drivers,* dan system-sistem operasi Android lainnya. *Linux karnel* yang digunakan Android adalah *linux karnel relase* 2.6 (Nazruddin Safaat H,2011:6-8).

II.7.2. Versi Android

Telepon pertama yang memakai system operasi Android adalah HTC Dream, yang dirilis pada 22 Oktober 2008, Pada penghujung tahun 2010 diperkirakan hampir semua vendor seluler didunia menggunakan Android sebagai *operating system.* Adapun versi-versi Android yang pernah dirilis sebagai berikut (Nazruddin Safaat H,2011:11-12) :

1. Android 1.1 *Bender*

Pada 9 Maret 2009, *Google* merilis Android versi 1.1. Android versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam, alarm, *voice search* (pencarian suara), pengiriman pesan dengan *Gmail*, dan pemeberitahuan *email.*

2. Android 1.5 *Cupcake*

Pada pertengahan Mei 2009, *Google* kembali merilis telepon seluler dengan menggunakan Android dan SDK *(Software Development Kit)* dengan versi 1.5 *(Cupcake).* Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengupload video ke *youtube* dan gambar ke *pisaca* langsung dari telepon, dukungan *Bluetooth A2DP,* kemampuan terhubung secara otomatis ke *headset Bluetooth,* animasi layar, dan *keyboard* pada layar yang dapat disesuaikan dengan system.

3. Android 1.6 *Donut*

Donut dirilis pada September dengan menampilkan proses pencarian yang baik disbanding sebelumnya, penggunaan baterai indicator dan control applet *VPN*. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus, kamera, *camcorder* dan galeri yang diintegrasikan, *CDMA/EVDO,* 802.IX, *VPN, Gestures,* dan *Text to engine,* kemampuan dial kontak, teknologi *text to change speech* (tidak tersedia pada semua ponsel, pengadaan resolusi *VWGA).*

4. Android 2.0/2.1 *Éclair*

Pada 3 Desember 2009 kembali diluncurkan ponsel Android dengan versi 2.0/2.1 (*Éclair*), perubahan yang dilakukan adalah pengoptimalan *hardware,* peningkatan *Google Maps* 3.1.2, perubahan *UI* dengan *browser* baru dan dukungan *HTML5,* daftar kontak yang baru, dukungan *flash* untuk kamera 3.2 MP, digital *Zoom,* dan *Bluetooth* 2.1.

5. Android 2.2 *Froyo*

Pada bulan Mei 2010 Android versi 2.2 Rev 1 diluncurkan. Android inilah yang sekarang sangat banyak beredar dipasaran, salah satunya adalah dipakai di *Samsung FX Tab* yang sudah ada dipasaran.

6. Android 2.3 *Ginferbread*

Android versi 2.3 diluncurkan pada Desember 2010, hal-hal yang direvisi dari versi sebelumnya adalah kemampuan seperti berikut *SIP based VoIP, Near Field Communications, Gyroscope* dan *sensor, Multiple cameras support, Mixable audio effects, Download manager.*

II.8. Pengertian Java

Java adalah bahasa pemrograman *Object Oriented Programming* (OOP) yang dibuat oleh Sun Microsystem. Java dirancang untuk menjadi bahasa yang memiliki kemampuan tinggi dalam hal portabilitas dan pemanfaatan jaringan tanpa mengabaikan kestabilan, kemanan, serta kemudahan dari sisi desain dan pemrograman aplikasi.

Sebutan Java 2 diberikan untuk Java versi 1.2 dan versi berikutnya. Awal mula bahasa Java adalah Oak. Oak dikembangkan oleh *Sun Microsystem* yang kemudian berganti nama menjadi java. Awalnya java dikembangkan menjadi sebuah bahasa pemrograman yang *platform independence* sehingga dapat dipakai dan di-embededkan ke perangkat-perangkat elektronik rumah tangga seperti televisi dan *microwave*, tetapi pada perkembangannya java dapat digunakan untuk pemrograman secara umum (Dyan Hari Widodo;2011 : 4).

II.9. *Intellij Idea*

Intellij IDEA adalah sebuah *code centric Integrated Development Environment* (IDE) yang diproduksi oleh JetBrains. Selain untuk pengembangan aplikasi desktop, *Intellij IDEA* juga mendukung pengembangan aplikasi *mobile*.

Salah satunya adalah pengembangan aplikasi *mobile* berbasis Android. Untuk pengembangan aplikasi Android ini, *Intellij IDEA* telah menyediakan Android SDK saat instalasi. Sehingga dengan *Intellij* pengguna dapat langsung mencoba aplikasi android pada emulator yang disediakan. Untuk mendukung pengembangan aplikasi berbasis android, *Intellij* menyediakan beberapa fitur, yaitu :

1. Memudahkan pengguna untuk membuat aplikasi Android menggunakan *New Project Wizard*.
2. Membantu pengguna untuk melakukan pencarian aplikasi android dengan dukungan tampilan *tree view*.
3. Memudahkan pengguna untuk membuat elemen-elemen aplikasi android dan melakukan pengaturan *string*, warna, dan lain-lain dengan melakukan integrasi antara hal tersebut dengan *R. Java file.*
4. Menjalankan aplikasi android via emulator (Hendra ; 2011 : 42).

