

## BAB 2

### TINJAUAN PUSTAKA

(Alfarisi, 2017) dalam penelitian yang berjudul “Sistem Prediksi Penjualan Gamis Toko Qitaz Menggunakan Metode Single Exponential Smoothing”. Pada penelitian ini, peneliti menerapkan Metode *single exponential smoothing* untuk mendapatkan laba yang besar pada suatu perusahaan dengan menentukan prediksi penjualan pada bulan berikutnya. Hasil peramalan yang dihasilkan dari Metode *single exponential smoothing* dengan menggunakan nilai  $\alpha = 0.1$  sampai  $0.9$ , maka didapat nilai error terkecil dengan menggunakan 4(empat) metode yang memperoleh hasil  $ME = 4$ ,  $MAD = 9$ ,  $MAPE=7\%$  dan  $MSE= 121$  ,yaitu dari nilai  $\alpha = 0.6$  dan  $\alpha = 0.7$ .

(Muhammad Noor Arridho\*1, 2020) dalam penelitian yang berjudul “Penerapan Metode Single Exponential Smoothing untuk Memprediksi Penjualan Katering pada Kedai Pojok Kedaung”. Pada penelitian ini, peneliti menerapkan Metode *single exponential smoothing* untuk memperoleh informasi prediksi penjualan dan nilai kesalahan serta persentase kesalahan menggunakan metode MAD dan MAPE. Berdasarkan penelitian yang telah dilakukan dan diuraikan, serta mengacu pada beberapa teori maka dapat ditarik kesimpulan bahwa dari jumlah data sebanyak 11 periode, dilakukan pengujian menggunakan metode MAD (*Mean Absolute Deviation*) dan MAPE

(*Mean Absolute Percentage Error*) dan didapatkan nilai akurasi kesalahan sebesar 26,77% dengan menggunakan bobot nilai  $\alpha$  (alpha) 0,3. Angka ini bisa berubah menjadi lebih baik atau buruk tingkat akurasinya dengan melakukan penambahan maupun pengurangan baik jumlah serta banyak data penjualan. Hasil suatu prediksi bukanlah suatu nilai yang pasti akan terjadi di periode yang akan datang. Mengingat banyak faktor-faktor di lapangan yang terkadang memberikan pengaruh pada hasil akhirnya. Seperti pengaruh perubahan harga jual yang banyak pada waktu tertentu.

(Ni Luh Wiwik Sri Rahayu Ginantra, 2019) dalam penelitian yang berjudul “Penerapan Metode Single Exponential Smoothing Dalam Peramalan Penjualan Barang”. Pada penelitian ini, peneliti menerapkan Metode *single exponential smoothing* untuk mengontrol jual beli barang yang keluar masuk agar memberikan kemudahan bagi pihak management dalam mengelola data stock, kontrol keuangan dan penghitungan laba yang akan langsung dapat diketahui oleh pemangku kepentingan. Dari hasil penelitian tersebut di dapatkan Hasil perhitungan dengan menggunakan rumus MAPE (Mean Absolut Percentage Error ) untuk tingkat akurasi dengan nilai alpha 0,1 adalah 2,62%.

(Dewi Rosa Indah, 2018) dalam penelitian yang berjudul “Sistem Forecasting Perencanaan Produksi dengan Metode Single Eksponensial Smoothing pada Keripik Singkong Srikandi Di Kota Langsa”. Pada penelitian ini, peneliti menerapkan Metode *single exponential smoothing* untuk melakukan peramalan penjualan keripik singkong agar dapat meningkatkan keuntungan dan menghindari terjadinya kelebihan maupun kekurangan dalam memproduksi keripik singkong.

Dari hasil penelitian tersebut di dapat disimpulkan bahwa peramalan penjualan keripik Srikandi akan mengalami kenaikan pada tahun 2016 yaitu peramalan penjualan keripik sebesar 32.533 dengan menggunakan alpha 0,2 dan alpha 0,4 sebesar 30.587.

