



BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1 Pengertian Komputer

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

Sedangkan Rosdiana (2016:1), “Secara definisi komputer diterjemahkan sebagai sekumpulan alat elektronik yang saling berkeja sama, dapat menerima data (*input*), mengolah data (*proses*) dan memberikan informasi (*output*) serta terkoordinasi dibawah *control* program yang tersimpan di memorinya”.

Dari kedua pendapat di atas, dapat disimpulkan pengertian komputer adalah peralatan elektronik yang saling berkerjasama yang dapat menerima data, memproses data dan memberikan informasi yang dipergunakan di berbagai pekerjaan yang dilakukan oleh manusia.

2.1.2 Pengertian Perangkat Lunak (*Software*)

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

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

Jadi, dapat disimpulkan bahwa perangkat lunak merupakan program yang berisikan sebuah instruksi yang terasosiasi dengan dokumentasi perangkat lunak yang diperintahkan dan diproses dengan bantuan perangkat keras.



2.1.3 Pengertian Basis Data (*Database*)

Lubis (2016:2), “Basis data merupakan gabungan *file* yang dibentuk dengan hubungan/relasi yang logis dan dapat diungkapkan dengan catatan serta bersifat *independent*”.

Sedangkan Fathansyah (dikutip Sagita dan Sugiarto, 2016:51), “Basis data adalah himpunan kelompok data atau arsip yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah”.

Berdasarkan penjelasan di atas, dapat disimpulkan bahwa basis data adalah himpunan kelompok data yang dibentuk dengan hubungan/relasi yang logis serta diorganisasikan sehingga dapat diakses dengan mudah dan cepat.

2.1.4 Pengertian Internet (*Interconnected Network*)

Internet merupakan jaringan luas yang menghubungkan komputer-komputer di seluruh dunia, dengan internet sebuah komputer bisa mengakses informasi atau data yang terdapat pada komputer lain di benua yang berbeda (Hidayatullah dan Kawistara dikutip Ayu dan Permatasari, 2018:19).

Menurut Pibriana dan Ricoida (2017:105), “Internet adalah penghubung antara organisasi dan pelanggannya, sehingga tercipta sebuah organisasi baru secara *visual*”.

Dari kedua pendapat di atas dapat disimpulkan bahwa internet merupakan jarring luas yang menghubungkan komputer-komputer dunia sehingga tercipta sebuah organisasi baru secara visual.



2.1.5 Metode Pengembangan Aplikasi

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

Juansyah (2015:2), “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”.

Sedangkan Solichin (2016:1) “Aplikasi atau perangkat lunak (*software*) merupakan bagian yang tidak terpisahkan dari suatu *system computer*, disamping keberadaan, pengguna (*brainware*), perangkat keras (*hardware*) dan jaringan (*networking*)”.

Zaki dan *Community* (Suhimarita dan Susianto, 2019:24), “Aplikasi merupakan komponen yang bermanfaat sebagai media untuk menjalankan pengolahan data ataupun berbagai kegiatan lainnya seperti pembuatan ataupun pengolahan dokumen dan file”.

Jadi, dari definisi di atas dapat disimpulkan bahwa, aplikasi adalah program atau komponen yang siap untuk digunakan tidak terpisahkan pada sistem komputer dengan tujuan melaksanakan suatu fungsi untuk mencapai suatu tujuan .

2.2.2 Pengertian Manajemen

Elbadiansyah (2018:1), manajemen merupakan suatu proses dalam rangka mencapai tujuan dengan bekerjasama melalui orang-orang dan sumber daya organisasi lainnya.

2.2.3 Pengertian Event

Rosyida (2018:8), *event* merupakan sebuah kegiatan yang dapat berupa budaya, pentas seni, carnival, tradisi dan lain sebagainya yang melibatkan lingkungan masyarakat didaerah yang menyelenggarakan *event* tersebut untuk melestarika kebudayaan turun menurun bahkan untuk menarik wisatawan baik dari dalam maupun luar negeri.



