BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Penelitian pertama yang dilakukan oleh Fresly Nandar Pabokory, Indah Fitri Astuti, Awang Harsa Kridalaksana (2015), Dari Program Studi Ilmu Komputer, FMIPA, Universitas Mulawarman pada jurnal penelitian yang berjudul "Implementasi Kriptografi Pengamanan Data Pada Pesan Teks, Isi File Dokumen, Dan File Dokumen Menggunakan Aagoritma Advanced Encryption Standard" Menjelaskan bahwa pengamanan data pada pesan teks, isi file dokumen, dan file dokumen dengan menggunakan algoritma Advanced Encryption Standard serta pendukung kemanan steganografi dalam penyembunyian pesan teks atau file dalam file citra dalam penggunaan Application Fres-CAESAS, *user* bebas untuk memproses pengamanan data informasinya (pesan rahasia) dengan melakukan teknik kriptografi yang terdapat beberapa macam keamanan, atau melakukan teknik *steganografi*, atau melakukan teknik kombinasi kriptografi dan *steganografi* yang di dalamnya terdapat beberapa tahapan keamanan pada sistem aplikasi *Fres-CAESAS,* atau melakukan kombinasi sesuai keinginan user dalam pengamanan sebuah data informasi atau pesan rahasia. Berdasarkan penggunaan *Application FresCAESAS*, bahwa sistem keamanan tersebut dibuat dengan keamanan yang berlapis-lapis atau bertahap-tahap atau serumit mungkin agar data informasi atau pesan rahasia tersebut dapat terjaga keamanannya, sehingga akan susah untuk

menjebol atau terjadinya kebocoran data informasi atau pesan rahasia tersebut. Ketika file rahasia yang disisipkan atau disembunyikan di dalam file citra (gambar) tersebut dapat dideteksi, file rahasia tersebut masih belum bisa terbaca informasinya karena masih dienkripsi file dokumennya, kemudian isi file dokumennya, dan selanjutnya pesan teksnya (*plaintext*). File citra (gambar) yang mengalami proses penyisipan sebuah pesan rahasia atau file rahasia tidak mengalami banyak perubahan. Gambar yang dihasilkan terlihat masih sama dengan citra aslinya, hanya berbeda pada ukurannya saja.

Peneliti yang kedua yang dilakukan oleh Mhd.Fachrul Fachrozi1, Hasanul Fahmi (2021), Program Studi Teknik Informatika, STMIK Pelita Nusantara Pada Jurnal Penelitian yang berjudul "Penerapan Metode AES-128 Untuk Pengamanan Data Absensi FingerPrint Di Balai Penelitian Sungei Putih" Menjelaskan bahwa recorddata absensi yang telah dijaga integritasnya ini tidak dapat dimanipulasi, karena satu *record* data saling terkait dengan *record* data yang lain pada *database*. Sehingga apabila satu *record* data diubah, maka akan merusak *record* data yang lain. Dengan adanya sistem keamanan *database* dapat mengambarkan bahwa data absensi tersebut dapat diamankan secara real atau data tersebut tidak bisa disadap ataupun dimanipulasi oleh orang yang tidak bertanggung jawab. dengan mengimplementasikan sebuah keamanannya dengan menggunakan algoritma AES 128. Penerapan metode AES-128 dalam keamanan data absensi pegawai dapat mempermudah admin melakukan keamanan data dan menjaga kerahasiaan data dari orang lain. Mempunyai pengamanan ganda yaitu *form login user* dapat memegang suatu *username* dan *password*. Jika dari pihak luar atau pihak yang

ingin membuka aplikasi tersebut maka tidak bisa karena cukup admin saja yang bisa membuka atau menjalankan aplikasi tersebut.

