



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Komputer

Menurut Situmorang & Maudiarti (2020:4) “Komputer secara sederhana adalah sebuah alat untuk melakukan proses perhitungan aritmatika, sedangkan secara umum adalah peralatan elektronika yang berfungsi sebagai peng-*input* data kemudian mengolahnya dan memberikan keluaran informasi dalam bentuk teks, gambar, suara, maupun video”.

Menurut Harmayani, dkk. (2021:2) “Komputer adalah alat elektronik yang terdiri dari rangkaian berbagai komponen yang saling terhubung sehingga membentuk suatu sistem kerja. Sistem di dalam komputer tersebut dapat melakukan pekerjaan secara otomatis berdasarkan program yang diperintahkan kepadanya sehingga mampu menghasilkan informasi berdasarkan data dan program yang ada”.

Dari beberapa pengertian di atas, penulis menyimpulkan bahwa komputer adalah suatu alat elektronik yang mengelola data menjadi suatu informasi yang berguna dan juga dapat melakukan perhitungan sehingga dapat mempermudah pekerjaan manusia di berbagai bidang.

2.1.2 Pengertian Internet

Menurut Wiwi & Syahlanisyiam (2022:14) “Internet secara harfiah adalah kumpulan komputer di seluruh dunia yang terhubung ke dalam sebuah jaringan. Internet bisa dianalogikan sebagai perpustakaan besar yang memuat beragam informasi yang dibutuhkan masyarakat”.

Menurut Gani (2018:71) “Internet merupakan kepanjangan dari *interconnected networking*, yang berarti hubungan komputer dengan berbagai tipe yang membentuk sistem jaringan yang mencakup seluruh dunia dengan melalui jalur telekomunikasi, telepon, satelit dan lainnya”.

Dari beberapa pengertian di atas, penulis menyimpulkan bahwa internet



adalah suatu jaringan komputer luas yang dapat menghubungkan komputer satu dengan komputer lainnya di seluruh dunia melalui telepon, satelit atau sistem komunikasi yang lain.

2.1.3 Pengertian Perangkat Lunak

Menurut Wibawanto (2018:193) menyatakan bahwa perangkat lunak merupakan program-program komputer yang digunakan untuk menjalankan pekerjaan sesuai dengan yang dikehendaki dan ditulis dengan bahasa khusus yang dimengerti oleh komputer.

Menurut Harmayani, dkk. (2021:7) “Software dalam komputer adalah komponen penting karena berperan menjalankan segala perintah yang masuk ke hardware. Software dapat diartikan juga sebagai suatu kumpulan data elektronik yang tersimpan dan diatur oleh komputer, bisa berupa program ataupun koneksi untuk menjalankan berbagai macam instruksi perintah”.

Dari beberapa pengertian di atas, penulis menyimpulkan bahwa perangkat lunak adalah program komputer yang berisi perintah-perintah yang ditulis dengan aturan tertentu untuk menjalankan tugas tertentu.

2.1.4 Pengertian Data

Menurut Rochman, dkk. (2019:2) “Data adalah suatu bahan mentah yang kelak dapat diolah lebih lanjut untuk menjadi suatu yang lebih bermakna. Data inilah yang nantinya akan disimpan dalam *database*”.

Menurut Siswandari (dalam Makbul, 2021:3) “Data dapat berarti fakta dari suatu objek yang diamati yang dapat berupa angka-angka dan kata-kata. Sedangkan jika dipandang dari sisi statistika, maka data merupakan fakta-fakta yang akan digunakan sebagai bahan kesimpulan”.

Dari beberapa pengertian di atas, penulis menyimpulkan bahwa data adalah sekumpulan fakta yang dapat berbentuk objek, orang, angka, simbol, tabel, huruf, kata, dan lain-lain yang dapat diolah menjadi informasi.



2.1.5 Pengertian Basis Data

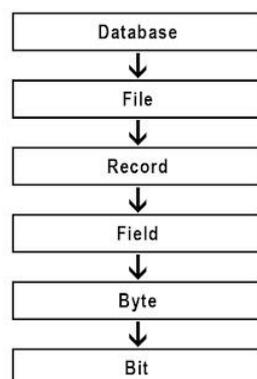
Menurut Jayanti dan Sumiarti (2018:2) menyatakan bahwa secara konsep basis data atau *database* adalah kumpulan dari data-data yang membentuk suatu berkas (file) yang saling terhubung (relation) dengan tata cara tertentu untuk membentuk suatu data baru atau sebuah informasi.

Menurut Rosa dan M. Shalahuddin (2018:43), "Basis data (*data base*) adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan".

Basis data dapat didefinisikan sebagai "sekumpulan data yang terintegrasi, yang diorganisasi untuk memenuhi kebutuhan para pemakai di dalam suatu organisasi". Maksud dari terintegrasi adalah, setiap data (yang nantinya kita sebut sebagai tabel) akan memiliki hubungan dengan data yang lainnya (data yang terhubung) (Jayanti dan Sumiarti, 2018).

Dalam implementasinya, untuk memudahkan dalam mengakses data, data disusun dalam suatu struktur logis yang menjelaskan bahwa:

- a. Kumpulan tabel menyusun basis data.
- b. Tabel tersusun atas sejumlah record.
- c. Sebuah record mengandung sejumlah field.
- d. Sebuah field disimpan dalam bentuk kumpulan bit.

