BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Penelitian terkait yang dilakukan oleh Raja Nasrul Fuad, Haikal Nando Winata, dkk (2017) yang berjudul **"*APLIKASI KEAMANAN FILE AUDIO WAV (WAVEFORM) DENGAN TERAPAN ALGORITMA RSA*.** Perangkat lunak (software) dapat melakukan penyandian data audio dengan menerapkan algoritma RSA dan struktur data audio WAV. Ukuran berkas audio WAV menjadi bertambah besar setelah dilakukan enkripsi menggunakan algoritma RSA berdasarkan besar kunci yang digunakan.

Penelitian terkait lain yang dilakukan oleh Iskandar Muda Siregar (2019) dengan judul **"*PENERAPAN ALGORITMA AFFINE CIPHER DAN ALGORITMA COLOUMNAR TRANSPOSITION DALAM KEAMANAN TEKS".*** Dengan adanya proses enkripsi dan dekripsi pada keamanan teks dengan menggunakan algoritma affine cipher dan algoritma coloumnar transposition, teks dapat diamankan dari orang – orang yang tidak berkepentingan. Proses pengaman teks dengan mengkombinasikan algoritma affine cipher dan algoritma coloumnar transposition bertujuan agar mempersulit analisis sandi. Perancangan aplikasi dengan visual basic studio 2008 bertujuan untuk mengimplementasikan hasil enkripsi dan dekripsi untuk keamanan teks.

Penelitian terkait lain yang dilakukan oleh Heri Santoso, M. Fakhriza (2018) dengan judul **"*PERANCANGAN APLIKASI KEAMANAN FILE AUDIO***

***FORMAT WAV (WAVEFORM) MENGGUNAKAN ALGORITMA RSA*".** Perangkat lunak (software) dapat melakukan penyandian data audio dengan menerapkan algoritma RSA dan struktur data audio WAV. Ukuran berkas audio WAV menjadi bertambah besar setelah dilakukan enkripsi menggunakan algoritma RSA berdasarkan besar kunci yang digunakan

Berdasarkan hasil penelitian yang terdahulu, maka dibuatlah kesimpulan untuk merancang sebuah aplikasi untuk menerapkan sebuah algoritma yang belum pernah digunakan sebelumnya untuk melakukan enkripsi dan dekripsi terhadap data *file* MS Word yaitu *Algoritma RSA dan Affine Cipher*. Sehingga pada penulisan skripsi ini dibuatlah sebuah judul **"*Penerapan Algoritma RSA dan Affine Cipher dalam Keamanan File MS Word*"**. Berdasarkan judul tersebut nantinya akan dihasilkan sebuah aplikasi untuk melakukan enkripsi dan dekripsi terhadap data File dokumen kontainer.

II.2. Landasan Teori

II.2.1. Aplikasi

Aplikasi adalah satu unit perangkat lunak yang dibuat untuk membantu meayani kebutuhan seseorang untuk melakukan aktivitas. Dengan adanya aplikasi maka kebutuhan akan pelayanan sebuah aktivitas menjadi lebih baik. Dimana setiap pekerjaan dapat dilakukan dengan mudah kalau menggunakan sebuah aplikasi (Agusdi Syafrizal: 2018).

II.2.2. Kriptografi

Kriptografi adalah suatu ilmu yang mempelajari bagaimana cara menjaga agar data atau pesan tetap aman saat dikirimkan, dari pengirim ke penerima tanpa

mengalami gangguan dari pihak ketiga. *Kriptografi* adalah ilmu pengetahuan dan seni menjaga *message* agar tetap aman. Tujuan penerapan *kriptografi* adalah untuk membuat sesuatu yang tersembunyi, dapat suatu pesan rahasia berupa teks, suara, gambar dan video. Di dalam sistem *kriptografi* terdapat 5 bagian yaitu

1. *Plaintext* adalah pesan atau data dalam bentuk aslinya teks yang dapat terbaca. *Plaintext* adalah masukan bagi algoritma enkripsi.
2. *Secret Key* adalah masukan bagi algoritma enkripsi merupakan nilai yang bebas terhadap teks asli dan menentukan hasil keluaran algoritma enkripsi.
3. *Chipertext* adalah keluaran algoritma enkripsi. *Ciphertext* dapat dianggap sebagai pesan tersembunyi yang akan terlihat acak.
4. *Algoritma Enkripsi* memiliki 2 masukan teks asli dan kunci rahasia. *Algoritma enkripsi* melakukan transpormasi terhadap teks asli sehingga menghasilkan teks sandi.
5. *Algoritma Dekripsi* memiliki 2 masukan yaitu teks sandi dan kunci rahasia. *Algoritma dekripsi* memulihkan kembali teks sandi menjadi teks asli bila kunci rahasia *algoritma enkripsi* sama dengan *algoritma dekripsi* (Guntur Tri Wibowo, dkk: 2015).

