



## 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 Media Pembelajaran

Menurut Bethany dikutip Mustaqim (2017:36), ”mengemukakan bahwa proses belajar adalah proses penyampaian pesan/materi dari pemberi pesan (guru) ke penerima pesan (peserta didik).



Menurut Arsyad (2005:3), “Istilah *media* berasal dari bahasa Latin yaitu *medius* yang berarti tengah, perantara, atau pengantar. Dalam bahasa Arab, media adalah perantara atau pengantar pesan dari pengirim kepada penerima pesan. Menurut AECT (*Association of Education and Communication Technology*) yang dikutip oleh Basyaruddin (2002) “media adalah segala bentuk yang dipergunakan untuk proses penyaluran informasi”.

Sedangkan menurut Djamarah (2006:136) pengertian lain media adalah alat bantu apa saja yang dapat dijadikan sebagai penyalur pesan guna mencapai tujuan pembelajaran. Dari definisi-definisi tersebut dapat dikatakan bahwa media merupakan sesuatu yang bersifat meyakinkan pesan dan dapat merangsang pikiran, perasaan, dan kemauan audiens (siswa) sehingga dapat mendorong terjadinya proses belajar pada dirinya.

### **2.2.3 Pengertian Digital**

Menurut Wikipedia Digital merupakan penggambaran dari suatu keadaan bilangan yang terdiri dari angka 0 dan 1 atau *off* dan *on* (bilangan biner). Semua sistem komputer menggunakan sistem digital sebagai basis datanya. Dapat disebut juga dengan istilah Bit (*Binary Digit*).

### **2.2.4 Pengertian Peninggalan Sejarah**

Peninggalan Sejarah adalah barang, benda, tulisan atau bangunan dari masa lalu yang memiliki nilai sejarah. Indonesia kaya dengan peninggalan-peninggalan bernilai sejarah. Peninggalan sejarah itu dapat berupa adat atau budaya, alat, benda atau barang, tulisan, maupun bangunan. Peninggalan sejarera berupa budaya, yaitu kebiasaan dari nenek moyang dan berlaku secara turun temurun.

### **2.2.5 Pengertian Objek Wisata**

Objek wisata adalah segala sesuatu yang ada di daerah tujuan wisata yang merupakan daya tarik agar orang-orang mau datang berkunjung ke tempat tersebut.



### 2.2.6 Pengertian Website

Menurut (Agus Hariyanto, 2015), *website* adalah: “*Web* dapat diartikan sebagai kumpulan halaman yang menampilkan informasi data teks, data gambar, data animasi, suara, video dan gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*)”.

Menurut Rohi Abdulloh (2015) *website* adalah: “Sekumpulan halaman yang terdiri dari beberapa halaman yang berisi informasi dalam bentuk data digital baik berupa teks, gambar, video, audio dan animasi lainnya yang disediakan melalui jalur koneksi internet”.

Dari penjelasan beberapa pendapat para ahli diatas maka dapat disimpulkan bahwa *website* sebuah tempat di mana sebuah informasi dapat di dapatkan melalui internet yang berupa sebuah halaman digital.

### 2.2.7 Pengertian Aplikasi Media Pembelajaran Digital Peninggalan Sejarah Objek Wisata Bukit Siguntang Berbasis Website

“Aplikasi Media Pembelajaran Digital Peninggalan Sejarah Objek Wisata Bukit Siguntang Berbasis Website adalah aplikasi berbasis website yang di buat untuk membantu masyarakat Provinsi Sumatera Selatan mendapatkan wawasan tentang prasejarah objek wisata di area Bukit Siguntang secara efisien tanpa harus mengunjungi lokasi”.



