



## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Teori Umum

##### 2.1.1 Sistem

Sistem dapat didefinisikan sebagai sekumpulan prosedur yang saling berkaitan dan saling terhubung untuk melakukan suatu tugas bersama-sama (Pratama, 2014:7).

Sistem adalah kumpulan dari obyek-obyek seperti orang, *resources*, konsep dan prosedur yang ditujukan untuk melakukan fungsi tertentu atau memenuhi suatu tujuan. Kemudian sistem juga merupakan kumpulan dari komponen yang berinteraksi bersama-sama secara kolektif untuk melaksanakan tugas (Pratiwi, 2016:4).

Berdasarkan definisi di atas, maka dapat disimpulkan bahwa pengertian sistem adalah sekumpulan komponen atau elemen yang dapat diartikan sebagai suatu kumpulan atau obyek-obyek yang terhubung satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan.

#### 2.2 Teori Judul

##### 2.2.1 Sistem Pendukung Keputusan

Menurut Latif, dkk (2018:2) “Mendefinisikan Sistem Pendukung Keputusan sebagai suatu informasi berbasis komputer yang menghasilkan berbagai alternatif keputusan untuk membantu dalam pengambilan keputusan. Sistem Pendukung Keputusan (SPK) menggunakan CBIS (*Computer Based Informmation System*) yang fleksibel, interaktif, dan dapat diadaptasi, yang dikembangkan untuk mendukung solusi atas masalah manajemen spesifikasi yang tidak terstruktur.

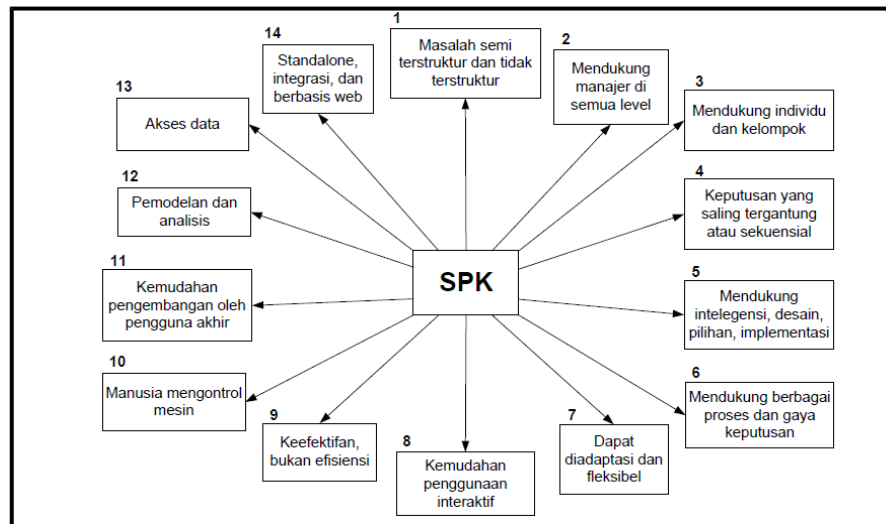
Berdasarkan definisi di atas, maka dapat disimpulkan Sistem Pendukung Keputusan (SPK) merupakan sistem yang berbasis komputer yang setiap



prosesnya membantu meme keputusan dengan memanfaatkan data yang sudah di bobotkan untuk pemecahan masalah dengan kondisi yang tak terstruktur.

### 2.2.1.1 Karakteristik dan Kemampuan Sistem Pendukung Keputusan

Pada Gambar ditunjukkan karakteristik dan kemampuan sebuah sistem pendukung keputusan.



**Gambar 2.1** Karakteristik dan Kapabilitas SPK

Sumber: (Pratiwi, 2016:7)

Karakteristik dan kemampuan sebuah sistem pendukung keputusan sebagai berikut:

1. Sistem Pendukung Keputusan menyediakan dukungan untuk pengambilan keputusan pada keadaan-keadaan semi terstruktur dan tidak terstruktur dengan menggabungkan penilai manusia dan informasi komputerisasi.
2. Menyediakan dukungan untuk tingkat manajerial mulai dari eksekutif sampai manajer.
3. Menyediakan dukungan untuk kelompok individu, problem-problem yang kurang terstruktur memerlukan keterlibatan beberapa individu dari departemen-departemen yang lain dalam organisasi.
4. Sistem pendukung keputusan menyediakan dukungan kepada independen atau keputusan yang berlanjut.



