



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Komputer

Menurut Harmayani, dkk (2021:2), menyatakan bahwa “Komputer adalah alat elektronik yang terdiri dari rangkaian berbagai komponen yang saling terhubung sehingga membentuk suatu sistem kerja”.

Menurut oleh Fauzi (2018:1), bahwa “Komputer adalah suatu peralatan elektronik yang dapat menerima input, mengolah *input (processing)*, memberikan informasi dengan menggunakan suatu program yang tersimpan di memori komputer dan dapat menyimpan program dari hasil pengolahan yang bekerja secara otomatis”.

Berdasarkan pendapat diatas, dapat disimpulkan bahwa komputer merupakan serangkaian alat elektronik yang saling bekerja sama mengolah informasi sehingga dapat menghasilkan keluaran yang sesuai perintah, rapi, dan teliti untuk memudahkan berbagai kegiatan manusia.

2.1.2 Pengertian Perangkat Lunak

Menurut Rianto (2021:5), menjelaskan bahwa " *Software* (Perangkat lunak) merupakan suatu data yang diprogram sedemikian rupa dan disimpan dalam bentuk digital yang tidak terlihat secara fisik tetapi tersimpan dalam media penyimpanan komputer”. Sementara menurut Risanto (2017:89), berpendapat bahwa “Perangkat lunak (*software*) adalah sekumpulan data elektronik yang disimpan dan diatur oleh komputer, data elektronik yang disimpan oleh komputer itu dapat berupa program atau intruksi yang akan menjalankan”.

Berdasarkan pendapat diatas, dapat disimpulkan bahwa perangkat lunak merupakan suatu data dalam berupa program yang disimpan di penyimpanan komputer.



2.1.3 Pengertian Perangkat Keras

Menurut Sudirman, dkk (2020:38), menjelaskan bahwa “Perangkat keras merupakan komponen atau elemen sistem informasi yang dapat terlihat dengan nyata. Perangkat keras merupakan unsur penting guna menunjang kinerja sistem informasi”.

2.1.4 Pengertian Data

Menurut oleh Pane, dkk (2020:7) bahwa “Data merupakan sekumpulan keterangan atau fakta mentah berupa simbol, angka, kata-kata, atau citra, yang didapatkan melalui proses pengamatan atau pencarian ke sumber-sumber tertentu”. Sementara menurut Pamungkas (2017:1) menjelaskan bahwa “Data merupakan nilai yang mempresentasikan deskripsi dari suatu objek atau kejadian”.

Berdasarkan pendapat di atas, dapat disimpulkan bahwa data merupakan sekumpulan fakta berupa simbol, angka, kata-kata atau citra yang didapatkan melalui pengukuran atau pengamatan.

2.1.5. Pengertian Basis Data (*Database*)

Menurut Risanto (2018:2), menyatakan bahwa “Basis data merupakan sekumpulan data yang terintegrasi, yang diorganisasi untuk memenuhi kebutuhan para pemakai di dalam suatu organisasi”.

Menurut Pamungkas (2017:2), menegaskan “Basis data merupakan kumpulan dari data terhubungan bersama-sama pada suatu media, yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, dan dengan software untuk melakukan manipulasi untuk kegunaan tertentu

Dari pendapat diatas, dapat disimpulkan bahwa basis data merupakan kumpulan data yang saling berhubungan untuk memenuhi informasi yang tersimpan di penyimpanan komputer secara sistematis agar program komputer dapat terus mengolahnya.



2.2 Pengertian Judul

2.2.1 Pengertian Aplikasi

Menurut oleh Pane, dkk (2020:53), menyatakan bahwa “Aplikasi adalah suatu perangkat lunak (*software*) atau program komputer yang beroperasi pada sistem tertentu yang diciptakan dan dikembangkan untuk melakukan perintah tertentu”.

Menurut Habibi dan Karnovi (2020:14), berpendapat bahwa “Aplikasi adalah sebuah program siap pakai yang bisa dipakai untuk menjalankan sejumlah perintah dari pengguna aplikasi itu sendiri”.