(Handoko, 2019) dalam penelitian yang berjudul “Prediksi Jumlah Penerimaan Mahasiswa Baru Dengan Metode Single Exponential Smoothing (Studi Kasus: Amik Royal Kisaran)”. Pada penelitian ini, peneliti menerapkan Metode *single exponential smoothing* untuk menilai prakiraan keadaan dimasa. di AMIK Royal Kisaran, ketika akan membuat jadwal kuliah sering terhambat karena tidak adanya perkiraan jumlah mahasiswa. Hasil ramalan yang diperoleh Pada Tahun Akademik 2018/2019 prediksi jumlah mahasiswa yang diterima untuk Program Studi Teknik Komputer sebanyak 30 Orang (Memiliki tingkat kesalahan yang terkecil  $MSE=1110,77$  dengan nilai  $\alpha=0,9$ ) dan Program Studi Manajemen Informatika sebanyak 89 Orang (Memiliki tingkat kesalahan yang terkecil  $MSE=7725,33$  dengan nilai  $\alpha=0,9$ )

(Agustinus Budi Santoso1, 2021) dalam penelitian yang berjudul “Penerapan Metode Single Exponential Smoothing Untuk Analisa Peramalan Penjualan”. Pada penelitian ini, peneliti menerapkan Metode *single exponential smoothing* untuk menangani dan mengontrol persediaan stok obat agar lebih optimal. Dari hasil penelitian tersebut di dapatkan Hasil Pengujian pada aplikasi yang menerapkan Metode *exponential smoothing* dengan pengujian beberapa faktor

penggunaan Alpha berbeda. Ditemukan bahwa penggunaan Alpha dengan 0,3 mendekati nilai kenyataan pada penjualan.

## 2.1 Forecasting (Peramalan)

Peramalan merupakan suatu bidang ilmu yang dijadikan alat untuk melakukan prediksi sesuatu dengan dasar data yang ada sebelumnya, dan diolah dengan cara tertentu. Peramalan pun dapat di implementasikan ke berbagai bidang ilmu termasuk bisnis dan industri, pemerintah, ekonomi, ilmu lingkungan, kedokteran, ilmu sosial, politik, dan keuangan (Maricar, 2019).

## 2.2 Single Exponential Smoothing

Metode *single exponential smoothing* merupakan pengembangan dari metode *single moving averages* dimana metode peramalan ini dilakukan dengan mengulang perhitungan secara terus menerus dengan menggunakan data terbaru dan setiap data diberi bobot. Metode *single exponential smoothing* mempertimbangkan bobot data sebelumnya dengan memberikan bobot pada setiap data periode untuk membedakan prioritas atas suatu data. Metode *single exponential smoothing* merupakan metode yang digunakan pada peramalan jangka pendek yang biasanya hanya 1 bulan ke depan yang mengasumsikan bahwa data berfluktuasi di sekitar nilai *mean* yang tetap tanpa *trend* atau pola pertumbuhan konsisten (Ni Luh Wiwik Sri Rahayu Ginantra, 2019).

Metode *single exponential smoothing* akan diterapkan pada perhitungan peramalan jumlah peserta didik baru pada SMK Muhammadiyah 6 Donomulyo. Menurut jurnal dari (Khairun Nisa Ulfa1, 2016). Perhitungan dapat dituliskan dalam rumus (1) :

$$F_t = F_{t-1} + \alpha(A_{t-1} - F_{t-1}) \dots\dots\dots(1)$$

dimana :  $F_t$  = Peramalan baru

$F_{t-1}$  = Peramalan sebelumnya

$\alpha$  = Konstanta penghalusan ( $0 \leq \alpha \leq 1$ )

$A_{t-1}$  = Permintaan aktual periode lalu

### 2.3 Ukuran Akurasi Peramalan

#### a) Mean Absolute Deviation (MAD)

*Mean Absolute Deviation (MAD)* merupakan perhitungan yang digunakan untuk menghitung rata-rata kesalahan mutlak (Maricar, 2019).

