



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Komputer

Robert H. Blissmer (2005) dalam Sutarman (2012:2), “Komputer adalah suatu alat elektronik yang mampu melakukan beberapa tugas, yaitu menerima, memproses sesuai instruksi, menyimpan perintah, hasil pengelolaan, serta menyediakan output dalam bentuk informasi”.

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

Berdasarkan definisi di atas, dapat disimpulkan bahwa komputer merupakan alat elektronik yang dapat menerima, memproses, mencetak, serta menyimpan data yang diolah dan beroperasi dibawah perintah manusia.

2.1.2 Pengertian Perangkat Lunak

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

Sedangkan menurut Kadir (2013:2), “Perangkat lunak berupa instruksi-instruksi yang ditujukan kepada komputer dapat melaksanakan tugas sesuai dengan kehendak pemakai”.

Berdasarkan pendapat di atas dapat disimpulkan bahwa perangkat lunak adalah perintah-perintah yang diarahkan kepada komputer sehingga dapat menjalankan tugas dengan sesuai keinginan *user*.

2.1.3 Pengertian Data

Menurut Pipin (2013:22), “ Data adalah kumpulan dari angka-angka maupun karakter-karakter yang tidak memiliki arti”.



Sedangkan menurut Indrajani (2015:69), “ Data merupakan fakta mentah tentang orang, tempat, kejadian, dan apapun yang penting bagi perusahaan, dimana data itu sendiri tidak memiliki arti”.

Menurut Fathansyah (2018:2) menyatakan bahwa “Data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya, yang diwujudkan dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya”.

Dari pendapat di atas dapat disimpulkan bahwa data adalah penjelasan atau informasi yang akurat dengan fakta di dunia nyata yang dapat diproses dan menghasilkan suatu informasi.

2.1.4 Pengertian Basis Data (*Database*)

Menurut Kadir (2013:264), “*Database* adalah suatu metode pengorganisasian data agar data dapat dimanipulasi, juga dapat diperoleh”.

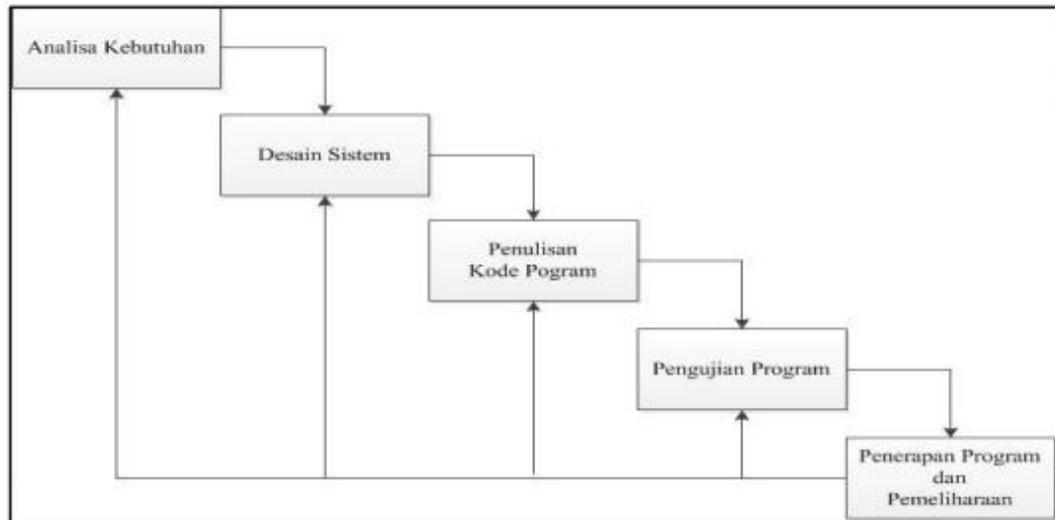
Menurut Abdulloh (2018:103) menegaskan bahwa “Basis data atau *database* adalah kumpulan informasi yang disimpan dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi”.

Menurut Kristanto (2018:79), “Basis data adalah kumpulan data, yang dapat digambarkan sebagai aktifitas dari satu atau lebih organisasi yang berelasi”.

Dari pendapat di atas dapat disimpulkan bahwa basis data atau *database* adalah sekumpulan data yang saling berhubungan disimpan di dalam komputer sehingga dapat diproses untuk memperoleh informasi yang dibutuhkan.

