



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Komputer

Menurut Sanders (2004:1), “Komputer adalah sistem elektronik untuk memanipulasi data yang cepat dan tepat serta dirancang dan diorganisasikan supaya secara otomatis menerima dan menyimpan data input, memprosesnya, dan menghasilkan output dibawah pengawasan suatu langkah-langkah instruksi-instruksi program yang tersimpan di memori (*stored program*)”.

Menurut Hamacher, et al (2004:2), “Komputer adalah mesin penghitung elektronik yang cepat dapat menerima informasi input digital, memprosesnya sesuai dengan suatu program yang tersimpan di memorinya (*stored program*) dan menghasilkan output informasi”.

Menurut Fouri (2004:2), “Komputer adalah suatu pemroses data (*data processor*) yang dapat melakukan perhitungan yang besar dan cepat, termasuk perhitungan aritmatika yang besar atau operasi logika tanpa campur tangan dari manusia mengoperasikan selama pemrosesan”.

Berdasarkan beberapa pengertian menurut para ahli di atas, penulis menyimpulkan bahwa Komputer adalah sistem elektronik yang dapat menerima dan menyimpan informasi digital yang besar secara cepat yang tersimpan di memori (*stored program*).

2.1.2 Pengertian Sistem

Menurut Romney dan Steinbart (2015) yang dikutip oleh Destiningrum (2017) Sistem adalah : “Serangkaian data atau lebih komponen yang saling terkait dan berinteraksi untuk mencapai tujuan”.

Menurut Jogiyanto (2005) yang dikutip oleh Destiningrum (2017) Sistem adalah : “Suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan,



berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu”.

Berdasarkan uraian di atas, penulis menyimpulkan bahwa Sistem adalah serangkaian data yang saling terhubung dan berinteraksi untuk melakukan suatu kegiatan guna mencapai tujuan tertentu.

2.1.3 Pengertian Informasi

Menurut Romney dan Steinbart (2015) yang dikutip oleh Destiningrum (2017) Informasi adalah : ”Data yang telah dikelola dan diproses untuk memberikan arti dan memperbaiki proses pengambilan keputusan”.

Menurut Ladjamuddin (2005) yang dikutip oleh Destiningrum (2017) Informasi adalah : “Informasi sebagai data yang telah diolah menjadi bentuk yang lebih berarti dan berguna bagi penerimanya untuk mengambil keputusan masa kini maupun yang akan datang”.

Berdasarkan uraian di atas, penulis menyimpulkan bahwa Informasi adalah data yang telah dioah dan diproses untuk memberikan arti bagi penerimanya dalam proses pengambilan keputusan di masa kini maupun masa yang akan datang.

2.1.4 Pengertian Sistem Informasi

Menurut Jogiyanto (2005) yang dikutip oleh Destiningrum (2017) Sistem Informasi adalah : “Suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan”.

Menurut Ladjamuddin (2005) yang dikutip oleh Destiningrum (2017) Sistem Informasi adalah : “Suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi yang mencapai suatu tujuan yaitu menyajikan informasi”.

Berdasarkan uraian di atas, penulis menyimpulkan bahwa Sistem Informasi adalah suatu sistem yang terdiri dari komponen-komponen dalam organisasi yang berguna untuk menyajikan informasi agar tercapainya suatu tujuan tertentu.



2.1.5 Pengertian Web

Dimuat dari jurnal Destiningrum dan Adrian (2017) bahwa menurut Abdulloh (2016) *Web* adalah : “Sekumpulan halaman yang terdiri dari beberapa halaman yang berisi informasi dalam bentuk data digital berupa text, gambar, video, audio, dan animasi lainnya yang disediakan melalui jalur koneksi internet”.

Dimuat dari jurnal Destiningrum dan Adrian (2017) bahwa menurut 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*)”.