Berdasarkan pendapat diatas, dapat disimpulkan bahwa aplikasi merupakan suatu program yang beroperasi pada sistem tertentu untuk melaksanakan perintah pengguna dengan tujuan dan manfaat tertentu.

2.2.2 Pengertian Absensi

Menurut Fitrianto (dalam Habibi, dkk 2019:1), menyatakan bahwa “Absensi adalah suatu kegiatan yang bertujuan untuk mengetahui tingkat kedisiplinan dari anggota dalam suatu instansi, instansi atau perusahaan”.

Menurut Haqi dan Setiawan (2019:104), menegaskan “Absensi adalah dokumen yang mencatat jam hadir setiap karyawan di perusahaan, dapat berupa daftar hadir biasa, dapat pula berbentuk kartu hadir yang diisi dengan mesin pencatat waktu”.

Berdasarkan pendapat diatas, dapat disimpulkan bahwa absensi merupakan dokumen atau data yang berupa daftar kehadiran biasa digunakan oleh lembaga atau instansi untuk kebutuhan tertentu.

2.2.3 Pengertian Pegawai

Menurut Soedaryono (dalam Pujiasri dan Budiningsih, 2018:4), berpendapat bahwa “Pegawai adalah seseorang yang melakukan penghidupannya dengan bekerja dalam kesatuan organisasi, baik kesatuan kerja pemerintah maupun kesatuan kerja swasta”. Sementara menurut Pujiasri dan Budiningsih (2018:4), menjelaskan bahwa “Pegawai adalah orang pribadi yang bekerja pada pemberi



kerja, baik sebagai pegawai tetap atau tidak, berdasarkan kesepakatan kerja baik tertulis maupun tidak tertulis, untuk melaksanakan suatu pekerjaan dalam jabatan atau kegiatan tertentu yang ditetapkan oleh pemberi kerja”.

Berdasarkan pendapat di atas, dapat disimpulkan bahwa pegawai merupakan seseorang yang bekerja di dalam instansi baik pemerintah maupun kerja swasta yang tetap ataupun tidak dan melakukan kegiatan tertentu sesuai ketentuan instansi tersebut.

2.2.4 Pengertian Website

Menurut Karnovi, dkk (2020:19) “*Website* atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara dan atau gabungan dari semuanya, baik bersifat statis maupun dinamis yang membentuk suatu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman”.

Menurut Abdulloh (2018:1), *Website* dapat diartikan sebagai kumpulan halaman yang berisi informasi data digital baik berupa teks, gambar, animasi, suara dan video atau gabungan dari semuanya yang disediakan melalui jalur koneksi internet sehingga dapat diakses dan dilihat oleh semua orang di seluruh dunia. Halaman *website* dibuat menggunakan bahasa standard yaitu *HTML*. Skrip *HTML* ini akan diterjemahkan oleh *web browser* sehingga dapat ditampilkan dalam bentuk informasi yang dapat dibaca oleh semua orang.

Dari kedua definisi di atas dapat penulis simpulkan bahwa *website* adalah kumpulan halaman yang saling terkait dimana didalamnya memuat konten berupa teks, gambar, animasi atau suara yang dapat diakses pengguna melalui jaringan internet.



2.2.5 Aplikasi Absensi Pegawai pada Kantor Badan Pusat Statistik Kota Palembang Berbasis Website

Aplikasi Absensi Pegawai pada Kantor BPS Kota Palembang Berbasis Website digunakan untuk mengisi daftar kehadiran pegawai, melihat data pegawai dan juga rekap absensi.

2.3 Teori Khusus

2.3.1 Kamus Data

Kamus Data merupakan katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Menurut Rusmawan (:36), mendefinisikan bahwa “Kamus data merupakan katalog fakta data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi”. Kamus data mendefinisikan bahwa elemen data dengan fungsi sebagai berikut:

1. Menjelaskan arti aliran data dan penyimpanan dalam DFD.
2. Mendeskripsikan komposisi paket data yang bergerak melalui aliran.
3. Mendeskripsikan nilai dan satuan yang relevan bagian penyimpanan dan aliran.
4. Mendeskripsikan hubungan antar penyimpanan yang akan menjadi titik perhatian dalam *entity relationship* diagram.

