



## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Teori Umum

##### 2.1.1 Pengertian Komputer

Irma (2016:1), “Komputer adalah sekumpulan alat elektronik yang saling bekerja sama, dapat menerima data (*input*), mengolah data (*proses*) dan memberikan informasi (*output*) serta terkoordinasi dibawah kontrol program yang tersimpan dimemorinya.”

Kadir (2017:2), “Komputer adalah peralatan elektronik yang bermanfaat untuk melaksanakan berbagai pekerjaan yang dilakukan oleh manusia.”

Penulis menyimpulkan bahwa Komputer adalah alat bantu pemrosesan data secara elektronik mulai dari *input*, *proses*, dan *output* yang bermanfaat untuk melaksanakan berbagai pekerjaan manusia.

##### 2.1.2 Pengertian Perangkat Lunak

Utami dan Asnawati (2015:2), “*Software* adalah perangkat lunak yang berisikan sebuah intruksi yang diperintahkan dan diproses dengan bantuan perangkat keras sehingga tanpa perangkat lunak, perangkat keras tidak bisa dipakai sehingga *software* dan *hardware* tidak bisa dipisahkan.”

Sukamto dan Shalahuddin (2018:2), “Perangkat Lunak (*Software*) adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain dan cara penggunaan (*user manual*).”

Jadi didapatkan kesimpulan bahwa Perangkat Lunak adalah program komputer yang diperintahkan dan diproses dengan bantuan perangkat keras yang terasosiasi dengan dokumentasi perangkat lunak untuk memudahkan pekerjaan manusia.



### 2.1.3 Pengertian Basis Data

Sukamto dan Shalahuddin (2018:43), “Basis data adalah sistem komputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan.”

Yanto (2016:11), “Basis data adalah kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redundansi), untuk memenuhi berbagai kebutuhan.”

Kesimpulannya, Basis Data adalah sistem komputerisasi yang saling berhubungan berfungsi sebagai media penyimpanan data dan menyediakan informasi saat dibutuhkan.

### 2.1.4 Metode Pengembangan Sistem

Penelitian ini menggunakan metode pengembangan perangkat lunak dengan *RUP (Rational Unified Process)*. Menurut Sukamto dan Shalahuddin (2018:125), “*RUP (Rational Unified Process)* adalah pendekatan pengembangan perangkat lunak yang dilakukan berulang-ulang (*iterative*), fokus pada arsitektur (*architecture-centric*), lebih diarahkan berdasarkan penggunaan kasus (*use case driven*)”. Adapun tahap-tahap (*fase*) dalam metode pengembangan *RUP* menurut Sukamto dan Shalahuddin (2018:128-131) adalah sebagai berikut:

#### 1. *Inception* (permulaan)

Tahap ini lebih pada memodelkan proses bisnis yang dibutuhkan (*bussiness modeling*) dan mendefinisikan kebutuhan akan sistem yang akan dibuat (*requirements*).

#### 2. *Elaboration* (perluasan/perencanaan)

Tahap ini lebih difokuskan pada perencanaan arsitektur sistem. Tahap ini juga dapat mendeteksi apakah arsitektur sistem yang diinginkan dapat dibuat atau tidak. Mendeteksi resiko yang mungkin terjadi dari arsitektur yang dibuat. Tahap ini lebih pada analisis dan desain sistem serta implementasi sistem yang fokus pada purwarupa sistem (*prototype*).



### 3. *Construction* (kontruksi)

Tahap ini fokus pada pengembangan komponen dan fitur-fitur sistem. Tahap ini lebih pada implementasi dan pengujian sistem yang fokus pada implementasi perangkat lunak pada kode program. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari *Initial Operational Capability Milestone* atau batas/tonggak kemampuan operasional awal.

### 4. *Transition* (transisi)

Tahap ini lebih pada *deployment* atau instalasi sistem agar dapat dimengerti oleh user. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari *Initial Operational Capability Milestone* atau batas/tongggak kemampuan operasional awal. Aktifitas pada tahap ini termasuk pada pelatihan *user*, pemeliharaan dan pengujian sistem apakah sudah memenuhi harapan *user*.

## 2.2 Teori Judul

### 2.2.1 Pengertian Aplikasi

Jogiyanto (2004:4) mengatakan, “Aplikasi merupakan program yang berisikan perintah-perintah untuk melakukan pengolahan data”. Jadi secara umum Aplikasi adalah suatu proses dari cara manual kemudian ke komputer dengan program data dapat diolah serta lebih berdaya guna secara optimal.”

Pengertian Aplikasi menurut Kamus Besar Bahasa Indonesia, “Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu.”