2.2 Teori Khusus

2.2.1 Metode Pengembangan Sistem



Gambar 2. 1 Ilustrasi Model *Waterfall*

Dalle, dkk (2020:398) menjelaskan tentang metode *waterfall* menyediakan pendekatan alur hidup *software* secara *sequential* atau terurut dimulai dari analisa, desain, pencodingan, uji coba, dan tahap pendukung. Jika langkah satu belum dikerjakan maka tidak akan bisa melakukan pengerjaan langkah 2, 3, dst.

1. Analisa

Sebuah analisis persyaratan sistem yang dilakukan dengan wawancara atau literatur belajar. Menghasilkan data yang berkaitan dengan kebutuhan klien dalam proses pembuatan sistem atau dapat dikatakan dokumen persyaratan pengguna, yang merupakan tujuan dari sistem analisis untuk menerjemahkannya ke dalam bahasa pemrograman.

2. Desain

Proses ini menerjemahkan persyaratan ke dalam desain perangkat lunak yang dapat diperkirakan sebelum pengkodean dibuat. Berfokus pada struktur data, arsitektur perangkat lunak, representasi antarmuka, dan retail prosedural. Menghasilkan sebuah dokumen yang disebut persyaratan perangkat lunak, yang merupakan apa yang programmer gunakan dalam kegiatan manufaktur sistemnya.

3. Coding dan Testing



Terjemahkan desain ke dalam bahasa pemrograman sehingga dapat dibaca oleh komputer. Setelah coding tahap berikutnya, pengujian. Pengujian dalam hal ini memiliki manfaat menemukan kesalahan dengan sistem dan kemudian memperbaikinya.

4. Penerapan

Step ini bisa dikatakan akhir dalam pembuatan sebuah sistem. Sesudah melakukan analisa, desain dan pengkodean maka sistem yang sudah jadi akan dipakai oleh *user*.

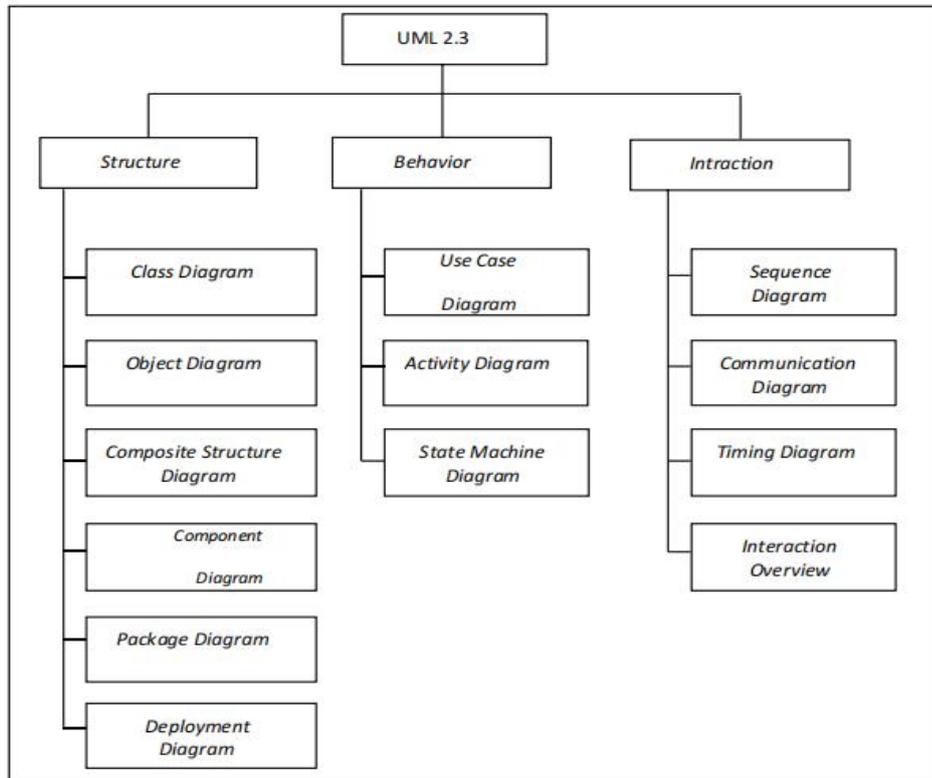
5. Pemeliharaan

Software yang sudah disampaikan kepada *client* pasti mengalami revisi. Revisi tersebut bisa karena mengalami kesalahan karena *software* harus menyesuaikan dengan daerahnya (*peripheral* atau sistem operasi baru. Atau karena *client* membutuhkan perkembangan fungsional.