Kamus data memiliki beberapa simbol untuk menjelaskan informasi tambahan sebagai berikut :

Tabel 2.1 Simbol-Simbol pada Kamus Data

No.	Simbol	Keterangan
1.	=	disusun atau terdiri dari
2.	+	Dan
3.	[]	baik ...atau...
4.	{ } ⁿ	n kali diulang/ bernilai banyak
5.	()	data opsional
6.	*...*	batas komentar

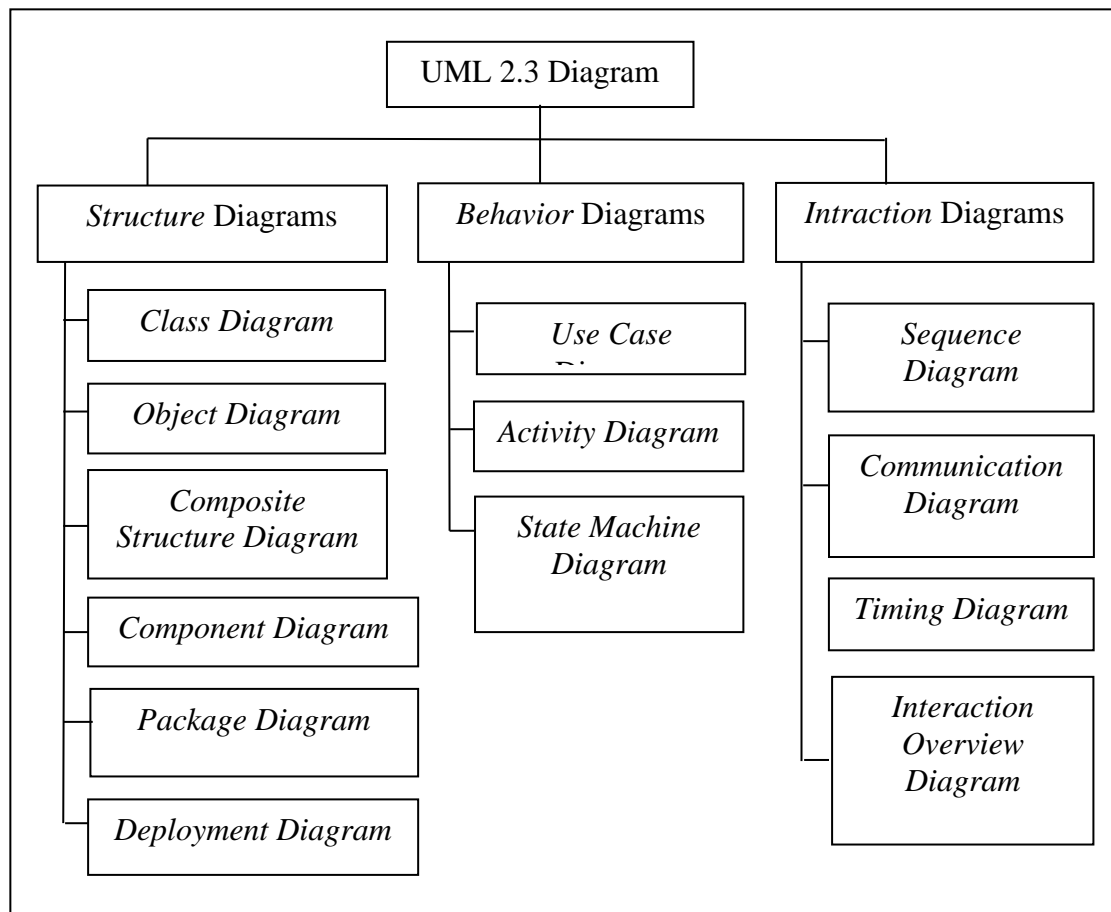
(Sumber : Sukamto dan Shalahuddin, 2016:74)



2.3.2 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 membangun 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



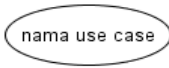
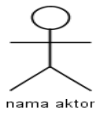
Penjelasan singkat dari pembagian kategori pada diagram UML menurut Sukamto dan Shalahuddin (2016:141) :

- 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.