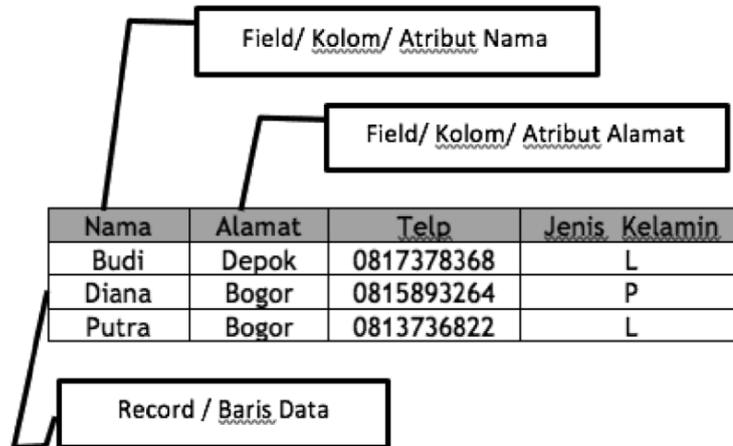


Gambar 2.1 Hierarchy Basis Data



-
- Basis data/ *Database*, merupakan kumpulan dari bermacam-macam tipe record yang memiliki hubungan antar record.
 - Berkas/ File, merupakan sekumpulan data rekaman yang berkaitan dengan suatu objek.
 - Record, merupakan sekumpulan field/atribut/data item yang saling berhubungan dengan obyek tertentu dengan panjang record yang tetap, semua field dalam record memiliki ukuran yang tetap. Variabel panjang record, field-field dalam record dapat memiliki ukuran yang berbeda (metode penandaan yang digunakan adalah : penanda akhir record, indikator panjang, dan tabel posisi record).
 - Field menyatakan data terkecil yang memiliki makna. Field merupakan implementasi dari suatu atribut data. Field merupakan unit terkecil dari data yang berarti (meaningful data) yang disimpan dalam suatu file atau basis data. Istilah lain untuk field yaitu elemen data, kolom item, dan atribut. Contoh field yaitu nama, alamat, telepon, dan jenis kelamin.
 - Byte , adalah bagian terkecil yang dialamatkan dalam memori. Byte merupakan kumpulan bit yang secara konvensi terdiri atas kombinasi delapan bit yang menyatakan sebuah karakter dalam memori (1 byte= 1 karakter) .
 - Bit , adalah sistem biner yang terdiri atas dua macam nilai, yaitu 0 dan 1. sistem biner merupakan dasar yang dapat digunakan untuk komunikasi antara manusia dan mesin, yang merupakan rangkaian komponen elektronik dan hanya dapat membedakan 2 macam keadaan, yaitu ada tegangan dan tidak ada tegangan yang masuk ke rangkaian tersebut

Berikut adalah contoh ilustrasi dari Basis Data :



Gambar 2.2 Contoh Ilustrasi Basis Data

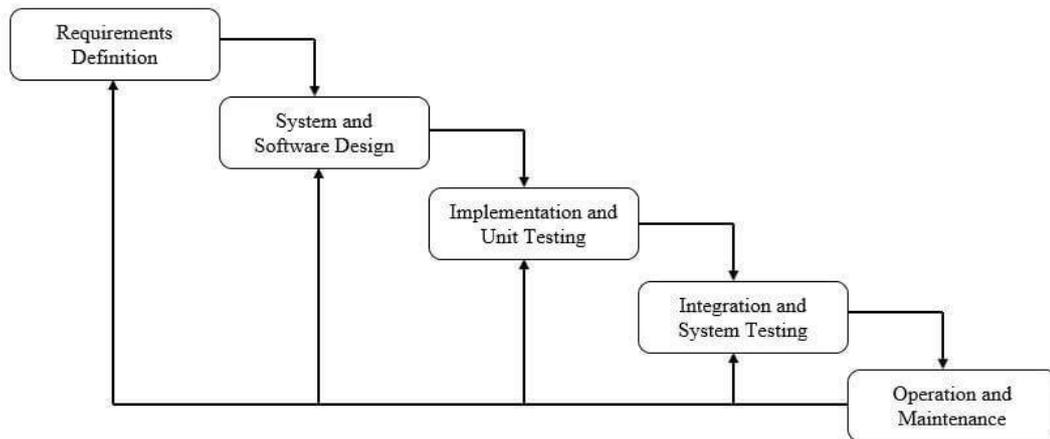
Dalam pengembangan suatu sistem informasi, basis data memiliki peran yang penting dan sangat diperlukan. Berikut merupakan alasan perlunya basis data:

- Salah satu komponen penting dalam suatu sistem informasi adalah basis data, karena basis data merupakan dasar dalam menyediakan informasi.
- Informasi dikatakan lebih bernilai jika memiliki manfaat yang efektif dibandingkan dengan biaya dalam mendapatkannya. Dalam hal ini, basis data akan menentukan kualitas informasi.
- Data akan dapat saling berelasi dengan mengimplementasikan basis data.
- Basis data mengurangi duplikasi data (data redundancy).
- Basis data dapat mengurangi pemborosan tempat penyimpanan luar.

2.1.6 Metode Pengembangan Sistem

Metode pengembangan sistem yang dilakukan dalam penelitian ini adalah Metode *Waterfall* yang mana metode tersebut menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut (Casro, dkk., 2020).

Berikut adalah gambar Model *Waterfall*:



Gambar 2.3 Metode *Waterfall*

1. *Requirements Definition*

Analisis Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu didokumentasikan, diawali dengan analisis proses bisnis usulan yang disesuaikan dengan kebutuhan *user*. Selanjutnya menganalisis kebutuhan sistem yaitu menganalisis *hardware* dan *software* apa yang sesuai dengan penelitian.

2. *System and Software Design*

Tahapan perancangan sistem mengalokasikan kebutuhan-kebutuhan sistem baik perangkat keras maupun perangkat lunak dengan membentuk arsitektur sistem secara keseluruhan. Perancangan perangkat lunak melibatkan identifikasi dan penggambaran abstraksi sistem dasar perangkat lunak dan hubungannya. Pada tahap ini peneliti merancang sistem yang akan dibangun, seperti *use case*, *sequence diagram*, *class diagram*, serta melakukan pengkodean sistem.

3. *Implementation and Unit Testing*

Desain yang telah dibuat harus diterjemahkan ke dalam program perangkat lunak. Tahapan ini akan menghasilkan program komputer yang merupakan representasi dari tahap desain.



4. *Integration and System Testing*

Tahap pengujian berfungsi untuk mengurangi kesalahan/*error* serta menghasilkan output yang sesuai dengan harapan. Pengujian dapat dilakukan dari sisi logik dan fungsional serta menguji semua bagian/fitur pada aplikasi. Metode pengujian yang digunakan pada penelitian ini adalah metode *blackbox testing* yang berfungsi untuk menguji fungsionalitas sistem yang dibangun.