Menurut jurnal dari (Ni Luh Wiwik Sri Rahayu Ginantra, 2019), *Mean Absolute Deviation (MAD)* merupakan metode untuk mengevaluasi metode peramalan menggunakan jumlah dari kesalahan-kesalahan yang *absolute*. *Mean Absolute Deviation (MAD)* mengukur ketepatan ramalan dengan merata-rata kesalahan dugaan (nilai *absolute* masing-masing kesalahan). MAD berguna ketika mengukur kesalahan ramalan dalam unit yang sama sebagai deret asli.

Nilai MAD dapat dihitung dengan menggunakan rumus(2) :

$$MAD = \frac{\sum |X_t - F_t|}{n} \dots\dots\dots(2)$$

Keterangan :  $X_t$  = Nilai Riil pada periode -t.

$F_t$  = Nilai Peramalan (*Forecast*) pada periode -t.

n = Jumlah periode peramalan yang terlibat.

#### b) Mean Square Error (MSE)

*Mean Square Error (MSE)* merupakan perhitungan yang digunakan untuk menghitung rata-rata kesalahan berpangkat (Maricar, 2019).

Menurut jurnal dari (Ni Luh Wiwik Sri Rahayu Ginantra, 2019), *Mean Squared Error* (MSE), metode lain untuk mengevaluasi metode peramalan. Masing-masing kesalahan atau sisa dikuadratkan. Kemudian dijumlahkan dan ditambahkan dengan dengan jumlah observasi. Pendekatan ini mengatur kesalahan peramalan yang besar karena kesalahan-kesalahan itu dikuadratkan. Metode itu menghasilkan kesalahan-kesalahan sedang yang kemungkinan lebih baik untuk kesalahan kecil, tetapi kadang menghasilkan perbedaan yang besar.

Nilai MSE dapat di hitung dengan menggunakan rumus (3) :

$$MSE = \frac{\sum (X_t - F_t)^2}{n} \dots \dots \dots (3)$$

Keterangan :  $X_t$  = Nilai Riil pada periode -t.

$F_t$  = Nilai Peramalan (Forecast) pada periode -t.

n = Jumlah periode peramalan yang terlibat.

#### c) *Mean Absolute Percentage Error* (MAPE)

*Mean Absolute Percentage Error* (MAPE) merupakan perhitungan yang digunakan untuk menghitung rata-rata persentase kesalahan mutlak (Maricar, 2019)

Menurut jurnal dari (Ni Luh Wiwik Sri Rahayu Ginantra, 2019), *Mean Absolute Percentage Error* (MAPE) dihitung dengan menggunakan kesalahan absolute pada tiap periode dibagi dengan nilai observasi yang nyata untuk periode itu. Kemudian merata-rata kesalahan persentase absolute tersebut. Pendekatan ini berguna ketika ukuran atau besar variable ramalan itu penting dalam mengevaluasi ketepatan ramalan. MAPE mengindikasikan seberapa besar kesalahan dalam meramal yang dibandingkan dengan nilai nyata.

Nilai MPE dapat dihitung dengan rumus MAPE (4)

$$PE_T = 100\% \times \left( \frac{(X_t - F_t)}{X_t} \right)$$

$$MAPE = \frac{\sum |PE_T|}{N} \dots\dots\dots (4)$$

Keterangan :  $X_t$  = Nilai Riil pada periode -t.

$F_t$  = Nilai Peramalan (Forecast) pada periode -t.

n = Jumlah periode peramalan yang terlibat.

## 2.4 Alat Bantu Dalam Perancangan

### 2.4.1 Unified Modeling Language (UML)

*UML* adalah bahasa pemodelan standar yang memiliki sintak dan semantik. (Isnayati, 2017).