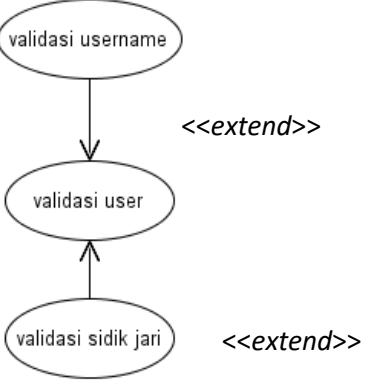
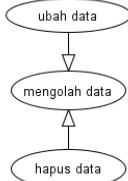
2.2.3 Pengertian Use Case Diagram

Menurut Sukamto dan Shalahuddin (2018:155), menjelaskan tentang “*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:

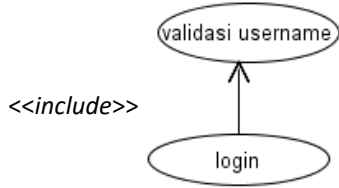
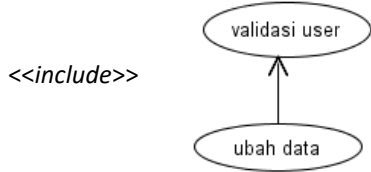
Tabel 2.2 Simbol-simbol pada *Use case* Diagram

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 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 aktor.</p>



<p>3</p>	<p>asosiasi / <u>association</u></p>	<p>komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i>.</p>
<p>4</p>	<p>ekstensi / <i>extend</i> <<extend>> ></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>  <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>5</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> 



		arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)
6	<p>menggunakan / <i>include / uses</i></p> <p><u><<include>></u> →</p> <p>→ <<uses>></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 include di <i>use case</i>:</p> <p><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:</p>  <pre> graph BT login((login)) -- <<include>> --> validasi_username((validasi username)) </pre> <p><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>  <pre> graph BT ubah_data((ubah data)) -- <<include>> --> validasi_user((validasi user)) </pre> <p>kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>





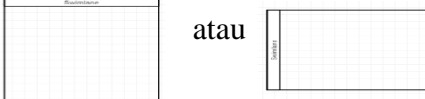
(Sumber : Sukamto dan Shalahuddin ,2018:156-158)



2.2.3 Pengertian *Activity Diagram*

Menurut Sukamto dan Shalahuddin (2018:161), menjelaskan tentang “*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 pada *Activity Diagram*

No	Simbol	Deskripsi
1	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
2	Aktivitas 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> atau 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

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

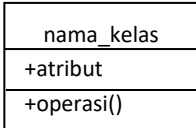



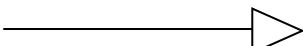


2.2.4 Pengertian *Class Diagram*

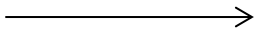
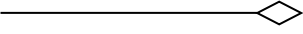
Menurut Sukamto dan Shalahuddin (2018:141), menjelaskan tentang “*Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. *Diagram Class* dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron”.

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

Tabel 2.4 Simbol-simbol pada *Class Diagram*

No	Simbol	Deskripsi
1	<p>kelas</p> 	Kelas pada struktur sistem
2	<p>antarmuka / <i>interface</i></p>  <p>nama_interface</p>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
3	<p>asosiasi / <i>association</i></p> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai <i>multiplicity</i>
4	<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>
5	<p>generalisasi</p> 	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>)

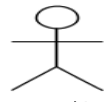
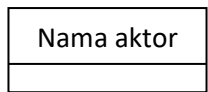
(Sumber : Sukamto dan Shalahuddin ,2018:146-147)