Menurut Juansyah (2015:1), ”Secara istilah pengertian Aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju.”

### 2.2.2 Pengertian Pemesanan

Menurut Edwin dan Chris dikutip Deni (2017:2), ”Pemesanan dalam arti umum adalah perjanjian pemesanan tempat antara 2 (dua) pihak atau lebih, perjanjian pemesanan tempat tersebut dapat berupa perjanjian atas pemesanan



suatu ruangan, kamar, tempat dan lainnya, pada waktu tertentu dan disertai dengan produk jasanya. Produk jasa yang dimaksud adalah jasa yang ditawarkan pada perjanjian pemesanan tempat tersebut.”

### 2.2.3 Pengertian *Tracking*

Prabowo (2018:2), “*Tracking* secara harfiah memiliki arti mengikuti jalan, atau dalam arti bebasnya ialah suatu kegiatan untuk mengikuti jejak suatu objek. Sistem pelacakan adalah suatu sistem yang mampu melacak atau mencari suatu hal dengan memberikan informasi tentang hal tersebut.”

Margaretta (2012:34), “*Tracking* merupakan mekanisme bagaimana memantau keberadaan objek yang bergerak dan jalurnya. Pengertian bergerak dalam perspektif geografi adalah perpindahan posisi suatu objek dari suatu koordinat ke koordinat lain. *Tracking* diperoleh dengan merekam data perpindahan tersebut.”

Jadi dapat penulis simpulkan pengertian *Tracking* dalam hal ini ialah kegiatan untuk memantau keberadaan sebuah kendaraan berdasarkan posisi yang didapatkan dari peralatan *Tracking*.

### 2.2.4 Pengertian *Laundry*

Amalia (2008:28), “*Laundry* menurut istilah adalah salah satu usaha yang bergerak di bidang jasa, lebih tepatnya jasa mencuci dan setrika baju, *laundry* kiloan adalah *laundry* dengan *system* pembayaran berdasarkan berat *material* yang di *laundry*. Berat *material* pada *laundry* kiloan di peroleh oleh cara menimbang berat *material*nya sebelum *material* tersebut di sebut di bawa oleh petugas *laundry*.”

Menurut Setyawati dikutip Mulyadi dkk (2019:50), “Usaha *laundry* adalah suatu jenis usaha yang bergerak di bidang jasa cuci dan setrika. Bisnis ini termasuk dalam kategori bisnis dengan perputaran yang cepat, maksudnya rentang waktu permintaan pelanggan antara permintaan pertama dan permintaan selanjutnya pada jasa ini yang memakan waktu relatif singkat. Lebih jelasnya,



pelanggan akan kembali menggunakan jasa ini ketika pakaian yang dikenakan sudah kotor. Selain itu bisnis *laundry* juga merupakan kategori bisnis yang berkelanjutan atau sebagai bisnis yang akan selalu dibutuhkan banyak orang. Manusia akan selalu mencuci pakaian selama kebutuhan sandang masih menjadi kebutuhan primer bagi manusia. Hal inilah yang membuat peluang bisnis *laundry* sangat menjanjikan.”

### 2.2.5 Pengertian Web

Menurut Surajino (2004:12), “*Website* atau situs dapat diartikan sebagai kumpulan halaman halaman yang digunakan untuk menampilkan informasi, teks, gambar diam atau bergerak, animasi, suara, dan atau gabungan dari semuanya itu, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling berkait dimana masing masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*).”

*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*. Dari *home page*, pengunjung dapat mengklik *hyperlink* untuk pindah kehalaman lain yang terdapat dalam *website* tersebut (Jhonsen dikutip Hendrianto 2014:3).

*World Wide Web* (biasa disingkat WWW) atau web merupakan salah satu aplikasi internet yang paling populer. Web adalah sebuah sistem dimana informasi dalam bentuk teks, gambar, suara dan lainnya yang tersimpan dalam sebuah *internet webserver* ditampilkan dalam bentuk HTML (*hypertext Markup language*) (Sutanta, 2005:546).

### 2.2.6 Pengertian Aplikasi Pemesanan Dan *Tracking Laundry* Berbasis Web Pada Nasaqila *Laundry* Palembang

“Aplikasi Pemesanan Dan *Tracking Laundry* Berbasis Web Pada Nasaqila *Laundry* Palembang adalah Aplikasi yang dibuat untuk membantu pelaku usaha *laundry* (Nasaqila *Laundry*) dalam menjual jasa dan mempromosikan jasa



*laundry*-nya yang dapat diakses oleh masyarakat luas dari kota Palembang untuk meningkatkan ekonomi Nasaqila *Laundry* Palembang.”

