

## **BAB II**

### **LANDASAN TEORI**

Pada landasan teori ini akan menjelaskan beberapa teori dan tinjauan pustaka yang melakukan penelitian di berbagai tempat untuk memperkuat dalam penyusunan Aplikasi reservasi pelayanan dan penyewaan pada lapangan futsal yang akan dibangun.

#### **2.1 Tinjauan Pustaka**

Terdapat beberapa penelitian terdahulu yang telah dilakukan dalam membuat Aplikasi reservasi pelayanan dan penyewaan pada lapangan futsal berbasis *web*. Contoh penelitian yang dilakukan oleh (Prasetya et al., 2018) yang berjudul “ Pengembangan Aplikasi Pemesanan Lapangan Futsal di Kota Malang Berbasis Android Menggunakan Metode Pengembangan *Extreme Programming*”. Penelitian ini membahas tentang aplikasi pemesanan lapangan futsal berbasis *Mobile Android*, android ini merupakan perangkat nomor 2 terbanyak yang digunakan di Indonesia berdasarkan survei Asosiasi Penyelenggara Jasa Internet Indonesia (APJII). Aplikasi ini akan dikembangkan menggunakan metode *Extreme Programming*. Metode ini merupakan metode yang paling sederhana dalam proses *desain* dan *codingan* dan lebih mengedepankan kebutuhan.

Penelitian Prasetya, Brata, dan Ananta mempunyai kesamaan dengan penelitian penulis berupa pemesanan lapangan futsal. Perbedaan dari penelitian ini, menggunakan aplikasi berbasis *android* dan metode yang digunakan yaitu *Extreme Programming*. Pada penelitian tersebut, belum adanya informasi penjadwalan futsal sehingga sulit bagi pengguna yang ingin melakukan pemesanan lapangan futsal karena tidak adanya informasi penjadwalan. Sedangkan penulis menggunakan aplikasi berbasis *web* dan metode yang digunakan yaitu *waterfall*. Dengan adanya informasi penjadwalan, pengguna dapat mengetahui penjadwalan lapangan futsal.

Pada penelitian selanjutnya yang dijelaskan (Bagaskara et al., 2021) dengan judul “Aplikasi Penyewaan *Sport Center D Milano Futsal*”. Pada penelitian ini menerapkan metode *Ground Research*, yang artinya metode ini digunakan berdasarkan fakta dan generalis empiris,

menetapkan konsep, membuktikan teori, mengembangkan teori, melakukan pengumpulan dan analisis data dalam waktu bersamaan. Hasil yang diperoleh dalam penelitian ini yaitu mempermudah pengguna dalam proses pengolahan data, penjadwalan yang lebih terorganisir, pembayaran lebih efektif, dan pembuatan laporan lebih mudah.

Penelitian dari Bagaskara, Aruan, dan Saputra ini memiliki kesamaan dalam pembuatan sistem yaitu melakukan pemesanan pada lapangan futsal. Perbedaan dari penelitian ini yaitu berbasis *Desktop*, dengan bahasa pemrograman *java*, peneliti lebih mengedepankan kebutuhan pengguna. Sedangkan penulis menggunakan berbasis *Website*.

Selanjutnya penelitian dari (Futsal et al., 2018) dengan judul “Pembuatan Aplikasi Penyewaan dan Penjualan Lapangan Futsal (SIP2SAL)”. Dalam penelitian ini, penulis mengembangkan sistem dari permasalahan yang terjadi seperti jadwal sewa dan reservasi berbenturan dan tidak cocoknya stok penjualan perlengkapan futsal dengan penjualan yang sudah dilakukan. Metode yang digunakan untuk membangun sistem tersebut ialah *prototype*.

Dari penelitian yang diungkapkan oleh (Futsal et al., 2018) memiliki persamaan dengan penelitian penulis yaitu tujuan untuk pemesanan lapangan futsal, pembuatan penjadwalan, dan pelayanan pada futsal. Yang membedakan dari sistem tersebut ialah pada metode yang dilakukan. Peneliti menggunakan metode *prototype*, sedangkan penulis menggunakan metode *waterfall*. Dari segi lain juga terdapat perbedaan pada aplikasi yang digunakan, peneliti menggunakan aplikasi berbasis desktop, yang mengharuskan pengguna untuk datang ke tempat futsal, sedangkan penelitian penulis sudah berbasis *website* tidak mengharuskan pengguna untuk datang langsung ke tempat futsal.