5. *Operation and Maintenance*

Biasanya (walaupun tidak selalu), tahapan ini merupakan tahapan yang paling panjang. Sistem di pasang dan digunakan secara nyata. *Maintenance* melibatkan pembetulan kesalahan yang tidak ditemukan pada tahapan-tahapan sebelumnya, meningkatkan implementasi dari unit sistem, dan meningkatkan layanan sistem sebagai kebutuhan baru (Purwanto dkk., 2022).



2.2 Teori Khusus

2.2.1 Pengertian UML (*Unified Modelling Language*)

Menurut Rosa & Shalahuddin (2018:133) “UML (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek”.

UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML adalah bahasa spesifikasi standar yang digunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak (Hendini, 2016).

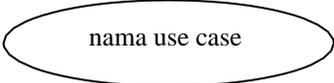
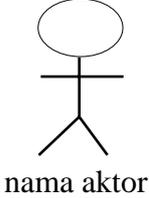
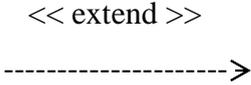
Pada perkembangan Teknik pemrograman berorientasi objek, munculah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan Teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language* (UML).

2.2.2 Pengertian *Use Case Diagram*

Menurut Rosa & Shalahuddin (2018:155) *Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Menurut Rosa & Shalahuddin (2018:156-158) simbol-simbol yang ada pada diagram *use case* yaitu:

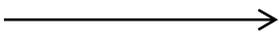
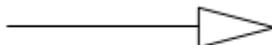
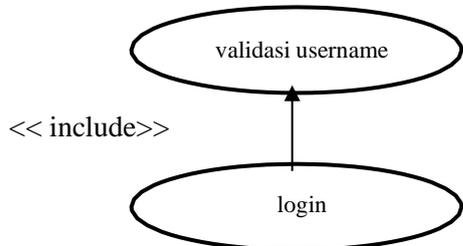
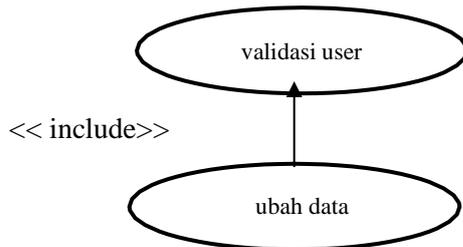
Tabel 2.1 Simbol-simbol *Use Case Diagram*

| No. | Simbol | Deskripsi |
|-----|--|--|
| 1. | <i>Uses case</i>  | Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukarpesan antar unit atau actor, biasanya dinyatakan dengan menggunakan kata kerja di awal di awal frase nama <i>use case</i> . |
| 2. | <i>Actor / actor</i>  | 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 actor adalah gambar orang, tapi actor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor. |
| 3. | <i>Asosiasi / 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. | <i>Ekstensi / 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</i> |

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

| | | |
|----|-----------------------------------|---|
| | | <p>case yang ditambahkan, misal</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> |
| 5. | Generalisasi / generalization | <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-simbol *Use Case Diagram*

| | | |
|-----------|--|--|
| <p>6.</p> | <p>Menggunakan / <i>include</i> / <i>uses</i></p> <p><< include >></p>  <p><< uses >></p>  | <p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini</p> <p>Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i>:</p> <ul style="list-style-type: none"> • <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan, misal pada kasus berikut:  <ul style="list-style-type: none"> • <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>Kedua interpretasi diatas dapat dianut salah satu atau keduanya tergantung ada pertimbangan dan interpretasi yang dibutuhkan.</p> |
|-----------|--|--|

2.2.3 Pengertian *Activity Diagram*

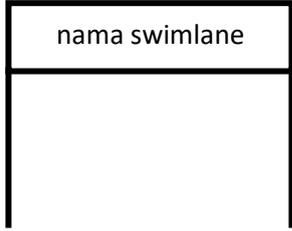
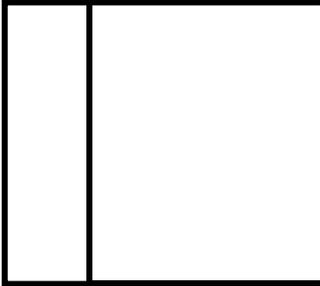
Rosa & Shalahuddin (2018:161) mengemukakan, “Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah

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 actor, jadi aktivitas yang dapat dilakukan oleh sistem. Menurut Rosa & Shalahuddin (2018:162—163) simbol- simbol yang ada pada diagram aktivitas:

Tabel 2.2 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. | Pecabangan / <i>decision</i>  | Asosiasi pecabangan 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 sistem, sebuah diagram aktivitas memiliki sebuah status akhir. |
| 6. | Swimlane | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi. |

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

| | | |
|--|--|--|
| |  | |
| | atau | |
| |  | |

2.2.4 Pengertian *Class Diagram*

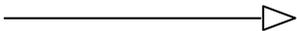
Menurut Rosa & Shalahuddin (2018:141) diagram kelas atau *class diagram* digunakan untuk menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.

Menurut Rosa & Shalahuddin (2018:146-147) simbol-simbol yang ada pada diagram kelas:

Tabel 2.3 Simbol-simbol *Class Diagram*

| No. | Simbol | Deskripsi |
|-----|--|---|
| 1. | Kelas  | Kelas pada struktur sistem. |
| 2. | Antarmuka / <i>interface</i>  nama_interface | Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek. |

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

| | | |
|----|---|--|
| 3. | Asosiasi / association  | Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> . |
| 4. | Asosiasi berarah / <i>directed association</i>  | Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> . |
| 5. | Generalisasi  | Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus). |
| 6. | Kebergantungan / <i>dependency</i>  | Relasi antar kelas dengan makna kebergantungan antar kelas. |
| 7. | Agregasi / aggregation  | Relasi antar kelas dengan makna semua-bagian (whole-part). |

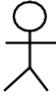
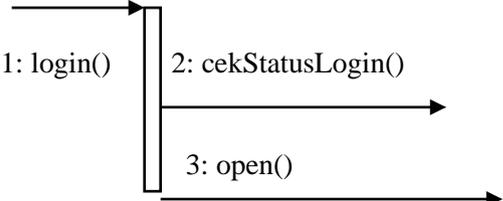
Sumber: Rosa & Salahudin, 2018.