5. Sistem pendukung keputusan memberikan dukungan kepada semua fase dalam proses pembuatan keputusan *intelligence*, *design*, *choice* dan *implementation*.
6. Sistem pendukung keputusan mendukung banyak proses dan gaya pengambilan keputusan.
7. Sistem pendukung keputusan bersifat *adaptive* terhadap waktu, sehingga pembuat keputusan harus reaktif dan bisa menghadapi perubahan-perubahan kondisi secara cepat dan merubah sistem pendukung keputusan menjadi fleksibel sehingga pengguna dapat menambah, menghapus, mengkombinasikan, merubah dan mengatur kembali terhadap elemen-elemen dasar.
8. Sistem pendukung keputusan mudah digunakan. Pengguna merasa nyaman, seperti *user friendly*, *fleksibel*. Kemampuan penggunaan grafik yang tinggi dan bahasa yang mudah dipahami untuk berinteraksi dengan mesin akan menaikkan efektifitas dari sistem pendukung keputusan.
9. Sistem pendukung keputusan menaikkan efektifitas pembuatan keputusan baik dalam hal ketetapan waktu dan kualitas bukan pada biaya pembuatan keputusan atau biaya penggunaan waktu computer.
10. Pembuat keputusan dapat mengontrol tahapan-tahapan pembuatan keputusan seperti pada tahap *intelligence*, *choice* dan *implementation* kemudian sistem pendukung keputusan diarahkan untuk mendukung pembuat keputusan bukan menggantikan posisinya.
11. Memungkinkan pengguna akhir dapat membangun sistem sendiri yang sederhana. Sistem yang besar dapat dibangun dengan bantuan dari spesialis sistem informasi.
12. Biasanya model-model digunakan untuk menganalisis situasi pengambilan keputusan. Kapabilitas pemodelan memungkinkan eksperimen dengan berbagai strategi yang berbeda dibawah konfigurasi yang berbeda.
13. Akses disediakan untuk berbagai sumber data, format, dan tipe, mulai dari sistem informasi geografis (GIS) sampai sistem berorientasi-objek.



14. Dapat dilakukan sebagai alat standalone yang digunakan oleh seorang pengambil keputusan pada satu lokasi dan di beberapa organisasi sepanjang rantai persediaan. Dapat diintegrasikan dengan DSS lain dan atau aplikasi lain, dan dapat didistribusikan secara internal dan eksternal dengan menggunakan networking dan teknologi Web.

### **2.2.2 Website**

*Website* merupakan kumpulan halaman yang menampilkan informasi data teks, data gambar diam atau gerak, data animasi, video dan gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana dihubungkan dengan jaringan-jaringan halaman (*hyperlink*) (Riyadi, Retnandi, & Deddy, 2013:3).

*Website* (Situs Web) merupakan kumpulan dari halaman-halaman web yang berhubungan dengan file-file lain yang terkait. Dalam sebuah website terdapat suatu halaman yang dikenal dengan sebutan home page. Home page adalah sebuah halaman yang pertama kali dilihat ketika seseorang mengunjungi website (Hendrianto, 2014:59).

Berdasarkan pengertian di atas, maka dapat disimpulkan *website* merupakan kumpulan halaman-halaman dan semua file yang terkait membentuk satu rangkaian yang saling berkaitan dimana dihubungkan dengan jaringan halaman.

### **2.2.3 Gizi**

Gizi merupakan rangkaian proses secara organik makanan yang dicerna oleh tubuh untuk memenuhi kebutuhan pertumbuhan dan fungsi normal organ, serta mempertahankan kehidupan seseorang (Mardalena, 2017:1).

Gizi (*nutriture/nutrition*) merupakan keseimbangan antara zat gizi yang masuk ke dalam tubuh dari makanan dengan zat gizi yang dibutuhkan oleh tubuh untuk keperluan proses metabolisme (Paramashanti, 2019:1).

Dari definisi di atas, maka dapat disimpulkan gizi adalah elemen yang dibutuhkan tubuh dalam masa tumbuh kembang terutama pada balita.