Berdasarkan uraian di atas, penulis menyimpulkan bahwa *Web* adalah kumpulan halaman yang berisi informasi data digital berupa *text*, gambar, video, audio, dan gabungan dari semuanya yang bersifat statis dan dinamis yang dihubungkan dengan jaringan halaman (*hyperlink*) melalui koneksi internet.

2.2 Teori Khusus

2.2.1 Pengertian Object Oriented Programming (OOP)

Dimuat dari jurnal Pahlevi (2018) bahwa menurut Nugroho (2010), *Object Oriented Programming* (OOP) atau Metodologi Berorientasi Objek adalah : “Metodologi berorientasi objek sebagai suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya”.

2.2.2 Pengertian Unified Modelling Language (UML)

Dimuat dari jurnal Suendri (2018) bahwa menurut Windu dan Grade (2013:2), “*Unified Modelling 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”.



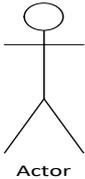
Dimuat dari jurnal Suendri (2018) Nugroho (2018:2), "*Unified Modelling Language (UML)* adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma berorientasi objek. Abstraksi konsep dasar UML terdiri dari *structural classification*, *dynamic behavior*, dan model *management* dapat kita pahami *main concepts* sebagai *term* 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*".

2.2.2.1 Pengertian Diagram Use Case

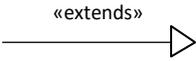
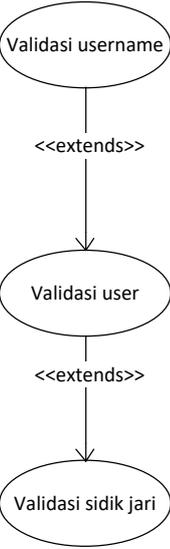
Diagram *Use Case* menunjukkan bagaimana sistem yang sedang dibangun digunakan untuk memenuhi kebutuhan pengguna. Sebuah *use case* menampilkan sebuah fungsionalitas yang diberikan oleh sistem. (Miles dan Hamilton, 2006:20) Yang ditekankan pada *use case* adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. (Dharwiyanti dan Wahono, 2003).

Berikut adalah simbol-simbol yang terdapat pada Diagram *Use Case*.

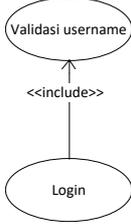
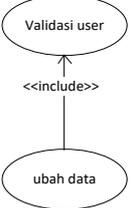
Tabel 2.1 Simbol-simbol pada Diagram *Use Case*

No.	Simbol	Deskripsi
1.		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i> .
2.		Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.

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

No.	Simbol	Deskripsi
3.	Assosiasi/ <i>association</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4.	Exstensi/ <i>extend</i> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu, mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek, biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misalnya:  <p>Arah panah mengarah pada <i>use case</i> yang ditambahkan, biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.</p>
5.	Generalisasi/ <i>generalization</i> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.

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

No.	Simbol	Deskripsi
6.	Menggunakan <i>include / uses</i> <<include>>  <<uses>> 	Relasi <i>use case</i> 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> : <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, misal pada kasus berikut: <div style="text-align: center; margin: 10px 0;">  <pre> graph BT Login((Login)) -- <<include>> --> ValidasiUsername((Validasi username)) </pre> </div> - <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 di jalankan, misal pada kasus berikut: <div style="text-align: center; margin: 10px 0;">  <pre> graph BT ubahData((ubah data)) -- <<include>> --> ValidasiUser((Validasi user)) </pre> </div> <p>Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>

Sumber: Rosa dan Shalahudin (2014:156).

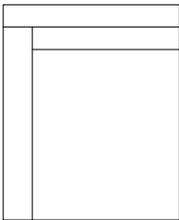


2.2.2.2 Pengertian Diagram Activity

Rosa dan Shalahudin (2014:161), diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Berikut adalah simbol-simbol yang terdapat pada Diagram Activity.