UML merupakan suatu pemodelan atau sebuah bahasa standar untuk merancang, mendokumentasikan, dan membuat model semua jenis sistem perangkat lunak khususnya sistem berorientasi objek. Aplikasi ini dapat berjalan pada semua perangkat keras, semua sistem operasi dan jaringan, serta dapat ditulis dalam semua bahasa pemrograman. UML menggunakan *class* dan *operation* sehingga lebih tepat digunakan untuk penulisan perangkat lunak dalam bahasa berorientasi objek, seperti C++, Java, C#, atau VB.NET. Notasi UML menggambarkan berbagai diagram peranti lunak dan diturunkan dari tiga notasi, yaitu OOD, OMT, dan OOSE (Munif, 2020).

### 2.4.2 Diagram Unified Modeling Language (UML)

Menurut Rosa A.S dan M. Shalahudin, 2018: 137 dalam (Daniel Dido Jantce TJ Sitingjak, 2020), “UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah system dengan menggunakan diagram dan teks-teks pendukung. UML hanya sebagai pemodelan”. UML terdiri dari 13 diagram yang

dikelompokkan dalam 3 kategori. Berikut ini penjelasan singkat dari pembagian kategori tersebut :

1. *Structure diagram*, yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan. *Structure diagram* terdiri dari *class diagram*, *object diagram*, *component diagram*, *composite structure diagram*, *package diagram* dan *deployment diagram*.
2. *Behavior diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem. Behavior diagram terdiri dari *Use case diagram*, *Activity diagram*, *State Machine System*.
3. *Interaction diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem. *Interaction diagram* terdiri dari *Sequence Diagram*, *Communication Diagram*, *Timing Diagram*, *Interaction Overview Diagram*.

### **2.4.3 Use Case Diagram**

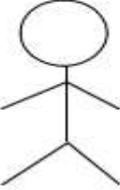
Use Case Diagram merupakan gambaran fungsional dari sistem yang dapat menunjukkan fitur-fitur yang disediakan oleh sistem serta aktor yang dapat mengakses fitur tersebut (Anindya Palmitraazzah\*, 2017).

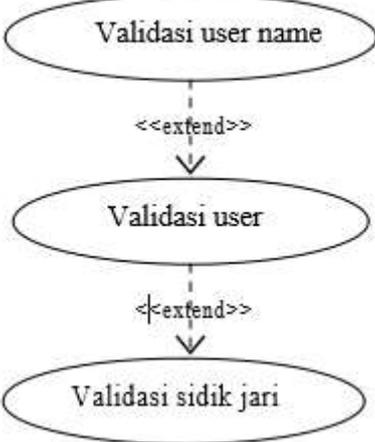
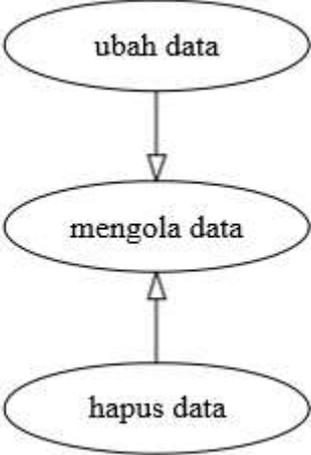
Rosa A.S dan M. Shalahudin, 2018: 137 dalam (Julianto Simatupang1, 2019), Diagram use case merupakan pemodelan untuk kelakuan system informasi yang akan dibangun. Use case mendeskripsikan sebuah interaksi anatra satu atau