II.10. *Genymotion*

Genymotion merupakan satu set alat yang menyediakan lingkungan virtual Android yang cocok untuk pengembang Android. *Genymotion* adalah salah satu emulator Android yang merupakan penerus dari *software* AndroVM yang

fungsinya adalah untuk menjalankan sistem operasi android di atas PC atau laptop.

Genymotion dibangun di berdasarkan proyek *open-source* sebelumnya : AndroVM, untuk memberikan lingkungan yang Android yang realistis dan berfungsi penuh berdasarkan berbagai ponsel dan tablet yang berbeda. Kamu harus mendaftar untuk mendapatkan akun gratis *Genymotion* sebelum kamu dapat mulai men-download mesin virtual *Genymotion*.

Emulator android *genymotion* menggunakan arsitektur virtualization x86 sehingga jauh lebih efisien. Emulator *Genymotion* juga mengambil kelebihan dari akselerasi *hardware "OpenGL*", sehingga memungkinkan kamu untuk menguji aplikasi anda dengan *performa* 3D yang menakjubkan. *Genymotion* dikembangkan dengan tujuan utama untuk para pengembang, penguji dan demonstran apps Android, tapi *Genymotion* bagus untuk gamer dan siapa saja yang ingin mencoba Android tanpa harus membeli ponsel atau tablet (Fauzan ; 2014 : 256).

II.11. UML (*Unified Modeling Language*)

Menurut Windu Gata (2013 : 4) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). 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

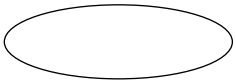
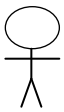
saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan UML adalah sebagai berikut :

1. *Use case* Diagram

Use case diagram merupakan pemodelan untuk kelakukan (*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. Simbol-simbol yang digunakan dalam *use case* diagram, yaitu :

Tabel II.1. Simbol *Use Case*

Gambar	Keterangan
	*Use case* menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukan pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama *use case*.
	Aktor adalah *abstraction* dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasikan aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan *use case*, tetapi tidak memiliki control terhadap *use case*.
	Asosiasi antara aktor dan *use case*, digambarkan dengan garis tanpa panah yang mengindikasikan

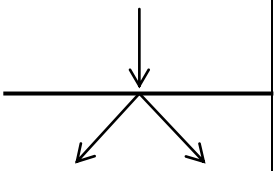
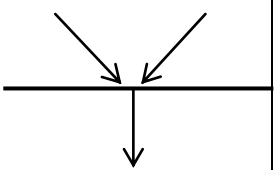
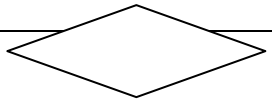
	siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.
	Asosiasi antara aktor dan *use case* yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.
	Include, merupakan di dalam *use case* lain (*required*) atau pemanggilan *use case* oleh *use case* lain, contohnya adalah pemanggilan sebuah fungsi program.
	Extend, merupakan perluasan dari *use case* lain jika kondisi atau syarat terpenuhi.

(Sumber : Windu Gata ; 2013 : 4)

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram,* yaitu :

Tabel II.2. Simbol *Activity Diagram*

Gambar	Keterangan
	*Start point*, diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	*End point*, akhir aktifitas.
	*Activites*, menggambarkan suatu proses/kegiatan bisnis.
	*Fork* (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	*Join* (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	*Decision Points,* menggambarkan pilihan untuk

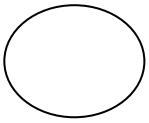
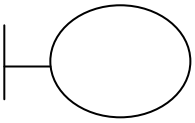
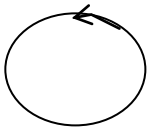
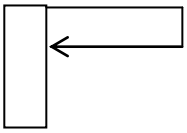
	pengambilan keputusan, *true*, *false*.
New Swimline	*Swimlane*, pembagian *activity* diagram untuk menunjukkan siapa melakukan apa.

(Sumber : Windu Gata ; 2013 : 6)

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram*, yaitu :

Tabel II.3. Simbol *Sequence Diagram*

Gambar	Keterangan
	*Entity Class*, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	*Boundary Class*, berisi kumpulan kelas yang menjadi *interface* atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan *form* cetak.
	*Control class*, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	*Message*, simbol mengirim pesan antar *class*.
	*Recursive*, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	*Activation*, *activation* mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.

	Lifeline, garis titik-titik yang terhubung dengan objek, sepanjang *lifeline* terdapat *activation*.

(Sumber : Windu Gata ; 2013 : 7)

4. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggng 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.

Tabel II.4. *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 : Windu Gata ; 2013 : 8)