Lalu penelitian yang dilakukan oleh (Sutanto et al., 2020) dengan judul “Perancangan Reservasi Pariwisata dan Kamar Hotel Berbasis *Web*”. Pada penelitian ini memiliki kesamaan dengan penelitian yang dilakukan sebelumnya yaitu membahas tentang pelayanan, pengelolaan, dan pemesanan hotel yang berbasis *web*. Pada pembahasan ini, penulis berhasil merancang aplikasi yang sesuai dengan permasalahan yang ada seperti promosi harga, promosi tour (paket wisata), serta memberikan informasi-informasi terkait jenis kamar dan fasilitas

yang disediakan oleh pihak hotel. Untuk perbedaannya terletak pada metode yang digunakan. Susanto, Lidwan, dan Ridwan menggunakan metode FAST (*Framework For The Application System Thinking*), metode ini digunakan untuk pengembangan dan pemeliharaan Aplikasi.

## 2.2 Aplikasi

Aplikasi merupakan program yang berisikan perintah-perintah untuk melakukan pengolahan data. Aplikasi secara umum adalah suatu proses dari secara manual yang ditransformasikan ke komputer dengan membuat sistem atau program agar data diolah lebih berguna secara optimal. Menurut kamus komputer eksekutif, pengertian aplikasi merupakan pemecahan masalah yang biasanya berpacu pada sebuah komputasi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan. Aplikasi biasanya berupa perangkat lunak yang berbentuk software yang berisi kesatuan perintah atau program yang dibuat untuk melaksanakan sebuah pekerjaan yang diinginkan. Selain itu aplikasi juga mempunyai fungsi sebagai pelayanan kebutuhan beberapa aktivitas yang dilakukan oleh manusia seperti sistem untuk software jual beli, permainan atau game online, pelayanan masyarakat dan hampir semua proses yang dilakukan oleh manusia dapat dibantu dengan menggunakan suatu aplikasi. Lebih dari satu aplikasi jika digabungkan akan menjadi satu paket atau sering juga disebut dengan *application suite*, dimana aplikasi tersebut memiliki posisi antar muka yang mempunyai kesamaan sehingga dapat dengan mudah digunakan atau dipelajari penggunaan tiap aplikasi tersebut (Leplingard et al., 2003).

## 2.3 Reservasi

Reservasi adalah pemesanan suatu fasilitas yang dilakukan sebelum calon pelanggan menggunakan fasilitas atau layanan jasa. Sebutan lain dari reservasi ialah *booking*, yang berarti memesan sebelum pelanggan datang ke tempat layanan jasa. Menurut kamus besar Bahasa Indonesia (KKBI), definisi reservasi adalah proses, pembuatan, dan cara memesan (tempat, barang, dan sebagainya) kepada orang lain (Office et al., 2019).

## 2.4 Website

*Website* adalah suatu kumpulan halaman dalam domain yang berisi berbagai informasi yang bertujuan untuk dibaca dan dilihat oleh pengguna internet melalui sebuah mesin pencari (Adani, 2020b). Pada *website* sendiri memiliki tiga jenis *website* diantaranya *website* statis, dinamis, dan interaktif. Fungsi dari *website* bergantung jenisnya untuk apa membuatnya, tujuan dan fungsinya harus menjadi salah satu faktor utama dalam memilih fitur yang ingin dibutuhkan.

## 2.5 PHP ( Hypertext Proprecessor )

PHP merupakan singkatan dari *Hypertext Proprecessor*, yaitu Bahasa pemrograman yang hampir mirip dengan *Javascript* dan *Python*. PHP awalnya diciptakan oleh **Rasmus Lerdorf** untuk memantau orang yang sedang mengunjungi *home page* miliknya. Seiring waktu yang berjalan, Bahasa php ini makin populer, dan **Rasmus Lerdorf** merilis sebagai proyek *open source* (Faradilla, 2022).