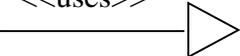
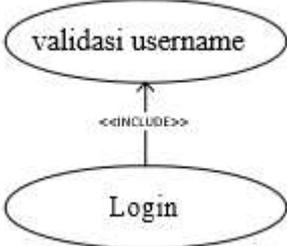
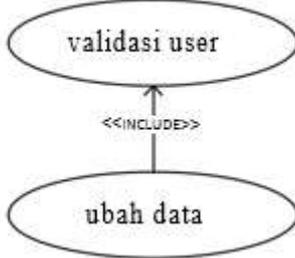
lebih actor dengan sistem informasi yang akan dibangun. Use case digunakan untuk mengetahui fungsi apa saja yang ada pada sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Berikut ini adalah simbol-simbol diagram use case, seperti yang terlihat pada tabel 2.1 dibawah ini :

Tabel 2. 1 Simbol – Simbol Pada Use Case

Sumber Rosa A.S dan M. Shalahudin, 2018: 137 dalam (Julianto Simatupang1, 2019).

No	Simbol	Deskripsi
1.	Use Case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama use case.
2.	Aktor / <i>Actor</i> 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
3.	Assosiasi/ <i>association</i> 	Komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor.
4.	Ekstensi / <i>extend</i> 	Relasi use case tambahan kesebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu, mirip dengan prinsip inheritance pada pemrograman berorientasi objek, biasanya use case tambahan memiliki nama depan yang sama dengan use case yang ditambahkan, misalkan

No	Simbol	Deskripsi
		 <p>Arah panah mengarah pada use case yang ditambahkan, biasanya use case yang menjadi extend-nya merupakan jenis yang sama dengan use case yang menjadi induknya.</p>
5.	<p>Generalisasi/ <i>Generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya :</p>  <p>arah panah mengarah pada use case yang menjadi generalisasinya (umum)</p>

No	Simbol	Deskripsi
6.	Menggunakan/ <i>include/uses</i> <<include>>   <<uses>> 	<p>Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini</p> <p>Ada dua sudut pandang yang cukup besar mengenai include di use case :</p> <p>Include berarti use case yang ditambahkan akan selalu di panggil saat use case tambahan dijalankan, misal pada kasus berikut :</p> <div style="text-align: center;">  <pre> graph BT     Login((Login)) -- "&lt;&lt;INCLUDE&gt;&gt;" --&gt; validasi_username((validasi username))           </pre> </div> <p>Include berarti use case yang tambahan akan selalu melakukan pengecekan apakah use case yang di tambahkan telah dijalankan sebelum use case tambahan di jalankan, misal pada kasus berikut :</p> <p>Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p> <div style="text-align: center;">  <pre> graph BT     ubah_data((ubah data)) -- "&lt;&lt;INCLUDE&gt;&gt;" --&gt; validasi_user((validasi user))           </pre> </div>

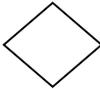
#### 2.4.4 Activity Diagram

Rosa A.S dan M. Shalahudin, 2018: 137 dalam (Julianto Simatupang1, 2019), Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran

kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Penekanan pada diagram aktivitas adalah menggambarkan aktivitas sistem atau aktivitas yang dapat dilakukan oleh sistem, bukan apa yang dilakukan aktor. Berikut adalah simbol-simbol activity diagram, seperti terlihat pada tabel 2.2 berikut ini:

Tabel 2. 2 Simbol Activity Diagram

Sumber Rosa A.S dan M. Shalahudin, 2018: 137 dalam (Julianto Simatupang1, 2019).

1.	Status Awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.	Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.	Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.	Status Akhir 	Status akhir yang dilakukan oleh sistem, sebuah diagram aktivitas sebuah status akhir.
6.	Swimlane	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

		
--	---	--

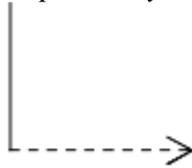
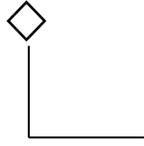
#### 2.4.5 Class Diagram

Rosa A.S dan M. Shalahudin, 2018: 137 dalam (Julianto Simatupang1, 2019), *Class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Berikut adalah simbol - simbol *class diagram*, seperti terlihat pada tabel 2.3 berikut ini:

Tabel 2. 3 Class Diagram

Sumber Rosa A.S dan M. Shalahudin, 2018: 137 dalam (Julianto Simatupang1, 2019).