2.2.5 Pengertian *Sequence Diagram*

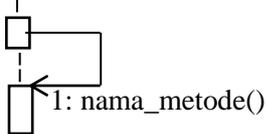
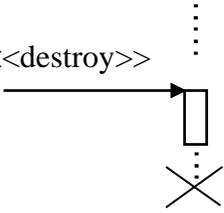
Menurut Rosa & Shalahuddin (2018:165) Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah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

Menurut Rosa & Shalahuddin (2018:165-167) simbol-simbol yang ada pada *sequence diagram* yaitu:

Tabel 2.4 Simbol-simbol *Sequence Diagram*

| No. | Simbol | Deskripsi |
|-----|--|--|
| 1. | <p>Aktor</p>  <p>atau</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> <u>nama aktor</u> </div> <p>tanpa waktu aktif</p> | <p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang kan 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> |
| 2. | <p>Garis hidup / lifeline</p>  | <p>Menyatakan kehidupan suatu objek.</p> |
| 3. | <p>Objek</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> <u>nama objek : nama kelas</u> </div> | <p>Menyatakan objek yang berinteraksi pesan.</p> |
| 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> |

Lanjutan Tabel 2.4 Simbol-simbol *Sequence Diagram*

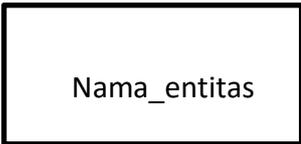
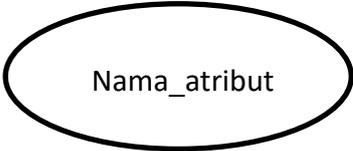
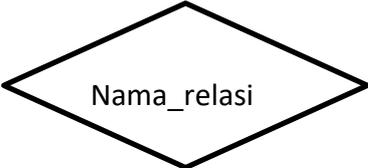
| No. | Simbol | Deskripsi |
|-----|--|---|
| 5. | Pesan tipe create <<create>>  | Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat. |
| 6. | Pesan tipe call 1: nama_metode()  | Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri,  Arah panah mengarah pada objek yang memiliki operasi / metode, karena ini memanggil operasi / metode maka operasi / metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi. |
| 7. | Pesan tipe send 1: masukan  | Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim. |
| 8. | Pesan tipe return 1: keluaran  | Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian. |
| 9. | Pesan tipe destroy <<destroy>>  | Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy. |

2.2.6 Pengertian *Entity Relationship Diagram*

Rosa, Shalahuddin (2018:50), ERD adalah pemodelan awal basis data yang dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional.

Rosa, Shalahuddin (2018:50) menjelaskan simbol-simbol yang digunakan dalam ERD, yaitu :

Tabel 2.5. Simbol-Simbol *Entity Relationship Diagram*

| No. | Simbol | Deskripsi |
|-----|---|--|
| 1. | Entitas/ entity  | Entitas merupakan data inti yang akan disimpan. |
| 2. | Atribut  | Field atau kolom data yang butuh disimpan dalam suatu entitas. |
| 3. | Relasi  | Relasi yang menghubungkan antar entitas. |
| 4. | Asosiasi / association  | Penghubung antar himpunan relasi dan entitas dimana kedua ujungnya memiliki multiplicity kemungkinan jumlah pemakaian. |

(Sumber: Rosa, Shalahuddin (2018:50))



2.2.7 Pengertian Kamus Data

Rosa dan Shalahudin, (2018:73), “Kamus data (*data dictionary*) adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingamasukan (*input*) dan keluaran (*output*) dapat dipahami secara umum”.

Rosa dan Shalahudin, (2018:74), menjelaskan simbol-simbol yang di gunakan dalam kamus data, yaitu:

Tabel 2.6 Simbol-simbol dalam Kamus Data

| Simbol | Keterangan |
|------------------|---------------------------------|
| = | Disusun atau terdiri atas |
| + | Dan |
| [] | Baik....atau.... |
| { } ⁿ | n kali diulang/bernilai banyak. |
| () | Data opsional |
| *...* | Batas komentar |

Sumber: Rosa &Salahudin, 2018.

2.2.8 Pengertian *Blackbox Testing*

Menurut Wijaya & Astuti (2021:23) “*Black box testing* merupakan pengujian kualitas perangkat lunak yang berfokus pada fungsionalitas perangkat lunak. Pengujian black box bertujuan untuk menemukan fungsi yang tidak benar, kesalahan antarmuka, kesalahan pada struktur data, kesalahan performansi, kesalahan inisialisasi dan terminasi”.

Blackbox Testing adalah sebuah metode yang dipakai untuk menguji sebuah software tanpa harus memperhatikan detail software. Pengujian ini hanya memeriksa nilai keluaran berdasarkan nilai masukan masing- masing. Tidak ada upaya untuk mengetahui kode program apa yang output pakai. Tahapan pertama pada pengujian dengan menggunakan *Blackbox Testing* adalah mengidentifikasi masukan lalu diuji agar kita mengetahui letak kesalahannya (Ningrum, dkk., 2019).

Dari beberapa pengertian di atas, penulis menyimpulkan bahwa *Blackbox Testing* adalah sebuah metode pengujian aplikasi atau program yang sedang



dikembangkan dengan mengamati hasil *input* dan *output* dari *software* tanpa mengetahui struktur kode dari perangkat lunak.

2.3 Pengertian Judul

2.3.1 Pengertian Aplikasi

Menurut Habibi & Karnovi (2020:14) aplikasi adalah sebuah program siap pakai yang dipakai untuk menjalankan sebuah perintah dari pengguna aplikasi itu sendiri. Dengan tujuan untuk memperoleh hasil yang lebih akurat dan sesuai dengan tujuan pembuatan aplikasi tersebut.

Menurut Firdaus & Ramadhan (2020:14) menyatakan bahwa aplikasi adalah suatu program yang siap digunakan dan dibuat untuk melaksanakan suatu fungsi bagi penggunanya serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang dituju.

Adapun pengertian aplikasi dari beberapa ahli menurut Zaki dan *Smitdev Community*, Aplikasi merupakan komponen yang bermanfaat sebagai media untuk menjalankan pengolahan data ataupun berbagai kegiatan lainnya seperti pembuatan ataupun pengolahan dokumen dan file.