II.2. Landasan Teori

Berikut ini ada landasan teori yang berkaitan dengan yang di ambil dari berbagai jurnal dan buku :

II.2.1 Aplikasi

Secara istilah pengertian aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanankan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju. Menurut kamus computer eksekutif, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang di harapkan. (Stevanus, 2020 : 29)

II.2.2. Database

Database adalah kumpulan informasi yang disimpan di dalam komputer secara sistematik sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. *Database* adalah representasi kumpulan fakta yang saling berhubungan disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan. *Database* merupakan sekumpulan informasi yang saling berkaitan pada suatu subjek tertentu pada tujuan tertentu pula. *Database* adalah susunan *record* data operasional lengkap dari suatu organisasi atau perusahaan,

yang diorganisir dan disimpan secara terintegrasi dengan menggunakan metode tertentu dalam komputer sehingga mampu memenuhi informasi yang optimal yang dibutuhkan oleh para pengguna. (Helmud, Ellya, 2021 : 81)

II.2.3. Dokumen Digital

Dokumen merupakan suatu sarana transformasi informasi dari satu orang ke orang lain atau dari suatu kelompok ke kelompok lain. Dokumen meliputi berbagai kegiatan yang diawali dengan bagaimana suatu dokumen dibuat, dikendalikan, diproduksi, disimpan, didistribusikan, dan digandakan. Dokumen digital merupakan setiap informasi elektronik yang dibuat, diteruskan, dikirimkan, diterima, atau disimpan dalam bentuk analog, digital, elektromagnetik, optikal, atau sejenisnya, yang dapat dilihat, ditampilkan dan/atau didengar melalui komputer atau sistem elektronik, termasuk tetapi tidak terbatas pada tulisan, suara atau gambar, peta, rancangan, foto atau sejenisnya, huruf, tanda, angka, kode akses, simbol atau perforasi yang memiliki makna atau arti atau dapat dipahami oleh orang yang mampu memahaminya. (Pabokory, Fresly Nandar . dkk. 2015 : 21)

II.2.4. Kriptografi

Kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya. Dalam ilmu kriptografi, terdapat dua buah proses yaitu melakukan enkripsi dan dekripsi. Pesan yang akan dienkripsi disebut sebagai *plaintext* (teks biasa). Disebut demikian karena informasi ini dengan mudah dapat dibaca dan dipahami oleh siapa saja. Algoritma yang dipakai untuk mengenkripsi dan mendekripsi sebuah *plaintext*

melibatkan penggunaan suatu bentuk kunci. Pesan *plaintext* yang telah dienkripsi (atau dikodekan) dikenal sebagai *ciphertext* (teks sandi).

Di dalam kriptografi kita akan sering menemukan berbagai istilah atau terminology. Beberapa istilah yang harus diketahui yaitu :

1. Pesan, *Plainteks*, dan *Cipherteks* Pesan (*message*) adalah data atau informasi yang dapat dibaca dan dimengerti maknanya. Nama lain untuk pesan adalah (*plaintext*) atau teks jelas (*cleartext*).

2. Pengirim dan Penerima Komunikasi data melibatkan pertukaran pesan antara dua entitas. Pengirim (sender) adalah entitas yang mengirim pesan kepada entitas lainnya. Penerima (*receiver*) adalah entitas yang menerima pesan.

3. Enkripsi dan dekripsi Proses menyandikan *plainteks* menjadi *cipherteks* disebut enkripsi (*encryption*) atau *enciphering* (standard nama menurut ISO 74982). Sedangkan proses mengembalikan *cipherteks* menjadi *plainteks* semula disebut dekripsi (*decryption*) atau *deciphering* (standard nama menurut ISO 74982).

4. Cipher dan kunci Algoritma kriptogarfi disebut juga cipher, yaitu aturan untuk enkripsi dan dekripsi, atau fungsi matematika yang digunakan untuk enkripsi dan dekripsi. Beberapa cipher memerlukan algoritma yang berbeda untuk enkripsi dan dekripsi. Konsep matematis yang mendasari algoritma kriptografi adalah relasi antara dua buah himpunan yang berisi elemen-elemen plainteks dan himpunan yang berisi *cipherteks*.