Tabel 2.2 Simbol-simbol pada Diagram Activity

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 dimana jika ada pilihan aktivitas lebih dari satu.
4.	Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.	Status akhir 	Status akhir yang dilakukan oleh sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.	<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Sumber: Rosa dan Shalahudin (2014:162).

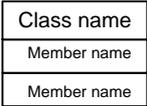


2.2.2.3 Pengertian Diagram Class

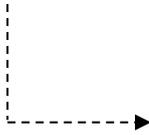
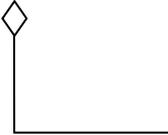
Diagram *Class* disebut sebagai bagian yang digunakan untuk membangun dan mendefinisikan objek-objek. Sehingga setiap hal yang dibangun dari sebuah *class* disebut sebagai objek atau *instances*. Deskripsi dari sebuah *class* mencakup dua bagian: informasi yang akan dimiliki oleh objek dan *behaviour* yang akan didukung objek tersebut. Ini yang membedakan *object-oriented* dengan yang lainnya, bahwa *state information* dan *behaviour* digabungkan dalam satu definisi *class*, dimana objek-objek dapat terbentuk dari *class* tersebut. (Miles dan Hamilton, 2006:64-65).

Berikut ini adalah simbol-simbol yang terdapat pada Diagram *Class*.

Tabel 2.3 Simbol-simbol pada Diagram *Class*

No.	Simbol	Deskripsi
1.	<p><i>Kelas</i></p> 	Kelas pada struktur sistem.
2.	<p>Antar muka/interface</p> 	Sama dengan konsep interface dalam pemrograman berorientasi objek.
3.	<p>Asosiasi / association</p> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity.
4.	<p>Asosiasi berarah / directed Association</p> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity.

Lanjutan Tabel 2.3 Simbol-simbol pada Diagram *Class*

No.	Simbol	Deskripsi
5.		Relasi antar kelas dengan makna generalisasi - spesialisasi (umum khusus).
6.		Relasi antar kelas dengan makna kebergantungan antar kelas.
7		Relasi antar kelas dengan makna semua-bagian (whole-part).

Sumber: Rosa dan Shalahudin (2014:162).

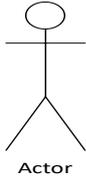
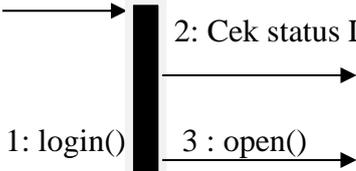
2.2.2.4 Pengertian Diagram *Sequence*

Diagram *Sequence* memodelkan bagaimana sistem yang sedang dibangun mendatangi tugasnya. Diagram ini menangkap semua urutan interaksi pada bagian-bagian dari sistem. Dengan menggunakan *sequence diagram*, pengembang bisa menjelaskan interaksi apa yang akan dipanggil ketika sebuah *use case* dieksekusi dan dalam urutan seperti apa interaksi-interaksi tersebut akan terjadi. (Miles dan Hamilton, 2006:108-109).

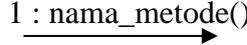
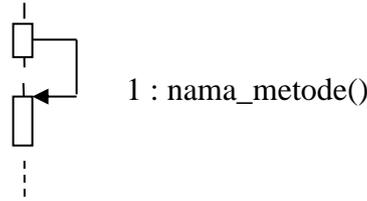
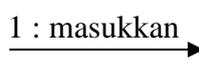
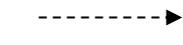
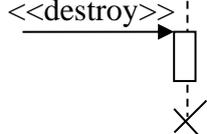


Berikut ini adalah simbol-simbol yang terdapat pada Diagram *Sequence*.