## 2.3 Teori Khusus

### 2.3.1 Pengertian *Unified Modeling Language (UML)*

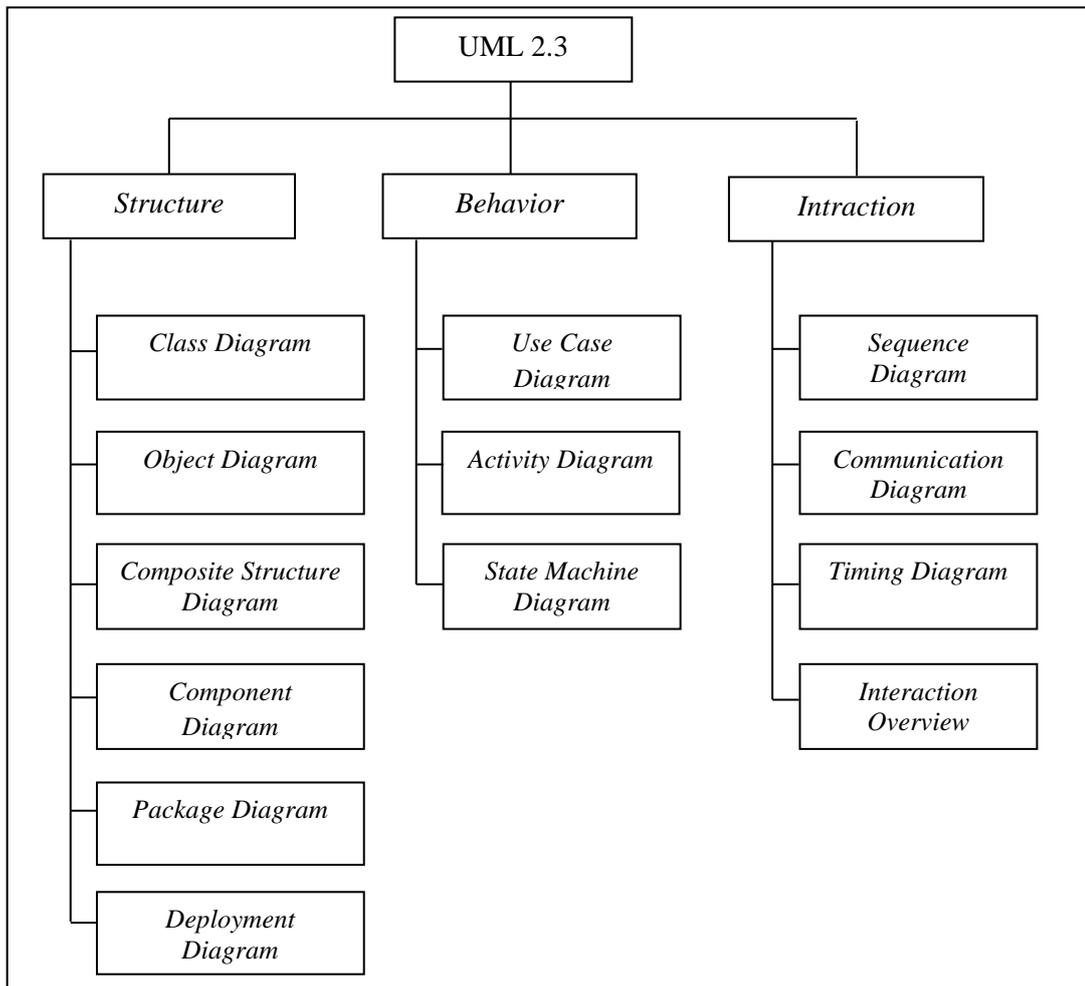
Sukamto dan Shalahuddin (2018:133), menjelaskan tentang pengertian *Unified Modeling Language* sebagai berikut :

“*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.”



**Gambar 2.1** Tampilan Logo UML

Menurut Sukamto dan Shalahuddin (2018:140), “Pada UML 2.3 terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori”. Pembagian kategori dan macam-macam diagram Menurut Sukamto dan Shalahuddin tersebut dapat dilihat pada gambar dibawah:



**Gambar 2.2** Kategori dan Macam-macam Diagram UML

Penjelasan singkat dari pembagian kategori pada diagram UML menurut Sukamto dan Shalahuddin (2018:141) :

- 1) *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
- 2) *Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
- 3) *Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.