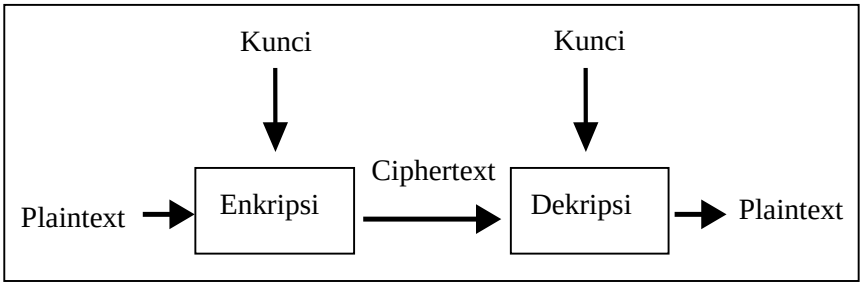
Enkripsi dan dekripsi merupakan fungsi yang memetakan elemen-elemen antara dua himpunan tersebut. Misalkan P menyatakan plainteks dan C menyatakan *cipherteks*, maka :

$E(P) = C$ → fungsi enkripsi E memetakan P ke C

$D(C) = P$ → fungsi dekripsi D memetakan C ke P

Karena proses enkripsi kemudian dekripsi mengembalikan pesan ke pesan asal, maka persamaan $D(E(P)) = P$ harus benar.

Kriptografi mengatasi masalah keamanan data dengan menggunakan kunci, yang dalam hal ini algoritma tidak dirahasiakan lagi, tetapi kunci harus tetap dijaga kerahasiaannya. Kunci (key) adalah parameter yang digunakan untuk transformasi enkripsi dan dekripsi. Kunci biasanya berupa string atau deretan bilangan. Dengan menggunakan kunci K, maka fungsi enkripsi dan dekripsi dapat dilihat pada Gambar II.1. dibawah ini.



Gambar II.1. Fungsi Enkripsi dan Dekripsi

(Sumber : (Pabokory, Fresly Nandar . dkk. 2015 : 22))

kriptografi bertujuan untuk member layanan keamanan. Yang dinamakan aspek-aspek keamanan:

1. Kerahasiaan (*confidentiality*) Adalah layanan yang ditujukan untuk menjaga agar pesan tidak dapat dibaca oleh pihak-pihak yang tidak berhak.

2. Integritas data (data *integrity*) Adalah layanan yang menjamin bahwa pesan masih asli atau belum pernah dimanipulasi selama pengiriman.

3. Otentikasi (*authentication*) Adalah layanan yang berhubungan dengan identifikasi, baik mengidentifikasi kebenaran pihak-pihak yang berkomunikasi (*user autehentication*).

4. *Non-repudiation* Adalah layanan untuk menjaga entitas yang berkomunikasi melakukan penyangkalan.

II.2.5. Algoritma AES (Advanced Encryption Standard)

Perlu diketahui bahwa pada tahun 2000, algoritma *Rijndael* terpilih sebagai algoritma kriptografi yang selain aman juga efisien dalam implementasinya dan dinobatkan sebagai AES. Nama *Rijndael* sendiri berasal dari gabungan nama penemunya, tetapi penulis pokus membahas pada algoritma Advanced Encryption Standard (AES).

Advanced Encryption Standard (AES) merupakan algoritma *cryptographic* yang dapat digunkan untuk mengamakan data. Algoritma AES adalah *blokchipertext* simetrik yang dapat mengenkripsi (*encipher*) dan dekripsi (*decipher*) informasi. Enkripsi merubah data yang tidak dapat lagi dibaca disebut *ciphertext*, sebaliknya dekripsi adalah merubah *ciphertext* data menjadi bentuk semula yang kita kenal sebagai *plaintext*. Algoritma AES mengunakan kunci kriptografi 128, 192, dan 256 bits untuk mengenkrip dan dekrip data pada blok 128 bits. Pemilihan ukuran blok data dan kunci akan menentukan jumlah proses yang harus dilalui untuk proses enkripsi dan dekripsi. Perbandingan jumlah proses

yang harus dilalui untuk masing - masing masukan. Dapat dilihatkan pada tabel II.1. Jumlah proses berdasarkan bit blok dan kunci

