



## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Teori Umum

##### 2.1.1. Pengertian Komputer

Menurut Wahyudin dan Munir (2018:1), Komputer adalah suatu alat elektronik yang mampu melakukan beberapa tugas, yaitu menerima *input*, memproses *input* sesuai dengan instruksi yang diberikan, menyimpan perintah-perintah dan hasil pengolahannya, serta menyediakan *output* dalam bentuk informasi.

Menurut William dalam Irma (2016:2), Komputer adalah suatu pemrosesan data yang dapat melakukan perhitungan yang benar dan cepat, termasuk perhitungan aritmatika yang besar atau operasi logikan tanpa campur tangan dari manusia dalam pengoperasiannya selama pemrosesan.

Dari definisi diatas maka Komputer dapat diartikan sebagai suatu alat elektronik yang dapat melakukan beberapa tugas secara benar dan cepat dengan cara menyimpan *input*, melakukan pemrosesan, dan menghasilkan *output* berdasarkan intruksi program yang tersimpan.

##### 2.1.2. Pengertian Informasi

Menurut Rusmawan (dalam Gaol, 2019:31), Informasi adalah data yang telah diproses atau diolah ke dalam bentuk yang sangat berarti untuk penerimanya dan merupakan nilai yang sesungguhnya atau dipahami dalam tindakan atau keputusan yang sekarang atau nantinya.

Selain itu Menurut Hutahaeen (2015:9), Informasi adalah data yang diolah menjadi bentuk yang lebih berarti bagi penerimanya.

Dari definisi diatas maka Informasi adalah data yang telah diproses menjadi bentuk yang lebih berarti, dan mudah dipahami dalam tindakan atau keputusan bagi penerimanya.



### 2.1.3. Pengertian Data

Menurut Rusmawan (2019:34), Data adalah catatan atas kumpulan fakta. Data merupakan bentuk jamak dari datum, berasal dari bahasa latin yang berarti sesuatu yang diberikan.

Menurut Hutahean (2015:8), Data adalah bahan mentah bagi informasi, dirumuskan sebagai kelompok lambing-lambang tidak acak menunjukkan jumlah-jumlah, tindakan-tindakan, hal-hal dan sebagainya.

Dari definisi diatas maka Data merupakan bahan mentah yang belum diolah dan harus diolah melalui proses sehingga dapat menghasilkan informasi yang dapat berguna bagi pengguna/user.

### 2.1.4. Pengertian Database

Menurut Pamungkas (2017:2). *Database* atau basis data merupakan suatu kumpulan data terhubung yang disimpan secara bersama-sama pada suatu media, yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, dan dengan *software* untuk melakukan manipulasi untuk kegunaan tertentu. Database dapat diartikan juga sebagai sekumpulan data yang disusun dalam beberapa tabel yang saling memiliki relasi maupun berdiri sendiri.

Menurut Sukamto (2018:43), Basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat.

Dari definisi diatas maka *Database* adalah sekumpulan data yang disusun dalam beberapa tabel untuk menyimpan data agar dapat diakses dengan cepat dan mudah bagi pengguna/user.

## 2.2. Teori Khusus

### 2.2.1. Pengenalan *Unified Modelling Language*

Menurut Sukamto (2018:13), *Unified Modeling Language* (UML) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan

---

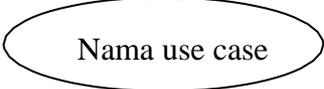
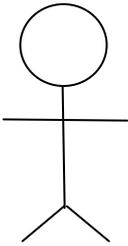


menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek.