II.2.3. Metode Algoritma RSA (Rivest Shamir Adleman)

RSA merupakan salah satu dari Public Key Cryptosystem yang sangat sering digunakan untuk memberikan kerahasiaan terhadap keaslian suatu data digital. Keamanan enkripsi dan dekripsi data model ini terletak pada kesulitan untuk memfaktorkan modulus n yang sangat besar. Dalam kriptografi, RSA adalah algoritma untuk enkripsi kunci publik. Algoritma ini adalah algoritma pertama

yang diketahui paling cocok untuk menandai (signing) dan untuk enkripsi dan salah satu penemuan besar pertaman dalam kriptografi kunci publik. RSA masih digunakan secara luas dalam protokol-protokol perdagangan elektronik dan dipercayai sangat aman karena diberikan kunci-kunci yang cukup panjang dan penerapan-penerapannya yang sangat mutakhir.

Algoritma pembentukan kunci :

1. Tentukan p dan q bernilai dua bilangan prima besar, acak dan dirahasiakan, $p \neq q$, p dan q memiliki ukuran yang sama.
2. Hitung n = p x q, dan hitung $\phi(n) = (p - 1) \times (q - 1)$, bilangan integer n disebut (RSA) modulus.
3. Tentukan e bilangan prima acak yang memiliki syarat : $1 < e < 1(n)$, GCD(e, i(n)) = 1, disebut e relatif prima terhadap i(n), bilangan integer n disebut (RSA) enciphering component, sehingga menghasilkan Dd (Ee(m)) = Ee(Dd(c)) ≡ md mod n. (Indra Gunawan, 2018;125).

II.2.4. Affine cipher

Affine cipher adalah perluasan dari caesar cipher, yang menggalikan plainteks dengan sebuah nilai dan menambahkannya dengan sebuah pergeseran. Secara matematis enkripsi plainteks P menghasilkan cipherteks C dinyatakan dengan fungsi kongruen

C=((a x P) +k)mod 26.....(1)

Dimana 26 adalah jumlah alphabet, persamaan 1 digunakan pada proses enkripsi. Proses dekripsi menggunakan persamaan 2 di bawah ini :

P=a-1(Ci-k) mod 26.....(2)

a adalah bilangan bulat yang harus relatif prima dengan26. Dengan kata lain great common divisior gcd(a,26) harus sama dengan 1. (Rinaldi Munir, 2016).

a. Enkripsi

Encrypt atau enkripsi merupakan sebuah teknik yang dilakukan mengacak data asli menjadi kode rahasia sehingga menyulitkan orang yang tidak berkepentingan untuk mengakses dan mengetahui data yang asli.

b. Dekripsi

Decryption atau dekripsi adalah kebalikan dari enkripsi, dimana berfungsi untuk mendeskripsikan data yang telah dienkripsi sehingga data yang telah menjadi kode rahasia diubah kembali menjadi data biasa atau aslinya (Arisantoso, dkk; 2017).

Data adalah sesuatu yang belum mempunyai arti bagi penerimanya dan masih memerlukan adanya suatu pengolahan. Data bisa berwujud suatu keadaan, gambar, suara, huruf, angka, matematika, bahasa ataupun simbol-simbol lainnya yang bisa kita gunakan sebagai bahan untuk melihat lingkungan,obyek, kejadian ataupun suatu konsep (Eka Iswandy; 2015).

