BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Penelitian yang dilakukan oleh Sholeh, dkk (2013) mengenai Mengamankan Skrip Pada Bahasa Pemograman PHP Dengan Menggunakan Kriptografi Base64, Sholeh, dkk (2013) menyimpulkan bahwa dengan adanya cara pengamanan ini, pengembang aplikasi yang menggunakan bahasa pemograman PHP dapat menyembunyikan skrip PHP supaya tidak mudah disalin, diubah sebagian/ seluruhnya oleh orang yang tidak berhak.

Penelitian yang dilakukan oleh Mukhlis (2013) mengenai Modifikasi *Nihilist Cipher*, Mukhlis menyimpulkan bahwa Algoritma ini cukup kuat pada beberapaserangan seperti *ciphertext-only attack*,*plaintext-only attack*, *chosen-plaintext attack*. Algorima ini lebih kompleks karenamenggunakan bilangan *M*.

Berdasarkan kedua penelitian di atas maka tidak ditemukan adanya kesamaan judul dan sistem pada penelitian ini, oleh karena itu referensi dari kedua penelitian di atas dapat digunakan sebagai 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).

II.2.2. 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. (Sadikin, 2017 : 5).

II.2.2.1. 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 iadapat 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 *monitoring* terhadap lalu lintas data pada jaringan. Data-data lalu lintas jaringan dikumpulkan dan kemudian dianalisis sehingga penyerang dapat mengetahui maksud data-data itu. (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 seranagn aktif yang dilakukan oleh penyerang dengan cara penyerang mengambil alih (menirukan) perilaku pengirim atau penerima. Misalnya pada saat Alice ingin membuat kunci bersama dengan 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 1000 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 meminta 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. (Sadikin, 2017 : 8).

II.2.3. Kriptografi

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 kriptografi 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. (Sadikin, 2017 : 9).

II.2.3.1. Jenis-JenisKriptografi

Berikut ini adalah pembagian kriptografi menurut Sadikin (2017 : 9) :

1. Kriptografi Klasik

Kriptografi klasik merupakan kriptografi yang digunakan pada zaman dahulu sebelum komputer ditemukan atau sudah ditemukan namun belum secanggih sekarang. Kriptografi ini melakukan pengacakan huruf pada kata terang / *plaintext*. Kriptografi ini hanya melakukan pengacakan pada huruf A - Z, dan sangatlah tidak disarankan untuk mengamankan informasi-informasi penting karena dapat dipecahkan dalam waktu singkat. Biarpun telah ditinggalkan, kriptografi klasik tetap dapat ditemui disetiap pelajaran kriptografi sebagai pengantar kriptografi modern.Semenjak ditemukannya komputer digital, metode kriptografi klasik yang bekerja dengan mengacak huruf semakin kehilangan posisinya dan digantikan dengan metode yang lebih baru, dimana yang diacak adalah bit dari huruf bersangkutan. Era kriptografi modern pun dimulai.

Berdasarkan teknik pengenkripsian, kriptografi klasik terbagi menjadi 2 yaitu:

a. Metode substitusi, yang dibagi lagi menjadi 2 yaitu:
    1. *Monoalphabetic*, setiap huruf pesan disubstitusi oleh satu huruf kunci
    2. *Polyalphabetic*, setiap huruf pesan disubstitusi oleh beberapa huruf kunci dengan pola tertentu.

Metode substitusi adalah metode enkripsi dengan mengganti tiap-tiap huruf pesan dengan kunci tertentu menjadi huruf lain.

b. Metode transposisi

Metode transposisi adalah metode enkripsi dengan memindahkan posisi tiap-tiap huruf pesan dengan pola tertentu. Contohnya adalah *Blocking Cipher* dan *Permutation*.

2. Kriptografi Modern

Kriptografi modern menggunakan gagasan dasar yang sama seperti kriptografi klasik (permutasi dan transposisi) tetapi penekanannya berbeda. Pada kriptografi klasik, kriptografer menggunakan algoritma yang sederhana, yang memungkinkan cipherteks dapat dipecahkan dangan mudah (melalui penggunaan statistik, terkaan, intuisi, dan sebagainya). Algoritma kriptografi modern dibuat sedemikian kompleks sedemikian sehingga kriptanalis sangat sulit memecahkan cipherteks tanpa mengetahui kunci.