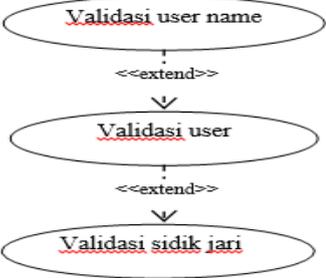
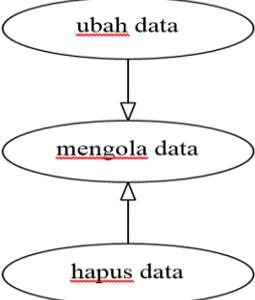
### 2.2.2. Usecase Diagram

Sukanto (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 aktor 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. Berikut adalah simbol-simbol yang ada pada diagram *use case* :

**Tabel 2.1.** Simbol-simbol diagram *use case*

| No. | Simbol  | Deskripsi   |
|-----|---|---|
| 1.  |  | Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i> .  |
| 2.  |  | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor. |
| 3.  |  | Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.  |

Lanjutan Tabel 2.1. Simbol-simbol diagram *use case*

| No. | Simbol   | Deskripsi  |
|-----|--|--|
| 4.  | Exstensi/ <i>extend</i><br><br><<extend>>  | <p>Relasi <i>use case</i> tambahan kesebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu, mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek, biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, missal</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/ <i>generalization</i><br><br> | <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>  |

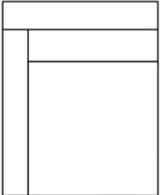




### 2.2.3. Activity Diagram

Sukamto (2018:161), diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Berikut adalah simbol-simbol yang ada pada diagram aktivitas :

**Tabel 2.2.** Simbol-simbol *activity diagram*

| No. | Simbol  | Deskripsi   |
|-----|---|---|
| 1.  | Status awal<br>                    | Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.             |
| 2.  | Aktivitas<br>                    | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.                  |
| 3.  | Percabangan/ <i>decision</i><br> | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.                         |
| 4.  | Penggabungan/ <i>join</i><br>    | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.                |
| 5.  | Status akhir<br>                 | Status akhir yang dilakukan oleh sistem, sebuah diagram aktivitas memiliki sebuah status akhir. |
| 6.  | Swimlane<br>                     | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.            |

**Sumber :** Sukamto (2018:162-163)

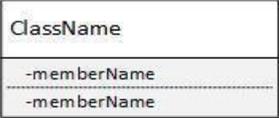
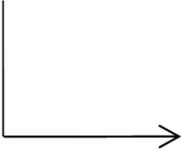
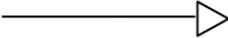


### 2.2.4. Class Diagram

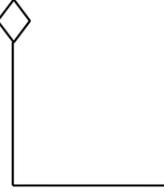
Sukamto (2018:141), diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan *method* atau operasi. Berikut penjelasan atribut dan *method* :

1. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau *method* adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

**Tabel 2.3** Simbol-simbol *class diagram*

| No. | Simbol   | Deskripsi  |
|-----|--|--|
| 1.  | Kelas<br>  | Kelas pada struktur sistem   |
| 2.  | Antarmuka/ <i>interface</i><br>                   | Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek   |
| 3.  | Asosiasi/ <i>association</i><br>                  | Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>   |
| 4.  | Asosiasi berarah/ <i>directed association</i><br> | Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> |
| 5.  | Generalisasi<br>                                  | Relasi antar kelas dengan makna generalisasi – spesialisasi (umum - khusus)  |

Lanjutan Tabel 2.3 Simbol-simbol *class diagram*

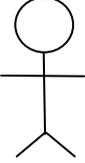
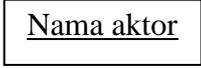
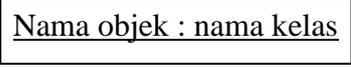
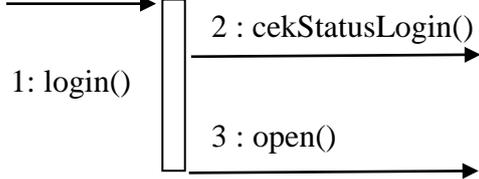
| No. | Simbol  | Deskripsi  |
|-----|---|--|
| 6.  |  | Relasi antar kelas dengan makna kebergantungan antar kelas         |
| 7.  |  | Relasi antar kelas dengan makna semua-bagian ( <i>whole-part</i> ) |

Sumber : Sukamto (2018:146-147)

