



BAB II TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Pengertian Komputer

Menurut Krisbiantoro (2018:1), Komputer merupakan alat yang dipakai untuk mengolah dan memproses data menurut perintah yang telah dirumuskan”. Sedangkan, menurut Yahfizham (2019:13), “Komputer adalah suatu perangkat yang digunakan untuk perhitungan atau mengendalikan operasi-operasi yang dinyatakan dalam bentuk numerik atau logika”. Dari pendapat tersebut dapat diambil intisaryanya bahwa komputer adalah suatu perangkat elektronik yang digunakan untuk mengolah dan memproses serta mengendalikan operasi-operasi yang dinyatakan dalam bentuk numerik ataupun logika.

2.1.2. Fungsi Komputer

Menurut Zakaria (2019), Komputer memiliki 4 fungsi utama, yaitu :

1. *Input*

Fungsi pertama adalah *input*, yaitu menerima data atau informasi dari sumber luar. Contoh paling mudah dari fungsi ini adalah informasi dari ketikan *keyboard* atau klik *mouse*. Komputer juga bisa mendapatkan informasi dari peralatan lain atau komputer lain.

2. *Processing*

Fungsi utama dari komputer adalah melakukan pemrosesan. Yang diproses adalah berbagai macam data dan informasi yang diberikan oleh perangkat input. Kemudian data yang tersimpan dalam memori internal akan diproses untuk menghasilkan informasi baru yang nantinya akan dikirim ke perangkat *output*.

3. *Output*

Setelah diproses, informasi yang dihasilkan akan menjalankan fungsi *output*. Hasilnya akan tersedia untuk digunakan oleh pengguna atau perangkat lainnya. Contoh dari perangkat *output* adalah monitor komputer, printer dan *speaker*.

Perangkat tersebut yang menyajikan *output* dari komputer untuk dimanfaatkan oleh pengguna atau perangkat lain.

4. *Storage*

Fungsi komputer yang terakhir adalah sebagai tempat untuk menyimpan informasi. Penyimpanan yang dilakukan berbeda-beda tergantung bagaimana informasinya akan digunakan. Ada yang disimpan dalam memori utama komputer, ada juga yang disimpan dalam memori tambahan. Informasi yang disimpan juga bisa dibedakan menjadi data pengguna dan data instruksi. Data instruksi inilah yang biasanya dikenal dengan sebutan program atau *software* komputer.

2.1.3. Pengertian Perangkat Lunak

Menurut Mulyani dalam (Fitriadi, 2018). “*Software*, Yaitu kumpulan program komputer yang digunakan untuk memproses data”.

Menurut (Mulyani 2016). “*Software* adalah istilah umum yang digunakan untuk mendeskripsikan kumpulan program-program komputer yang terdiri dari prosedur-prosedur dan dokumentasi untuk melakukan tugas tertentu”.

2.1.4. Jenis-Jenis Perangkat Lunak

Menurut Biztech Proxis Group (2020), ada dua klasifikasi utama perangkat lunak yaitu:

1. Perangkat Lunak Sistem

Perangkat lunak sistem membantu pengguna dan perangkat keras untuk berfungsi dan berinteraksi satu sama lain. Pada dasarnya, ini adalah perangkat lunak untuk mengelola perilaku perangkat keras komputer sehingga dapat menyediakan fungsionalitas dasar yang dibutuhkan oleh pengguna. Dengan kata sederhana, kita dapat mengatakan bahwa perangkat lunak sistem adalah perantara atau lapisan tengah antara pengguna dan perangkat keras.

2. Perangkat Lunak Aplikasi

Perangkat Lunak Aplikasi, juga dikenal sebagai program pengguna akhir atau program produktivitas adalah perangkat lunak yang membantu pengguna dalam menyelesaikan tugas-tugas seperti melakukan penelitian online, mencatat



catatan, menyetel alarm, merancang grafik, membuat log akun, melakukan perhitungan, atau bahkan bermain game.