II.2.5. *WEB*

Web merupakan salah satu sumber daya internet yang berkembang pesat. Pendistribusian informasi web dilakukan melalui pendekatan hyperlink, yang memungkinkan suatu teks, gambar, ataupun objek yang lain menjadi acuan untuk membuka halaman-halaman yang lain. Melalui pendekatan ini, seseorang dapat

memperoleh informasi dengan beranjak dari satu halaman ke halaman lain. (Abdul Kadir, 2005)

II.2.6. *HTML*

HyperText Markup Language (HTML) adalah sebuah bahasamarkupyangdigunakan untuk membuat sebuah halaman web, menampilkan berbagai informasidi dalamsebuah Penjelajah web Internet dan formating hypertext sederhana yangditulis kedalam berkasformat ASCII agar dapat menghasilkan tampilan wujudyang terintegerasi. Dengan kata lain,berkas yang dibuat dalam perangkat lunakpengolah kata dan disimpan kedalam format ASCIInormal sehingga menjadihome page dengan perintah-perintah HTML. (Abdul Kadir, 2005)

II.2.7. *PHP*

PHP adalah bahasa yang dirancang secara khusus untuk penggunaan pada *Web*. PHP adalah *tool* untuk pembuatan halaman web dinamis. Pada awalnya PHP merupakan kependekan dari Personal *Home Page* (Situs Personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama FI (*Form Interpreted*), yang wujudnya berupa sekumpulan *script* yang digunakan untuk mengolah data form dari *web*. Saat ini PHP adalah singkatan dari PHP: *Hypertext Preprocessor*, sebuah kepanjangan rekursif, yakni permainan kata dimana kepanjangannya terdiri dari singkatan itu sendiri: PHP: *Hypertext Preprocessor* (Ahmad Lutfi, 2017 ; 105).

II.2.8. *Database*

Database atau biasa disebut basis data merupakan kumpulan data yang saling berhubungan. Data tersebut biasanya terdapat dalam tabeltabel yang saling berhubungan satu sama lain, dengan menggunakan field/kolom pada tiap tabel yang ada" (Agus Prayitno dan Yulia Safitri, 2015; 2).

II.2.9 *SQL*

SQL adalah bahasa standard untuk melakukan berbagai operasi data pada database, diantaranya mendefinisikan tabel, menampilkan data dengan kriteria tertentu, menambahkan data hingga menghapus data tertentu. Penggunaan SQL pada beberapa bahasa pemrograman secara umum relatif sama. (Agus Prayitno dan Yulia Safitri, 2015; 2).

II.2.10. *MySQL*

MySQL adalah salah satu aplikasi DBMS (*Database Management System*) yang sudah sangat banyak digunakan oleh para pemrogram aplikasi web. Dalam sistem database tak relasional, semua informasi disimpan pada satu bidang luas, yang kadangkala data di dalamnya sangat sulit dan melelahkan untuk diakses. Tetapi *MySQL* merupakan sebuah sistem database relasional, sehingga dapat mengelompokkan informasi ke dalam tabel-tabel atau grup-grup informasi yang berkaitan. Setiap tabel memuat bidang-bidang yang terpisah, yang mempresentasikan setiap bit informasi. MySQL menggunakan indeks untuk mempercepat proses pencarian terhadap baris informasi tertentu. MySQL memerlukan sedikitnya satu indeks pada tiap tabel. Biasanya akan menggunakan suatu *primary key* atau pengenal unik untuk membantu penjejakan data (Ahmad Lutfi, 2017; 106).

II.2.11. *Unified Modeling Language* (UML)

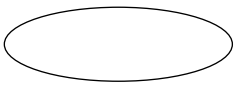
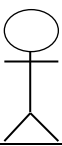
Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). UML adalah satu alat bantu yang sangat handal didunia pengembnagan sistem yang berorientasi objek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam membentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain (Munawar ; 2018 : 49).

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 tipikal interaksi antara (pengguna) sebuah *system*dengan sistemnya sendiri melalui sebuah cerita bagaima sebuah sistem yang dipakai (Munawar ; 2018 : 89).