## 2.3 Teori Khusus

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

“Unified Modeling Language (UML) adalah Bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan system berorientasi objek dan juga alat untuk mendukung pengembangan sistem” (Windu dan Grade, 2013:2). (Adi, 2018:2), “Unified Modeling Language (UML) adalah bahasa pemodelan untuk system atau perangkat lunak yang berparadigma berorientasi objek. Abstraksi konsep dasar UML terdiri dari structural classification, dynamic behaviour, dan model management dapat kita pahami main concepts sebagai team yang akan muncul pada saat membuat diagram dan view adalah kategori dari diagram tersebut, UML mendefinisikan diagram-diagram sebagai Use Case diagram, Class diagram, Statechart diagram, Activity diagram, Sequence diagram, Collaboration diagram, Component diagram, dan Deployment diagram”

### 2.3.2 Jenis-Jenis Diagram UML

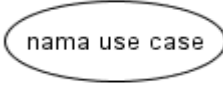


#### 2.3.2.1 Pengertian *Use case Diagram*

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

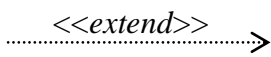
“*Use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan system.”

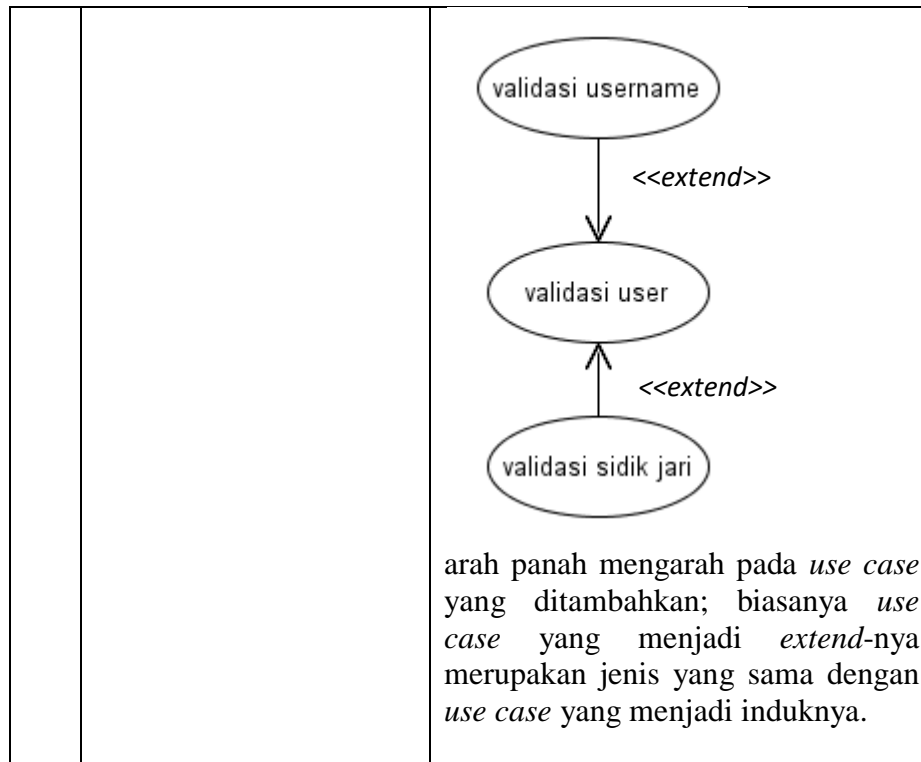
Adapun simbol-simbol yang digunakan dalam *Use case* adalah sebagai berikut:

Tabel 2.1. Simbol-simbol *Use case* Diagram

No	Simbol	Deskripsi
1	<p><i>Use case</i></p> 	<p>fungsionalitas 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


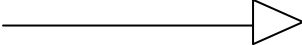
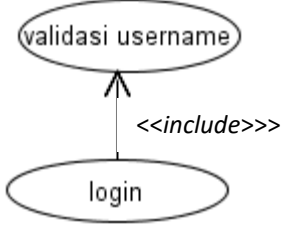
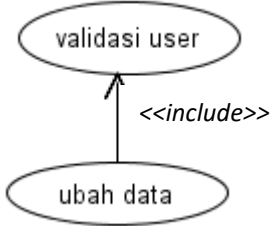
No	Simbol	Deskripsi
4	<p>ekstensi / <i>extend</i></p> 	<p>relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang di tambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misalnya,</p>

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> <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.