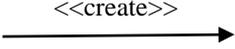
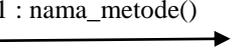
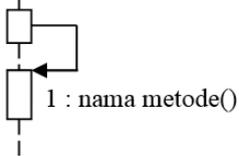
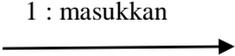
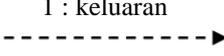
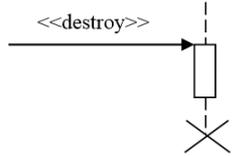
### 2.2.5. Pengertian *Sequence Diagram*

Sukamto (2018:165), diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dengan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan 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*. Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup dalam diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak. Berikut adalah simbol-simbol yang ada pada diagram sekuen :

Tabel 2.4. Simbol-simbol *sequence diagram*

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

Lanjutan Tabel 2.4. Simbol-simbol *sequence diagram*

| No. | Simbol   | Deskripsi  |
|-----|--|--|
| 5.  | <p>Pesan tipe <i>create</i></p>     | Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat   |
| 6.  | <p>Pesan tipe <i>call</i></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> |
| 7.  | <p>Pesan tipe <i>send</i></p>     | Menyatakan bahwa suatu objek mengirimkan data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.  |
| 8.  | <p>Pesan tipe <i>return</i></p>   | Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.  |
| 9.  | <p>Pesan tipe <i>destroy</i></p>  | Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada <i>create</i> maka ada <i>destroy</i>   |

Sumber: : Sukamto (2018:165-167)



## **2.3. Teori Judul**

### **2.3.1. Pengertian Aplikasi**

Menurut Santoso (2015:9) Aplikasi adalah suatu kelompok *file* (*form, class, report*) yang bertujuan untuk melakukan aktivitas tertentu yang saling terkait.

Menurut Kadir (2017:3) Aplikasi lebih sering disebut untuk menyatakan perangkat lunak. Di kalangan profesional teknologi informasi, istilah program biasa digunakan untuk menyatakan hasil karya mereka yang berupa instruksi-instruksi untuk mengendalikan komputer. Di sisi pemakai, hal seperti itu biasa disebut sebagai aplikasi.

Dari definisi diatas maka Aplikasi adalah suatu kelompok *file* dalam komputer yang bertujuan untuk melakukan aktivitas tertentu dengan sarana komputer.

### **2.3.2. Pengertian Kuesioner**

Menurut Umar (2002:101), *kuesioner* adalah sebuah alat pengumpulan data yang nantinya data tersebut akan diolah untuk menghasilkan informasi tertentu. Sedangkan menurut Djaali dan Muljono (2007:64), *kuesioner* adalah alat pengumpul data yang berbentuk pertanyaan yang akan diisi atau dijawab oleh responden.

Berdasarkan pengertian diatas dapat disimpulkan bahwa pengertian *kuesioner* merupakan sebuah alat pengumpulan data yang berbentuk pertanyaan yang dijawab oleh responden untuk menghasilkan informasi tertentu.

### **2.3.3. Pengertian Stakeholder**

Menurut Sumarto (2003:135), *Stakeholder* adalah individu, kelompok, atau organisasi, perempuan maupun laki-laki, yang memiliki kepentingan dari suatu aktivitas atau proyek.

Menurut Thomsett (2008:80) *Stakeholder* adalah orang, kelompok, atau organisasi yang berada di luar kendali anda yang berkecimpung di dalam proyek anda.



Berdasarkan definisi diatas dapat disimpulkan bahwa *Stakeholder* dalam tulisan kali ini adalah seorang individu ataupun kelompok yang memiliki kepentingan dari perusahaan yang memakai jasa lulusan Jurusan Bahasa Inggris Politeknik Negeri Sriwijaya.

#### **2.3.4. Service Quality (SERVQUAL)**

Menurut Maryono (2018:128), Service Quality adalah suatu metode deskriptif guna menggambarkan tingkat kepuasan pelanggan. Metode ini dikembangkan tahun 1985 oleh A. Parasuraman dkk. Adanya fenomena umum bahwa pencapaian kualitas dalam hal produk dan layanan menduduki posisi sentral. Kualitas layanan adalah perbandingan antara harapan (*expectation*) dengan kinerja sebenarnya (*performance*).