Algoritma kriptografi modern umumnya beroperasi dalam mode bit ketimbang mode karakter. Operasi dalam mode *bit* berarti semua data dan informasi (baik kunci, plainteks, maupun cipherteks) dinyatakan dalam rangkaian (string) bit biner, 0 dan 1. Algoritma enkripsi dan dekripsi memproses semua data dan informasi dalam bentuk rangkaian *bit*. Rangkaian *bit* yang menyatakan plainteks dienkripsi menjadi cipherteks dalam bentuk rangkaian *bit*, demikian sebaliknya. Berikut ini adalah jenis-jenis kriptografi modern :

a. Fungsi *Hash*

Fungsi *hash* adalah fungsi yang melakukan pemetaan npesan dengan panjang sembarang ke sebuah teks khusus yang disebut *message digest* dengan panjang

tetap. Fungsi *hash* umumnya dipakai sebagai nilai uji (*check value*) pada mekanisme keutuhan data.

b. Penyandian dengan kunci simetrik (*Symmetric Key Encipherment*)

Penyandian dengan kunci simetrik adalah penyandian yang kunci enkripsi dan kunci simetrik adalah penyandian yang kunci enkripsi dan kunci dekripsi bernilai sama. Kunci pada penyandian simetrik diasumsikan bersifat rahasia hanya pihak yang melakukan enkripsi dan dekripsi yang mengetahui nilainya. Oleh karena itu penyandian dengan kunci simetrik disebut juga penyandian dengan kunci rahasia *secret key encipherment*.

II.2.4. Metode *Nihilist Cipher*

Nihilist cipher pertama kali dikembangkan oleh para*Russian Nihilist*, yaitu orang-orang Rusia yangmendukung cara kekerasan untuk mencapaiperubahan politik yang diinginkan, dalam hal inimenggulingkan kekuasaan Tsar Alexander II diRusia.

Mereka memanfaatkan algoritma *Nihilist*untuk berkomunikasi dan mengorganisasikan parateroris untuk melawan para pendukung Tsar pada tahun 1880-an.Selain itu, algoritma ini juga banyak digunakan oleh*First Chief Directorate*, sebuah divisi dari KGB(badan intelejen Rusia) untuk berkomunikasi paracalon mata-mata mereka. Serta digunakan pulauntuk berkomunikasi dengan para sekutu mereka. Dengan diterapkan cara tersebut maka kecurian pesan dapat di atasi. (Mukhlis, 2013 :2). Berikut ini adalah langkah-langkah metode *Nihilist Cipher*.

a. Langkah I:

Persiapkan 2 kata kunci, dengan syarat:

a. Kata Kunci I : ≤ 25 huruf

b. Kata Kunci II : ≤ plainteks

b. Langkah II:

Misalkan, kata kunci I adalah KUNCI, masukkankata ini ke dalam baris I *Polybius Square*, kemudiandiikuti dengan huruf lainnya yang belum ada dalamkata kunci :

Tabel II.1. Tabel kunci

	1	2	3	4	5
1	K	U	N	C	I
2	A	B	D	E	F
3	G	H	L	M	O
4	P	Q	R	S	T
5	V	W	X	Y	Z

c. Langkah III:

Lalu, misalkan kata kunci II adalah KRIPTO,kemudian berdasarkan tabel kunci diatas maka,koordinat yang berkoresponden dengan kata kunci :

K	R	I	P	T	O
11	43	15	41	45	35

Gambar II.1. Koordinat Kata Kunci II

d. Langkah IV:

Misalkan plainteksnya adalah MATA KULIAH,kemudian lakukan langkah III pada plainteks, sehingga :

M	A	T	A	K	U	L	I	A	H
34	21	45	21	11	12	33	15	21	32

Gambar II.2. Koordinat Plainteks

e. Langkah V:

Lakukan operasi pertambahan antara koordinatplainteks dengan kata kunci II, sehingga akan didapat *cipherteks* :

kt	11	43	15	41	45	35	11	43	15	41
pt	34	21	45	21	11	12	33	15	21	32
ct	45	64	60	62	56	47	44	58	36	73

Gambar II.3. Hasil Chiperteks dengan *Nihilist*

3. 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. (Sadikin, 2017 : 9).
4. Penyandian dengan kunci asimetrik (*Asymmetric Key Encipherment*).

   Penyandian dengan kunci asimetrik atau sering juga disebut dengan penyandian kunci publik (*public key*) adalah penyandian dengan kunci enkripsi dan dekripsi berbeda nilai. Kunci enkripsi yang juga disebut dengan kunci publik (*public key*) bersifat terbuka. Sedangkan, kunci dekripsi yang juga disebut kunci privat (*private key*) bersifat tertutup/rahasia. (Sadikin, 2017 : 10).