### 2.2.4 Metode *Weighted Product* (WP)

Menurut Latif, Jamil, dkk (2018:26), “Metode *Weight Product* (WP) merupakan salah satu metode yang sederhana dengan perkalian untuk menghubungkan rating atribut dimana setiap atribut harus dipangkatkan dengan bobot atribut yang bersangkutan. Hal tersebut dinamakan normalisasi”.

Dari definisi di atas, maka dapat disimpulkan Metode *Weight Product* (WP) merupakan metode di mana setiap atribut harus dinormalisasikan.

Adapun algoritma penyelesaian dari metode *Weight Product* (WP) yaitu sebagai berikut: (Latif, Jamil, dkk, 2018:26)

1. Mendefinisikan terlebih dahulu kriteria-kriteria yang akan dijadikan sebagai tolak ukur penyelesaian masalah.
2. Menormalisasi setiap nilai alternatif (nilai vektor).

$$S_i = \prod_{j=1}^n X_{ij}^{w_j}$$

Sedangkan  $\sum w_j = 1$  serta  $w_j$  adalah pangkat bernilai positif untuk atribut keuntungan, dan bernilai negatif untuk atribut biaya.

Keterangan :

S : Preferensi alternative dianalogikan sebagai vector S

X : Nilai Kriteria

W : Bobot Kriteria atau sub kriteria

i : Alternatif ( dimana  $i=1,2,\dots,n$ )

j : Kriteria

n : Banyaknya Kriteria

3. Menghitung nilai bobot preferensi pada setiap alternatif.

Prefensi relatif dari setiap alternatif, diberikan sebagai berikut :

(Nofriansyah, 2014:48)

$$V_i = \frac{\prod_{j=1}^n X_{ij}^{w_j}}{\prod_{j=1}^n (X_{j*})^{w_j}};$$



Keterangan :

V : Preferensi alternative dianalogikan sebagai vector V

X : Nilai Kriteria

W : Bobot Kriteria atau sub criteria

i : Alternatif

j : Kriteria

N? : Banyaknya kriteria

\* : Banyaknya kriteria yang telah dinilai pada vector S

4. Melakukan perangkaian.

### 2.2.5 Balai Pengobatan

Menurut Peraturan Menteri Kesehatan Republik Indonesia Nomor 920/Men.Kes/Per/XII86 Tentang Upaya Pelaksanaan Kesehatan Swasta Di Bidang Medik, Balai Pengobatan adalah tempat untuk memberikan pelayanan medik secara rawat jalan.

### 2.2.6 Pengertian Sistem Pendukung Keputusan Berbasis *Website* untuk Menentukan Status Gizi Balita Menggunakan Metode *Weighted Product (WP)* pada Balai Pegobatan Talang Kelapa Palembang

Sistem Pendukung Keputusan penentuan gizi balita berbasis *website* merupakan aplikasi yang digunakan untuk menentukan status gizi pada balita dengan menggunakan aplikasi sistem pendukung keputusan dengan metode perhitungan atau algoritma *weighted product*, mengelola data pasien balita dan dokter oleh admin pada Balai Pengobatan Talang Kelapa Palembang.

## 2.3 Teori Khusus

### 2.3.1 Unified Modeling Language

#### 2.3.1.1 Pengenalan *Unified Modeling Language (UML)*

*Unified Modeling Language (UML)* adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*,



membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.

UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek (Sukamto dan Shalahuddin, 2018:133).

### 2.3.1.2 Sejarah UML

Bahasa pemrograman berorientasi objek yang pertama dikembangkan dikenal dengan nama Simula-67 yang dikembangkan pada tahun 1967. Perkembangan aktif dari pemrograman berorientasi objek mulai menggeliat ketika berkembangnya bahasa pemrograman Smalltalk pada awal 1980-an yang kemudian diikuti dengan perkembangan bahasa pemrograman berorientasi objek yang lainnya seperti C objek, C++, Eiffel, dan CLOS.