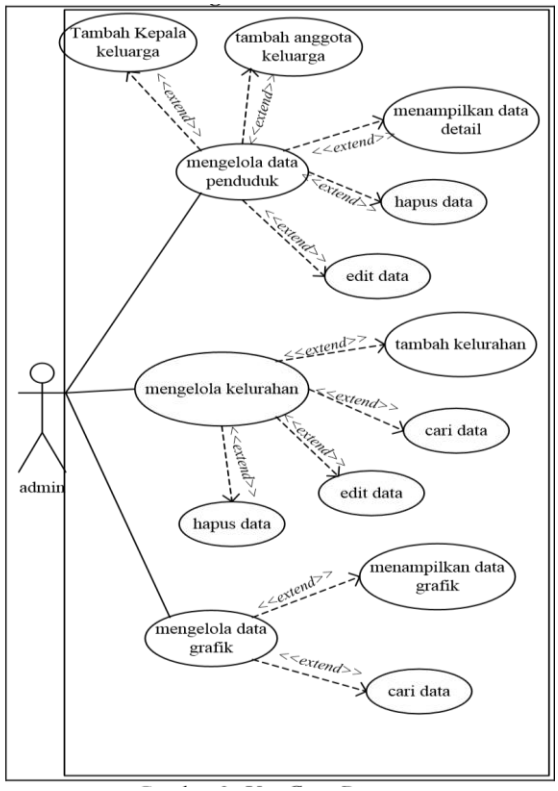
Tabel II.3. Simbol *Use Case Diagram*

Gambar	Keterangan
	*Use Case* menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar 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 mengindikasikan aliran data.
——→	Asosiasi antara aktor dan *Use Case* yang menggunakan panah terbuka untuk mengindikasikan 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 : Munawar ; 2018 : 93)

Contoh kasus;



(*Sumber* :*Winda Aprianti & Umi Maliha 2016:24)*

Use Case diagram yang disajikan pada Gambar mendeskripsiskan interaksi aktor, yaitu admin sistem informasi data penduduk yang dapat mengelola data penduduk,

mengelola kelurahan, dan mengelola data grafik. Pengelolaan data penduduk meliputi tambah kepala keluarga, tambah anggota keluarga, menampilkan, mengedit, dan menghapus data. Sedangkan pengelolaan kelurahan meliputi tambah, cari, *edit*, dan hapus data kelurahan.

2. *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.

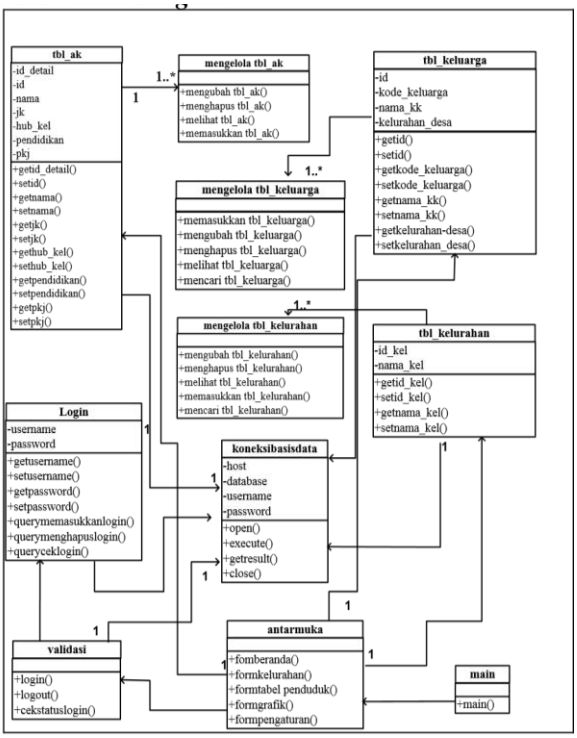
Class diagram merupakan diagram statis dari suatu aplikasi. Class Diagram tidak hanya digunakan untuk memvisualisasikan, menggambarkan, dan mendokumentasikan berbagai aspek sistem tetapi juga untuk membangun kode eksekusi *(executable code)* dari aplikasi perangkat lunak (Munawar ; 2018 : 101).

Tabel II.4. Simbol *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 :* Munawar ; 2018 : 101*)

Contoh kasus:



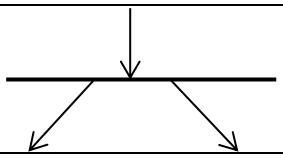
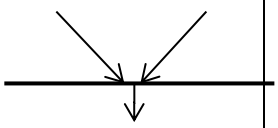
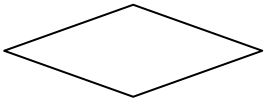
(*Sumber* :*Winda Aprianti & Umi Maliha 2016:24)*

Class diagram disajikan pada Gambar 1 terdiri dari 11 kelas yang meliputi kelas *Main*, Antarmuka, *login*, KoneksiBasisData, *Validasi*, tbl_keluarga, mengelola tbl_keluarga, tbl_kelurahan, mengelola tbl_kelurahan, tbl_ak, dan mengelola tbl_ak