II.2.5. *Keystream Generator*

Keystream merupakan *bit-bit* kunci yang digunakan untuk enkripsi/dekripsi. *Keystream* umumnya dikenal pada *cipher*aliran, yang termasuk ke dalam algoritma kriptografi modern. *Keystream* tersebut dibangkitkan oleh *Keystream generator*. *Keystream generator* menerima masukan kunci U dan akan menghasilkan kunci (*keystream*) yang digunakan untuk enkripsi/dekripsi. Pengirim dan penerima pesan harus memiliki kunci U yang sama. Kunci U tersebut harus dijaga kerahasiaannya. Dengan menggunakan *keystream generator*, kunci U semula akan menghasilkan kunci (*keystream*) yang akan digunakan untuk enkripsi/dekripsi. Pendekatan terhadap prinsip tersebutlah yang akan digunakan untuk memodifikasi *Vigenere Cipher*. Penerapan pendekatan tersebut pada *Vigenere Cipher* dilakukan pada kunci U semula yang panjangnnya lebih pendek dari pada panjang plainteks. (Abhirama, 2014 : 2).

II.2.6. *Hypertext Preprocessor* (PHP)

PHP adalah singkatan dari PHP *Hypertext Preprocessor* yang digunakan sebagai bahasa *script server-side* dalam pengembangan *web* yang disisipkan pada dokumen HTML.PHP adalah bahasa *scripting* yang menyatu dengan HTML dan dijalankan pada *server side*. Artinya semua sintaks yang kita berikan akan sepenuhnya dijalankan pada *server* sedangkan yang dikirimkan ke *browser* hanya hasilnya saja. (Warnman dan Zahni, 2013 : 31).

II.2.7. *Hypertext Markup Language* (HTML)

Hypertext Markup Language (HTML) adalah suatu bahasa yang dikenali oleh *web browser* untuk menampilkan informasi dengan lebih menarik dibandingkan dengan tulisan teks biasa (*plaint text*). Sedangkan *web browser* adalah bahasa program komputer yang digunakan untuk membaca HTML, kemudian menerjemahkan dan menampilkan hasilnya secara *visual* ke layar komputer. (Nugraha, dkk, 2014 : 175).

II.2.8. MySQL

MySQL adalah *Relation Database Management System* (RDBMS) yang didistribusikan secara gratis di bawah lisensiGPL (*General Public License).* MySQL merupakan turunan dari salah satu konsep utama dalam *database* sejak lama, yaitu SQL (*Structure Query Language)*. SQL merupakan salah satu konsep pengoperasian *database,* terutama sebagai seleksi dan pemasukan data,yang memungkinkan pengoperasian datanya dikerjakan dengan mudah secara otomatis. (Inayah, dkk, 2015 : 5).

II.2.9. *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.

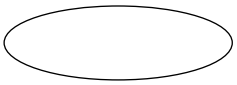
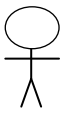
UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem. (Urva dan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

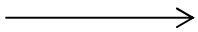
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan UML adalah sebagai berikut:

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.2. Simbol *Use Case*

Gambar	Keterangan
	*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*.
	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.

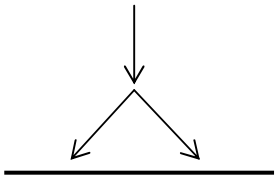
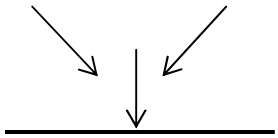
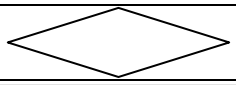
(Sumber:Urva dan Siregar, 2015 : 94)

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.3. 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.

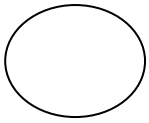
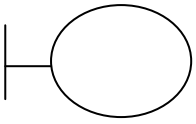
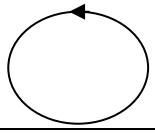
	*Fork* (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	*Join* (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	*Decision Points,* menggambarkan pilihan untuk pengambilan keputusan, *true, false.*
New Swimline	*Swimlane*, untuk menunjukkan siapa melakukan apa.

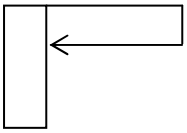
(Sumber : Urva dan Siregar, 2015 : 94)

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.4. Simbol *Sequence Diagram*

Gambar	Keterangan
	*EntityClass,* 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 : 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.5. *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)

II.2.10. Sistem *Login*

Log masuk (bahasa Inggris: *login*, juga biasa disebut sebagai *log in*, *log on*, *logon*, *signon*, (*sign on*, *signin*, *sign in*) adalah proses untuk mengakses komputer dengan memasukkan identitas dari akun pengguna dan kata sandi guna mendapatkan hak akses menggunakan sumber daya komputer tujuan.Pada saat melakukan *login* untuk masuk ke dalam sistem, *user* akan diminta memasukkan identitas *user* seperti identitas *user id* dan *password* sebagai antisipasi dalam hal pengamanan sistem (Agus, 2014 : 173).