2.2.5 Pengertian *Sequence Diagram*

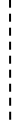

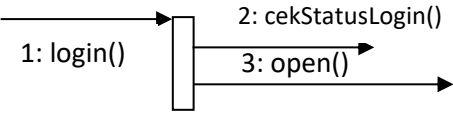


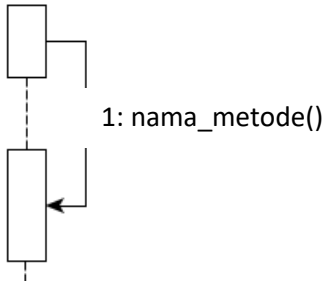
Menurut Sukamto dan Shalahuddin (2018:141), menjelaskan tentang “Diagram *sequence* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram *sequence* maka harus diketahui objek-objek 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 interaksi jalannya 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* :


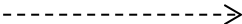
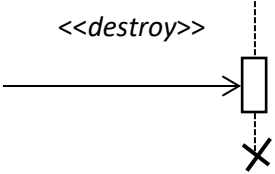
Tabel 2.5 Simbol-simbol pada *Sequence Diagram*

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 aktor</p>



2	<p>Garis hidup / <i>lifeline</i></p> 	menyatakan kehidupan suatu objek
3	<p>Objek</p> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 5px auto;"> <p>Nama objek : nama kelas</p> </div>	menyatakan objek yang berinteraksi pesan
4	<p>Waktu aktif</p> 	<p>menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya</p>  <p>maka cekStatusLogin () dan open() dilakukan di dalam metode login() aktor tidak memiliki waktu aktif</p>
5	<p>Pesan tipe <i>create</i></p> <p style="text-align: center;"><code><<create>></code></p> 	menyatakan suatu objek membuat objek yang lain, arah panah mengarah kepada objek yang dibuat
6	<p>Pesan tipe <i>call</i></p> <p>1 : nama_metode()</p> 	<p>menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,</p> 



		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	<p>Pesan tipe <i>send</i> 1: masukan</p> 	menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
8	<p>Pesan tipe <i>return</i> 1: keluaran</p> 	menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
9	<p>Pesan tipe <i>destroy</i></p> 	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 PHP

Menurut Enterprise (2019:1) menyatakan bahwa “*PHP* merupakan bahasa pemrograman yang digunakan untuk membuat aplikasi berbasis *website*”. Sedangkan interaktif artinya *PHP* dapat memberi *feedback* bagi *user*. Sedangkan menurut Rusli, Ahmar dan Rahman (2019:63), “*PHP* adalah bahasa *scripting server* alat yang ampuh untuk membuat halaman web dinamis dan interaktif”.



Berdasarkan definisi diatas, dapat disimpulkan bahwa PHP adalah bahasa pemrograman yang digunakan untuk membuat aplikasi dengan halaman yang dinamis dan interaktif.

2.4.2 Pengertian MySQL

Menurut Fitri (2020:2), “*MySQL* merupakan *database engine* atau *database server* yang mendukung bahasa *database SQL* sebagai bahasa interaktif dalam mengelola data”. Sedangkan Setyawan dan Pratiwi (2020:52) mengungkapkan bahwa, *MySQL* merupakan sebuah *database management system* (manajemen basis data) menggunakan perintah dasar *SQL (Structured Query Language)* yang cukup terkenal”.

Berdasarkan beberapa definisi diatas, dapat disimpulkan bahwa *MySQL* adalah sebuah *database* sistem manajemen basis data *SQL*.

2.4.3 Pengertian XAMPP

Menurut Habibi dan Suryanah (2020:53), “*XAMPP* adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program.”. Sedangkan menurut Haqi (2019:8), “*XAMPP* adalah perangkat lunak (*free software*) bebas, yang mendukung untuk banyak sistem operasi, yang merupakan kompilasi dari beberapa program”. Berdasarkan beberapa definisi di atas dapat ditarik kesimpulan bahwa *XAMPP* adalah aplikasi *web server* yang dapat digunakan untuk membangun suatu program pada berbagai sistem operasi.

2.4.4 Pengertian PhpMyAdmin

Menurut Habibi (2019:42), “Fungsi *PhpMyAdmin* pada web hosting sesungguhnya, peran *PhpMyAdmin* dalam aplikasi *XAMPP* juga diplot sebagai pengatur konfigurasi *MySQL*”.