2.2.4 Pengertian Android

Menurut Juansyah (2015:2), “Android adalah sebuah sistem operasi perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi.”.

Dewanti dan Permana (2017:396), menjelaskan mengenai pengertian android sebagai berikut:

“Android adalah sebuah sistem operasi pada *handphone* yang bersifat terbuka dan berbasis pada sistem operasi Linux. Android bisa digunakan oleh setiap orang yang ingin menggunakannya pada perangkat mereka. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang digunakan untuk bermacam peranti bergerak.”

Android merupakan sistem operasi yang paling diminati di masyarakat karena memiliki kelebihan seperti sifat *open source* yang memberikan kebebasan para pengembang untuk menciptakan aplikasi (Anggaraeni dalam Astuti et al, 2017:59).

Dari beberapa penjelasan di atas, dapat disimpulkan bahwa android adalah sistem operasi perangkat *mobile* berbasis linux yang bersifat *open source* sehingga dapat dimanfaatkan pengembang untuk membuat aplikasi.

2.2.5 Pengeritan Aplikasi Manajemen *Event* pada My Office Co-working Space Berbasis Android

Aplikasi Manajemen *Event* pada My Office Co-working Space adalah Aplikasi yang dibuat untuk membantu My Office Co-working Space dalam mengelola *event* dengan aplikasi berbasis android yang disediakan melalui jalur koneksi internet.

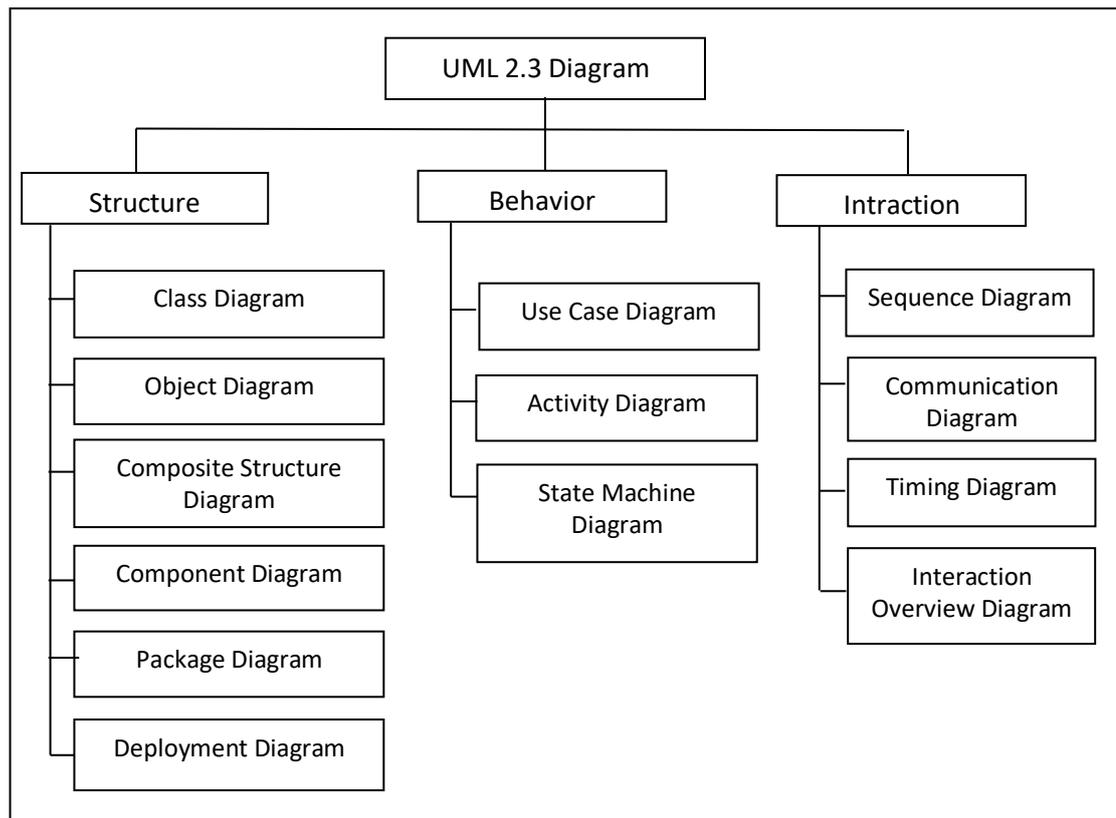


2.3 Teori Khusus

2.3.1 Pengertian UML (*Unified Modeling Language*)

Menurut Gata dan Grace (dikutip Hendini 2016:108), “*Unified Modeling Language* (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membanngun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem”.