Tabel II.1. Jumlah proses berdasarkan bit blok dan kunci

Panjang Kunci Dalam bit	Panjang Kunci (Nk) Dalam Word	Ukuran Blok Data (Nb) Word	Jumlah Proses (Nr)
128	4	4	10
192	6	4	12
256	8	4	14

(Sumber : (Pabokory, Fresly Nandar . dkk. 2015 : 23))

Blok-blok data masukan dan kunci dioperasikan dalam bentuk array. Setiap anggota array sebelum menghasilkan keluaran ciphertext dinamakan dengan state. Setiap state akan mengalami proses yang secara garis besar terdiri dari empat tahap yaitu, *AddRoundKey*, *SubBytes*, *ShiftRows*, dan *MixColumns*. Kecuali tahap *MixColumns*, ketiga tahap lainnya akan diulang pada setiap proses sedangkan tahap MixColumns tidak akan dilakukan pada tahap terakhir.

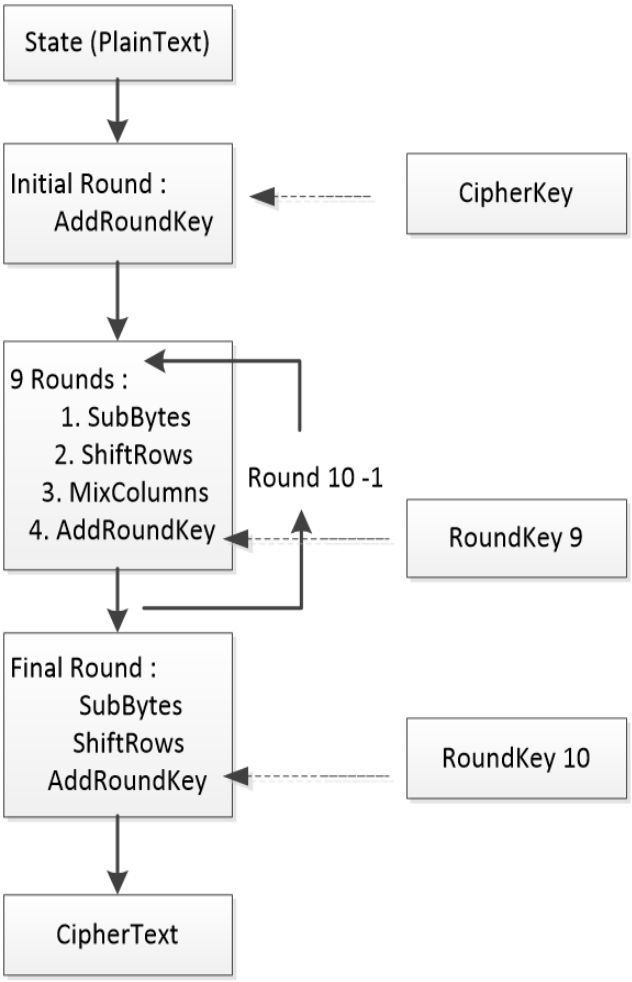
Karena terjadi beberapa tahap dalam proses enkripsi, maka diperlukan *subkey-subkey* yang akan dipakai pada tiap tahap. Pengembangan jumlah kunci yang akan dipakai diperlukan karena kebutuhan *subkey-subkey* yang akan dipakai dapat mencapai ribuan bit, sedangkan kunci yang disediakan secara default hanya 128-256 bit. Jumla total kunci yang diperlukan sebagai *subkey* adalah sebanyak Nb(Nr+1), dimana Nb adalah besarnya blok data dalam satuan *word*. Sedangkan Nr adalah jumlah tahapan yang harus dilalui dalam satuan *word*. Sebagai contoh,

bilamana digunakan 128 bit (4 *word*) blok data dan 128 bit (4 *word*) kunci maka akan dilakukan 10 kali proses.

Dengan demikian dari rumus didapatkan 4(10+1)=44 *word*=1408 bit kunci. Untuk melakukan pengembangan jumlah kunci yang akan dipakai dari kunci utama maka dilakukan *key schedule*. (Pabokory, Fresly Nandar . dkk. 2015 : 23).