Sekitar lima tahun setelah Smalltalk berkembang, maka berkembang pula metode pengembangan berorientasi objek. Karena banyaknya metodologi- metodologi yang berkembang pesat saat itu, maka muncullah ide untuk membuat sebuah bahasa yang dapat dimengerti semua orang. Maka dibuat bahasa yang merupakan gabungan dari beberapa konsep, seperti konsep *Object Modeling Technique* (OMT) dari Rumbaugh dan Booch (1991), konsep *The Classes, Responsibilities, Collaborators* (CRC) dari Rebecca Wirfs-Brock (1990), konsep pemikiran Ivar Jacobson, dan beberapa konsep lainnya dimana James R. Rumbaugh, Grady Booch, dan Ivar Jacobson bergabung dalam sebuah perusahaan yang bernama Rational Software Corporation menghasilkan bahasa yang disebut dengan *Unified Modeling Language* (UML).

Pada tahun 1996, *Object Management Group* (OMG) mengajukan proposal agar adanya standarisasi pemodelan berorientasi objek dan pada bulan September 1997 UML diakomodasi oleh OMG sehingga sampai saat ini



UML telah memberikan kontribusinya yang cukup besar di dalam metodologi berorientasi objek dan hal-hal yang terkait di dalamnya (Sukamto dan Shalahuddin, 2014:138).

### 2.3.1.3 Diagram UML

Menurut Sukamto dan Shalahuddin (2018:140), UML terdiri dari 13 macam 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*.

### 2.3.1.4 Use Case Diagram

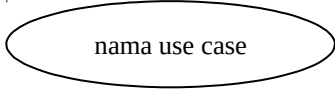
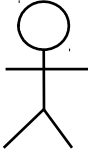

Menurut Sukamto dan Shalahuddin (2018:155), *Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.





Menurut Sukamto dan Shalahuddin (2018:156-158) simbol-simbol yang ada pada diagram *use case* yaitu:

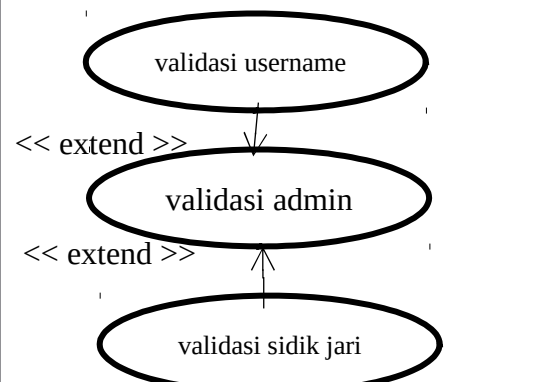
**Tabel 2.1** Simbol-simbol *Use Case* Diagram

No.	Simbol	Deskripsi
1.	<p><i>Uses case</i></p>  <p>nama use case</p>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau actor, biasanya dinyatakan dengan menggunakan kata kerja di awal di awal frase nama <i>use case</i> .
2.	<p><i>Actor / actor</i></p>  <p>nama aktor</p>	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 actor adalah gambar orang, tapi actor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
3.	<p><i>Asosiasi / association</i></p> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.

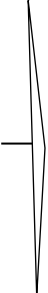
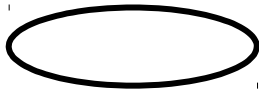


4.	Ekstensi / <i>extend</i>  << extend >> ----->	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use</i>
----	--	--

Lanjutan **Tabel 2.1** Simbol-simbol *Use Case Diagram*

No.	Simbol	Deskripsi
		<p>case yang ditambahkan, misal</p>  <p>arah panah mengarah pada <i>use case</i> yang ditambahkan; biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.</p>



5.	Generalisasi generalization 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya:  mengelola hapus data Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)
----	---	---

Lanjutan **Tabel 2.1** Simbol-simbol *Use Case Diagram*

No.	Simbol	Deskripsi
6.	Menggunakan / <i>include</i> / <i>uses</i>  << include >>  << uses >>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambhaknya memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i> : <ul style="list-style-type: none"> <li>• <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan, misal pada kasus berikut:    <div style="text-align: center;">validasi</div> </li> </ul>




		<pre>&lt;&lt; include&gt;&gt;</pre> <ul style="list-style-type: none"> <li>• Include berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang di tambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, misal pada kasus berikut:  <pre>login validasi user</pre> </li> </ul> <pre>&lt;&lt; include&gt;&gt;</pre> <pre>ubah data</pre> <p>Kedua interpretasi diatas dapat dianut salah satu atau keduanya tergantung ada pertimbangan dan interpretasi yang dibutuhkan.</p>
--	--	--

Sumber: Sukamto dan Salahuddin, 2018.