2.1.5. Pengertian Data

Menurut Kristanto (2018:7), “Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata”. Dantes et.al. (2018:3) menuliskan, “Data adalah nilai/*value* yang dapat merepresentasikan deskripsi dari suatu objek atau kejadian (*event*). Objek yang dimaksud dapat berupa manusia (pegawai, siswa, pembeli, pelanggan), barang dan hewan”.

2.1.6. Pengertian Basis Data (*Database*)

Menurut Abdulloh (2018:103), “Database atau basis data adalah kumpulan informasi yang disimpan dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi”.

Menurut Dantes et.al. (2018:22), “Basis data adalah kumpulan data yang saling berhubungan. Basis data adalah kelompok data (arsip) yang saling berhubungan dan tanpa pengulangan (redudansi) yang disimpan dalam media elektronik dan diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan dengan cepat dan mudah”.

2.2. Teori Judul

2.2.1. Pengertian Aplikasi

Menurut Habibie dan Karnovi (2020:14), menyatakan bahwa “Aplikasi adalah sebuah program siap pakai yang dipakai untuk menjalankan sejumlah perintah dari pengguna aplikasi itu sendiri. Dengan tujuan untuk memperoleh hasil yang lebih akurat dan sesuai dengan tujuan pembuatan aplikasi tersebut”. Idcloudhost (2020) menuliskan, “Aplikasi adalah perangkat lunak yang menggabungkan beberapa fitur tertentu dengan cara yang dapat diakses oleh pengguna”.



2.2.2. Pengertian Pendataan

Pendataan telah digunakan atau diterapkan dalam melakukan suatu pekerjaan, agar bisa dijadikan suatu bahan untuk dijadikan dasar kajian. Karena pentingnya pendataan seseorang tidak akan bisa melangkah ketahap selanjutnya tanpa melakukan pendataan, maka dari itu diperlukan pendataan. Menurut Antoro (2016:6) menyatakan bahwa “pendataan adalah kegiatan untuk mengumpulkan data dan informasi”. Sedangkan Menurut Biro Pusat Statistik dalam Pauziah (2013:191) “pendataan adalah proses pembuktian yang ditemukan dari hasil penelitian yang dapat di jadikan dasar kajian atau pendapat”. Berdasarkan pernyataan diatas dapat disimpulkan pendataan adalah suatu proses pengumpulan data untuk pembuktian yang ditemukan dari hasil penelitian.

2.2.3. Pengertian Aset

Menurut Siregar (2018:178), menyebutkan bahwa “aset adalah barang (thing) atau sesuatu barang (anything) yang mempunyai nilai ekonomis (economic value), nilai komersial (commercial value) atau nilai tukar (exchange value) yang dimiliki oleh badan usaha, instansi atau individu/perorangan”. Menurut Supriyati (2016:44), Aset Tetap digolongkan menjadi dua antara lain adalah :

1. Aset Berwujud (Tangible Asset) Merupakan aset yang digunakan dalam jangka waktu lama dan bentuk fisiknya memberikan kegunaan dari aset tersebut. Contohnya adalah tanah, bangunan, kendaraan, mesin, peralatan, batubara, dan barang lainnya.
2. Aset Tak Berwujud (Intangible Asset) Aset yang digunakan bukan karena fisiknya, tetapi karena kepemilikan atas aset tersebut sehingga kita memiliki hak untuk melakukan sesuatu. Contohnya: hak paten, hak cipta, merek dagang, waralaba, dan royalti.



2.2.4. Pengertian *Right of Way* (ROW)

Menurut MKJI (1997) ruas Jalan, kadang-kadang disebut juga Jalan raya atau daerah milik Jalan (*right of way*). Pengertian Jalan meliputi badan Jalan, trotoar, drainase dan seluruh perlengkapan Jalan yang terkait, seperti rambu lalu lintas, lampu penerangan, marka Jalan, median, dan lain lain.

2.2.5. Pengertian *Website*

Menurut Sebok, Vermat, dan tim (2018:70), “*Website* adalah kumpulan halaman yang saling terhubung yang di dalamnya terdapat beberapa item seperti dokumen dan gambar yang tersimpan di dalam *web server*”. Hutahaean dan Azhar (2018:21), “*Website* (situs web) merupakan kumpulan dari halaman-halaman web yang berhubungan dengan file-file lain yang terkait”.