## 2.3.2 Jenis-Jenis Diagram UML

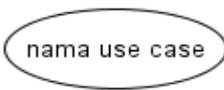
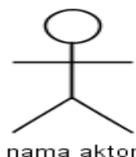
### 2.3.2.1 Pengertian *Use case* Diagram

Sukamto dan Shalahuddin (2018:155), menjelaskan tentang *use case* diagram sebagai berikut :

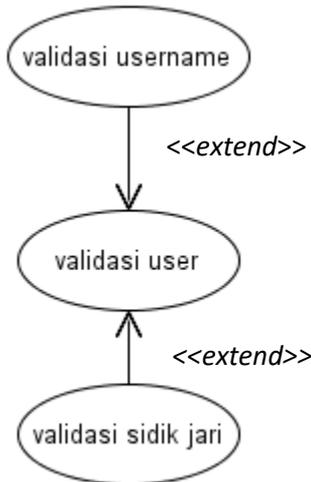
“*Use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan system.”

Adapun simbol-simbol yang digunakan dalam *Use case* adalah sebagai berikut:

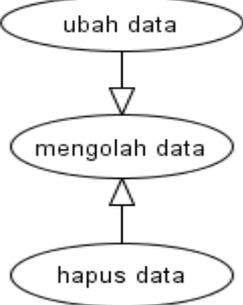
**Tabel 2.1.** Simbol-simbol *Use case* Diagram

No	Simbol	Deskripsi
1	<p><i>Use case</i></p> 	<p>fungsi yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal-awal frase nama <i>use case</i>.</p>
2	<p>aktor / <i>actor</i></p> 	<p>orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan 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 <i>actor</i>.</p>
3	<p>asosiasi / <i>association</i></p> 	<p>komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan <i>actor</i>.</p>

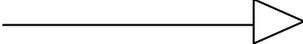
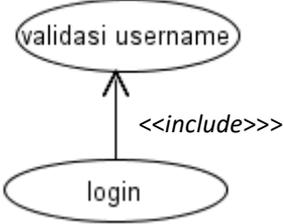
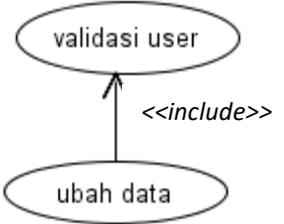
Lanjutan Tabel 2.1. Simbol-simbol *Use case* Diagram

No	Simbol	Deskripsi
4	ekstensi / <i>extend</i>  	<p>relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang di tambahkan 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 case</i> yang ditambahkan, misalnya,</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>

Lanjutan Tabel 2.1. Simbol-simbol *Use case* Diagram

No	Simbol	Deskripsi
5	Generalisasi/ <i>generalization</i> 	<p>hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya,</p> <div style="text-align: center;">  </div> <p>misalnya:            arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum).</p>

Lanjutan Tabel 2.1. Simbol-simbol *Use case* Diagram

No	Simbol	Deskripsi
6	<p>menggunakan / <i>include</i> / <i>uses</i></p> <p><code>&lt;&lt;include&gt;&gt;</code>    <code>&lt;&lt;uses&gt;&gt;</code>  </p>	<p>relasi tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan 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>:</p> <ul style="list-style-type: none"> <li>• <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu di panggil saat <i>use case</i> tambahan dijalankan, misalnya pada kasus berikut:</li> </ul>  <ul style="list-style-type: none"> <li>• <i>Include</i> 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:</li> </ul>  <p>kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>

Sumber: Sukamto dan Shalahuddin (2018:156)



Ada dua hal utama pada *use case* yaitu:

1. Aktor merupakan 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.
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

### 2.3.2.2 Pengertian *Class Diagram*

Sukamto dan Shalahuddin (2018:141), menjelaskan tentang *class diagram* sebagai berikut :

“*Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Diagram kelas dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron.”

Adapun simbol-simbol yang digunakan dalam *class diagram* adalah sebagai berikut:

**Tabel 2.2** 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 2.2 Simbol-simbol *Class Diagram*

No	Simbol	Deskripsi
3	asosiasi / <i>association</i> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai <i>multiplicity</i> .
4	asosiasi berarah/ <i>directed association</i> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5	generalisasi 	Relasi antarkelas dengan makna generalisasi – spesialisasi (umum khusus).
6	kebergantungan/ <i>dependency</i> 	Relasi antarkelas dengan makna kebergantungan antar kelas.
7	agregasi / <i>aggregation</i> 	Relasi antarkelas dengan makna semua-bagian ( <i>whole-part</i> ).