3. Diagram Aktivitas (*Activity Diagram*)

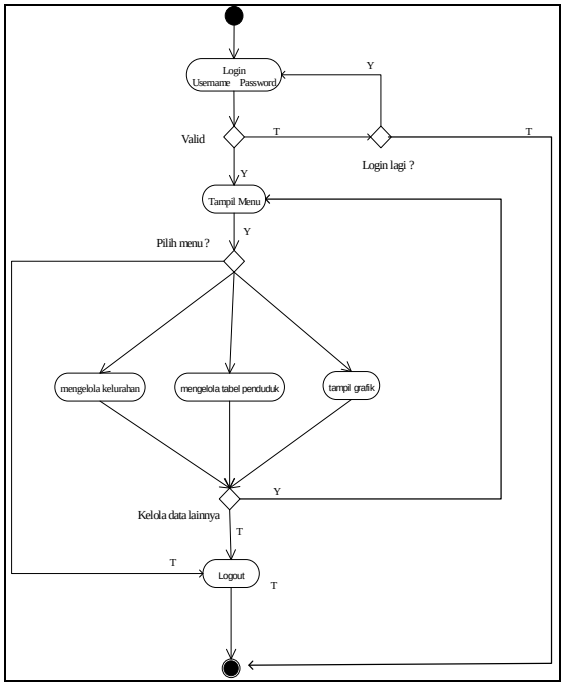
Activity Diagram bagian penting dari UML yang menggambarkan aspek dinamis dari sistem. Logika prosedural, proses bisnis dan aliran kerja suatu bisnis bisa dengan mudah dideskripsikan dalam *activity diagram* (Munawar ; 2018 : 137).

Tabel II.5. Simbol Diagram Aktivitas

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*.
	*Swimlane*, pembagian *activity* diagram untuk menunjukkan siapa melakukan apa.

(*Sumber : Munawar ; 2018 : 137*)

Contoh kasus;



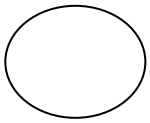
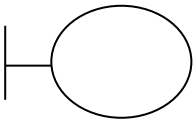
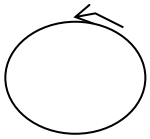
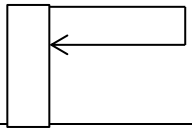
(Sumber :Winda Aprianti & Umi Maliha 2016:25)

Activity diagram yang diilustrasikan pada Gambar merupakan diagram yang menggambarkan *workflow* (aliran kerja) dari Sistem Informasi Kepadatan Penduduk di Setiap Kelurahan atau Desa pada Badan Pemberdayaan Masyarakat dan Pemerintah Desa (BPMPD) Studi Kasus pada Kecamatan Bati-Bati Kabupaten Tanah Laut.

4. Diagram Urutan (*Sequence Diagram*)

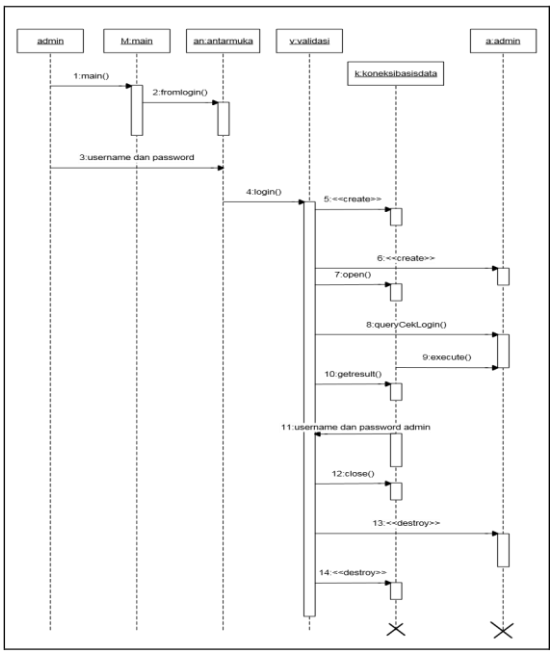
Sequence diagram adalah salah satu *interaction diagram*. Karena sequence diagram mengacu kepada obyek, maka sbelum membuat diagram ini class diagram sudah harus teridentifikasi (Munawar ; 2018 : 186).

Tabel II.6. Simbol Diagram Urutan

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,* 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 : Munawar ; 2018 : 186)

Contoh kasus :



(*Sumber* :*Winda Aprianti & Umi Maliha 2016:25)*

Urutan proses pada *Sequence login* pada Gambar dimulai dari admin sebagai pengguna yang masuk ke antarmuka untuk masuk ke *form login*. Setelah itu *admini* memasukkan *username* dan *password* menuju validasi untuk *login* setelah itu data dikirim ke *database*. (Sumber :Winda Aprianti & Umi Maliha 2016:25)