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:



Sumber : Sukamto dan Shalahuddin (2018:140)

Gambar 2.1 Macam-macam Diagram UML



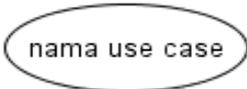
Penjelasan singkat dari pembagian kategori pada diagram UML menurut Sukamto dan Shalahuddin (2016:141) :

- 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.

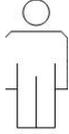
2.3.2 Pengertian Use Case Diagram

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 sistem”. Adapun simbol-simbol yang digunakan dalam *use case* adalah sebagai berikut:

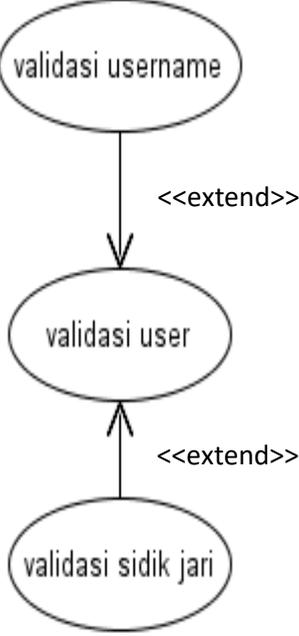
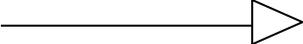
Tabel 2.1 Simbol-simbol pada *Use case* Diagram

Simbol	Deskripsi
<p style="text-align: center;"><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>

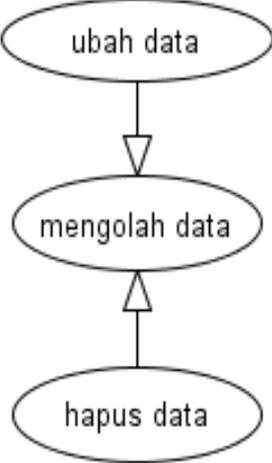
Lanjutan **Tabel 2.1** Simbol-simbol pada *Use case* Diagram

Simbol	Deskripsi
<p data-bbox="453 472 616 501">aktor / <i>actor</i></p> 	<p data-bbox="759 517 1362 927">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 aktor</p>
<p data-bbox="395 999 671 1028">asosiasi / <i>association</i></p> 	<p data-bbox="759 1010 1299 1099">komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i>.</p>
<p data-bbox="427 1144 639 1173">ekstensi / <i>extend</i></p> 	<p data-bbox="759 1144 1362 1554">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 pada *Use case* Diagram

Simbol	Deskripsi
	 <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>
<p>Generalisasi / <i>generalization</i></p> 	<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, misalnya:</p>

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

Simbol	Deskripsi
	 <p>arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)</p>
<p>menggunakan / <i>include</i> / <i>uses</i></p> <p><code><<include>></code> </p> <p><code><<uses>></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</p> <p>ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i>:</p> <ul style="list-style-type: none"> • <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:

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

Simbol	Deskripsi
	<div data-bbox="911 488 1198 719" data-label="Diagram"> <pre> graph BT login((login)) -- "<<include>>" --> validasi_username((validasi username)) </pre> </div> <p data-bbox="759 763 1369 1016"><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:</p> <div data-bbox="954 1066 1198 1294" data-label="Diagram"> <pre> graph BT ubah_data((ubah data)) -- "<<include>>" --> validasi_user((validasi user)) </pre> </div> <p data-bbox="759 1339 1369 1541">kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>

Sumber : Sukamto dan Shalahuddin (2018:156-158)



