

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian terdahulu yang dilakukan oleh Zulfitri Yani, Devi Gusmita, dan Nurmaliana Pohan pada tahun 2019 dalam penelitiannya yang berjudul “Seleksi Penerimaan Karyawan Baru Menggunakan Metode TOPSIS”. Pada penelitian ini membahas tentang sistem pendukung keputusan yang membantu perusahaan pada bagian penerimaan karyawan di PT Panel Indofurn Cab dalam pengambilan keputusan untuk menentukan karyawan baru. Model yang digunakan pada penelitian ini adalah metode TOPSIS. Melalui penerapan metode ini, peneliti dapat membandingkan setiap calon karyawan berdasarkan kriteria yang menentukan calon karyawan dianggap memenuhi kualifikasi. Adapun kriteria yang digunakan dalam pengambilan keputusan yaitu pendidikan, pengalaman bekerja, skill/kemampuan, usia, dan karakter ^[2].

Penelitian serupa dilakukan oleh Yustika Indah Purwanti pada tahun 2020 dalam penelitiannya yang berjudul “Sistem Pendukung Keputusan Penerima Bantuan Rumah Tak Layak Huni (RTLH) Di Kota Binjai Dengan Metode Simplemultiple Attribute Rating Technique (SMART) (STUDI KASUS: DINAS SOSIAL BINJAI)”. Pada sistem berjalan saat ini pihak Dinas Sosial masih mengalami kesulitan seperti dalam pengolahan datanya membutuhkan ketelitian, sehingga mungkin terjadinya rangkap data dan juga kesalahan dalam penentuan penduduk yang harus diutamakan, sehingga diperlukannya suatu sistem pendukung keputusan untuk menentukan siapa yang berhak untuk didahulukan dalam mendapatkan bantuan RTLH ^[3].

Penelitian selanjutnya yang dilakukan oleh Ramos Somya dan Andre Wahyudi pada tahun 2020 yang berjudul “Sistem Pendukung Keputusan Perekrutan Karyawan Menggunakan Metode TOPSIS di PT Visionet Data Internasional” yang membahas permasalahan dalam proses perekrutan karyawan yang disebabkan oleh banyak faktor seperti banyaknya kriteria penilaian yang menjaid tolak ukur dalam proses penerimaan karyawan. Untuk mengatasi permasalahan tersebut, dibuat sebuah sistem pendukung keputusan yang menerapkan salah satu metode MCDM. TOPSIS merupakan metode MCDM yang dapat digunakan pada sistem pendukung keputusan tersebut. Hasil yang didapatkan dalam

penelitian ini adalah sebuah sistem pendukung keputusan yang dapat digunakan oleh HRD untuk membantu proses rekrutmen karyawan ^[4].

Penelitian yang lain dilakukan oleh Nezard Dafian Nugraha dan Arief Wibowo pada tahun 2020 melakukan penelitian dengan judul “Implementasi Metode TOPSIS untuk Pemilihan Karyawan Terbaik”. Penelitian ini dilakukan untuk membangun model penilaian karyawan menggunakan metode TOPSIS yang diharapkan memiliki hasil yang akurat dan dapat dipertanggung jawabkan serta dapat diimplementasikan dalam bentuk aplikasi pemilihan karyawan terbaik. Salah satu metode yang digunakan dalam pengambilan keputusan karyawan terbaik ini adalah metode TOPSIS. Dalam hal ini dikarenakan metode TOPSIS mampu melakukan perankingan terhadap alternatif terpilih ^[5].

Penelitian sejenis yang dilakukan oleh Muhammad Andri Imawan, dkk pada tahun 2019 dalam penelitiannya yang berjudul berjudul “Aplikasi Perekrutan Karyawan Menggunakan Metode TOPSIS Berbasis Web pada PT. Smesco Indonesia”. Perusahaan ini membutuhkan sistem pengambilan keputusan dalam penerimaan calon karyawan kontrak dalam setiap tahun karena ini mengalami kesulitan dalam memproses perhitungan penerimaan calon karyawan. Oleh karena itu, penelitian ini bertujuan untuk mengimplementasikan program aplikasi *Decision Support Systems* (DSS) berbasis web yang dapat membantu HRD PT. Smesco Indonesia dalam proses penilaian terhadap calon karyawan dengan cepat dan akurat. Salah satu metode yang digunakan adalah metode TOPSIS. Hasil diimplementasikan aplikasi ini dapat mengambil keputusan setiap calon karyawan mendapat laporan penilaian yang detail dan transparan, bersifat online dan realtime, serta HRD mendapatkan penilaian karyawan yang objektif sesuai dengan bobot kriteria yang ditentukan ^[6].

Penelitian berikutnya yang dilakukan oleh Dwi Marisa Efendi, dkk pada tahun 2021 dalam penelitiannya yang berjudul “Sistem Pendukung Keputusan Pemilihan Karyawan Terbaik Menggunakan Metode Technique Order Of Preference By Similarity To Ideal Solution (TOPSIS)”. Tujuan dilakukannya penelitian ini adalah meningkatkan kualitas karyawan sehingga diperlukan pemilihan karyawan terbaik sebagai motivasi untuk karyawan agar dapat lebih berkompeten dalam melakukan pekerjaan masalah yang terjadi pada kantor POS Kuningan. Metode yang digunakan adalah metode TOPSIS dengan menggunakan aplikasi Borland Delphi 7, dan dengan menggunakan metode pengembangan sistem yaitu waterfall. Sehingga akan menghasilkan

sebuah pengambilan keputusan calon karyawan terbaik secara terkomputerisasi dengan cara yang lebih efektif dan tidak menurut sudut pandang saja, atau kepada karyawan yang memiliki kedekatan hubungan tertentu dengan kepala kantor POS Kuningan^[7].