II.2.5.1. Enkripsi AES (Advanced Encryption Standard)

Proses enkripsi pada algoritma Adavanced Encryption Standard terdiri dari 4 jenis transformasi bytes, yaitu *SubBytes*, *ShiftRows*, *Mixcolumns*, dan *AddRoundKey*. Pada awal proses enkripsi, input yang telah dicopykan ke dalam state akan mengalami transformasi byte *AddRoundKey*. Setelah itu, *state* akan mengalami transformasi *SubBytes*, *ShiftRows*, *MixColumns*, dan *AddRoundKey* secara berulangulang sebanyak Nr. Proses ini dalam algoritma AES disebut sebagai *round function*. *Round* yang terakhir agak berbeda dengan round-round sebelumnya dimana pada round terakhir, state tidak mengalami transformasi MixColumns. Diagram alur proses enkripsi pada algoritma Advanced Encryption Standard dapat dilihat pada Gambar II.2. dibawah ini

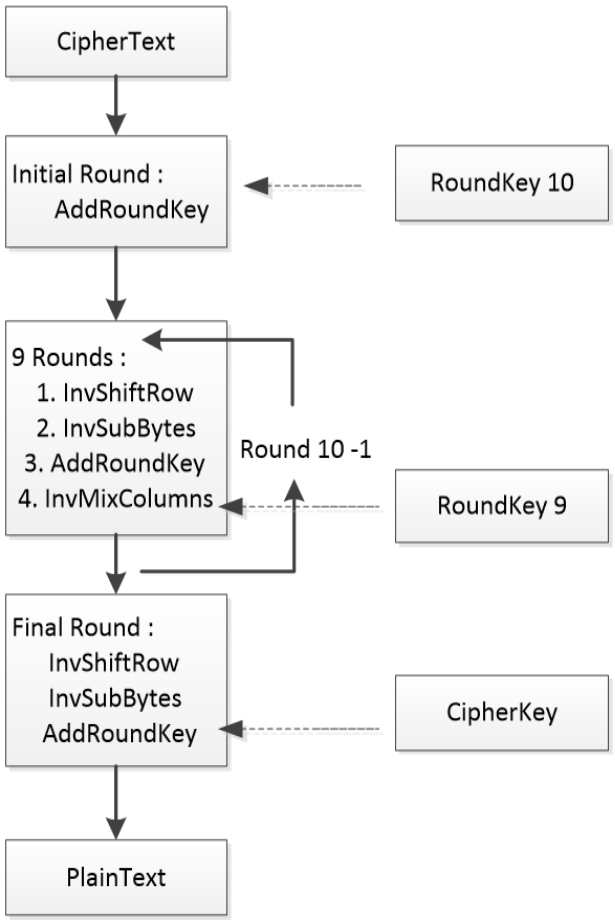


Gambar II.2. Diagram Alur Proses Enkripsi

(Sumber : (Pabokory, Fresly Nandar . dkk. 2015 : 24))

II.2.5.2. Dekripsi AES (Advanced Encryption Standard)

Transformasi cipher dapat dibalikkan dan diimplementasikan dalam arah yang berlawanan untuk menghasilkan inverse cipher yang mudah dipahami untuk algoritma AES. Transformasi byte yang digunakan pada *invers cipher* adalah *InvShiftRows*, *InvSubBytes*, *InvMixColumns*, dan *AddRoundKey*. Diagram alur proses dekripsi pada algoritma Advanced Encryption Standard dapat dilihat pada Gambar II.3. dibawah ini



Gambar II.3. Diagram Alur Proses Dekripsi

(Sumber : (Pabokory, Fresly Nandar . dkk. 2015 : 24))

II.2.5.3. Ekspansi Kunci AES (Advanced Encryption Standard)

Algoritma AES mengambil kunci *cipher*, K, dan melakukan rutin ekspansi kunci (*key expansion*) untuk membentuk *key schedule*. Ekspansi kunci menghasilkan total Nb(Nr+1) word. Algoritma ini membutuhkan set awal key yang terdiri dari Nb *word*, dan setiap *round* Nr membutuhkan data kunci sebanyak Nb *word*.