2.3. Teori Khusus

2.3.1. *Unified Modelling Language* (UML)

Menurut Haqi (2019:29), “UML merupakan singkatan dari “*Unified Modelling Language*” yaitu suatu metode permodelan secara visual untuk sarana perancangan sistem berorientasi objek, atau definisi UML yaitu sebagai suatu bahasa yang sudah menjadi standar pada visualisasi, perancangan dan juga pendokumentasian sistem *software*”.

Menurut Sa’ad (2020:46), “*Use Case Diagram* membantu menentukan fungsi dan fitur dari perangkat lunak. Dalam diagram ini, gambar yang menyerupai boneka kayu mewakili aktor yang berhubungan dengan kategori dari pengguna. Di dalam diagram *Use Case*, para aktor terhubung oleh garis ke *Use Case* yang mereka kerjakan.

2.3.2. Jenis-Jenis Diagram *Unified Modeling Language* (UML)

2.3.2.1. Pengertian *Use Case Diagram*

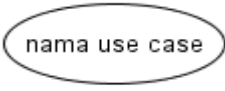
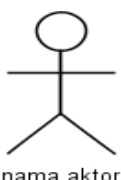

Menurut Setyawati, dkk (2021:13), “*Use case diagram* merupakan konsep informasi dengan menggunakan fungsi apa saja yang ada di dalam sistem informasi & siapa saja *actor* yang terlibat”.



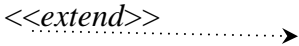
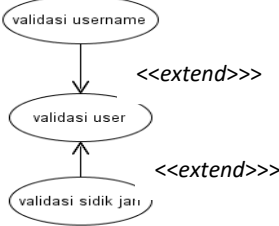
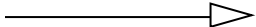
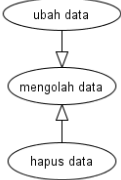
Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

- a. **Aktor** merupakan 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.
- b. **Use case** merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor. Simbol-simbol pada *Use Case Diagram* adalah sebagai berikut:

Tabel 2.1 Simbol-Simbol *Use Case Diagram*

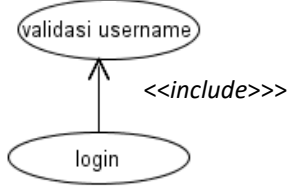
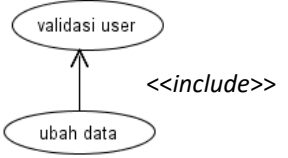
No	Simbol	Deskripsi
1.		<p>fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal-awal frase nama <i>use case</i>.</p>
2.		<p>orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi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

Lanjutan Tabel 2.1 Simbol-Symbol *Use Case* Diagram

4.	ekstensi / <i>extend</i> 	 <p>relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang di tambahkan dapat berdiri sendiri; 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 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> 	<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>  <p>arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)</p>

Lanjutan Tabel 2.1 Simbol-Symbol *Use Case* Diagram



6.	<p>menggunakan / include / uses</p> <p>.....></p> <p><<include>></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>a. <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>b. <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>
----	---	---

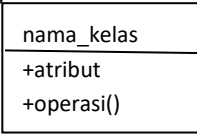



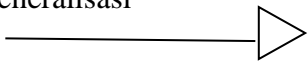

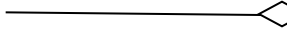
Sumber: Sukamto dan Shalahuddin (2018:156)



2.3.2.2. Pengertian Class Diagram

Menurut Haqi (2019:32), “Class diagram yaitu salah satu jenis diagram pada UML yang digunakan untuk menampilkan kelas-kelas maupun paket-paket yang ada pada suatu sistem yang nantinya akan digunakan. Jadi diagram ini dapat memberikan sebuah gambaran mengenai sistem maupun relasi yang terdapat pada sistem tersebut”. Simbol-simbol pada 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>  nama_interface	Sama dengan konsep interface dalam pemrograman berorientasi objek.
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>).