Dari beberapa pengertian aplikasi di atas, penulis menyimpulkan bahwa aplikasi adalah perangkat lunak yang siap digunakan untuk melakukan perintah dari penggunanya seperti mengelola data, membuat ataupun mengelola dokumen dan file.

2.3.2 Pengertian Pelayanan

Pelayanan adalah setiap aktivitas yang dilakukan untuk memenuhi kebutuhan dan keinginan dari pihak lain. Pelayanan terhadap pelanggan sangat penting dilakukan perusahaan karena tanpa pelayanan yang bagus maka pelanggan tidak akan mau membeli produk atau jasa yang akan ditawarkan. Berikut pengertian pelayanan menurut beberapa ahli:

Menurut Kotler dalam Laksana (2019:85), pelayanan adalah setiap tindakan atau kegiatan yang dapat ditawarkan oleh satu pihak kepada pihak lain, yang pada dasarnya tidak berwujud dan tidak mengakibatkan kepemilikan apapun.



Menurut Casro, dkk. (2020:168) “Pelayanan adalah aktifitas yang tidak dapat digambarkan secara tersendiri karena pada bersifat *intangible* (tidak teraba), pelayanan merupakan bentuk pemenuhan kebutuhan dan tidak harus terikat pada penjualan produk atau pelayanan lain”.

Berdasarkan beberapa pengertian diatas, dapat disimpulkan bahwa pelayanan adalah suatu kegiatan yang dilakukan untuk orang lain yang sesuai dengan yang diinginkan dan diharapkan oleh konsumen.

2.3.3 Pengertian Pengaduan

Menurut Tolba, dkk. dalam Amalia, dkk. (2020) menguraikan bahwa pengaduan adalah semua komunikasi tertulis, lisan atau elektronik yang digunakan pelanggan untuk menyatakan ketidaksenangan mereka mengenai kekurangan atau kegagalan suatu produk atau jasa. Pengaduan menjadi sebuah sumber informasi berharga tentang bagaimana dan dimana kesalahan organisasi dalam melakukan pelayanan.

Pelayanan Dalam lingkup pelayanan publik, Peraturan Presiden Republik Indonesia Nomor 76 Tahun 2013 tentang Pengelolaan Pengaduan Pelayanan Publik juga memberikan penjelasan yang spesifik. Pada Pasal 1 angka 8 peraturan tersebut, pengaduan dimaknai sebagai penyampaian keluhan yang disampaikan pengadu kepada pengelola pengaduan pelayanan publik atas pelayanan pelaksana yang tidak sesuai dengan standar pelayanan, atau pengabaian kewajiban dan/atau pelanggaran larangan oleh penyelenggara.

Dari beberapa pengertian diatas, penulis menyimpulkan bahwa pengaduan adalah penyampaian atau pemberitahuan yang disampaikan oleh pengadu kepada penyedia layanan tentang standar pelayanan yang disediakan.

2.3.4 Pengertian Keluhan

Menurut Luddington dalam Hadi & Artadita (2022: 2) menyatakan bahwa keluhan adalah masukan untuk perusahaan yang sifatnya negatif, masukan dapat dilakukan secara tertulis atau lisan.



Menurut Rayasa, dkk. (2019:234) menyatakan bahwa keluhan (*complain*) adalah sebuah kesalahan, masalah, stres, frustrasi, tuntutan dan sejenisnya yang selalu dianggap negatif bagi kedua belah pihak yaitu pelanggan dan perusahaan. Keluhan merupakan bagian dari proses purna beli dimana berawal dari konsumen yang merasakan ketidakpuasan setelah menerima pelayanan atau melakukan transaksi.

Dari beberapa pengertian diatas, penulis menyimpulkan bahwa keluhan adalah pengaduan atau penyampaian ketidakpuasan, ketidaknyamanan, kejengkelan, dan kemarahan atas pelayanan jasa atau produk.

2.3.5 Pengertian Jasa

Kotler dan Ketler dalam Wijaya (2018:2) mengemukakan pengertian jasa sebagai setiap tindakan atau kegiatan yang dapat ditawarkan satu pihak kepada pihak lain, yang bersifat tidak berwujud dan tidak mengakibatkan kepemilikan apa pun.

Definisi jasa menurut Alex dalam Wijaya (2018:2) adalah produk tidak berwujud yang melibatkan perbuatan, kinerja, atau usaha yang secara fisik tidak dapat dimiliki.

Berdasarkan beberapa definisi di atas, maka jasa pada dasarnya merupakan proses aktivitas yang diberikan kepada konsumen yang tidak berwujud dan memiliki nilai bagi konsumen karena dapat memenuhi kebutuhannya.

2.3.6 Pengertian Porter

“Porter adalah sebutan bagi seseorang yang biasanya membawakan barang bawaan orang lain” (Torch.id, 2022). Porter termasuk ke dalam profesi yang ada di Indonesia dan dapat ditemukan pada berbagai tempat transportasi umum. Orang-orang yang bekerja sebagai porter umumnya tidak memiliki pemasukan atau gaji yang tetap seperti karyawan tetap lainnya. Mereka hanya mengandalkan masyarakat yang ingin menggunakan jasa porter untuk membawa berbagai barang bawaan. Tarif yang dikeluarkan untuk menggunakan porter berbeda-beda tergantung lembaga porter atau kesepakatan yang dibuat (Torch.id, 2022).



2.3.7 Pengertian *Website*

Menurut Awal (2019:1) “*Website* adalah halaman informasi yang disediakan melalui jaringan internet sehingga bisa di akses diseluruh dunia Selama masih tetkoneksi dengan intrenet. *Website* merupakan komponen atau kumpulan komponen yang terdiri dari teks, gambar, suara animasi, sehingga lebih merupakan media informasi yang menarik untuk dikunjungi.”

Menurut Elgamar dalam Novria, dkk. (2022) *Website* merupakan sebuah media yang memiliki banyak halaman yang saling terhubung (*hyperlink*), berfungsi memberikan informasi berupa teks, gambar, video, suara dan animasi atau penggabungan dari semuanya. Karakteristik utama dari *website* adalah halaman-halaman yang saling terhubung dan dilengkapi dengan domain sebagai alamat (url) atau *World Wide Web* (www) dan juga *hosting* sebagai media penyimpanan banyak data.