Keterangan : pada gambar 2.1 terdapat contoh script PHP sederhana

```
<html>
<head>
  <title>Percobaan PHP</title>
</head>
<body>
<?php echo '<p>Salam dari Hostinger</p>'; ?>
</body>
</html>
```

Gambar 2. 1 Contoh script php

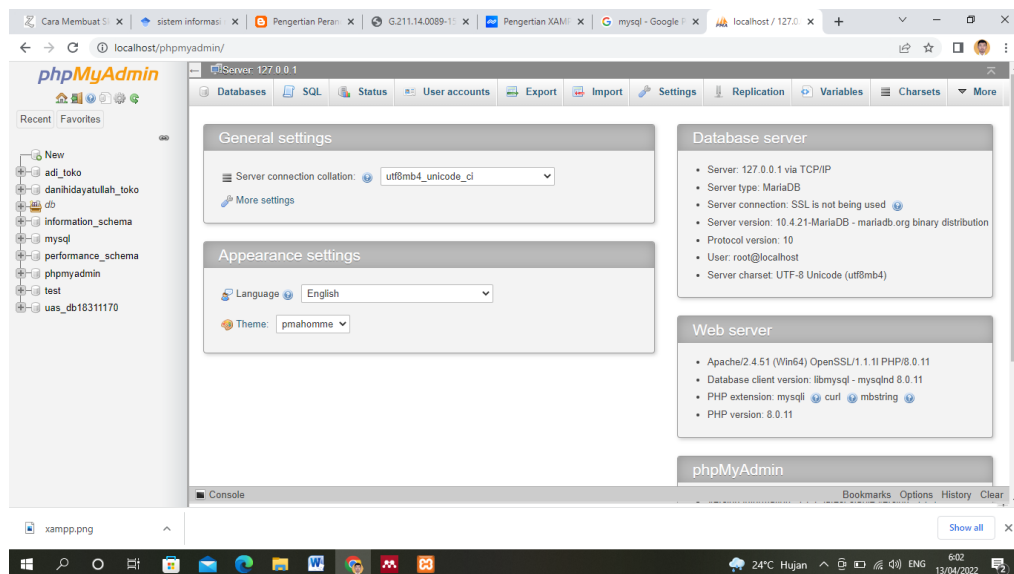
## 2.6 Framework

*Framework* atau kerangka kerja adalah istilah yang tidak asing lagi dalam dunia *developer*. Dalam pengertiannya, *Framework* adalah kerangka kerja untuk mengembangkan suatu aplikasi yang berbasis *website* maupun desktop. Kerangka kerja diciptakan untuk mempermudah dalam kinerja seorang programmer. Karena didalamnya hanya perlu menyusun komponen-komponen pemrograman saja. Untuk *framework* memiliki tiga jenis yaitu *Framework CSS*, *Framework JavaScript*, dan *Framework PHP* (Adani, 2020a).

## 2.7 MySQL

*MySQL* adalah sebuah *Database Management System* (DBMS) menggunakan perintah dasar *SQL* (*Structured Query Language*) yang *open source* dengan dua bentuk lisensi, yaitu *Free Software* (perangkat lunak bebas) dan *Shareware* (perangkat lunak berpemilik terbatas) (Ariata, 2019). Untuk tampilannya dapat dilihat pada gambar dibawah ini.

Keterangan : pada gambar 2.2 menjelaskan gambar dari *phpmyadmin*.



Gambar 2. 2 Tampilan *phpmyadmin*

### 2.7.1 Kelebihan *MySQL*

1. Mendukung integrase dengan Bahasa pemrograman  
*MySql* membantu mengembangkan perangkat lunak yang lebih efektif dengan menggunakan berbagai macam Bahasa pemrogramana.
2. Tidak membutuhkan RAM besar  
*MySQL* dapat dijalankan pada server dengan kapasitas walaupun hanya 1 Gb.
3. Mendukung *multi user*  
*MySQL* dapat dipakai beberapa user dengan waktu yang bersamaan tanpa terganggu.
4. *Open Source*  
*MySQL* merupakan sistem manajemen database gratis, apalagi