2.3.3 Pengertian *Activity Diagram*

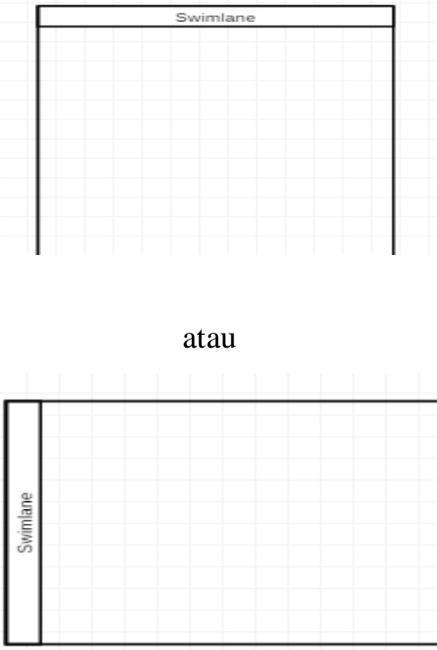
Sukanto dan Shalahuddin (2018:161), menjelaskan tentang *activity diagram* sebagai berikut :

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 2.2 Simbol-simbol pada *Activity Diagram*

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
Aktivitas aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan / <i>decision</i> 	Asosiasi percabangan di mana jika ada pilihan aktivitas lebih dari satu
Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir

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

Simbol	Deskripsi
<p data-bbox="533 450 660 479"><i>Swimlane</i></p>  <p data-bbox="568 835 625 864">atau</p>	<p data-bbox="884 450 1334 591">Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi</p>

Sumber : Sukamto dan Shalahuddin (2018:162-163)

2.3.4 Pengertian *Class Diagram*

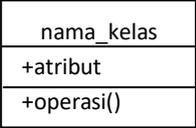
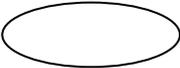
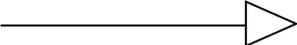
Sukamto dan Shalahuddin (2018:141), menjelaskan tentang *class diagram* sebagai berikut :

Class Diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. *Diagram Class* 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 2.3 Simbol-simbol pada *Class Diagram*

Simbol	Deskripsi
<p>kelas</p> 	Kelas pada struktur sistem
<p>antarmuka / <i>interface</i></p>  <p>nama_interface</p>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
<p>asosiasi / <i>association</i></p> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai <i>multiplicity</i>
<p>asosiasi berarah / <i>directed association</i></p> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
<p>generalisasi</p> 	Relasi antarkelas dengan makna generalisasi – spesialisasi (umum khusus)
<p>kebergantungan / <i>dependency</i></p> 	Relasi antarkelas dengan makna kebergantungan antar kelas

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

Simbol	Deskripsi
agregasi / <i>aggregation</i> 	Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>)

Sumber : Sukamto dan Shalahuddin (2018:146-147)

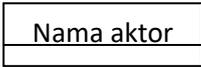
2.3.5 Pengertian *Sequence Diagram*

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

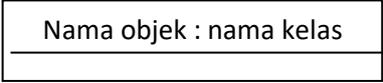
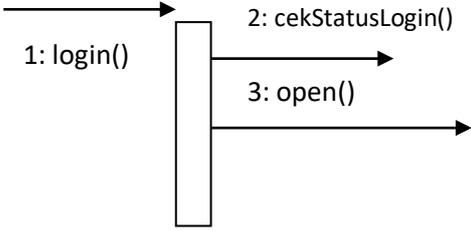
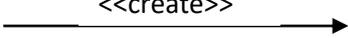
Diagram *sequence* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram *sequence* maka harus diketahui objek-objek yang terlibat dalam sebuah beserta metode-metode yang dimiliki kelas yang diinstansikan menjadi objek itu. Membuat diagram *sequence* juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

Banyaknya diagram *sequence* yang harus digambarkan adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksinya pesan sudah dicakup pada diagram *sequence* sehingga semakin banyak *use case* yang didefinisikan maka diagram *sequence* yang harus dibuat juga semakin banyak. Berikut simbol-simbol pada *Sequence Diagram* :