Dari beberapa pengertian di atas, penulis menyimpulkan bahwa *website* adalah halaman yang saling terhubung dan berisi informasi data digital yang berfungsi memberikan informasi berupa teks, gambar, video, suara, atau berbagai bentuk media lainnya.

2.3.8 Pengertian Aplikasi Pelayanan dan Pengaduan Keluhan Jasa Porter pada CV Buana Adhi Karya Berbasis *Website*

Aplikasi pelayanan dan pengaduan keluhan jasa porter pada CV Buana Adhi Karya berbasis *Website* adalah aplikasi yang digunakan oleh pelanggan atau pengguna jasa porter sebagai media dalam menyampaikan keluhan yang terjadi ketika menggunakan jasa porter dari CV Buana Adhi Karya.

2.4 Teori Program

2.4.1 Pengertian *Bootstrap*



Gambar 2.4 Logo *Bootstrap*

Menurut Abdulloh (dalam Susilawati, dkk. 2020) “*Bootstrap* merupakan salah satu *framework* CSS yang sangat populer di kalangan pecinta pemrograman *website*. Dengan menggunakan *bootstrap*, proses desain *website* tidak dibuat dari nol, sehingga proses desain *website* lebih cepat dan mudah. Bahkan tanpa kita membuat skrip CSS sedikit pun, kita sudah dapat membuat desain *website* yang bagus.”

Menurut Suprayogi dan Rahmanesa (2019 : 120) “*Bootstrap* adalah *front-end framework* yang bagus dan luar biasa yang mengedepankan tampilan untuk *mobile device* (Handphone, smartphone dll.) guna mempercepat dan mempermudah pengembangan *website*. *Bootstrap* menyediakan HTML, CSS dan *Javascript* siap pakai dan mudah untuk dikembangkan.”

Berdasarkan beberapa pengertian di atas, dapat disimpulkan bahwa *Bootstrap* adalah kerangka kerja CSS yang *open source* dan bebas, digunakan untuk merancang tampilan situs *web* agar lebih mudah dan lebih efisien.

2.4.2 Pengertian *Framework CodeIgniter*



Gambar 2.5 Logo *CodeIgniter*



Menurut Sallaby & Kanedi (dalam Ridwan, dkk., 2022) mengatakan bahwa *CodeIgniter* adalah sebuah *framework* yang dibuat menggunakan bahasa pemrograman PHP yang bertujuan untuk memudahkan para *programmer web* untuk membuat atau mengembangkan aplikasi berbasis *web*. *CodeIgniter* memiliki eksekusi tercepat dibandingkan dengan *framework* lainnya. *CodeIgniter* bersifat *open source* dan menggunakan model basis MVC (*Model View Controller*), yang merupakan model konsep modern saat ini. Metode MVC (*Model View Controller*) terdapat tiga komponen menurut Yesputra, dkk. (dalam Ridwan, dkk., 2022) yaitu:

1. *Model*, mengelola basis data (RDBMS) seperti MySQL ataupun Oracle RDMS. *Model* berhubungan dengan *database* sehingga biasanya dalam model akan berisi class ataupun fungsi untuk membuat (*create*), melakukan pembaruan (*update*), menghapus data (*delete*), mencari data (*search*), dan mengambil data (*select*) pada *database*. Selain itu juga *model* akan berhubungan dengan perintah-perintah *query* sebagai tindak lanjut dari fungsi-fungsi di atas.

2. *View*, bagian *User Interface* atau bagian yang nantinya merupakan tampilan untuk *end-user*. *View* bisa berupa halaman HTML, CSS, Javascript, JQuery dan AJAX, karena metode yang dipakai merupakan MVC sehingga *view* tidak boleh terdapat pemrosesan data ataupun pengaksesan yang berhubungan dengan *database*, sehingga *view* hanya menampilkan data-data hasil dari *Model* dan *Controller*.

3. *Controller*, penghubung antara *view* dan *model*, maksudnya ialah karena *model* tidak dapat berhubungan langsung dengan *view* ataupun sebaliknya, jadi *controller* inilah yang digunakan sebagai jembatan keduanya. Sehingga tugas *controller* ialah sebagai pemrosesan data atau Alur Logic Program, menyediakan variabel yang akan ditampilkan di *view*, pemanggilan *model* sehingga *model* dapat mengakses *database*, *error handling* validasi atau cek terhadap suatu input data.

Berdasarkan uraian di atas, dapat disimpulkan bahwa *Framework CodeIgniter* adalah sebuah *framework* yang menggunakan bahasa pemrograman PHP yang digunakan untuk mempermudah *programmer* dalam mengembangkan atau membuat suatu *web*. *CodeIgniter* menggunakan model basis MVC atau terdiri dari *Model*, *View* dan *Controller*.

2.4.3 Pengertian CSS



Gambar 2.6 Logo CSS

Menurut Abdulloh (2019:45) “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 Lewenusa (2020:1) “CSS (*Cascading Style Sheet*) secara sederhana adalah sebuah metode yang digunakan untuk mempersingkat penulisan tag HTML, seperti *font*, *color*, *text* dan tabel menjadi lebih ringkas sehingga tidak terjadi pengulangan penulisan”.

Berdasarkan beberapa pengertian di atas, dapat disimpulkan bahwa CSS adalah sebuah Bahasa sederhana yang digunakan untuk menambahkan gaya/*styling* (misalnya *font*, warna, spasi dan lain-lain) ke sebuah halaman *web*.

2.4.4 Pengertian HTML



Gambar 2.7 Logo HTML

Menurut Hidayatullah (dalam Awal, 2019:69) menyatakan bahwa HTML adalah suatu metode untuk mengimplementasikan konsep *Hypertext* dalam suatu

naskah atau dokumen. HTML tidak tergolong ke dalam bahasa pemrograman karena sifatnya yang memberikan tanda pada suatu naskah atau program.

Menurut Khozaimi (2020:1) “*Hypertext Markup Language (HTML)* merupakan standard bahasa yang diaplikasikan dalam penampilan halaman *web* dan terdiri dari beberapa elemen yang diwakili beberapa tag”.