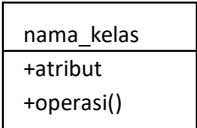
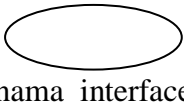
### 2.3.2.2 Pengertian *Class Diagram*

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)


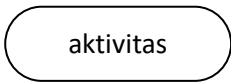



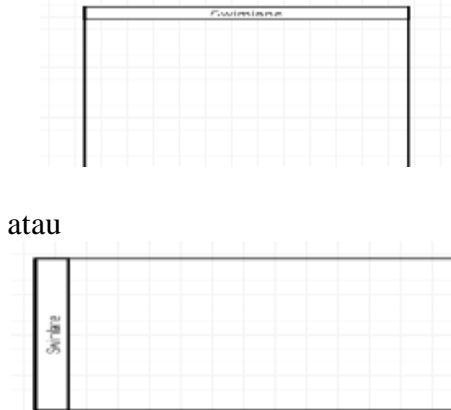
### 2.3.2.3 Pengertian *Activity Diagram*

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



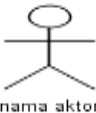
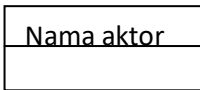
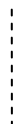
menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah beserta metode-metode yang dimiliki kelas yang diinstansikan menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *Use case*.”

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.

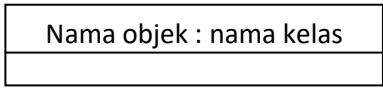

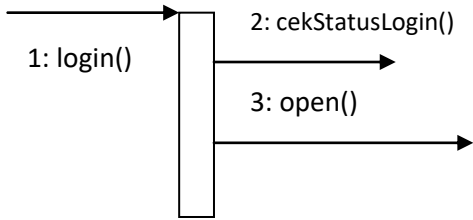
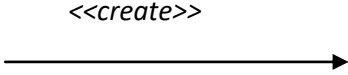
Dapat penulis simpulkan bahwa *Sequence diagram* adalah penggambaran skenario dari sebuah objek yang ada pada *use case* yang meliputi rangkaian langkah-langkah aktivitas dari objek berdasarkan waktu hidup objek dan pesan-pesan yang diterima maupun yang dikirimkan objek kepada objek lainnya.

Berikut simbol-simbol pada *Sequence Diagram* :


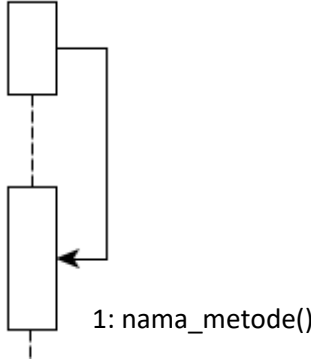
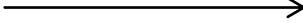

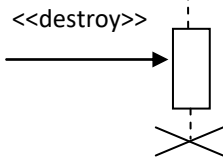
**Tabel 2.4** Simbol-simbol pada *Sequence Diagram*

No	Simbol	Deskripsi
1	<p><i>Actor</i></p>  <p>nama aktor</p> <p>atau</p>  <p>Nama aktor</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 Basis Data

(Kristanto, 2004:3), “Kumpulan file-file yang mempunyai kaitan antara satu file dengan file yang lain sehingga membentuk satu bangunan data untuk menginformasikan satu perusahaan, instansi dalam batasan tertentu”.

(Harison dan Syarif, 2016:43), “Database sebagai kumpulan informasi yang bermanfaat yang diorganisasikan kedalam tata cara yang khusus”.

Dalam praktek, penggunaan istilah database (Elmasri, 2016) lebih dibatasi pada artiimplisit yang khusus, yaitu:

- a) Basis data merupakan penyajian suatu aspek dari dunia nyata.
- b) Basis data merupakan kumpulan data dari berbagai sumber yang secara logika mempunyai arti implisit, sehingga data yang terkumpul secara acak dan tanpa mempunyai arti, tidak dapat disebut sebagai database.
- c) Database perlu dirancang, dibangun, dan data dikumpulkan untuk suatu tujuan. Database dapat digunakan oleh beberapa pemakai dan beberapa aplikasi yang sesuai dengan kepentingan pemakai (Waljiyanto dikutip Harison dan Syarif, 2016).

### 2.4.2 Pengertian XAMPP

(Sidik, 2017:67), “XAMPP secara default menyatukan antara HTML dan PHP, dengan direktori penyimpanan script terletak di c:\xampp\htdocs. Direktori c:\xampp\htdocs ini disebut sebagai direktori root (root directory), isi dari direktori ini yang akan ditampilkan jika kita mengakses lewat browser web, dengan URL http://localhost, jika kita mengakses server web dari komputer yang sama”.