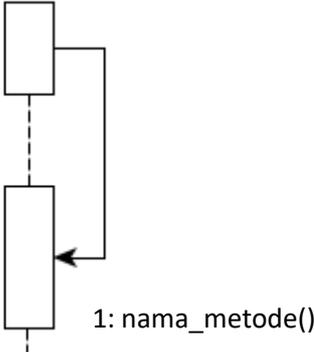
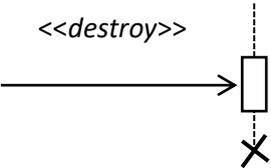
Tabel 2.4 Simbol-simbol pada *Sequence Diagram*

Simbol	Deskripsi
Actor  nama aktor atau  Nama aktor tanpa waktu aktif	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 aktor

Lanjutan Tabel 2.4 Simbol-simbol pada *Sequence Diagram*

Simbol	Deskripsi
Garis hidup / <i>lifeline</i> 	menyatakan kehidupan suatu objek
Objek 	menyatakan objek yang berinteraksi pesan
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 <code>cekStatusLogin ()</code> dan <code>open()</code> dilakukan di dalam metode <code>login()</code> aktor tidak memiliki waktu aktif
Pesan tipe <i>create</i> 	menyatakan suatu objek membuat objek yang lain, arah panah mengarah kepada objek yang dibuat

Lanjutan Tabel 2.4 Simbol-simbol pada *Sequence Diagram*

Simbol	Deskripsi
Pesan tipe <i>call</i> _ 1: nama_metode() →	menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,  arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus
	ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.
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
Pesan tipe <i>return</i> 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
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 Web

Menurut Hariyanto (dikutip Destianingrum dan Adrian, 2017:32), “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*)”

Web merupakan layanan yang paling banyak digunakan di internet untuk menyampaikan informasi karena sifatnya mendukung multimedia tidak hanya disampaikan melalui teks, tapi juga gambar, video dan suara (Swara dan Pebriadi, 2016:32).

Dari kedua pendapat di atas web dapat diartikan sebagai kumpulan halaman yang sifatnya mendukung multimedia yaitu tidak hanya disampaikan melalui teks, namun juga melalui gambar, animasi, suara, video bahkan gabungan dari semuanya baik bersifat statis maupun dinamis.

2.4.2 Pengertian HTML (*Hyper Text Markup Language*)

Kustiyahningsih dan Anamisa (dikutip Sasongko, 2017:147), Menjelaskan bahwa HTML kependekan dari *Hyper Text Markup Language* yang *file* teks murni yang dapat dibuat dengan editor teks sembarang. Dokumen ini dikenal sebagai web *page*. *File-file* HTML ini berisi instruksi-instruksi yang kemudian diterjemahkan oleh browser yang ada dikomputer *client (user)* sehingga isi formasinya dapat ditampilkan secara visual dikomputer pengguna (*user*).

Hidayatulah dan Kawistara (dikutip Ayu dan Permatasari, 2018:20), menjelaskan tentang HTML sebagai berikut :

Hypertext Markup Language (HTML) adalah Bahasa standard yang digunakan untuk menampilkan halaman web. Yang bisa dilakukan dengan HTML yaitu:

1. Mengatur tampilan dan halaman web dan isinya.
 2. Membuat tabel dalam halaman web.
 3. Mempublikasikan dalam halam web secara *online*.
-



4. Membuat *form* yang bisa digunakan untuk menangani registrasi dan transaksi via web.

Contoh: Setiap dokumen HTML diawali dan diakhiri dengan tag HTML.

Dari penjelasan di atas dapat disimpulkan bahwa *Hyper Text Markup Language* (HTML) adalah bahasa standar yang merupakan teks murni yang berisi instruksi yang diterjemahkan oleh browser dan ditampilkan secara *visual* di komputer pengguna.