Tabel 2.4 Simbol-simbol pada Diagram *Sequence*

No.	Simbol	Deskripsi
1.	<p>Aktor</p>  <p>Atau</p>  <p>Tanpa waktu aktif</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan dalam menggunakan kata benda diawal frase nama aktor.</p>
2.	<p>Garis hidup/<i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek.</p>
3.	<p>Objek</p> 	<p>Menyatakan objek yang berinteraksi pesan.</p>
4.	<p>Waktuaktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya:</p>  <p>Maka cek StatusLogin() dan open() dilakukan didalam metode login(). Aktor tidak memiliki waktu aktif.</p>

Lanjutan Tabel 2.4 Simbol-simbol pada Diagram *Sequence*

No.	Simbol	Deskripsi
5.	Pesan tipe <i>create</i> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
6.	Pesan tipe <i>call</i> 	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> 	Menyatakan bahwa suatu objek mengirimkan data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8	Pesan tipe <i>return</i> 	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	Pesan tipe <i>destroy</i> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada create maka ada destroy.

Sumber: Rosa dan Shalahudin (2014:162).



2.2.3 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 tahapan-tahapan (fase) dalam metode pengembangan RUP menurut Sukamto dan Shalahuddin (2018:128-131) adalah sebagai berikut:

1. Inception (Pemulaan)

Tahap ini lebih pada memodelkan proses bisnis yang dibutuhkan (*business modelling*) dan mendefinisikan kebutuhan akan sistem yang akan dibuat (*requisites*).

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/tonggak kemampuan operasional awal. Aktivitas pada tahap ini termasuk pada pelatihan user, pemeliharaan dan pengujian sistem apakah sudah memenuhi harapan user.



2.3 Pengertian Judul

2.3.1 Pengertian Aplikasi

Indrajani (2018:3), “Aplikasi adalah program yang menentukan aktivitas pemrosesan informasi yang dibutuhkan untuk penyelesaian tugas-tugas khusus dari pemakai komputer.”

2.3.2 Pengertian Pelayanan

Dalam Kamus Bahasa Indonesia Daring (2016) dijelaskan bahwa :”Pelayanan adalah sebagai usaha untuk membantu menyiapkan atau mengurus apa yang diperlukan orang lain”.

2.3.3 Pengertian Online

Menurut Badan Pengembangan dan Pembinaan Bahasa Kementerian Pendidikan dan Kebudayaan menjelaskan bahwa : “Dalam bahasa Indonesia istilah *online* dipadankan menjadi dalam jaringan (daring), yaitu perangkat elektronik yang terhubung ke jaringan internet”.

2.3.4 Pengertian Kecamatan

Kecamatan diatur sesuai dengan ketentuan Pasal 1 angka 24 Undang-Undang Republik Indonesia Nomor 23 Tahun 2014 tentang Pemerintahan Daerah yang menyatakan bahwa “Kecamatan adalah bagian wilayah administratif di Indonesia dari daerah kabupaten/kota yang dipimpin oleh Camat”.

2.3.5 Pengertian Aplikasi Pelayanan Online pada Kantor Camat Jakabaring Kota Palembang

Aplikasi Pelayanan *Online* pada Kantor Camat Jakabaring Kota Palembang merupakan aplikasi yang berguna untuk memberikan kemudahan kepada pegawai bagian Pelayanan Umum Kantor Camat Jakabaring Kota Palembang dalam mengelola data administrasi masyarakat Kecamatan Jakabaring dalam memproses permohonan pelayanan masyarakat menjadi terkomputerisasi. Serta, Aplikasi Pelayanan *Online* pada Kantor Camat Jakabaring Kota Palembang ini dapat



mengurangi potensi kehilangan data dan kesalahan yang sering terjadi dalam mengolah data administrasi pelayanan masyarakat.

2.4 Teori Program

2.4.1 Pengertian PHP

Menurut Kurniawan (2010:2) yang dikutip dari jurnal Pahlevi (2018) menjelaskan bahwa, “PHP adalah bahasa pemrograman untuk dijalankan melalui halaman *web*, umumnya digunakan untuk mengolah informasi di internet. Sedangkan dalam pengertian lain PHP adalah singkatan dari *Hypertext Preprocessor* yaitu bahasa pemrograman *webserver-side* yang bersifat *open source* atau gratis. PHP merupakan *script* yang menyatu dengan HTML dan berada pada server”.