2.2.2 *Unified Modeling Language (UML)*

Menurut Sukamto dan Shalahuddin (2018:137), menjelaskan “Unified Modeling Language (UML) merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung”.

Pada UML terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian dan macam - macam diagram tersebut yaitu :



Gambar 2. 2 Diagram UML

Berikut merupakan penjelasan singkat dari pembagian kategori tersebut :

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. *Interactions diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun sistem interaksi antar subsistem pada suatu sistem.

2.2.3 Jenis-Jenis Diagram UML

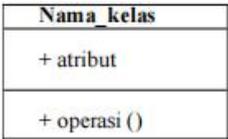
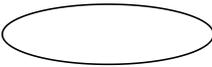
2.2.3.1 Pengertian *Class Diagram*

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

“*Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Diagram kelas dibuat agar pembuat program atau programmer membuat kelas-kelas sesuai rancangan didalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron.”

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

Tabel 2. 1 Simbol-Simbol *Class Diagram*

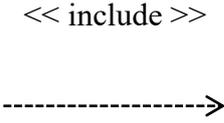
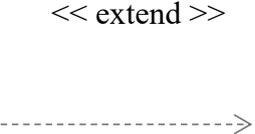
| No. | Simbol | Deskripsi |
|-----|---|---|
| 1 | Kelas  | Kelas pada struktur sistem. |
| 2. | Antarmuka / interface  | Sama dengan konsep interface dalam pemrograman berorientasi objek. |
| 3. | Asosiasi / association  | Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai <i>multiplicity</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 antarkelas |
| 7. | Agregasi / <i>aggregation</i>  | Relasi antarkelas dengan makna semua-bagian (<i>whloe-part</i>) |

Sumber : Sukamto dan Shalahuddin (2018:146)

2.2.3.2 Pengertian Use case Diagram

Sukamto dan Shalahuddin (2018:155), menjelaskan tentang *use case* diagram sebagai berikut : “Use Case merupakan pemodelan untuk kelakuan (behaviour) sistem informasi yang akan dibuat. Use Case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem.”

Tabel 2. 2 Simbol-simbol Use Case Diagram

| No. | Simbol | Nama | Deskripsi |
|-----|---|-----------------|--|
| 1. |  | Actor | Seseorang atau apa saja yang berhubungan dengan sistem |
| 2. |  | Use Case | Menggambarkan bagaimana seseorang menggunakan sistem |
| 3. |  | Relasi asosiasi | Relasi yang dipakai untuk menunjukkan hubungan antara aktor dan use case |
| 4. |  | Relasi include | Memungkinkan satu use case menggunakan fungsionalitas yang disediakan oleh use case lainnya |
| 5. |  | Relasi extend | Memungkinkan suatu use case secara optional menggunakan fungsionalitas yang disediakan oleh use case lainnya |

(Sumber: Rusmawan, 2019-73)

Ada dua hal utama pada *use case* yaitu :

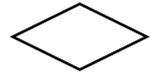
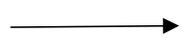
1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

2.2.3.3 Pengertian *Activity Diagram*

Menurut Nugroho dalam Rusmawan (2019:79) “*Activity diagram* digunakan untuk menampilkan rangkaian kegiatan, menunjukkan alur kerja dari suatu titik awal ke titik akhir keputusan, merinci banyak jalur yang ada dalam perkembangan peristiwa yang terkandung dalam kegiatan”.

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

Tabel 2. 3 Simbol-simbol *Activity Diagram*

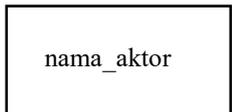
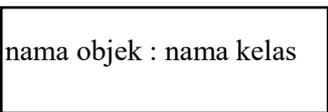
| No. | Simbol | Nama | Deskripsi |
|-----|---|-------------|--|
| 1. |  | Start state | Titik awal atau permulaan |
| 2. |  | End state | Titik akhir atau akhir dari aktivitas |
| 3. |  | Activity | <i>Activity</i> atau aktivitas yang dilakukan oleh Actor |
| 4. |  | Decision | Pilihan untuk mengambil keputusan |
| 5. |  | Interaction | Alur |

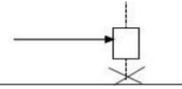
Sumber : Sukanto dan Shalahuddin (2018:162-163)

2.2.3.4 Pengertian *Sequence Diagram*

Menurut Sukamto dan Shalahuddin (2018:165), “ Diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah beserta metode-metode yang dimiliki kelas yang diinstansikan menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada use case”.