### 2.3.1.5 Activity Diagram

Sukamto dan Shalahuddin (2018:161) mengemukakan, “Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan actor, jadi aktivitas yang dapat dilakukan oleh sistem. “Menurut Sukamto dan Shalahuddin (2018:162—163) simbol-simbol yang ada pada diagram aktivitas:

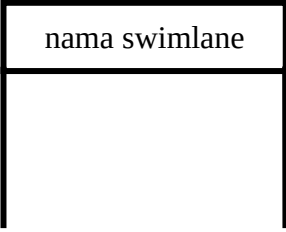
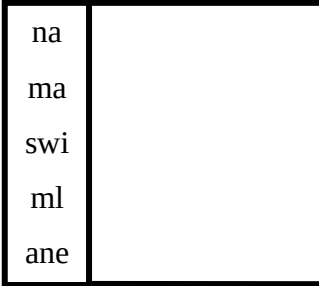
**Tabel 2.2** Simbol-simbol *Activity Diagram*

No.	Simbol	Deskripsi
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.	Pecabangan / <i>decision</i> 	Asosiasi pecabangan 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 sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.	Swimlane	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Lanjutan **Tabel 2.2** Simbol-simbol *Activity Diagram*

 <p>atau</p> 	
---	--

Sumber: Sukanto dan Shalahuddin, 2018.

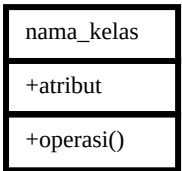



### 2.3.1.6 Class Diagram


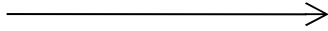
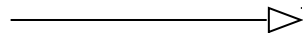
Menurut Sukamto dan Shalahuddin (2018:141) diagram kelas atau *class diagram* digunakan untuk menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.

Menurut Sukamto dan Shalahuddin (2018:146-147) simbol-simbol yang ada pada diagram kelas:

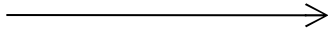
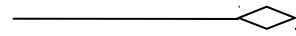
**Tabel 2.3** Simbol-simbol *Class Diagram*

No	Simbol	Deskripsi
1.	Kelas 	Kelas pada struktur sistem.
2.	Antarmuka / <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.

Lanjutan **Tabel 2.3** Simbol-simbol *Class Diagram*

3.	Asosiasi / <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
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 <i>multiplicity</i> .
5.	Generalisasi 	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 ( <i>whole-part</i> ).


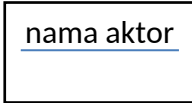
Sumber: Sukamto dan Shalahuddin, 2018.

### 2.3.1.7 Sequence Diagram


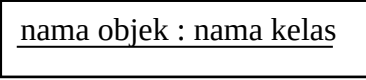

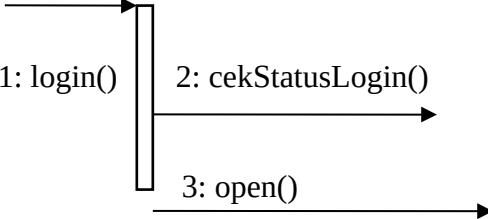
Menurut Sukamto dan Shalahuddin (2018:165) Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam 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*.

Menurut Sukamto dan Shalahuddin (2018:165-167) simbol-simbol yang ada pada *sequen diagram* yaitu:

**Tabel 2.4** Simbol-simbol *Sequence Diagram*

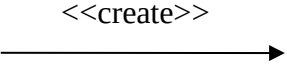
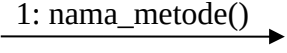
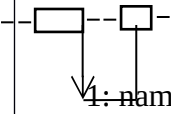
No.	Simbol	Deskripsi
1.	Aktor  atau  tanpa waktu aktif	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang kan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.



2.	Garis hidup / lifeline 	Menyatakan kehidupan suatu objek.
3.	Objek 	Menyatakan objek yang berinteraksi pesan.
4.	Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya 

Maka cekStatusLogin() dan open() dilakukan di dalam metode login(). Aktor tidak memiliki waktu aktif.