Hasil *key schedule* terdiri dari array 4 byte *word linear* yang dinotasikan dengan [wi]. *SubWord* adalah fungsi yang mengambil 4 byte *word input* dan mengaplikasikan S-Box ke tiap-tiap data 4 byte untuk menghasilkan *word output*. Fungsi *RotWord* mengambil *word* [a0, a1, a2, a3] sebagai *input*, melakukan permutasi siklik, dan mengembalikan *word* [a1, a2, a3, a0]. Rcon[i] terdiri dari nilai-nilai yang diberikan oleh [xi-1, {00}, {00}, {00}], dengan xi-1 sebagai pangkat dari x (x dinotasikan sebagai {02} dalam *field GF*(28). *Word* ke Nk pertama pada ekspansi kunci berisi kunci *cipher10*.

Setiap *word* berikutnya, w[i], sama dengan XOR dari *word* sebelumnya, w[i-1] dan *word* Nk yang ada pada posisi sebelumnya, w[i-Nk]. Untuk *word* pada posisi yang merupakan kelipatan Nk, sebuah transformasi diaplikasikan pada w[i-1] sebelum XOR, lalu dilanjutkan oleh XOR dengan konstanta *round*, Rcon[i]. Transformasi ini terdiri dari pergeseran siklik dari byte data dalam suatu *word RotWord*, lalu diikuti aplikasi dari *lookup* Tabel untuk semua 4 byte data dari *word SubWord*. (Pabokory, Fresly Nandar . dkk. 2015 : 23).

II.2.6. Hypertext Preprocessor (PHP)

PHP adalah *scripting* yang menyatu dengan HTML dan dijalankan pada *serverside.* Artinya semua sintak yang diberikan akan sepenuhnya dijalankan pada server sedang yang dikirim ke *browser* hanya hasilnya saja. Ketika seorang pengguna *internet* membuka situs yang menggunakan fasilitas *serverside scripting* PHP, maka terlebih dahulu *server* yang bersangkutan akan memproses semua perintah PHP di *server* lalu mengirimkan hasilnya dalam format HTML ke *web*

server pengguna *internet* tadi. Sehingga kode asli yang ditulis dengan PHP tidak terlihat di *browser* pengguna. PHP berfungsi mengambil informasi dari *form* berbasis *web* dan menggunakannya untuk berbagai macam fungsi, sebagai bahasa untuk mengidentifikasi seberapa banyak pengunjung menggunakan bahasa PHP, pengaturan *layout* dalam berbagai macam *browser* seperti *firefox*. Bahasa pemrograman PHP dalam *web* sangat luas, jadi semua tergantung pada pengguna, jika pengguna sangat teliti maka semua fungsi akan diketahui. (Wulandari dan Aprilia, 2015 : 43)

II.2.7. MySQL

MySQL adalah suatu perangkat lunak *database* relasi atau *Relational Database Management System* (RDBMS) yang didistribusikan gratis dibawah lisensi *General Public License* (GPL). Dimana setiap orang bebas mengunakan MySQL, namun tidak boleh dijadikan produk turunan yang dijadikan *closed source* atau komersial. (Muslihudin dan Larasati, 2014 : 34)

II.2.8. *Unified Modeling Language* (UML)

UML Merupakan Metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam industry perangkat lunak dan pengembangan sistem (Urva dan Siregar, 2015 : 93).

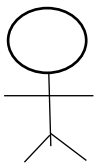
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan UML adalah sebagai berikut :

1. *Use Case* Diagram

Use Case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use Case* mendekripsikan sebuah interaksi anatara satu atau lebih *actor* 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.2. 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 actor, dan dinyatakan dengan menggunakan kata kerja awal nama *use case*
	Aktor adalah abstraction dari orang atau sistem yang lain mengktikan fungsi dari target sistem untuk mengidentifikasikan actor, harus ditentukan pembagian tenaga kerja dn tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang tau sistem bisa muncul dalam beberapa peran.

