BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Berikut ini dalah hasil penelitian dari para peneliti terdahulu yang di ambil dari kesimpulan :

Berdasarkan penelitian yang dilakukan oleh Setiawan (2011) mengenai Analisis Dan Perbandingan Algoritma Whirlpool Dan SHA-512 Sebagai Fungsi Hash, Setiawan menyimpulkan bahwa kedua algoritma memiliki kelebihan dan kekurangannya masing-masing. Algoritma hash SHA-512 yang memiliki algoritma yang sederhana, tetapi dapat menghasilkan nilai hash yang kuat, serta menggunakan memori yang lebih sedikit jika dibandingkan dengan algoritma hash Whirlpool. Tetapi, nilai hash yang dihasilkan memiliki kekuatan sebagai kata kunci yang tidak stabil. Terkadangnilai kata kunci yang diberikan sangat kuat, tetapi terkadang nilai kata kunci yang diberikan sangat lemah.

Berdasarkan penelitian Sutanto (2011) mengenai Algoritma Fungsi Hash Baru Dengan Menggabungkan MD5, SHA-1 Dan Penyertaan Panjang Pesan Asli, Sutanto menyimpulkan bahwa pada dasarnya algoritma ini seperti memodifikasi algoritma SHA-1. Namun algoritma di atas hanya merupakan salah satu contoh saja. Bisa juga menggunakan algoritma MD5 sebagai algoritma inti lalu dilakukan modifikasi dengan menambahkan beberapa pengoperasian dari algoritma SHA-1. Namun dengan menggunakan algoritma SHA-1 sebagai dasar dapat memberikan tingkat keamanan yang lebih.

Dari beberapa penelitian diatas mengenai Penelitian Terdahulu tidak ditemukan adanya kesamaan judul. Namun ada terdapat kesamaan penggunaan metode. Oleh karena itu penelitian di atas. Penelitian Terdahulu dapat digunakan sebagai tambahan referensi untuk penelitian ini.

II.2. Landasan Teori

II.2.1. Aplikasi

Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Aplikasi adalah Program yang dibuat oleh manusia yang berfungsi untuk menyelesaikan permasalahan-permasalahan masalah yang akan dihadapi (Zulfauzi, 2015 : 57).

Aplikasi merupakan rangkaian kegiatan atau perintah untuk dieksekusi oleh komputer atau suatu perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau *suite* aplikasi *(application suite)*. Dalam Aplikasi suatu paket biasanya memiliki antar kesamaan dan antar muka pengguna sehingga memudahkan pengguna untuk menggunakan tiap aplikasi (Sitohang, 2013 : 2).

II.2.2. Kriptografi

Kriptografi merupakan suatu bidang ilmu yang mempelajari tentang bagaimana merahasiakan suatu informasi penting ke dalam suatu bentuk yang tidak dapat dibaca oleh siapapun serta mengembalikannya kembali menjadi

informasi semula dengan menggunakan berbagai macam teknik yang telah ada sehingga informasi tersebut tidak dapat diketahui oleh pihak manapun yang bukan pemilik atau yang tidak berkepentingan. Sisi lain dari kriptografi ialah kriptanalisis (*Cryptanalysis*) yang merupakan studi tentang bagaimana memecahkan mekanisme kriptografi (Sitohang, 2013 : 3).

Kriptografi pada awalnya dijabarkan sebagai ilmu yang mempelajari bagaimana menyembunyikan pesan. Namun pada pengertian modern kriptografi adalah ilmu yang bersandarkan pada teknik matematika untuk berurusan dengan keamanan informasi seperti kerahasian, keutuhan data dan otrntikasi entitas. Jadi pengertian kriptografi modern adalah tidak saja berurusan hanya dengan penyembunyian keamanan informasi. Berikut adalah beberapa rangkuman yang berkembang pada kriptograffi modern :