Tabel 2. 4 Simbol-simbol pada *Sequence Diagram*

| No. | Simbol | Keterangan |
|-----|---|--|
| 1. | Aktor atau   | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. |
| 2. | Garis hidup/ <i>lifeline</i>  | Menyatakan kehidupan suatu objek |
| 3. | Objek  | Menyatakan objek yang berinteraksi pesan |
| 4. | Waktu aktif  | Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya |
| 5. | Pesan tipe <i>call</i> 1 : nama_metode()  | Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri |

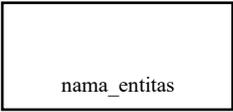
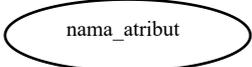
| | | |
|----|--|---|
| 6. | <p>Pesan tipe <i>send</i></p> <p>1 : masukan</p>  | <p>Menyatakan bahwa suatu objek mengirimkan data.masukan.informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim</p> |
| 7. | <p>Pesan tipe <i>return</i></p> <p>1 : keluaran</p>  | <p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian</p> |
| 8. | <p>Pesan tipe <i>destroy</i></p> <p><<destroy>></p>  | <p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada Destroy</p> |

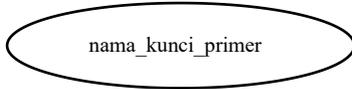
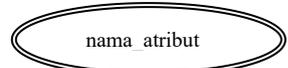
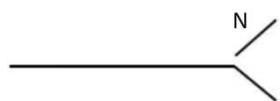
Sumber : Sukamto dan Shalahuddin (2018:165)

2.2.4 Entity Relationship Diagram (ERD)

Puspitasari (2016:229), “Entity Relationship Diagram (ERD) adalah suatu rancangan atau bentuk hubungan suatu kegiatan di dalam sistem yang berkaitan langsung dan mempunyai fungsi di dalam proses tersebut”.

Tabel 2. 5 Simbol-Simbol Entity Relationship Diagram (ERD)

| No. | Simbol | Deskripsi |
|-----|--|---|
| 1. | <p>Entitas/<i>entity</i></p>  | <p>Entitas merupakan data inti yang akan tersimpan; bakal tabel pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.</p> |
| 2. | <p>Atribut</p>  | <p><i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas.</p> |

| | | |
|----|--|--|
| 3. | <p>AtributKunciPrimer</p>  | <p><i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama).</p> |
| 4. | <p>AtributMultinilai/<i>Multival</i> <i>ue</i></p>  | <p><i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.</p> |
| 5. | <p>Relasi</p>  | <p>Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.</p> |
| 6. | <p>Asosiasi/<i>Association</i></p>  | <p>Penghubung antara relasi dan entitas dimana dikedua ujungnya punya <i>multiplicity</i> ke-mungkinan jumlah pemakaian. Kemungkinan jumlah maksimum keterhubungan antara entitas yang lain disebut dengan <i>one to many</i></p> |

Sumber : Sukamto dan Shalahuddin (2018:50-51)

2.2.5 Kamus Data

Sukamto dan Shalahuddin (2018:73), “Kamus data adalah kumpulan daftar elemen yang mengalir pada sistem lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan)”. Kamus data memiliki beberapa simbol sebagai berikut :

Tabel 2. 6 Simbol-Simbol Kamus Data

| No. | Simbol | Keterangan |
|-----|------------------|--------------------------------|
| 1. | = | Disusun terdiri dari |
| 2. | + | Dan |
| 3. | [] | baik...atau... |
| 4. | { } ⁿ | n kali diulang/bernilai banyak |



| | | |
|----|------|----------------|
| 5. | () | data opsional |
| 6. | *..* | batas komentar |

Sumber : Sukamto dan Shalahuddin (2018:74)

2.3 Teori Judul

2.3.1 Pengertian Implementasi

Menurut Winarno (2014:147), “Implementasi dipandang secara luas mempunyai makna pelaksanaan undang-undang di mana berbagai aktor, organisasi, prosedur, dan teknik bekerja bersama-sama untuk menjalankan kebijakan dalam upaya untuk meraih tujuan-tujuan kebijakan atau program - program”.