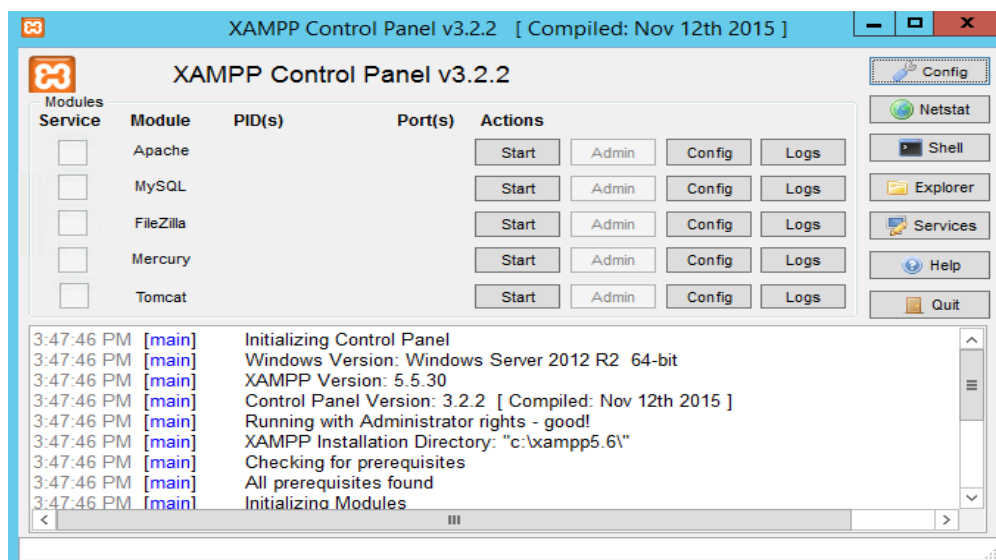
pengelolaannya dibawah oracle sehingga kualitasnya termasuk baik.

## 5. Keamanan yang terjamin

*MySQL* mempunyai fitur keamanan yang cukup baik.

## 2.8 Xampp

*Xampp* merupakan *web server* yang bersifat *open source*. *Xampp* sendiri bisa digunakan pada sistem operasi *Windows*, *Linux*, maupun *MacOs*. *Website* sendiri membutuhkan beberapa pengelolaan yang terdapat pada *xampp* seperti *Apache*, *Mysql/MariaDB*, *PHP*, dan *Perl*. Fungsi dari *xampp* adalah untuk mengatur *database phpMyadmin*, instal *WordPres Offline*, dan menjalankan laravel pada komputer. *Xampp* merupakan *software local* yang mudah digunakan serta ringan untuk dijalankan (Kurniawan, 2020). Untuk tampilan *xampp* bisa dilihat dibawah ini.



Gambar 2.3 Tampilan *xampp*

## 2.9 UML (Unified Modelling Language)

Uml (*Unified Modelling Language*) merupakan sebuah metode permodelan secara *visual* untuk perancangan sistem yang berorientasi objek (Aplikasinya, 2020). Adapaun beberapa fungsinya yaitu untuk menghubungkan metode rancangan terbaik, menghubungkan Bahasa *visual*, sebagai model sistem berorientasi objek, dan untuk membuat Bahasa rancangan permodelan. Adapun jenis diagram yang terdapat pada *uml* adaah sebagai berikut.


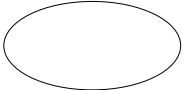

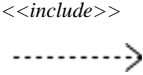
### 2.9.1 Komponen diagram UML

Komponen-komponen yang terdapat pada *Uml* yang akan digunakan adalah sebagai berikut (Irawan et al., 2019).

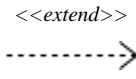

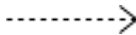
#### 1. Use Case Diagram

Merupakan penggambaran interaksi yang diharapkan oleh sistem. use case diagram juga dapat men-deskripsikan tipe interaksi antara si pemakai sistem dengan sistemnya.

Tabel 2. 1 Use case diagram

NO	GAMBAR	NAMA SIMBOL	KETERANGAN
1		<i>Actor</i>	<i>System</i> atau <i>external</i> entitas yang menyediakan maupun menerima informasi dari <i>system</i> .
2		<i>Use case</i>	Sistem yang disediakan sebagai unit-unit yang saling bertukar antar unit atau actor.
3		<i>Association</i>	Komunikasi antar actor dan use case yang berpartisipasi pada use case
4		<i>include</i>	Relasi <i>use case</i> tambahan ke semua <i>use case</i> dimana yang ditambahkan <i>use case</i> memerlukan <i>use case</i> ini.