1. Fungsi *Hash.* Fungsi *Hash* adalah fungsi yang melakukan pemetaan pesan dengan panjang sembarang ke sebuah teks khusus yang sisebut *message digest* degan panjang tetap. Fungsi *Hash* umumnya dipakai sebagai nilai uji *check value*) pada mekanisme keutuhan data.
2. Penyandian dengan kunci simetrik (*symmetric key encipherment*). Penyandian kunci dengan simetrik adalah penyandian yang kunci enkripsi dan kunci deskripsi bernilai sama. Kunci pada penyandian simetrik diasumsikan bersifat rahasia hanya piahak yang melakukan enkripsi dan deskripsi yang mengetahui nilainya. Oleh karena itu penyadian dengan kunci simetrik disebut juga penyandian dengan kunci rahasia *secret key encipherment.*

Penyandian dengan kunci asimetrik (A*symmetric key encipherment*). Penyandian dengan kunci asimetrik atau sering juga disebut kunci publik (*public key*) adalah Penyandian dengan enkripsi dan deskripsi berbeda nilai. Kunci enkripsi disebut dengan kunci publik (*public key*) bersifat terbuka. Sedangkan, kunci deskripsi disebut privat (*private key*) bersifat tertutup/rahasia. (Rifki Sadikin, 2017 : 9).

II.2.3. Keamanan Jaringan

Untuk mewujudkan layanan keamanan jaringan pengembang sistem dapat menggunakan mekanisme keamanan jaringan. Rekomendasi ITU-T(X.800) juga mendefinisikan beberapa mekanisme keamanan jaringan. Berikut ini adalah beberapa jenis mekanisme keamanan jaringan :

1. *Encipherment*

Encipherment merupakan mekanisme keamanan jaringan yang digunakan untuk menyembunyikan data. Mekanisme *Encipherment* dapat menyediakan layanan kerahasiaan data (*confidentiality)* meskipun dapat juga digunakan untuk layanan lainnya. Untuk mewujudkan mekanisme *Encipherment* teknik kriptografi dan steganografi dapat digunakan. Kriptografi merupakan kumpulan teknik untuk menyembunyikan pesan dengan mengubah pesan ini menjadi pesan tersembunyi. Sedangkan steganografi merupakan kumpulan teknik untuk menyembunyikan pesan pada media lain misalnya gambar, suara, atau video.

2. Keutuhan Data

Mekanisme keutuhan data digunakan untuk memastikan keutuhan data pada unit data atau pada suatu aliran (*stream)* data unit. Cara yang digunakan adalah dengan menambahkan nilai penguji (*check value)* pada data asli. Jadi jika sebuah data akan dikirim nilai penguji dihitung terlebih dahulu dan kemudian data dan penguji dikirim bersamaan. Penerima dapat menguji apakah ada perubahan data atau tidak dengan cara menghitung nilai penguji data yang dikirim dan membandingkan nilai penguji yang dihitung dengan nilai penguji yang dikirim bersamaan data asli. Bila sama penerima dapat menyimpulkan data tidak berubah.

3. *Digital Signature*

Digital Signature merupakan mekanisme keamanan jaringan yang menyediakan cara bagi pengirim data untuk "menandatangani" secara elektronik sebuah data dan penerima dapat memverifikasi "tanda tangan" itu secara elektronik. *Digital Signature* ditambahkan pada data unit dan digunakan sebagai bukti pengirim dan menghindari pemalsuan (*forgery)* tanda tangan.

4. *Authentication Exchange*

Mekanisme ini memberikan cara agar dua entitas dapat saling meng-otentikasi dengan cara bertukar pesan untuk saling membuktikan identitas.

5. *Traffic Padding*

Traffic Padding menyediakan cara untuk pencegahan analisis lalu lintas data pada jaringan yaitu dengan menambah data palsu pada lalu lintas data.

6. *Routing Control*

Routing Control menyediakan cara untuk memilih dan secara terus menerus mengubah alur (*rote)* pada jaringan komputer antara pengirim dan penerima. Mekanisme ini menghindarkan komunikasi dari penguping (*eavedropper).*

7. Notarisasi

Notarisasi (*notarizatio)* menyediakan cara untuk memilih pihak ketiga yang terpercaya sebagai pengendali komunikasi antar pengirim dan penerima.

8. Mekanisme Kendali Akses

Mekanisme kendali akses memberikan cara bagi penggna untuk memperoleh hak akses sebuah data. Misalnya dengan tabel relasi pengguna dan otoritasinya (kemampuan aksesnya).