Sumber: Sukamto dan Shalahuddin (2018:146)

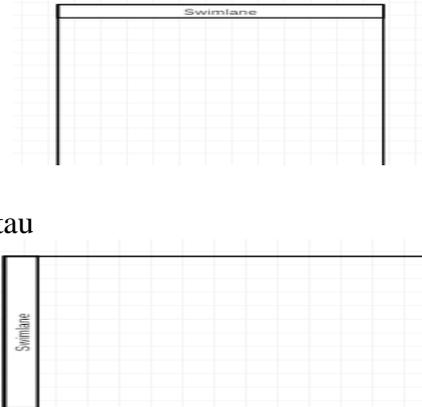
### 2.3.2.3 Pengertian *Activity Diagram*

Sukamto dan Shalahuddin (2018:161), menjelaskan tentang *activity diagram* sebagai berikut :

“*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 aktor, jadi aktivitas yang dapat dilakukan oleh sistem.”

Adapun simbol-simbol yang digunakan dalam *activity diagram* adalah sebagai berikut:

Tabel 2.3 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	Percabangan / <i>decision</i> 	Asosiasi percabangan di mana 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	<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Sumber: Sukamto dan Shalahuddin (2018:164)

### 2.3.2.4 Pengertian *Sequence Diagram*

(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



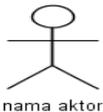
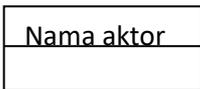
menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah beserta metode-metode yang dimiliki kelas yang diinstansikan menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *Use case*.”

Banyaknya diagram sekuen yang harus digambarkan adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

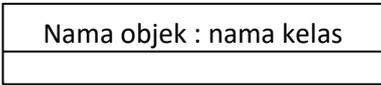
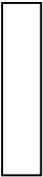
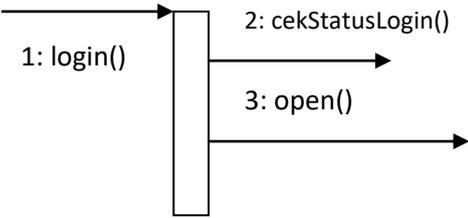
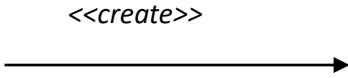
Dapat penulis simpulkan bahwa *Sequence diagram* adalah penggambaran skenario dari sebuah objek yang ada pada *use case* yang meliputi rangkaian langkah-langkah aktivitas dari objek berdasarkan waktu hidup objek dan pesan-pesan yang diterima maupun yang dikirimkan objek kepada objek lainnya.

Berikut simbol-simbol pada *Sequence Diagram* :

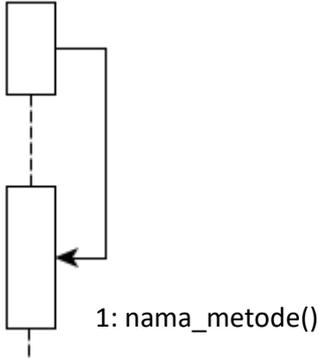
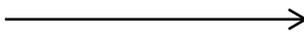
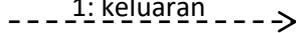
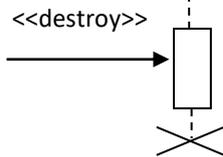
**Tabel 2.4** Simbol-simbol pada *Sequence Diagram*

No	Simbol	Deskripsi
1	<p><i>Actor</i></p>  <p>atau</p>  <p>tanpa waktu aktif</p>	<p>orang, proses, atau sistem lain yang berinteraksi dengan 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 <i>actor</i>.</p>
2	<p>Garis hidup / <i>lifeline</i></p> 	<p>menyatakan kehidupan suatu objek.</p>

Lanjutan Tabel 2.4 Simbol-simbol pada *Sequence Diagram*

No	Simbol	Deskripsi
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 ,  <p>maka <i>cekStatusLogin ()</i> dan <i>open()</i> dilakukan di dalam metode <i>login()</i> aktor tidak memiliki waktu aktif.</p>
5	Pesan tipe <i>create</i> 	menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.

Lanjutan Tabel 2.4 Simbol-simbol pada *Sequence Diagram*

No	Simbol	Deskripsi
6	Pesan tipe <i>call</i> <<create>> 	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> 1: masukan 	menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8	Pesan tipe <i>return</i> --- 1: keluaran ---> 	menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu.
9	Pesan tipe <i>destroy</i> <<destroy>> 	menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .

Sumber: Sukamto dan Shalahuddin (2018:165-167)



## 2.4 Teori Program

### 2.4.1 Pengertian GPS

*GPS (Global Positioning System)* adalah sistem satelit navigasi dan penentuan posisi yang dimiliki dan dikelola oleh Amerika Serikat. Sistem ini didesain untuk memberikan posisi dan kecepatan tiga dimensi serta informasi mengenai waktu. GPS terdiri dari 3 segmen yaitu segmen angkasa, kontrol/pengendali, dan pengguna. Segmen angkasa terdiri dari 24 satelit yang beroperasi dalam 6 orbit pada ketinggian 20.200 km dengan periode 12 jam (satelit akan kembali ke titik yang sama dalam 12 jam). Segmen Kontrol/Pengendali terdapat pusat pengendali utama yang terdapat di Colorado Springs, dan 5 stasiun pemantau lainnya dan 3 antena yang tersebar di bumi ini. Pada sisi pengguna dibutuhkan penerima *GPS* yang biasanya terdiri dari penerima, *prosesor*, dan antena. (Ginting, 2018:187-188).

### 2.4.2 Pengertian Google Maps API

*Google Maps* merupakan layanan dari *google* yang mempermudah penggunaannya untuk melakukan kemampuan pemetaan untuk aplikasi yang dibuat. Sedangkan *Google Maps API* memungkinkan pengembangan untuk mengintegrasikan *Google Maps* ke dalam situs web. Dengan menggunakan *Google Maps API* memungkinkan untuk menanamkan situs *Google Maps* ke dalam situs eksternal, di mana situs data tertentu dapat dilakukan *Overlay* 11.



**Gambar 2.3** Tampilan Google Maps



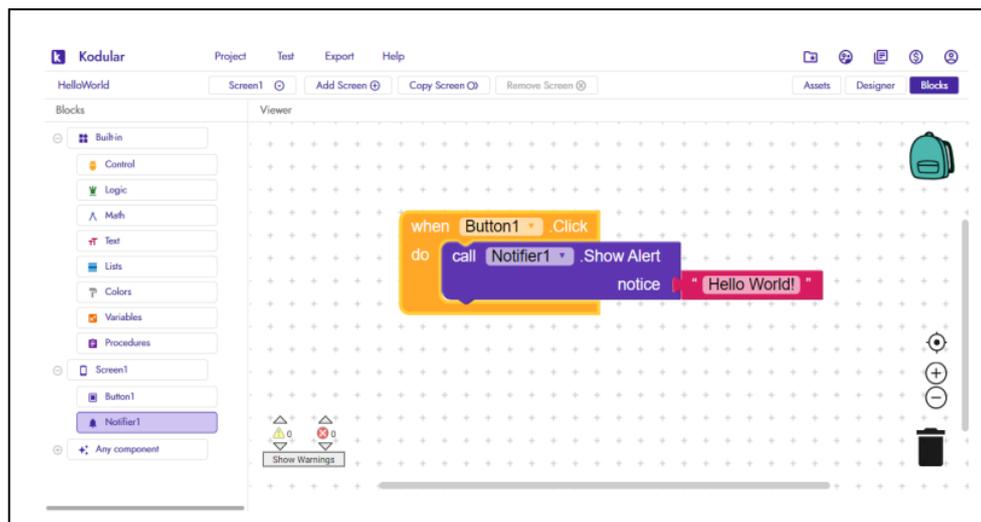
Meskipun pada awalnya hanya *javascript API*, *API Maps* sejak diperluas untuk menyertakan sebuah *API* untuk *Adobe Flash* aplikasi, layanan untuk mengambil gambar peta status, dan layanan *web* untuk melakukan *geocoding*, menghasilkan petunjuk arah mengemudi, dan mendapatkan profil *elevasi*. (Juansyah, 2015:4).

*Google Maps* adalah layanan gratis yang diberikan oleh *Google* dan sangat populer. *Google Maps* adalah suatu peta dunia yang dapat kita gunakan untuk melihat suatu daerah. Dengan kata lain, *Google Maps* merupakan suatu peta yang dapat dilihat dengan menggunakan suatu browser. Kita dapat menambahkan fitur *Google Maps* dalam *web* yang telah kita buat atau pada blog kita yang berbayar maupun gratis sekalipun dengan *Google Maps API*. *Google Maps API* adalah suatu library yang berbentuk *JavaScript*. (Kindarto, 2008).

### 2.4.3 Pengertian Kodular

Kodular merupakan *platform open-source* yang dapat membuat aplikasi android dengan berbasis web. *Platform* ini berfungsi untuk membuat berbagai jenis aplikasi Android dengan mudah dan cepat tanpa memprogram dengan tulisan. Kodular didesain dengan tampilan yang sederhana dan praktis. Kodular *platform* berbasis web yang dapat membuat berbagai jenis aplikasi Android tanpa dan Kodular juga bahasa pemrograman tingkat tinggi. *Platform* ini tidak perlu menulis kode-kode program konvensional yaitu kode yang berbasis teks untuk membuat aplikasi androidnya.