Berdasarkan beberapa pengertian di atas, dapat disimpulkan bahwa HTML adalah sebuah bahasa yang menggunakan tanda-tanda tertentu (*tag*) untuk menyatakan kode-kode yang harus ditafsirkan oleh *browser* agar dapat menampilkan halaman dengan benar.

2.4.5 Pengertian MySQL



Gambar 2.8 Logo MySQL

Menurut Hidayatullah (dalam Novria, dkk. 2022) menyatakan bahwa MySQL adalah salah satu aplikasi *Database Management System (DBMS)* yang sudah sangat banyak digunakan oleh pemrogram aplikasi *web*. Kelebihan MySQL adalah dapat diakses secara gratis, handal, *update*, dan memiliki banyak forum yang memfasilitasi jika ada kendala.

Harianto, dkk (2019:13-14) mengatakan, “MySQL adalah salah satu jenis *database server* yang sangat terkenal dan banyak digunakan untuk membangun aplikasi *web* yang data base sebagai sumber dan pengelolaan datanya”.

Kepopuleran MySQL antara lain karena MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses database-nya sehingga mudah untuk digunakan, kinerja query cepat, dan mencakupi untuk kebutuhan database perusahaan-perusahaan skala menengah hingga yang kecil. Selain itu MySQL juga bersifat *open source* dan *free*. Terdapat tiga perintah SQL, yaitu DDL, DML, dan DCL.

1. *Data Definition Language* (DDL) DDL merupakan perintah SQL yang berhubungan dengan pendefinisian suatu struktur database, dalam hal ini database dan tabel. Beberapa perintah dasar yang termasuk DDL ini antara lain CREATE, ALTER, dan DROP.
2. *Data Manipulation Language* (DML) DML merupakan perintah SQL yang berhubungan dengan memanipulasi atau pengolahan data atau record dalam tabel. Perintah SQL yang termasuk dalam DML antara lain SELECT, INSERT, UPDATE, dan DELETE.
3. *Data Control Language* (DCL) DCL merupakan perintah SQL yang berhubungan dengan pengaturan hak akses user MySQL, baik terhadap server, database, tabel maupun field. Perintah SQL yang termasuk dalam DCL antara lain GRANT, dan REVOKE. Jadi, disimpulkan bahwa MySQL adalah database yang banyak digunakan oleh pengguna untuk membangun aplikasi *web*.

Berdasarkan beberapa pengertian di atas, dapat disimpulkan bahwa *MySQL* adalah sebuah sistem manajemen basis data yang menggunakan Bahasa SQL sebagai Bahasa penghubung antara perangkat lunak dengan *database server*.

2.4.6 Pengertian PHP



Gambar 2.9 Logo PHP

Menurut Hidayatullah (2021:257) “PHP adalah bahasa *scripting* yang bersifat *open source*, sangat cocok digunakan untuk pengembangan *web* dan dapat disematkan ke dalam HTML.”

Menurut Enterprise (2018:1-2) PHP merupakan bahasa pemrograman untuk membuat *website* yang dinamis dan interaktif. Dinamis artinya *website*

tersebut bisa berubah - ubah tampilan dan kontennya sesuai kondisi tertentu. Interaktif artinya PHP dapat memberi *feedback* bagi *user*.

Secara khusus, PHP dirancang untuk membentuk aplikasi *web* dinamis. Artinya, ia dapat membentuk suatu tampilan berdasarkan permintaan terkini. Misalnya, kita dapat menampilkan isi *database* ke halaman *web*. Pada prinsipnya PHP mempunyai fungsi yang sama dengan script-script seperti ASP (*Active Server Page*), *cold fusion*, ataupun Perl. Namun, perlu diketahui PHP sebenarnya bisa dipakai secara *common line* artinya script PHP (*Hypertext Preprocessor*) dapat dijalankan tanpa melibatkan *web server* maupun *browser*.

2.4.7 Pengertian *Sublime Text*



Gambar 2.10 Logo *Sublime Text*

“*Sublime Text* merupakan perangkat lunak *text editor* yang digunakan untuk membuat atau mengedit suatu aplikasi. *Sublime Text* memiliki plug in tambahan yang memudahkan programmer. Selain itu, *Sublime Text* juga memiliki desain yang simpel dan keren sehingga terlihat elegan untuk sebuah *syntax editor*” (Supono, 2018:14).

Menurut Karmeila (dalam Susilawati, dkk., 2020) “*Sublime Text 3* adalah sebuah software yang dikembangkan oleh Jon Skinner. Beliau merupakan seorang programmer dari Australia. *Sublime Text 3* merupakan aplikasi *text editor* untuk menulis kode. Banyak sejumlah bahasa program yang ada pada aplikasi ini. Diantaranya PHP, CSS, C, C++, HTML, ASP, Java, dan sebagainya. Tentu saja, *software* ini bisa lebih memudahkan pekerjaan pengguna saat membuat sebuah program.”

Berdasarkan beberapa pengertian di atas, dapat disimpulkan bahwa *Sublime Text* adalah perangkat lunak yang berfungsi sebagai *editor* kode yang digunakan seorang *programmer* untuk membuat suatu program.

2.4.8 Pengertian *phpMyAdmin*



Gambar 2.11 Logo *phpMyAdmin*

phpMyAdmin adalah alat perangkat lunak gratis yang ditulis dalam PHP yang dimaksudkan untuk menangani administrasi MySQL melalui internet (Yudayanto, 2018:5).

Menurut Risal, dkk. (2020:33) *phpMyAdmin* adalah perangkat lunak bebas yang ditulis dalam bahasa pemrograman PHP yang digunakan untuk menangani administrasi MySQL melalui *world wide web*. *phpMyAdmin* mendukung berbagai operasi MySQL, diantaranya mengolah basis data, tabel-tabel, fields, relasi, indeks, *user*, *permissions*, dan lain-lain.

Berdasarkan beberapa pengertian di atas, dapat disimpulkan bahwa *phpMyAdmin* adalah perangkat lunak yang bersifat *open source* dan gratis yang ditulis dengan bahasa pemrograman PHP yang digunakan untuk menangani segala administrasi MySQL.