Hubungan antar mekanisme dan layanan jaringan menjelaskan bahwa untuk mewujudkan sebuah layanan keamanan jaringan dibutuhkan mekanisme yang tepat dan tidak semua mekanisme keamanan jarinagn digunakan untuk mewujudkan sebuah layanan keamanan jaringan. Misalnya untuk otentikasi diperlukan beberapa mekanisme keamanan jaringan yaitu *encipherment, digital signature, dan authentication exchanges.* Ketika melakukan analisis kebutuhan terhadap keamanan jaringan, pengembang harus cermat memilih layanan keamanan jaringan yang tepat untuk memenuhi kebutuhan itu. (Rifki Sadikin, 2017 : 5).

II.2.3.1. Serangan Keamanan Jaringan

Sistem keamanan jaringan yang dioperasionalkan pada jaringan publik rentan terhadap serangan oleh siapapun. Orang yang berusaha meruntuhkan keamanan jaringan disebut sebagai penyerang (penyerang). Penyerang menyerang sistem keamanan jaringan untuk mengalahkan tujuan layanan keamanan jaringan. Misalnya penyerang pada layanan kerahaisaan data ingin mengungkap isi teks asli sehingga ia dapat mengungkap teks sandi lainnya. Secara umum serangan pada sistem keamanan jaringan dapat dikatagorikan menjadi 2 jenis : serangan pasif (*passive attack*) dan serangan aktif (*active attack*).

1. Serangan Pasif

Pada serangan pasif, penyerang hanya mengumpulkan data yang melintas pada jaringan publik (jaringan yang bisa diakses oleh penyerang). Serangan pasif tidak melakukan modifikasi data yang melintas atau merusak sistem, penyerang hanya punya keamanan membaca saja (*read only*). Lalu berdasarkan data yang dikumpulkan, penyerang melakukan analisis untuk mengagalkan tujuan layanan keamanan jaringan. Karena tidak melakukan perubahan data dan menganggu sistem, serangan pasif susah untuk dideteksi namun serangan pasif dapat dicegah dengancra misalnya selalu menggunakan sandi (*encryption)* ketika pengirim pesan. Oleh karena itu, penekanan untuk mengatasi serangan pasif lebih pada pencegahan daripada pendeteksian. Berikut ini beberapa jenis serangan yang digolongkan sebagai serangan pasif :

a. *Snooping*

Snooping merujuk pada kegiatan yang bermasksud mendapatkan data yang tengah terkirim pada jaringan biasanya melalui akses yang tak berwenang. Contoh aktivitas *Snooping* misalnya sebuah email disadap oleh penyerang. Untuk mengalahkan penyerang sehingga aktivitas Snooping tidak bermakna data yang dikirim dibuat tidak kaset mata (*monintelligible*) dengan menggunakan mekanisme penyandian (*encipherment*).

b. *Traffic Analysis*

Traffic Analysis merupakan kegiatan serangan pasif dengan melakukan *monotoring* terhadap lalu lintas data pada jaringan. Data-data lalu lintas jaringan dikumpulkan dan kemudian dianalisis sehingga penyerang dapat mengetahui maksud data-data itu. (Rifki Sadikin, 2017 : 7).

2. Serangan Aktif

Sebuah serangan aktif (active attack) dapat mengakibatkan perubahan data yang terkirim dan jalannya sistem terganggu. Pada serangan aktif seakan-akan penyerang memperoleh kemampuanan untuk mengubah data pada lalu lintas data selain kemampuan baca. Jens-jenis serangan aktif adalah sebagai berikut:

a. *Masquerade*

Masquerade adalah serangan aktif yang dilakukan oleh penyerang dengan cara penyerang mengambil alih (menirukan) perilaku pengirim atau penerima. Misalnya pada saat Alice ingin membuat kunci bersama denagn Bob, Eve mengambil alih peran Bob sehingga Alice tidak sadar bahwa ia mengirim psean ke EVE bukan pada Bob.

b. *Modification*

Modification adalah serangan aktif yang dilakukan oleh penyerang dengan cara penyerang mengambil alih jalur komunikasi untuk mengubah atau menghapus atau menunda pesan yang sedang dikirim untuk keuntungan penyerang. Contohnya sebuah pesan "Kirim 1000o ke Akun Alice" diubah oleh Eve menjadi "Kirim 10000 ke Akun Eve".