Kodular merupakan *platform* yang memprogram dengan cara *drag & drop* dan *Visual Blocks Programming (VBP)* untuk mengembangkan aplikasi-aplikasi yang berjalan di sistem operasi Android. *Platform* ini merupakan *platform* yang bersifat *open-source* dengan proses pembuatan yang mudah. Fitur *VBP* ini mampu mentransformasikan pengkodean bahasa pemrograman berbasis teks ke dalam bahasa visual dalam bentuk blok-blok kode program (Wolber, 2014).



**Gambar 2.4** Tampilan Kodular

*Sumber : www.kodular.io*

#### 2.4.4 Pengertian Android

Menurut Desmira (2016:6). Android adalah sistem operasi untuk telepon seluler yang berbasis *Linux*. Android menyediakan *platform* terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, Google Inc. membeli Android Inc, pendatang baru yang membuat peranti lunak untuk ponsel. Android adalah sistem operasi berbasis *linux* yang digunakan untuk ponsel (telepon seluler) mencakup sistem operasi, *middleware* dan aplikasi.

Menurut Giyartono dkk (2015:2), “Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka.”

Menurut Sari (2015:151), “Android adalah sistem operasi berbasis kernel Linux yang pada awalnya dikembangkan oleh Android, Inc, yang didukung *Google* finansial dan kemudian dibeli pada tahun 2005. Android ini diresmikan pada tahun 2007 seiring dengan berdirinya *Open Handset Alliance* konsorsium *hardware*, *software*, dan perusahaan telekomunikasi yang ditujukan untuk memajukan standar perangkat selular.”



**Gambar 2.5** Logo Android

*Sumber: www. graetnewsnetwork.com*

#### **2.4.5** Pengertian *Firebase*

*Firebase* adalah penyedia layanan *cloud* dengan *back-end* sebagai servis yang berbasis di San Fransisco, California. *Firebase* membuat sejumlah produk untuk pengembangan aplikasi *Mobile* ataupun web. *Firebase* didirikan oleh Andrew Lee dan James Tamplin pada tahun 2011 dan diluncurkan dengan *cloud database* secara *realtime* di tahun 2012 .

Produk utama dari *Firebase* yakni suatu *database* yang menyediakan *API* untuk memungkinkan pengembang menyimpan dan mensinkronisasi data lewat *multiple client*. Perusahaan ini diakuisi oleh *Google* pada Oktober 2014.

*Firebase* adalah penyedia layanan *realtime database* dan *backend* sebagai layanan. Suatu aplikasi yang memungkinkan pengembang membuat *API* untuk disinkronisasikan untuk *client* yang berbeda-beda dan disimpan pada *cloud*-nya *Firebase*. *Firebase* memiliki banyak *library* yang memungkinkan untuk mengintegrasikan layanan ini dengan *Android*, *Ios*, *Javacript*, *Java*, *Objective-C* dan *Node.JS*. *Database Firebase* juga bersifat bisa diakses lewat *REST API*. *REST API* tersebut menggunakan protokol *Server-Sent Event* dengan membuat koneksi *HTTP* untuk menerima push notification dari server. Pengembang menggunakan *REST API* untuk *post* data yang selanjutnya *Firebase client library* yang sudah diterapkan pada aplikasi yang dibangun yang akan mengambil data secara *realtime*.



**Gambar 2.6** Logo *Firebase*

*Sumber : www.arthnugraha.com*

#### **2.4.6 Pengertian XAMPP**

Dadan dan Kerendi (2015:28), “*XAMPP* adalah salah satu aplikasi *web server apache* yang terintegrasi dengan *mysql* dan *phpmyadmin*.”

Madcoms (2016:186), “*XAMPP* adalah sebuah paket kumpulan software yang terdiri dari *Apache*, *MySQL*, *PhpMyAdmin*, *Perl*, *Filezilla* dan lain-lain.”

Jadi, penulis simpulkan pengertian *XAMPP* adalah sebuah aplikasi perangkat lunak pemrograman dan *database* yang di dalamnya terdapat berbagai macam aplikasi pemrograman yang terdiri dari *Apache*, *MySQL*, *PhpMyAdmin*, *Perl*, *Filezilla* dan lain-lain.

#### **2.4.7 Pengertian MySQL**

Sukamto dan Shalahuddin (2018:46), “*SQL (Structured Query Language)* adalah bahasa yang digunakan untuk mengelola data pada *RDBMS*. *SQL* awalnya dikembangkan berdasarkan teori aljabar relasional dan kalkulus.”