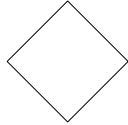
Tabel 2. 2 *Use case diagram* lanjutan

NO	GAMBAR	NAMA SIMBOL	KETERANGAN
5		<i>Extend</i>	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri.
6		<i>Generalization</i>	Hubungan generalisasi dan spesialisasi antar dua buah use case.
7		<i>dependency</i>	Hubungan dimana terdapat perubahan pada suatu elemen maka berpengaruh pada elemen lainna.

## 2. *Class Diagram*.


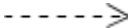
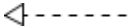


*Class diagram* merupakan sesuatu yang bisa membantu dalam memvisualisasikan struktur dalam setiap kelas yang terdapat dalam sebuah sistem. *Class diagram* adalah salah satu jenis diagram berbentuk struktur pada model UML, *class diagram* juga bagian terpenting dalam UML.

Tabel 2. 3 *Class Diagram*

NO	GAMBAR	NAMA SIMBOL	KETERANGAN
1		<i>Generalization</i>	Hubungan antar objek anak sebagai perilaku dan struktur data.
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.




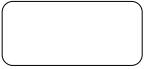
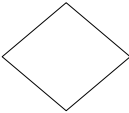


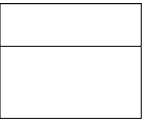
Tabel 2. 4 *Class* Diagram lanjutan

NO	GAMBAR	NAMA SIMBOL	KETERANGAN
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut dan operasi yang sama.
4		<i>Dependency</i>	Hubungan perubahan yang terjadi pada suatu elemen mandiri.
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Collaboration</i>	Urutan aksi-aksi yang ditampilkan sistem untuk menghasilkan sesuatu yang terukur bagi suatu aktor.
7		<i>Association</i>	Hubungan dimana terdapat perubahan pada suatu elemen maka berpengaruh pada elemen lainnya.

### 3. *Activity* Diagram

*Activity* diagram merupakan *state* diagram yang khusus sebagian besar dari *state* adalah *action* dan sebagian besar transisi *trigger*. Oleh karena itu, *activity* diagram tidak menggambarkan perilaku internal dari sebuah sistem tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari secara umum. Berikut ini merupakan simbol dari *activity* diagram.

Tabel 2. 5 Activity Diagram


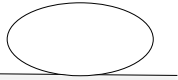




NO	GAMBAR	NAMA SIMBOL	KETERANGAN
1		Status awal	Sebuah diagram aktivitas memiliki sebuah status awal.
2		aktivitas	Aktivitas yang dilakukan sistem, biasanya diawali kata kerja.
3		Percabangan / <i>Decision</i>	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
4		Penggabungan / <i>join</i>	Penggabungan yang mana lebih dari satu aktivitas lalu digabungkan menjadi satu.
5		Status akhir	Status akhir yang dilakukan sistem.
6		<i>Swimlane</i>	<i>Swimlane</i> memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

#### 4. Sequence Diagram

*Sequence* diagram merupakan penggambaran interaksi antar objek didalam dan disekitar sistem (termasuk pengguna, *display/form*) berupa pesan (*message*). *Sequence* terdiri dari dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence* diagram

biasa digunakan dalam penggambaran skenario atau langkah-langkah dari sebuah *event* untuk menghasilkan sebuah *output* tertentu. Diagram ini secara khusus berasosiasi dengan *use case* diagram. Berikut ini simbol-simbol yang terdapat pada sequence diagram.

Tabel 2. 6 *Sequence Diagram*

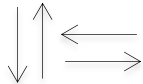





NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Meggambarkan orang yang sedang interaksi dengan sistem.
2		<i>Entity class</i>	Mengambarkan hubungan kegiatan yang akan dilakukan.
3		<i>Boundary classs</i>	Menggambarkan sebuah penggambaran dari <i>form</i>
4		<i>Control class</i>	Menggambarkan penghubungan antar boundary dengan tabel.
5		<i>Focus control and life line</i>	Menggambarkan tempat mulai dan berakhirnya sebuah pesan ( <i>message</i> ).
6		<i>message</i>	Menggambarkan pengiriman pesan.