c. *Replay*

Replay adalah serangan aktif yang terdiri atas pencatatan secara pasif data unit dan transmisi ulang untuk menimbulkan efek yang diinginkan penyerang. Contohnya Eve pernah meimnta Bob mengirim 10000 ke Eve, lalu Bob mengirim pesan "Kirim 10000 ke Eve" ke Bank, Eve mencatat pesan "Kirim 10000 ke Eve" dan mengirim ulang ke Bank.

d. *Denial Of Service*

Denial Of Service adalah serangan aktif yang bertujuan agar sistem menjadi collapse sehingga tidak mampu memberikan respon atau layan yang semestinya kepada pengguna. Serangan ini biasanya dilakukan dengan membuat *server* menjadi *overload* dengan permintaan bodong (*dummy*). Untuk mengembangkan sistem keaman jarinagn yang aman, perancang keamanan jarinagn harus menganalisis kemungkinan serangan-serangan atas layanan keaman jaringan. Biasanya sebuah sistem keamanan jaringan dikatakan aman bila sistem itu mampu bertahan terhadap serangan aktif. (Rifki Sadikin, 2017 : 8).

II.2.4. Metode SHA-512

Fungsi yang terjadi pada algoritma SHA-2 mirip dengan yang terjadi pada SHA-1. Proses yang terjadi pada SHA-512 adalah :

+ Sebelum pemrosesan

- Mengubah string masukan menjadi kumpulan bit 0 dan 1 - Penambahan bit ‘1’ ke pesan

- Penambahan k bit ‘0’,dimana k adalah nilai minimum > = 0 sehingga pesan yang memiliki kelipatan 1024 bit

- Tambahkan panjang pesan, dalam bit, sebagai 64-bit integer.

+ Pada saat pemrosesan

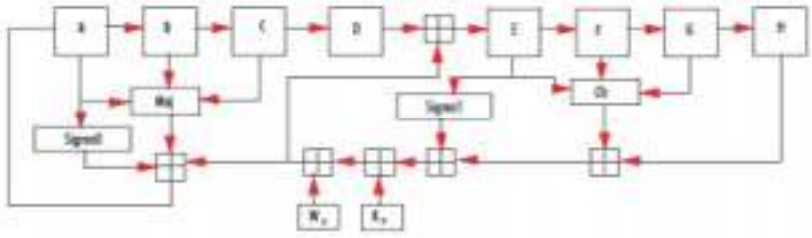
- Memiliki 8 buah penyangga yang berukuran 64 bit - Memiliki 80 buah konstanta yang berukuran 64 bit

- Bagi kumpulan bit setiap 1024 bit

- Pada tiap kumpulan 1024 bit tersebut, bagi tiap bit tersebut menjadi 16 bagian

- Perpanjang jadi 80 bagian, dengan menggunakan fungsi Gamma0 dan Gamma1

- Operasi yang terjadi pada tiap putaran adalah :



Gambar 2.1. Operasi fungsi *hash* SHA-2

(Sumber : Setiawan, 2011 : 2)

Nilai A sampai H adalah 8 buah penyangga yang telah ditentukan sebelumnya. Fungsi Maj yang ada pada gambar diatas adalah fungsi yang menerima 3 buah variabel dengan aturan melakukan operasi : (((A ∨ B) ∧ C) ∨ (A ∧ B)).

Sedangkan fungsi Sigma0 adalah fungsi yang melakukan operasi : (ROR (x,28) ⊕ ROR (x,34) ⊕ ROR (x,39)). Yang dimaksud dengan ROR adalah rotasi bit yang ada ke kanan. Sehingga, yang terjadi pada fungsi Sigma0 adalah rotasikan x ke kanan sebanyak 2 kali , kemudian dilakukan operasi xor dengan x yang dirotasi ke kanan sebanyak 13 kali dan xor dengan x yang dirotasi kanan sebanyak 22 kali. Fungsi Sigma1 juga memiliki operasi yang mirip, yaitu : (ROR (x,14) ⊕ ROR (x,18) ⊕ ROR (x,41)). Fungsi Gamma0 dan Gamma1 adalah fungsi yang dilakukan untuk memperbanyak jumlah potongan pada tiap bagian menjadi 80 bagian dari 16 bagian. Fungsi ini mirip dengan fungsi sebelumnya, yaitu fungsi Sigma0 dan Sigma1.Fungsi Gamma0 memiliki persamaan (ROR (x,1) ⊕ ROR (x,8) ⊕ R (x,7)). Fungsi R berbeda dengan fungsi rotasi kanan. Fungsi R merupakan fungsi untuk melakukan shift bit ke kanan sebanyak yang telah ditentukan. Sedangkan, fungsi Gamma1 memiliki persamaan (ROR (x,19) ⊕ ROR (x,61) ⊕ R (x,6)).