2.4.3 Pengertian PHP (*Hypertext Preprocessor*)

Menurut Madcoms (dikutip Ayu dan Permatasari, 2018:20) “PHP (*Hypertext Preprocessor*) adalah bahasa *script* yang dapat ditanamkan atau disisipkan ke dalam HTML. PHP banyak dipakai untuk membuat program situs web dinamis”.

Sedangkan Menurut Hastanti et al (2015:4), “PHP adalah bahasa pemrograman skrip sederhana yang digunakan untuk pemrosesan HTML form di dalam halaman web. Strukturnya sangat sederhana sehingga PHP dapat dengan mudah dipelajari *programmer* pemula bahkan orang tanpa latar belakang teknologi informasi”.

Dari kedua pendapat tersebut dapat disimpulkan bahwa *Hypertext Preprocessor*) adalah bahasa pemrograman dengan skrip yang sederhana yang dapat ditanamkan, disisipkan atau digunakan ke dalam HTML. PHP biasanya digunakan untuk membuat web dinamis.

Nugroho (dikutip 2018:74), menjelaskan tentang beberapa kelebihan dari PHP sebagai berikut :

Beberapa kelebihan PHP dari bahasa pemrograman web, antara lain:

1. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. PHP memiliki tingkat akses yang lebih cepat.
3. PHP memiliki tingkat *lifecycle* yang cepat sehingga selalu mengikuti perkembangan teknologi internet.



4. PHP juga mendukung akses ke beberapa database yang sudah ada baik yang bersifat *free/gratis* ataupun komersial. *Database* itu antara lain : MySQL, PostgreSQL, infomix, dan MicrosoftSQL Server. Web server yang mendukung PHP dapat ditemukan dimana mana dari mulai Apache, IIS, AOservlet, phttp. Fhttp. PWS, Lighttpd hingga Xitami dengan konfigurasi yang *relative* mudah.

2.4.4 Pengertian CSS (*Cascading Style Sheet*)

Menurut Saputra dan Agustin (dikutip Sagita dan Sugiarto 2016:51), “CSS atau yang memiliki kepanjangan *Cascading Style Sheet*, merupakan suatu bahasa pemrograman web yang digunakan untuk mengendalikan dan membangun berbagai komponen dalam web sehingga tampilan web akan lebih rapi, terstruktur, dan seragam”.

Sedangkan Djaelangkara et al (2015:88), menjelaskan bahwa *Cascading Style Sheet* (CSS) adalah aturan untuk mengendalikan beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam. Pada umumnya CSS dipakai untuk memformat tampilan halaman web yang dibuat dengan bahasa HTML dan XHTML.

Dari kedua penjelasan di atas dapat disimpulkan bahwa *Cascading Style Sheet* (CSS) adalah sebuah bahasa pemrograman web atau aturan untuk mengendalikan dan membangun beberapa komponen dalam sebuah web agar terstruktur, lebih rapih dan seragam.

2.4.5 Pengertian Javascript

Menurut Hidayatullah dan Kawistara (dikutip Ayu dan Permatasari, 2018:21), “*JavaScript (js)* ialah suatu bahasa *scripting* yang digunakan sebagai fungsionalitas dalam membuat suatu web”.



Kustiyahningsih dan Anamisa (2016:74), menjelaskan tentang *JavaScript* sebagai berikut :

“*JavaScript* adalah bahasa yang berbentuk kumpulan skrip berjalan pada suatu dokumen HTML. Bahasa ini adalah bahasa pemrograman untuk memberikan kemampuan tambahan terhadap HTML dengan mengizinkan pengeksekusian perintah-perintah disisi *user* variabel atau fungsi dengan nama *TEST* berbeda dengan variabel dengan nama *test* dan setiap instruksi diakhiri dengan artinya disisi browser bukan disisi server web. *JavaScript* adalah bahasa yang “*case sensitive*” artinya membedakan penamaan variabel dan fungsi yang menggunakan huruf besar dan huruf kecil, contoh karakter titik koma”.

Dari kedua pendapat di atas, maka dapat disimpulkan bahwa *javascript* adalah suatu bahasa yang berbentuk kumpulan skrip yang merupakan kemampuan tambahan sebagai fungsionalitas dalam membuat suatu web.