2.4.9 Pengertian XAMPP



Gambar 2.12 Logo XAMPP

Menurut Putra (2020:29) XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan komplikasi dari beberapa program



yang berfungsi sebagai *server* yang berdiri sendiri (local host) , yang terdiri atas Apache, MySQL *database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl.

Menurut Hidayatullah (2021:161) menyatakan bahwa XAMPP ialah paket *software* yang di dalamnya terdapat *server MySQL* dan didukung oleh PHP sebagai bahasa pemrograman untuk membuat *website* dinamis serta terdapat *web server apache* yang dapat dijalankan di beberapa platform seperti OS X, Windows, Linux, Max.

XAMPP berguna untuk menjalankan Apache, MariaDB, dan PHP pada *localhost* atau komputer tanpa harus ada koneksi internet. Dengan adanya XAMPP akan mempermudah pekerjaan *frontend* dan *backend developer*, karena mereka dapat melakukan testing pada program sebelum nantinya akan diupload ke server online *website*.

XAMPP disebut juga sebagai *standalone server* atau *server* yang dapat berdiri sendiri sehingga memudahkan pengguna saat menjalankan proses pengeditan, desain, dan pengembangan aplikasi. Penggunaan XAMPP dirasa mampu menghemat anggaran karena dapat menggantikan peran *web hosting* dengan cara menyimpan file *website* ke dalam *localhost* agar bisa dipanggil atau dihubungkan melalui browser.

Adapun XAMPP memiliki kepanjangan X (*cross platform*), A (Apache), M (MySQL/MariaDB), P (PHP), dan P (Perl). Kepanjangan XAMPP tersebut merupakan program yang tersedia pada *software* ini.

a. X (Cross Platform)

Kode ini merupakan kode penanda untuk *software cross platform* atau yang bisa berjalan di banyak sistem operasi, seperti Windows, Linux, MacOS, dan Solaris.

b. A (Apache)

Berarti *web server* gratis dan bisa dikembangkan oleh banyak orang (*open source*) untuk menciptakan halaman *web* yang benar. Dengan menggunakan *web server*, pengguna dapat menjalankan file yang berisi bahasa pemrograman



PHP pada *localhost*.

c. M (MySQL/MariaDB)

MySQL adalah salah satu aplikasi *database server* dengan bahasa pemrograman *structured query language* (SQL) yang berfungsi untuk mengelola data secara terstruktur dan sistematis. Misalnya, mengolah, mengedit, dan menghapus daftar melalui database. *MySQL* juga dapat digunakan di *localhost* tanpa memerlukan sambungan internet. Dengan demikian, *developer* maupun *programmer* dapat membuat aplikasi berbasis *web* di komputernya. Sedangkan, MariaDB merupakan sistem manajemen *database* yang merupakan bagian dari pengembangan mandiri dari *MySQL*.

d. P (PHP)

Hypertext Preprocessor (PHP) merupakan bahasa pemrograman khusus berbasis *web* untuk kebutuhan pada sisi *server* atau *back end* sehingga dapat digunakan untuk membuat halaman *web* menjadi lebih dinamis dengan menerapkan *server-side scripting*. Selain itu, PHP juga mendukung manajemen sistem pada Oracle, Postgresql, Microsoft Access, dan lain sebagainya.

e. P (Perl)

Perl adalah bahasa pemrograman yang dapat digunakan untuk segala kebutuhan atau *cross platform*. Perl ini bisa berjalan di dalam banyak sistem operasi sehingga sangat fleksibel dan banyak digunakan. Bahasa pemrograman ini juga berfungsi sebagai penunjuk eksistensi dari PHP. Saat ini, Perl banyak digunakan untuk keperluan pengembangan aplikasi hingga *web server* dan banyak juga digunakan untuk *website development* pada sistem berbasis *Content Management System* (CMS), seperti WordPress.

Dalam penggunaannya, XAMPP memiliki banyak sekali manfaat. Berikut beberapa fungsi dari tool *web server* open source ini.

1. Mengkonfigurasi pengaturan database PhpMyAdmin
2. Menjalankan *framework* PHP secara offline
3. Melakukan proses install WordPress offline



4. Melakukan pengujian fitur dan mengakses *web* tanpa internet

Adapun Komponen penting XAMPP adalah sebagai berikut :

- a. **Control panel**, merupakan layanan yang digunakan untuk mengelola XAMPP. Mulai dari mengatur penggunaan *database*, mengupload file, melakukan konfigurasi proyek *web*, dan sejumlah fungsi fitur lain. Control panel ini kurang lebih sama dengan fitur yang terdapat pada cPanel atau Plesk yang digunakan untuk kebutuhan *hosting* yang bersifat *online*.
- b. **HTDocs**, merupakan komponen XAMPP yang berbentuk *folder public* dan berfungsi sebagai tempat penyimpanan folder serta file yang dapat ditampilkan melalui *browser*.
- c. **Config**, merupakan menu yang digunakan untuk melakukan konfigurasi dasar pada XAMPP, seperti mengatur aplikasi *editor* teks dan *browser* yang akan digunakan secara default oleh XAMPP.
- d. **PhpMyAdmin**, merupakan komponen XAMPP yang berfungsi untuk mengelola *database* melalui *browser*.
- e. **Netstat**, merupakan menu yang digunakan untuk memastikan port yang digunakan oleh XAMPP telah digunakan oleh aplikasi lain atau belum. Bila port standar XAMPP sudah digunakan oleh aplikasi lain, maka XAMPP tidak dapat berfungsi dengan baik. Dengan demikian, pengguna harus mengubah port tersebut dengan yang lainnya yang masih belum digunakan.

Berdasarkan penjelasan di atas, dapat disimpulkan XAMPP adalah perangkat lunak yang bersifat *open source* yang berisi kompilasi beberapa program yang dapat digunakan untuk menguji program yang dibuat tanpa terhubung ke internet. Melalui XAMPP, pengguna dapat mengelola *database* yang berada di *localhost* tanpa memerlukan akses internet sehingga jika koneksi internet terganggu dan tidak dapat mengakses *web server*, pengguna tidak lagi perlu khawatir.