Pada gambar terlihat ada Wx dan Kx. Yang dimaksud dengan nilai Kx adalah nilai konstanta yang telah ditentukan pada tiap putaran. Sedangkan, nilai Wx adalah nilai dari tulisan yang akan di-hash yang sudah ditambah bitnya dan diperpanjang ukurannya sampai 80 buah potongan. Untuk mendapatkan nilai W ke 16 sampai dengan 79 (jika nilai W pertama adalah nilai W ke 0), operasi yang dilakukan adalah melakukan operasi penambahan antara Gamma1 dari W[i-2]

dengan W[i-7] dan juga ditambah dengan hasil penjumlahan antara Gamma0 dari W[i-15] dan W[i-16].

Kemudian, terdapat juga fungsi Ch. Fungsi ini melakukan operasi (G ⊕ (E ∧ (F ⊕ G))). Setelah melakukan operasi-operasi yang ada, nilai selanjutnya dapat ditentukan. Penentuan nilai selanjutnya dilakukan dari H sampai ke A. Nilai H ditentukan dari nilai G, nilai G ditentukan dari nilai F, nilai F ditentukan dari nilai E, nilai E ditentukan dari penjumlahan dari D dan operasi penjumlahan, nilai D ditentukan dari nilai C, nilai C ditentukan dari nilai B, nilai B ditentukan dari nilai A,dan nilai A diperoleh dari penjumlahan.

Fungsi *hash* diperoleh dengan menggabungkan nilai-nilai hasil penjumlahan sebelumnya. Fungsi hash SHA- 512 melakukan operasi *hash* yang sama dengan operasi SHA-2 pada umumnya. Yang menjadi perbedaan dengan algoritma hash SHA-2 yang lainnya adalah angkanya memiliki panjang 64 bit, sedangkan SHA-256 hanya memiliki panjang 32 bit. Selain itu, pada SHA-512 terdapat 80 buah putaran, sedangkan SHA-256 hanya memiliki 64 putaran. Pada SHA-512, nilai 8 buah penyangganya dan konstanta penambahnya diperpanjang mencapai 64 bit. Operasi *shift* dan rotasi yang dilakukan pada SHA-512 juga berbeda. (Setiawan, 2011 : 2).

II.2.5. Fungsi *Hash*

Fungsi *hash* adalah fungsi yang menerima masukan berupa *string* yang panjangnya sembarang dan memberikan nilai berupa suatu *string* yang memiliki ukuran panjang yang tetap. Fungsi yang dilakukan pada fungsi *hash* disebut juga

sebagai fungsi kompresi. Berbeda dengan fungsi kompresi biasa (seperti pada *file zip*), fungsi hash tidak bisa mendekompresi *file* hasil *hash* yang telah dilakukan. Dengan kata lain, kompresi yang dilakukan adalah kompresi satu arah. Dalam kriptografi, fungsi hash harus memiliki 2 properti untuk dapat berguna : mereka harus satu arah dan harus kebal terhadap kolisi. Satu arah mengartikan bahwa keluaran dari fungsi *hash*, mempelajari sesuatu yang berguna tentang masukan adalah tidak mungkin. Hal ini adalah bagian penting dari *hash*, karena mereka sering digunakan bersamaan dengan data seed RNG dan *user password*. Kebal terhadap kolisi mengartikan bahwa jika diberikan suatu keluaran dari hash, mencari *input* lain yang menghasilkan *output* yang sama adalah tidak mungkin. (Setiawan, 2011 : 2)

II.2.6. *Visual Basic*

Visual Basic 2010 merupakan versi perbaikan dan pengembangan dari versi pendahulunya, yaitu *Visual Basic* 2008. Beberapa pengembangan yang terdapat didalamnya antara lain dukungan terhadap *library* terbaru dari *Microsoft*, yaitu *.Net FrameWork 4.0*, dukungan terhadap aplikasi berbasis Cloud *Computing*, serta perluasan dukungan terhadap *database-database*, baik *standalone* maupun *database server*. (Sari, dkk, 2015 : 2).