*Service Quality* adalah ukuran seberapa baik suatu layanan menemui kecocokan dengan harapan pelanggan (*Lewis and Booms, 1983*). Penyelenggaraan kualitas layanan berarti melakukan kompromi dengan harapan pelanggan dengan tata cara yang konsisten.

Menurut Parasuraman dkk dalam Noer (2016), ada Lima dimensi pokok suatu kualitas jasa pada metode *Service Quality*, yaitu sebagai berikut:

1. Tampilan Elemen Fisik/Berwujud (*Tangible*)

Dimensi ini mencakup ketersediaan fasilitas fisik, peralatan, sumber daya manusia, materi-materi untuk komunikasi yang merupakan bukti nyata dari pelayanan.

2. Keandalan (*Reliability*)

Dimensi ini mencakup kemampuan perusahaan dalam memberikan layanan yang akurat sejak pertama kali tanpa membuat kesalahan dan menyampaikan jasanya sesuai dengan waktu yang telah disepakati.

3. Daya Tanggap (*Responsiveness*)

Dimensi ini mencakup kemampuan dan kesediaan para karyawan dalam membantu para pelanggan dan merespon permintaan mereka, serta menginformasikan kapan jasa akan diberikan dan memberikan jasa secara cepat.



#### 4. Jaminan/Keyakinan (*Assurance*)

Dimensi ini mencakup perilaku karyawan yang dapat menumbuhkan kepercayaan dari pelanggan terhadap perusahaan sehingga perusahaan bisa menciptakan rasa aman bagi pelanggannya. Jaminan ini berarti bahwa para karyawan menguasai pengetahuan dan keterampilan dalam menangani setiap pertanyaan atau masalah pelanggan serta selalu bersikap sopan.

#### 5. Empati (*Empathy*)

Hal ini mencakup pemahaman masalah oleh perusahaan kepada para pelanggannya dan bertindak demi kepentingan pelanggan serta memberikan perhatian personal kepada para pelanggannya dan juga jam operasi yang nyaman.

Berdasarkan pengertian diatas dapat disimpulkan bahwa Metode *Service Quality* (SERVQUAL) adalah salah satu metode yang dapat digunakan dalam pembuatan Aplikasi *Kuesioner* dengan mengukur kualitas layanan barang atau jasa berdasarkan Lima dimensi tersebut.

Setiawan dkk (2015:19) menjelaskan bahwa Metode *Service Quality* (SERVQUAL) dibangun atas adanya perbandingan dua faktor utama yaitu persepsi pelanggan atas layanan yang diterima (*Perceived Service*) dengan layanan sesungguhnya diharapkan pelanggan (*Expected Service*). Jika kenyataan lebih dari yang pelanggan harapkan, maka layanan dapat dikatakan bermutu, sedangkan jika kenyataan kurang dari yang pelanggan harapkan, maka dikatakan tidak bermutu.

Rumus kepuasan Pelanggan :

$$Q = P - E$$

Keterangan :

Q : Kualitas pelayanan pelanggan

E : Harapan Pelanggan atas Kualitas pelayanan (Harapan)

P : Pelayanan Sesungguhnya diterima (Kenyataan)



Model konsep *Service Quality* (SERVQUAL) muncul dari 5 GAP kesenjangan diantaranya :

Gap 1, antara persepsi manajemen dan ekspektasi pelanggan.

Gap 2, antara persepsi manajemen dan spesifikasi kualitas layanan.

Gap 3, antara spesifikasi kualitas layanan dan penyampaian jasa.

Gap 4, antara penyampaian jasa komunikasi eksternal

Gap 5, antara persepsi ekspektasi pelanggan.

Berdasarkan ruang lingkup sistem yang ada, penulis kali ini memfokuskan untuk menghitung nilai persepsi pelanggan untuk mendapatkan GAP ke 5.

## 2.4.1 Teori Program

### 1.1.1. Pengertian PHP

Menurut Sidik (2017:5), PHP: *HyperText Preprocessor* merupakan bahasa utama *script serverside* yang disisipkan pada HTML yang dijalankan di *server*, dan juga bisa digunakan untuk membuat aplikasi desktop.