## 2.10 Flowchart

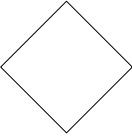


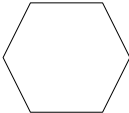


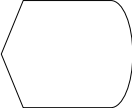
*Flowchart* atau bagan alur merupakan diagram yang menampilkan langkah-langkah maupun keputusan untuk melakukan suatu proses pada program. Setiap langkah digambarkan dalam bentuk diagram dan dihubungkan dengan

garis maupun panah. Dengan menggunakan bagan alur ini, maka sebuah proses akan lebih jelas, ringkas, dan mengurangi salah penafsiran. Fungsi dari *flowchart* sendiri ialah memberi gambaran jalannya sebuah proses program satu dengan yang lainnya.

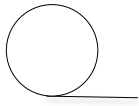


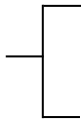

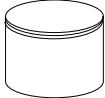
Tabel 2. 7 *Flowchart*

NO	GAMBAR	NAMA SIMBOL	KETERANGAN
1		<i>Flow</i>	Simbol yang digunakan untuk menghubungkan antara simbol satu dengan yang lainnya.
2		terminator	Simbol untuk memulai dan mengakhiri dari suatu kegiatan
3		<i>connector</i>	Pada simbol ini berguna untuk penyambungan proses dalam lembar halaman yang sama.
4		<i>connctor</i>	Simbol ini berguna untuk keluar masuk atau penyambung proses pada halaman yang berbeda.
5		<i>processing</i>	Simbol ini menunjukkan pengolahan yang dilakukan oleh computer.
6		<i>Manual operation</i>	Simbol ini menunjukkan pengolahan yang tidak dilakukan oleh komputer.

Tabel 2. 8 *Flowchart* Lanjutan

NO	GAMBAR	NAMA SIMBOL	KETERANGAN
7		<i>decision</i>	Simbol ini menunjukkan proses pemilihan berdasarkan kondisi.
8		<i>Input output</i>	Pada simbol ini menyatakan suatu proses <i>input</i> dan <i>output</i> tanpa tergantung dengan jenis peralatannya.
9		<i>Manual input</i>	Simbol ini menunjukkan data yang masuk secara manual <i>on-line keyboard</i> .
10		<i>Preparation</i>	Simbol ini untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan <i>storage</i> .
11		<i>Predefine proses</i>	Simbol ini untuk pelaksanaan suatu bagian (sub-program) / prosedur.
12		<i>Disk and On-line storage</i>	Simbol ini menyatakan <i>input</i> yang berasal dari <i>disk</i> atau disimpan ke <i>disk</i> .
13		<i>Display</i>	Simbol ini menyatakan peralatan <i>output</i> yang digunakan yaitu layar, plotter, printer, dan sebagainya.

Tabel 2. 9 *Flowchart* Lanjutan

NO	GAMBAR	NAMA SIMBOL	KETERANGAN
14		<i>Magnetic tape unit</i>	Simbol ini menyatakan <i>input</i> yang berasal dari pita magnetic atau <i>output</i> disimpan ke pita magnetic.
15		<i>Punch Card</i>	Simbol ini menyatakan <i>input</i> berasal dari kartu atau <i>output</i> ditulis ke kartu.
16		Dokumen	Simbol ini menyatakan <i>input</i> berasal dari dokumen dalam bentuk kertas atau <i>output</i> dicetak dikertas.
17		<i>Comment</i>	Menunjukkan informasi tambahan mengenai langkah dalam suatu proses.
18		<i>Offline stroge</i>	Berfungsi menunjukkan bahwa data akan disimpan di tempat tertentu berdasarkan : T = Tanggal N = Nomor/ <i>Number</i> A = Abjad
19		Database	Menujukan penyimpanan pada data.

## 2.11 Sistem Basis Data

Basis data atau *database* memiliki penegrtian masing-masing. Adapaun basis adalah sebagai markas atau gudang, tempat bersarang atau berkumpul. Sedangkan data adalah sesuatu yang bersifat nyata yang mewakili suatu objek

seperti manusia, barang, hewan, peristiwa, konsep, keadaan, dan sebagainya yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya (Gadek, 2021).