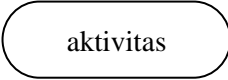



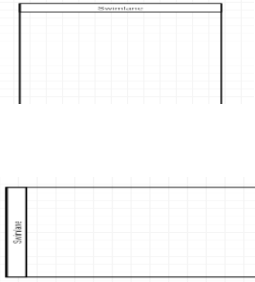
Sumber: Sukamto dan Shalahuddin (2018:146)



2.3.2.3. Pengertian Activity Diagram

Menurut Haqi (2019:31), “*Activity diagram* atau diagram aktivitas yaitu salah satu jenis diagram pada UML yang dapat memodelkan proses-proses apa saja yang terjadi pada sistem”. Simbol-simbol pada *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> 	<i>Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.</i>


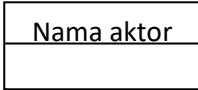

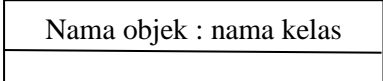
Sumber: Sukamto dan Shalahuddin (2018:162)




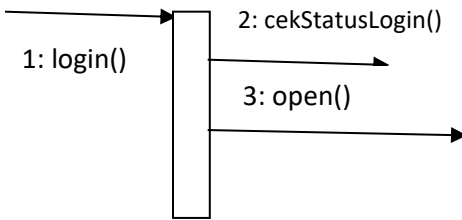


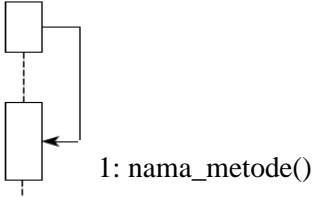
2.3.2.4. Pengertian Sequence Diagram

Menurut Haqi (2019:31), “*Sequence* diagram yaitu salah satu jenis diagram pada UML yang menjelaskan interaksi objek yang berdasarkan urutan waktu, *sequence* diagram juga dapat menggambarkan urutan atau tahapan yang harus dilakukan untuk dapat menghasilkan sesuatu seperti pada *use case* diagram”. Simbol-simbol pada *Sequence* Diagram adalah sebagai berikut:

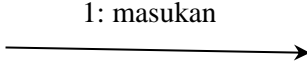
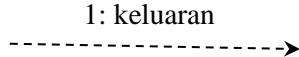
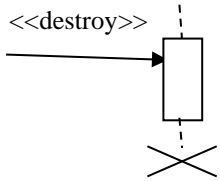
Tabel 2.4 Simbol-Simbol *Sequence* Diagram

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>
3.	<p>Objek</p>  <p>Nama objek : nama kelas</p>	<p>menyatakan objek yang berinteraksi pesan.</p>

Lanjutan Tabel 2.4 Simbol-Simbol *Sequence Diagram*

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 cek Status Login () 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;"><<create>></p> 	<p>menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.</p>
6.	<p>Pesan tipe <i>call</i></p> <p style="text-align: center;">1: nama_metode()</p> 	<p>menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,</p>  <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.</p>

Lanjutan Tabel 2.4 Simbol-Simbol *Sequence Diagram*

7.	Pesan tipe <i>send</i> 	menyatakan bahwa suatu objek mengirimkan data/masukan/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, 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 *Visual Studio Code*

Menurut Nurhayati (2022:34), “*Visual Studio Code (VS Code)* adalah sebuah teks editor ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi *multiplatform*, artinya tersedia juga untuk versi Linux, Mac, dan Windows”.

Menurut Salamah (2021:1) *Visual Studio Code (VS Code)* ini adalah sebuah teks editor ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi *multiplatform*, artinya tersedia juga untuk versi Linux, Mac, dan Windows. Teks editor ini secara langsung mendukung bahasa pemrograman JavaScript, Typescript, dan Node.js, serta bahasa pemrograman lainnya dengan bantuan plugin yang dapat dipasang via marketplace *Visual Studio Code* (seperti C++, C#, Python, Go, Java, dst).