Sedangkan menurut Usman dan Nurdin (2002:70), “Implementasi adalah suatu aktivitas, aksi, tindakan atau lebih dari itu terdapat mekanisme sistem dengan kegiatan terencana untuk mencapai tujuan”.

2.3.2 Pengertian Warehouse Management System

Menurut Emmett (2005), “Manajemen Warehouse dirancang bertujuan untuk mengontrol kegiatan pergudangan. Yang diharapkan dari pengontrolan ini adalah terjadinya pengurangan biaya-biaya yang ada di dalam gudang, pengambilan dan pemasukan barang ke gudang yang efektif dan efisien, serta kemudahan dan keakuratan informasi stock barang di gudang. Sistem informasi mengenai manajemen pergudangan ini sering disebut dengan *Warehouse Management System (WMS)*”.

2.3.3 Pengertian Pengelolaan

Menurut Poerwadarminta (2006), “Pengelolaan adalah proses yang membantu merumuskan kebijakan dan tujuan organisasi atau proses yang memberikan pengawasan pada semua hal yang terlibat dalam pelaksanaan kebijaksanaan dan pencapaian tujuan”.

Sedangkan menurut Syamsi (2008), “Pengelolaan adalah proses, cara, perbuatan pengelolaan yang membantu merumuskan kebijakan dan tujuan



organisasi atau yang memberikan pengawasan suatu hal yang terlibat dalam pelaksanaan kebijakan dan pencapaian tujuan dengan menggunakan tenaga orang lain”.

2.3.4 Pengertian Gudang

Menurut Purnomo (2004), “Gudang atau storage merupakan tempat menyimpan barang baik bahan baku yang akan dilakukan proses manufacturing maupun barang jadi yang siap dipasarkan. Sedangkan pergudangan tidak hanya kegiatan penyimpanan barang saja melainkan proses penanganan barang mulai dari penerimaan barang, pencatatan, penyimpanan, pemilihan, penyortiran, pelebelan, sampai dengan proses pengiriman”.

2.3.5 Pengertian Website

Menurut Bekti (2015:35), “*Website* merupakan kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman”.

Sedangkan menurut Lutfhi (2013:1), “*Website* (lebih dikenal dengan sebutan situs) adalah sejumlah halaman web yang memiliki topik saling terkait, terkadang disertai pula dengan berkas-berkas gambar, video atau jenis-jenis berkas yang lainnya”.

2.4 Teori Program

2.4.1 Pengertian XAMPP

Madcoms (2014:186), “Xampp adalah sebuah paket kumpulan software yang terdiri dari Apache, MySQL, PhpMyAdmin, Perl, Filezilla, dan lain-lain”.

Aryanto (2016:4), “Xampp merupakan sebuah aplikasi perangkat lunak pemrograman dan database yang didalamnya terdapat berbagai macam aplikasi pemrograman seperti: Apache, HTTP, MySQL, database, bahasa pemrograman PHP dan Perl”.



2.4.2 Pengertian MySQL

Menurut Raharjo (2015:16), “MySQL adalah software RDMS (atau server database) yang dapat mengelola database dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak user (multi-user), dan dapat melakukan suatu proses secara sinkron atau berbarengan (multi-threaded)”.

2.4.3 Pengertian PHP

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

Menurut Huda dan Rahayu (2022:14), “PHP adalah sebuah bahasa pemrograman web berbasis server (server-side) yang mampu memarsing kode PHP dari kode web dengan ekstensi .php, sehingga menghasilkan tampilan website yang dinamis di sisi client (browser)”.

2.4.4 Pengertian PHPMyAdmin

Menurut Arief dalam Firliana *et.al.* (2020:446), “phpMyAdmin adalah salah satu aplikasi GUI (Graphic User Interface) yang digunakan untuk mengelola database MySQL”.

Sedangkan menurut Yudhanto dan Prasetyo (2019:173), “*phpMyAdmin* merupakan sebuah program aplikasi yang dibuat menggunakan program *PHP* untuk mempermudah kita dalam mengakses *database MySQL*”.

2.4.5 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”.

Sedangkan menurut Saputra (2019:2), “HTML atau *Hyper Text Markup Language* merupakan sebuah bahasa pemrograman terstruktur yang



dikembangkan untuk membuat laman *website* yang dapat diakses atau ditampilkan menggunakan *web browser* (peramban web)”.