2.4.6 Pengertian *Sublime Text*

Menurut Bos (dikutip Pahlevi et al, 2018:29), “*Sublime Text* merupakan salah satu *text editor* yang sangat *powerful* yang dapat meningkatkan produktivitas dan mengembangkan kualitas kode yang tinggi”.

Sedangkan menurut Annisak et al (2017:2), “*Sublime text* adalah teks *editor* berbasis Python, sebuah teks *editor* yang elegan, kaya fitur, *cross platform*, mudah dan *simple* yang cukup terkenal di kalangan *developer* (pengembang), penulis *source code* dan *desaigner*”.

Berdasarkan kedua pendapat di atas, dapat disimpulkan bahwa *sublime text* merupakan sebuah teks *editor* berbasis *Python* yang sangat *powerful*, elegan, kaya fitur, *cross platform*, mudah dan *simple* yang dapat meningkatkan produktivitas dan mengembangkan kualitas kode yang tinggi dan berguna bagi kalangan pengembang, penulis *source code* dan *desaigner*.



2.4.7 Pengertian MySQL

Kurniawan (dikutip Pahlevi et al, 2018:28), menjelaskan mengenai MySQL sebagai berikut:

“MySQL adalah salah satu jenis *database* yang banyak digunakan untuk membuat aplikasi berbasis web yang dinamis. MySQL termasuk jenis RDBMS (*Relational Database Management System*). MySQL ini mendukung Bahasa pemrograman PHP. MySQL juga mempunyai *query* atau bahasa SQL (*Structured Query Language*) yang *simple* dan menggunakan *escape character* yang sama dengan PHP”.

Sedangkan Lavarino dan Yustanti (2016:73), “MySQL (*MY Structure Query Language*) adalah salah satu *Database Management System* (*DBMS*) dari sekian banyak *DBMS* seperti Oracle, MS SQL, Postagre SQL, dan lainnya. MySQL berfungsi untuk mengolah Basis Data menggunakan bahasa SQL. MySQL bersifat *open source* sehingga kita bisa menggunakannya secara gratis”.

Berdasarkan penjelasan di atas, dapat disimpulkan bahwa MySQL merupakan salah satu jenis *Database Management System* (*DBMS*) yang digunakan untuk membuat aplikasi berbasis web dinamis yang bersifat *open source*.

2.4.8 Pengertian XAMPP

Madcoms (dikutip Ayu dan Permatasari, 2018:19), “XAMPP adalah sebuah paket kumpulan *software* yang terdiri dari *Apache*, *MySql*, *PhpMyAdmin*, *PHP*, *Perl*, *Filezilla* dan lain”.

Sidik (dikutip Suhimarita dan Susianto, 2019:25), menyatakan bahwa Xampp (X(Windows/Linux) Apache MySQL PHP dan Perl) merupakan paket *server* web PHP dan *database* MySQL yang paling populer dikalangan pengembang web dengan menggunakan PHP dan MySQL sebagai *databasenya*.

Berdasarkan penjelasan di atas, dapat disimpulkan bahwa XAMPP merupakan sebuah paket kumpulan aplikasi server web PHP dan *database* MySQL.



2.4.9 Pengertian Apache Cordova

Tompoh et al (2016:2), “*Apache Cordova* atau disebut saja cordova adalah *open-source framework* untuk pengembangan aplikasi *mobile*. Yang memungkinkan pengembang menggunakan teknologi web standar seperti HTML5, CSS3, dan *JavaScript* untuk pengembangan *cross platform*”.

Harmadya et al (2015:112), “*Apache Cordova* adalah *framework* untuk membuat aplikasi pada berbagai macam *platform* seperti Android, Blackberry, Iphone atau Windows Phone menggunakan HTML5, JQuery, JQuery Mobile dan CSS3.

Dari kedua pendapat di atas dapat ditarik kesimpulan bahwa *apache cordova* adalah *framework* yang bersifat *open source* untuk membuat aplikasi pada berbagai macam *platform (cross platform)*.