2.4.2. Pengertian HTML

Menurut Enterprise (2018:21), “HTML adalah Bahasa markup (*markup language*) seperti yang ada di dalam singkatan HTML itu sendiri. Itu artinya, HTML adalah bahasa struktur untuk menandai bagian-bagian dari sebuah halaman”.

Menurut Kaban (2019:4), “HTML merupakan singkatan dari Hyper Text Markup Language yaitu bahasa standar web yang dikelola penggunaannya oleh W3C (World Wide Web Consortium) berupa tag-tag yang menyusun setiap elemen dari website”.

2.4.3. Pengertian CSS

Menurut Abdulloh (2018:45), “CSS adalah singkatan dari *Cascading Style Sheet* yaitu dokumen web yang berfungsi mengatur elemen *HTML* dengan berbagai property yang tersedia sehingga dapat tampil dengan berbagai gaya yang diinginkan”.

Menurut Sulistiono (2018:2), “CSS (*Cascading Style Sheet*) merupakan aturan untuk mengendalikan beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam. CSS bukan bahasa pemrograman. Sama halnya seperti style dalam aplikasi pengolahan kata, Microsoft Word yang dapat mengatur beberapa style, misalnya heading, subbab, bodytext, footer, images, dan style lainnya untuk dapat digunakan bersama-sama dalam beberapa berkas (file)”.

2.4.4. Pengertian PHP

Menurut Arief dalam Sa’ad (2020:24), “PHP adalah salah satu bahasa *server-side-scripting* yang menyatu dengan HTML untuk membuat halaman web yang dinamis. Karena PHP merupakan *server-side-scripting* maka sintaks dan perintah-perintah PHP akan dieksekusi di *server* kemudian hasilnya akan dikirimkan ke browser dengan format HTML”.

Menurut Enterprise (2018:1), “*PHP* merupakan bahasa pemrograman yang digunakan untuk membuat *website* dinamis dan interaktif. Dinamis artinya, *website* tersebut bisa berubah-ubah tampilan dan kontennya sesuai kondisi tertentu”.



2.4.5. Pengertian *CodeIgniter*

Menurut Arrhioui et al., (2017), “*CodeIgniter* adalah kerangka kerja pengembangan aplikasi PHP berdasarkan arsitektur yang terstruktur. *CodeIgniter* memiliki tujuan untuk memberikan alat bantu yang dibutuhkan seperti *helpers and libraries* untuk mengimplementasikan tugas yang biasa dilakukan. dengan demikian, pengembangan proyek menjadi lebih mudah dan cepat. dan pengembang tidak perlu menulis lagi dari awal”.

Menurut Saputra (2018:2), “*CodeIgniter* merupakan *toolkit* bagi orang yang ingin membangun aplikasi *web* menggunakan PHP. Tujuannya adalah membuat pengembangan proyek menjadi lebih cepat dibandingkan dengan menulis kode awal”.

2.4.6. Pengertian MySQL

Menurut Enterprise (2018:2), “*MySQL* merupakan server yang melayani database. Untuk membuat dan mengolah database, kita dapat mempelajari pemrograman khusus yang disebut *query* (perintah) *SQL*”.

Menurut Sa’ad (2020:35), “*MySQL* adalah suatu *RDBMS (Relational Data Base Management System)*, yaitu aplikasi sistem yang menjalankan fungsi pengolahan data”.

2.4.7. Pengertian XAMPP

Menurut Roza, dkk (2020:82), *XAMPP* adalah Perangkat lunak bebas, yang mendukung banyak sytem operasi, merupakan komplikasi dari beberapa program. Salah satu paket installisasi Apache, PHP dan MySQL instan yang dapat kita gunakan untuk membantu proses installasi ketiga produk tersebut.

Menurut Mawaddah dan Fauzi (2018:2), “*XAMPP* adalah software web server apache yang di dalamnya tertanam server *MySQL* yang didukung dengan bahasa pemrograman PHP untuk membuat website yang dinamis. *XAMPP* sendiri dapat berjalan pada berbagai macam platform seperti Windows, Linux, Mac OS X dan Solaris”.