### 2.11.1 Komponen dasar basis data

#### 1. Data

Dapat diartikan representasi fakta dunia nyata yang mewakili suatu objek yang direkam dalam bentuk jangkak, huruf, simbol, teks, gambar, bunyi, atau kombinasinya.

#### 2. Hardware

Terdiri dari semua peralatan peangkat keras computer yang digunakan untuk mengelola sistem basis data.

#### 3. Software

Merupakan sebagai perantara antara pemakai dengan data fisik pada basis data, dapat berupa database *management system* (DBMS) atau program-program aplikasi dan prosedur-prosedur.

#### 4. *User* (pemakai)

Pada bagaian user ini terbagi menjadi 4 klasifikasi diantaranya :

##### a. *System engineer*

Tenaga ahli yang bertanggung jawab atas pemasangan sistem basis data.

##### b. Administrartor basis data

Tenaga ahli yang memiliki tugas untuk mengontrol sistem basis data secara keseluruhan, meramalkan kebutuhan bais data, merencanakan, dan mengaturnya.

##### c. Programmer

Merupakan pembuat program aplikasi yang diperlukan oleh pemakai akhir dengan menggunakan data yang terdapat dalam sistem basis data.

### 2.11.2 Istilah-istilah basis data

Dalam penyusunan basis data terdapat beberapa istilah yaitu :

#### 1. Entitas

Suatu objek yang dapat dibedakan dari lainnya yang dapat diwujudkan

dalam basis data. Contoh entitas dalam lingkungan universitas terdiri dari mahasiswa, mata kuliah.

2. Atribut (elemen data)

Karakteristik dari suatu entitas. Contoh dari entitas mahasiswa atributnya terdiri dari npm, nama, alamat, tanggal lahir.

3. Kunci elemen data (*key data element*)

Merupakan tanda pengenal yang secara unik mengidentifikasi entitas dari suatu kumpulan entitas.

## 2.12 Pengujian sistem

### 2.12.1 Pengujian *black box*

Dalam melakukan pengujian sistem, terdapat banyak cara untuk melakukan pengujiannya salah satu contoh yang akan dibahas adalah pengujian *black box*. *Black box testing* atau sering disebut dengan *Behavioral Testing* merupakan pengujian yang dilakukan untuk mengamati dari hasil *input* dan *output* pada perangkat lunak tanpa mengetahui struktur kodenya. Pengujian ini dilakukan pada akhir pembuatan perangkat lunak (Hanifah et al., 2016). Pada *black box testing*, memiliki beberapa teknik yang biasa digunakan untuk menguji perangkat lunak diantaranya:

1. *All pair testing*

Teknik ini juga dikenal dengan *pairwise testing*. Pengujian ini dilakukan untuk semua kemungkinan kombinasi dari seluruh pasangan berdasarkan *input* parameteranya.

2. *Boundary value analysis*

Teknik ini berfokus pada pencarian *error* dari luar atau sisi dalam perangkat lunak.

3. *Cause-effect graph*

Teknik pengujian ini dilakukan dengan menggunakan teknik grafik sebagai patokannya. Grafik menggambarkan relasi antara efek dan penyebab dari *error*.

4. *Equivalence partitioning*

Pada teknik ini bekerja dengan cara membagi data *input* dari beberapa



perangkat lunak menjai beberapa partisi data.

5. *Fuzzing*

Teknik ini merupakan teknik mencari *bug* dalam perangkat lunak dengan memasukkan data yang tidak sempurna.

6. *Orthogonal arraytesting*

Pada teknik ini digunakan jika *input* berukuran kecil, namun cukup berat jika digunakan dalam skala besar.

7. *State transition*

Teknik ini berguna dalam melakukan pengujian terhadap mesin dan navigasi dalam bentuk grafis.

### 2.12.2 Kerangka pengujian *black box*

Untuk kerangka pengujian *black box* dapat dilihat pada contoh tabel dibawah ini.

Tabel 2. 10 Tabel contoh pengujian *black box*

<b>Data Masukan</b>	<b>Yang Diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
Klik menu lihat jadwal lapangan	Dapat memperlihatkan informasi jadwal lapangan	Sesuai yang diharapkan	[ ] Diterima [ ] Ditolak