II.2.7. *Unified Modeling Language* (UML)

Menurut Windu Gata (2013) 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.

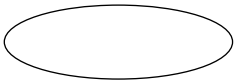
UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem. (Urva dan Siregar, 2015 : 93).

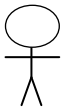
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan UML adalah sebagai berikut :

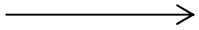
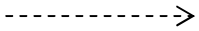
1. *Use Case* Diagram

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 dapat dilihat pada tabel II.4 dibawah ini :

Tabel II.4. Simbol *Use Case*

Gambar	Arti	Keterangan
	*Use Case*	*Use case* menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan

		aktor, dan dinyatakan dengan menggunakan kata kerja di awal nama *use case*.
	Actor (Aktor)	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*.
	Garis (*Line*)	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.

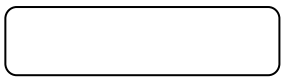
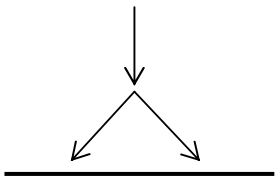
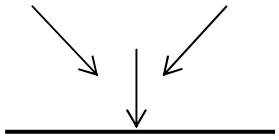
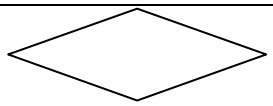
	Panah (*Arrow*)	Asosiasi antara aktor dan *use case* yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.
	*Include*	*Include,* merupakan di dalam *use case* lain (*required*) atau pemanggilan *use case* oleh *use case* lain, contohnya adalah pemanggilan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

(Sumber : Urva dan Siregar, 2015 : 94)

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* dapat dilihat pada tabel II.5 dibawah ini :

Tabel II.5. Simbol *Activity Diagram*

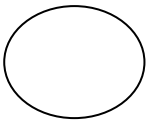
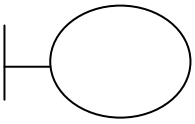
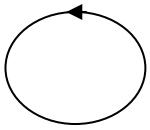
Gambar	Arti	Keterangan
	Titik Awal (*Start Point*)	*D*iletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	Titik Akhir (*End Point*)	*A*khir aktifitas.
	Aktifitas (*Activites*)	Menggambarkan suatu proses/kegiatan bisnis.
	*Fork* (Percabangan)	Digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	*Join* (Penggabungan)	Digunakan untuk menunjukkan adanya dekomposisi.
	Titik Keputusan (*Decision Points*)	Menggambarkan pilihan untuk pengambilan keputusan, *true, false.*
	*Swimline*	*Swimlane*, untuk menunjukkan siapa melakukan apa.

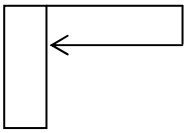
(Sumber : Urva dan Siregar, 2015 : 94)

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* dapat dilihat pada tabel II.6 dibawah ini :

Tabel II.6. Simbol *Sequence Diagram*

Gambar	Arti	Keterangan
	Kelas Entitas (*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.
	Kelas Tampilan (*Boundary Class*)	Berisi kumpulan kelas yang menjadi *interface* atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan *formentry* dan *form* cetak.
	Kelas Kendali (*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.
	Pesan (*Message*)	Simbol mengirim pesan antar *class*.
	Pengiriman Pesan (*Recursive*)	Menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	Aktifasi (*Activation*)	*Activation* mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	Garis Putus (*Lifeline*)	Garis titik-titik yang terhubung dengan objek, sepanjang *lifeline* terdapat *activation*.

(Sumber : Urva dan Siregar, 2015 : 95)

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 yang dapat dilihat pada tabel II.7 dibawah ini :

Tabel II.7. *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 dan Siregar, 2015 : 95)