2.4.2 Pengertian HTML

Dimuat dari jurnal Pahlevi (2018) bahwa menurut Henderson (2009:232), “HTML (*HyperText Markup Language*) merupakan bahasa yang digunakan untuk mendeskripsikan struktur sebuah halaman *web*. HTML berfungsi untuk mempublikasikan dokumen *online*. *Statement* dasar dari HTML disebut *tags*. Sebuah *tag* dinyatakan dalam sebuah kurung siku (<>). *Tags* yang ditujukan untuk sebuah dokumen atau bagian dari suatu dokumen haruslah dibuat berupa pasangan. Terdiri dari *tag* pembuka dan *tag* penutup. Dimana *tag* penutup menggunakan tambahan tanda garis miring (/) di awal nama *tag*”.

Dalam buku karangan Jubilee Enterprise yang berjudul Pengenalan HTML dan CSS (2016), “Tag adalah teks atau mungkin lebih enak disebut dengan istilah kode yang ditulis di antara tanda < dan >. Tag berfungsi untuk menentukan bagaimana sebuah website akan dibuat. Dengan kata lain, Tag mengontrol bentuk teks, paragraf, tabel, dan banyak lagi. Contoh-contoh tag antara lain: , <u>, <i>, <p>, dan seterusnya. Variasi lain dari Tag adalah Two-sided tag, yaitu tag, yang ditulis berpasangan. Jadi, penulisannya seperti ini: ... , <p> ... </p>, dan seterusnya.”.



Struktur script HTML sederhana ialah dengan menulis tag-tag sebagai berikut di awal script :

- a. <html>
- b. <head>
- c. <body>

Berikut contoh susunan script HTML paling sederhana yang bisa dicoba :

```
<!DOCTYPE HTML>
<html>
<head>
<title>Referensi Kilat HTML</title>
</head>

<body>
<h1>Buku Pintar HTML</h1>
<p>Ini adalah struktur script HTML sederhana</p>
</body>
</html>
```

2.4.3 Pengertian XAMPP

Anonim (2014:72), "XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP, dan Perl. XAMPP adalah tool yang menyediakan paket perangkat lunak dalam satu buah paket".

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.

Localhost adalah alamat komputer yang mengacu diri sendiri, yang bisa juga diganti dengan menggunakan alamat IP (IP address) 127.0.0.1, sehingga URL bisa ditulis dengan <http://127.0.0.1>. Jika komputer yang digunakan memiliki alamat IP yang tetap, maka kita bisa menggunakan alamat IP tersebut untuk mengakses server web".



2.4.4 Pengertian phpMyAdmin

Menurut Nugroho (2013:71), “phpMyAdmin adalah tools yang dapat digunakan dengan mudah untuk manajemen database MySQL secara visual dan Server MySQL, sehingga kita tidak perlu lagi harus menulis query SQL setiap akan melakukan perintah operasi database”.

2.4.5 Pengertian Sublime Text

Supono dan Virdiandry (2016:14), “Sublime Text merupakan perangkat lunak text editor yang digunakan untuk membuat atau meng-edit suatu aplikasi. Sublime Text mempunyai fitur plugin tambahan yang memudahkan programmer. Selain itu, Sublime Text juga memiliki desain yang simple dan keren menjadikan Sublime Text berkesan elegan untuk sebuah syntax editor.”

Menurut Bos (2014:12) yang dikutip dari jurnal Pahlevi (2018), “*Sublime Text* merupakan salah satu *text* editpr yang sangat *powerful* yang dapat meningkatkan produktivitas dan mengembangkan kualitas kode yang tinggi”.