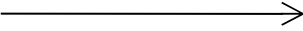
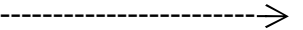
Lanjutan **Tabel 2.4** Simbol-simbol *Sequence Diagram*

No.	Simbol	Deskripsi
5.	Pesan tipe create 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
6.	Pesan tipe call 	Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri, 

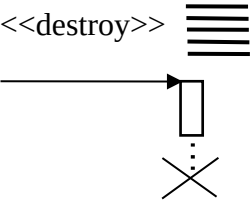
Arah panah mengarah pada objek yang





		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 send 1: masukan 	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8.	Pesan tipe return 1: 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.

Lanjutan **Tabel 2.4** Simbol-simbol *Sequence Diagram*

9.	Pesan tipe destroy <<destroy>> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy.
----	---	--

Sumber: Sukamto dan Shalahuddin, 2018.

### 2.3.2 *Black Box Testing*

Menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan (Sukamto dan Shalahuddin, 2018:275).



## 2.4 Teori Program

### 2.4.1 Bootstrap

Bootstrap merupakan salah satu framework CSS paling populer dari sekian banyak framework CSS yang ada. Bootstrap memungkinkan desain sebuah web menjadi responsif sehingga dapat dilihat dari berbagai macam ukuran *device* dengan tampilan tetap menarik (Abdulloh, 2018:261).

### 2.4.2 CSS (*Cascading Style Sheet*)

CSS adalah singkatan dari *Cascading Style Sheet* yaitu dokumen web yang berfungsi mengatur elemen HTML dengan berbagai property yang tersedia sehingga dapat tampil dengan berbagai gaya yang diinginkan (Abdulloh,2018:45).

### 2.4.3 Database

Basis data (*database*) dapat didefinisikan sebagai himpunan kelompok data saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah (Hidayatullah dan Kawistara, 2014:147).

### 2.4.4 HTML (*Hypertext Markup Language*)

*Hypertext Markup Language* (HTML) adalah bahasa standard yang digunakan untuk menampilkan halaman *web*. Yang bisa dilakukan HTML yaitu (Hidayatullah dan Kawistara, 2014:13).

1. Mengatur tampilan dari halaman *web* dan isinya.
2. Membuat tabel dalam halamn *web*.
3. Mempublikasikan halaman *web* secara *online*.
4. Membuat form yang bisa digunakan untuk menangani registrasi dan transaksi via *web*.
5. Menambahkan objek-objek seperti citra, audio, video, animasi, *java applet* dalam halaman *web*.
6. Menampilkan area gambar (*canvas*) di *browser*.



#### 2.4.5 MySQL

*MySQL* adalah salah satu aplikasi DBMS yang sudah sangat banyak digunakan oleh para pemrogram aplikasi web. Kelebihan dari *MySQL* adalah gratis, handal, selalu di-*update* dan banyak form yang memfasilitasi para pengguna jika memiliki kendala. *MySQL* juga menjadi DBMS yang sering dibundling dengan *web server* sehingga proses instalasinya jadi lebih mudah (Hidayatullah dan Kawistara, 2014: 180).

#### 2.4.6 PHP (*Hypertext Preprocessor*)

PHP *Hypertext Preprocessor* atau disingkat dengan PHP ini adalah suatu bahasa *scripting* khususnya digunakan untuk *web development*. Karena sifatnya yang *server side scripting*, maka untuk menjalankan PHP harus menggunakan *web server* (Hidayatullah dan Kawistara, 2014: 231).

#### 2.4.7 XAMPP

XAMPP merupakan merupakan paket php berbasis open source yang dikembangkan oleh sebuah komunitas *Open Source* dengan menggunakan XAMPP kita tidak perlu lagi melakukan penginstalan program yang lain karena semua kebutuhan telah disediakan oleh XAMPP. (Hendrianto, 2014:59).

XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache MySQL, PHP. XAMPP juga adalah tool yang menyediakan paket perangkat lunak dalam satu buah paket (Purba, 2015:34).

#### 2.4.8 Metode *Extreme Programming* (XP)

*Extreme Programming* (XP) merupakan sebuah proses rekayasa perangkat lunak yang cenderung menggunakan pendekatan berorientasi objek Prabowo (Spriyatna, 2018: 3).