Murya (2014:46), “*MySQL* adalah sebuah perangkat lunak sistem manajemen basis data *SQL (database management system)* atau *DBMS* yang *multi thread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia.”

Jadi dapat penulis simpulkan pengertian *MySQL* adalah bahasa yang digunakan untuk mengelola data *SQL (database management system)* atau *DBMS* yang *multithread*, *multi-user*.



### 2.4.8 Pengertian *PHP*

Murya (2014:65), “*PHP Hypertext Preprocessor* atau sering disebut *PHP* merupakan bahasa pemrograman berbasis *server-side* yang dapat melakukan *parsing script php* menjadi *script web* sehingga dari sisi *client* menghasilkan suatu tampilan yang menarik.”

Madcoms (2016:2), “*PHP (Hypertext preprocessor)* adalah bahasa *script* yang dapat ditanamkan atau disisipkan ke dalam *HTML*.”

Jadi, dapat penulis simpulkan pengertian *PHP* adalah bahasa pemrograman berbasis *server-side* yang bisa kita gunakan untuk membuat aplikasi web yang disisipkan pada *HTML*.



**Gambar 2.7** Tampilan Logo *PHP*

#### 2.4.8.1 Sintaks Dasar *PHP*

Yuana (2015:2), menjelaskan kode-kode *PHP* dituliskan diantara tanda berikut ini:

```
<?php
```

```
...
```

```
...
```

```
...
```

```
?>
```

Atau

```
<?
```

```
...
```

```
...
```

```
...
```

```
?>
```



Apabila membuat kode *PHP* dan berencana akan mendistribusikan ke pihak/orang lain, maka usahakan menggunakan sintaks `<?php ... ?>`. Hal ini dikarenakan untuk penggunaan kode yang menggunakan `<? ... ?>` terkadang tidak bisa dijalankan dalam *server* tertentu.

#### 2.4.8.2 Tipe Data *PHP*

Tipe data merupakan jenis dari suatu data yang akan diproses oleh bahasa pemrograman. Murya (2014:26), menjelaskan beberapa tipe data dalam *PHP*, sebagai berikut :

1. ***Integer*** merupakan tipe data yang berguna untuk menyimpan bilangan bulat. *Range* bilangan *integer* adalah antara -2.147.483.647 sampai dengan 2.147.483.647
2. ***Double Floating*** adalah tipe data yang berguna untuk menyimpan bilangan desimal. *Range* bilangan floating point antara 1e308 sampai dengan 1e308.
3. ***Boolean*** adalah tipe data yang paling sederhana, hanya berupa ***TRUE*** dan ***FALSE***.
4. ***String*** adalah tipe data yang terdiri dari kata, bias berupa kata tunggal maupun kalimat. Penulisan *string* harus diapit dengan tanda petik, baik berupa petik tunggal ('...') maupun petik ganda ("...").
5. ***Objek*** adalah tipe data dibuat dengan tujuan agar para *programmer* terbiasa dengan OOP. Tipe data ini bias berupa bilangan.
6. ***Array*** merupakan ***Tipe Compound Primitif***, terdapat pada bahasa pemrograman lain.
7. ***Null*** adalah tipe data yang tidak memuat apapun. Setiap variabel yang diset menjadi tipe data *Null*, ini akan menjadikan variabel tersebut kosong.
8. ***Resources*** tipe data spesial yang satu ini dikhususkan untuk menyimpan *resources*, sumber atau alamat.

#### 2.4.8.3 Pengertian *PHPMyAdmin*



Hikmah, dkk (2015:2), “*PHPMyAdmin* merupakan aplikasi yang dapat digunakan untuk membuat *database*, pengguna (*user*), memodifikasi tabel, maupun mengirim *database* secara cepat dan mudah tanpa harus menggunakan perintah (*command*) *SQL* .”

Rahman (2014:12), “*PHPMyAdmin* adalah aplikasi *PHP* sebagai administrator *MySQL*, *PHPMyAdmin* mendukung berbagai aktivitas *MySQL* seperti pengelolaan data, *table*, relasi antar *table* dan lain sebagainya.”

Jadi, penulis simpulkan pengertian *PHPMyAdmin* adalah aplikasi *PHP* sebagai administrator *MySQL* yang digunakan untuk membuat *database*, mengelola tabel, mengelola data, relasi antar tabel, dan mengirim *database* secara praktis tanpa harus menggunakan perintah (*command*) *SQL*.”



**Gambar 2.8** Tampilan logo *PHPMyAdmin*