Menurut Enterprise (2017:93), menyatakan bahwa XAMPP merupakan tool yang digunakan untuk membantu kita bekerja menggunakan MySQL Server, namun dengan tampilan antarmuka grafis yang lebih murah untuk siapa pun, terlebih untuk kalangan pemula.

Menurut Gunawan (2017:17), mengemukakan XAMPP adalah aplikasi web server instan yang dibutuhkan untuk membangun aplikasi berbasis web, fungsi XAMPP adalah sebagai server yang berdiri sendiri (localhost), yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl.

### **2.4.3 Pengertian MySQL**

Menurut Nugroho (2017:33), MySQL (My Structured Query Language) adalah: "Suatu sistem basis data relation atau Relational Database Management System (RDBMS) yang mampu bekerja secara cepat dan mudah digunakan. MySQL juga merupakan program pengakses database yang bersifat jaringan, sehingga dapat digunakan untuk aplikasi multi user (banyak pengguna). MySQL didistribusikan gratis dibawah lisensi GPL (General Public License). Dimana Politeknik Negeri Sriwijaya 20 setiap program bebas menggunakan MySQL namun tidak bias dijadikan produk turunan yang dijadikan closed source atau komersial".

(Sidik, 2017:301), "MySQL merupakan software database yang termasuk paling populer di lingkungan Linux, kepopuleran ini karena ditunjang karena performansi query dari databasenya yang saat itu bias dikatakan paling cepat, dan jarang bermasalah. MySQL telah tersedia juga di lingkungan Windows".

### **2.4.4 Pengertian PHP**

(Anonim, 2014:73), "Bahasa pemrograman PHP merupakan bahasa pemrograman untuk membuat website yang bersifat server-side scripting".

(Anonim, 2014:33), "PHP merupakan bahasa berbentuk script yang ditempatkan di dalam server baru kemudian diproses. Kemudian hasil pemrosesan dikirimkan kepada web browser klien".



### **2.4.5 Pengertian HTML**

(Harison dan Syarif, 2016:43) mengatakan bahwa, “Hyper Text Markup Language (HTML) adalah sebuah bahasa markup yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi di dalam sebuah Penjelajah web Internet dan formatting hyper text sederhana yang ditulis kedalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi”. Kaban (2019:4) HTML merupakan singkatan dari Hyper Text Markup Language yaitu bahasa standar web yang dikelola penggunaannya oleh W3C (World Wide web Consortim) berupa tag-tag yang menyusun setiap elemen dari website.

Hidayatullah dan Kawistra (2017:5), Hyper Text Markup Language (HTML) adalah bahasa standard yang digunakan untuk menampilkan halaman web. Sedangkan menurut Naista (2016:62) Hyper Text Markup Language atau lebih disingkat dengan HTML adalah bahasa markup yang digunakan untuk membuat sebuah halaman web.

### **2.4.6 Pengertian Sublime Text 3**

(Pahlevi, Omar, dkk, 2018:29), “Sublime Text merupakan salah satu editor yang sangat powerful yang dapat meningkatkan produktivitas dan mengembangkan kualitas kode yang tinggi”.

### **2.4.7 Pengertian CSS**

Pratama (2016:1), menyatakan bahwa CSS merupakan singkatan dari Cascading Style Sheet. CSS digunakan untuk mengubah tampilan (style) dari halaman web. Sedangkan menurut Abdullah (2016:2), CSS singkatan dari Cascading Style Sheets, yaitu skrip yang digunakan untuk mengatur desain website.

Menurut Madcoms (2011:100) CSS (Cascading Style Sheet) adalah sebuah fitur yang diperkenalkan sejak HTML versi 4.0 dan berfungsi untuk menangani masalah tampilan pada HTML seperti jenis, ukuran dan warna font, posisi teks, batas tulisan atau margin, warna background, dan sebagainya. Penting yang perlu diperhatikan adalah cara meletakkan CSS dan juga bahasa berbasis web lain untuk memudahkan manajemen file, editing, dan maintenance.