No	Simbol	Deskripsi
1	Kelas / <i>Class</i> 	Kelas pada struktur sistem
2	Antarmuka / <i>Interface</i> 	Sama dengan konsep interface dalam pemrograman berorientasi objek
3	Asosiasi / <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity
4	Asosiasi Berarah / <i>Directed Association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity

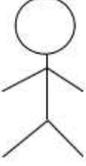
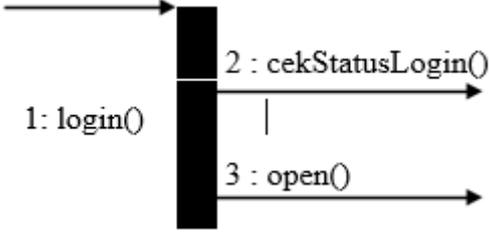
5	Generalisasi <i>Generalization</i> 	/ Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus)
6	Kebergantungan <i>Dependency</i> 	/ Relasi antar kelas dengan makna kebergantungan antar kelas
7	Agregasi / <i>aggregation</i> 	Relasi antar kelas dengan makna semua-bagian (whole-part)

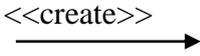
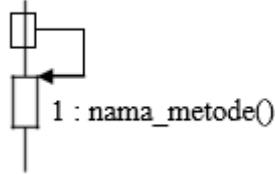
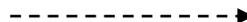
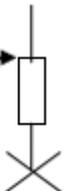
#### 2.4.6 Sequence Diagram

Rosa A.S dan M. Shalahudin, 2018: 137 dalam (Julianto Simatupang1, 2019), “Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada use case“. Berikut adalah simbol-simbol activity diagram, seperti terlihat pada tabel 2.4 berikut ini:

Tabel 2. 4 Simbol Sequence Diagram

Sumber Rosa A.S dan M. Shalahudin, 2018: 137 dalam (Julianto Simatupang1, 2019).

No	Simbol	Deskripsi
1	<p data-bbox="395 432 475 461">Aktor</p>  <p data-bbox="395 696 453 725">atau</p> <div data-bbox="475 721 692 813" style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;"> <p data-bbox="491 734 641 763">Nama Aktor</p> </div> <p data-bbox="464 898 625 909">_____</p> <p data-bbox="395 969 624 999">tanpa waktu aktif</p>	<p data-bbox="754 432 1359 712">Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan dalam menggunakan kata benda diawal frase nama aktor.</p>
2	<p data-bbox="395 1048 651 1077">Garis hidup/<i>lifeline</i></p> 	<p data-bbox="754 1048 1209 1077">Menyatakan kehidupan suatu objek</p>
3.	<p data-bbox="395 1238 480 1267">Objek</p> <div data-bbox="402 1272 711 1402" style="border: 1px solid black; padding: 5px; width: fit-content;"> <p data-bbox="418 1285 663 1357">Nama Objek : Nama Kelas</p> <p data-bbox="418 1368 695 1379">_____</p> </div>	<p data-bbox="754 1238 1305 1267">Menyatakan objek yang berinteraksi pesan</p>
4.	<p data-bbox="395 1462 563 1491">Waktu Aktif</p> 	<p data-bbox="754 1462 1359 1603">Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya</p> 

		Maka cekStatusLogin() dan open() dilakukan didalam metode login(). tidak memiliki waktu aktif
5.	Pesan tipe create  	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
6.	Pesan Tipe Call  1: nama_metode()  	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,    Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi
7.	Pesan tipe <i>send</i>  masukan  	Menyatakan bahwa suatu objek mengirimkan data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
8.	Pesan tipe <i>return</i>  keluaran  	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
9.	Pesan tipe <i>destroy</i>    	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada create maka ada destroy