2.4.5 Pengertian Adobe Dreamweaver CS5

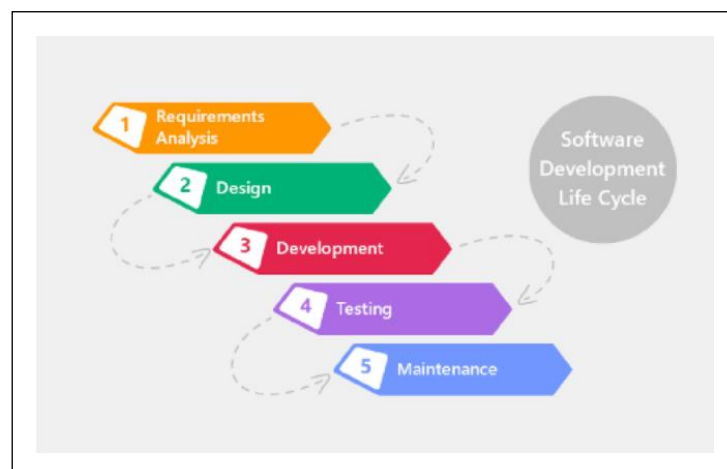
Menurut Rahmad dan Purnama (dalam Team, 2013:61), menegaskan bahwa “Adobe Dreamweaver CS5 merupakan salah satu program aplikasi yang digunakan untuk membangun sebuah *website*, baik secara grafis maupun dengan menuliskan kode sumber secara langsung”.

2.5 Metode Pengembangan Sistem

Metode pengembangan sistem sangat diperlukan sebagai suatu proses pengembangan sistem yang formal dan presisi yang memuat serangkaian aktivitas, metode, dan *tools* yang terotomatis dalam rangka mengembangkan sistem informasi atau *software* secara keseluruhan. Adapun metode pengembangan sistem yang digunakan dalam penyusunan laporan akhir ini adalah Metode *Waterfall*.

Menurut Pressman (dikutip Fathorini (2020:19), menegaskan bahwa Metode air terjun atau yang sering disebut metode *waterfall* sering dinamakan siklus hidup klasik (*classic life cycle*), nama model ini sebenarnya adalah “*Linear Sequential Model*”, dimana hal ini menggambarkan pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak, dimulai dengan spesifikasi kebutuhan pengguna lalu berlanjut melalui tahapan-tahapan perencanaan (*planning*).

Menurut Fathorini (2020:20-22), Tahapan-tahapan dalam metode pengembangan sistem *waterfall* adalah sebagai berikut :



Gambar 2.2 Tahapan-tahapan metode *waterfall*



Dalam pengembangannya metode *waterfall* memiliki beberapa tahapan yang berurut. Tahapan – tahapan dari metode *waterfall* adalah sebagai berikut:

1. *Requirement Analisis*

Tahap ini pengembang sistem yang diperlukan komunikasi yang bertujuan untuk memahami perangkat lunak yang diharapkan oleh pengguna dan batasan perangkat lunak tersebut.

2. *System Design*

Desain sistem membantu dalam menentukan perangkat keras (*hardware*) dan sistem persyaratan dan juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.

3. *Implementation*

Pada tahap ini, sistem pertama kali dikembangkan di program kecil yang disebut unit, yang terintegrasi dalam tahap selanjutnya. Setiap unit dikembangkan dan diuji untuk fungsionalitas yang disebut sebagai unit *testing*.

4. *Integration & Testing*

Seluruh unit yang dikembangkan dalam tahap implementasi diintegrasikan ke dalam sistem setelah pengujian yang dilakukan masing-masing unit. Setelah integrasi seluruh sistem diuji untuk mengecek setiap kegagalan maupun kesalahan.

5. *Operation & maintenance*

Tahap akhir dalam model *waterfall*. Perangkat lunak yang sudah jadi, dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada Langkah sebelumnya. Perbaikan implementasi unit sistem dan peningkatan jasa sistem sebagai kebutuhan baru.