Menurut Raharjo (2016:38), PHP adalah salah satu bahasa pemrograman skrip yang dirancang untuk membangun aplikasi. Ketika dipanggil dari *web browser*, program yang ditulis dengan PHP akan di-*parsing* di dalam *webserver* oleh *interpreter* PHP dan diterjemahkan ke dalam dokumen HTML, yang selanjutnya akan kembali ke *web browser*.



Sumber : <https://en.wikipedia.org/wiki/File:PHP-logo.svg>

Gambar 2.2 Logo PHP



### 1.1.2. Pengertian *MySql*



Sumber : <https://en.wikipedia.org/wiki/MySQL>

Gambar 2.3 Logo MySQL

Menurut Murya (2014:46), MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*Database Management System*) atau DBMS yang *mutiheard*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. Keistimewaan yang dimiliki MySQL, antara lain:

1. MySQL dapat berjalan stabil pada berbagai sistem operasi.
2. MySQL didistribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi GPL sehingga dapat digunakan secara gratis.
3. MySQL dapat digunakan beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah dan konflik.
4. MySQL memiliki kecepatan yang menakjubkan dalam menagani *query*.
5. MySQL memiliki ragam tipe data yang sangat kaya, seperti signed/unsigned integer, float, double char, text, date, timestamp, dan lain-lain.
6. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam perintah *query*.
7. MySQL memiliki beberapa lapisan keamanan seperti subnetmassk, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta sandi terenkripsi.
8. MySQL mampu menagaini basis data dalam skla besar, dengan jumlah rekaman lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris.
9. MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix soket, atau Named Pipes.
10. MySQL dapat mendeteksi pesan kesalahan pada klien.



11. MySQL memiliki antarmuka terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*)
12. MySQL dilengkapi dengan berbagai peralatan yang dapat digunakan untuk administrasi basis data.
13. MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE.

### 1.1.3. XAMPP



Sumber : [https://commons.wikimedia.org/wiki/File:Xampp\\_logo.svg](https://commons.wikimedia.org/wiki/File:Xampp_logo.svg)

Gambar 2.4 Logo XAMPP

Fungsinya adalah sebagai server yang berdiri sendiri (localhost), yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Selain itu juga fungsi XAMPP untuk web yaitu sebagai browser engine ataupun web server.

Menurut Hidayatullah dan Jauhari (2014:127), XAMPP adalah fasilitas untuk banyak sistem operasi seperti Windows, Linux, Mac, dan Solaris yang memungkinkan sebuah *web* dinamis bisa diakses secara local menggunakan *web server* local. Kata XAMPP sendiri terdiri dari :

1. X yang berarti *Cross Platform* karena XAMPP bisa dijalankan di Windows, Linux, Mac, dan Solaris.
2. A yang berarti Apache sebagai *web-server*-nya.
3. M yang berarti *MySQL* sebagai *Database Management System (DBMS)*
4. PP yang berarti *PHP* dan Perl sebagai bahasa yang didukungnya.



#### 1.1.4. *HyperText Markup Language (HTML)*

Menurut Winarno (2014:1) *Hyper Text Markup Language (HTML)* adalah sebuah bahasa untuk menampilkan konten di *web*. *HTML* mirip dengan teks biasa, hanya dalam dokumen ini tulisan mengandung beberapa instruksi yang ditandai dengan kode tertentu yang dikenal dengan tag tertentu.

Elemen *HTML* dimulai dengan *tag* awal, yang diikuti dengan isi elemen dan *tag* akhir. *Tag* terakhir termasuk simbol/diikuti oleh tipe elemen, misalnya `</HEAD>`. Sebuah elemen *HTML* dapat bersarang di dalam elemen lainnya. Sebuah dokumen *HTML* standar terlihat seperti ini :

```
<html>
  <head>
    <title> ini judul dokumen html </title>
  </head>
  <body>
    Teks ini adalah teks yang akan muncul
  </body>
</html>
```