	Asosiasi antara actor dan use case, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
	Asosiasi antara actor dan use case yang menggunakan panah terbuka untuk mengindinkasikan bila actor berintakasi 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.

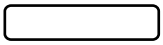
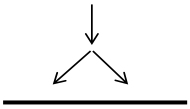
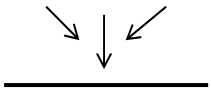
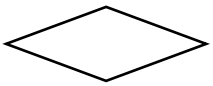
(Sumber : Urva dan Siregar, 2015 : 94)

2. Diagram Aktivitas (*Activity* Diagram)

Activity Diagram menggambarkan *Workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktifitas menggambarkan aktifitas sistem bukan apa yang dilakukan *actor*, jadi aktifitas yang dapat dilakukan oleh sistem (Urva dan siregar, 2015 : 94).

Simbol-simbol yang digunakan dalam *Activity* diagram dapat dilihat pada Tabel II.3. dibawah ini.

Tabel II.3. Simbol *Activity* Diagram

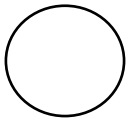
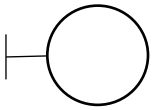
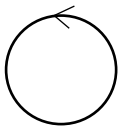
Gambar	Keterangan
	*Start point*, diletakan pada pojok kiri atas dan merupakan awal aktivitas.
	*End point*, akhir aktifitas
	*Activites*, menggambarkan proses/kegiatan bisnis.
	*Fork* (percabagan), digunakan untuk menunjukkan kegiatan yang dilakaukan secara parallel atau untuk menggabungan dua kegiatan pararel menjadi satu.
	*Join* (Penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	*Decision points*, menggambarkan pilihan untuk pengambilan keputusan, true, false.
	*Swimlane*, untuk menunjukan siapa melakukan apa.

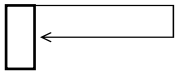
(Sumber : Urva dan Siregal, 2015 : 94)

3. Diagram Urutan (*Sequence* Diagram)

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antara objek. Oleh karena itu untuk menggambarkan diagram *sequence* maka harus diketahui objek-objek yang dilibatkan dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang di instasikan menjadi objek itu (Urva dan Siregar, 2015 : 95). Simbol-simbol yang digunakan dalam *Sequence* diagram dapat dilihat pada Tabel II.4. dibawah ini.

Tabel II.4. 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 interfase atau interaksi antara satu atau lebih actor dengan sistem, seperti tampilan formentry dan form cetak.
	*Control Class*, atau objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berabagai objek.

	*Message*, simbol mengirim pesan antar class.
	*Recursive*, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	Activation, mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	Lifeline, garis titik-titik yang berhubung dengan objek, sepanjang lifeline terdapat activation.

(Sumber : Urva dan Siregar, 2015 : 95)

4. *Class* Diagram (Diagram Kelas)

Merupakan hubungan antara kelas dan penjelasan detail tiap-tiap kelas didalam model desain dari suatu sistem, juga memperhatikan aturan-aturan dan tanggung jawab entitas yang menentukan perilakukan sistem. *Class* diagram juga menunjukan atribut-atribut dan operasi-operasi dari sebuah kelas dan constraint yang berhubungan dengan objek yang dikoneksikan. *Class* diagram secara khas meliputi : Kelas (*Class*), *Relasi*, *Assoclastions*, *Generalization* dan *Aggregation*, Atribut (*Attrabutes*), Operasi (*Operasiton*/*Method*), *Visibility*, tingkat akses objek

eksternal kepada suatu operasi atau atribut (Urva dan Siregar, 20215 : 95). Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *kardinaliti* yang dapat dilihat pada Tabel II.5. dibawah ini.

Tabel II.5. *Multiplicity Class* Diagram

Multiplicty	**Keterangan**
1	Satu dan hanya satu
0..*	Boleh tidak ada, satu atau lebih
1..*	Satu atau Lebih
0..1	Boleh tidak ada, maksimal satu
n..n	Batasan antara contoh 2.4 mempunyai arti minimal 2 maksimum 4.

(Sumber : Urva dan Siregal, 2015 : 95)