Penelitian ini bermaksud membangun sistem pendukung keputusan proses penerimaan karyawan menggunakan bahasa pemrograman PHP dan database *MySQL* serta dikembangkan dengan metode *waterfall*. Perbedaan penelitian ini dibandingkan dengan penelitian sebelumnya adalah adanya fitur tambahan yaitu rekomendasi oleh Manager, selain dari proses perangkingan. Dengan harapan adanya fitur ini dapat memastikan bahwa data yang dihasilkan oleh sistem dapat lebih sesuai dengan kebutuhan pengguna. Adapun penelitian yang akan dikembangkan berjudul “Automatisasi Pendukung Keputusan Proses Penerimaan Karyawan Dengan Metode Topsis (Studi Kasus: PT. Adi Sarana Logistik)”.

2.2 Landasan Teori

Landasan teori berisi hal-hal atau teori-teori yang berkaitan dengan permasalahan dan ruang lingkup permasalahan sebagai landasan dalam pembuatan laporan ini.

2.2.1 Sistem Pendukung Keputusan

Sistem merupakan suatu kumpulan jaringan kerja dari beberapa prosedur yang bertujuan untuk melakukan sesuatu kegiatan yang menyelesaikan suatu permasalahan. Sistem dapat diartikan sebagai kumpulan jaringan atau elemen yang saling berhubungan yang bertujuan memproses suatu masukan (*input*) yang kemudian diolah oleh sistem dan menghasilkan keluaran (*output*). Prosedur itu sendiri merupakan urutan operasi tulis menulis yang dapat melibatkan beberapa orang dalam pengoperasiannya^[8].

Keputusan merupakan suatu tindakan memilih strategi yang bertujuan untuk memecahkan suatu permasalahan. Pengambilan keputusan merupakan tindakan yang dapat memberikan solusi terbaik atas sesuatu yang terjadi. Tujuannya untuk mencapai target atau aksi tertentu yang harus dilakukan^[9].

Sistem pendukung keputusan merupakan sistem informasi interaktif yang menyediakan informasi, pemodelan, dan pemanipulasian data. Sistem ini digunakan untuk membantuk pengambilan keputusan

dalam situasi yang semistruktur dan situasi yang tidak terstruktur, di mana tidak ada seorang pun yang mengetahui cara pasti pendukung keputusan dibuat ^[10].

2.2.2 Metode TOPSIS (*Techniwue for Orders Reference by Similarity to Ideal Solution*).

Metode yang digunakan dalam sistem pendukung keputusan ini adalah metode TOPSIS. Metode *Technique for Order Preference by Similarity to Ideal Solution* (TOPSIS) merupakan sebuah metode yang mampu membantu proses pengambilan keputusan yang optimal untuk menyelesaikan masalah keputusan secara praktis. Hal ini karena konsep yang mudah dipahami dan sederhana, komputasi yang efisien, dan mampu mengukur kinerja relatif dari alternatif-alternatif keputusan dalam bentuk matematis sederhana^[11]. Prinsip metode ini adalah alternatif yang dipilih memiliki jarak terpendek dari solusi ideal positif dan jarak terjauh dari solusi ideal negatif ^[12].

Penyelesaian metode TOPSIS sebagai berikut ^[13]:

1. Membuat matriks keputusan yang ternormalisasi.

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}} \dots \dots \dots (1)$$

Dengan $i = 1, 2, \dots, m$; dan $j = 1, 2, \dots, n$.
 r_{ij} = matriks keputusan ternormalisasi
 x_{ij} = bobot kriteria ke j pada alternatif ke i
 i = alternatif ke i
 j = alternatif ke j

2. Membuat matriks keputusan yang ternormalisasi terbobot.

$$y_{ij} = w_{ij} r_{ij} \dots \dots \dots (2)$$

Dengan $i = 1, 2, \dots, m$; dan $j = 1, 2, \dots, n$.
 y_{ij} = matriks ternormalisasi bobot
 w_{ij} = matriks keputusan ternormalisasi
 r_{ij} = bobot kriteria ke- j
 Dimana w_j adalah bobot dari kriteria ke- j .

3. Membuat matriks solusi ideal positif dan matriks solusi ideal negatif. Berdasarkan rating bobot ternormalisasi maka dapat menentukan solusi ideal positif (A+) dan solusi ideal negatif (A-) untuk dapat menentukan solusi ideal sebelumnya harus ditentukan apakah atribut bersifat keuntungan (*benefit*) atau bersifat biaya (*cost*).

$$A^+ = (y_1^+, y_2^+ \dots, y_n^+) \dots \dots \dots (3)$$

$$A^- = (y_1^-, y_2^- \dots, y_n^-) \dots \dots \dots (4)$$

Dimana,

$$\left. \begin{aligned} y_j^+ &= \max_i y_{ij} \text{ jika } j \text{ adalah atribut keuntungan} \\ &\quad \min y_{ij} \text{ jika } j \text{ adalah atribut biaya} \end{aligned} \right\}$$

$$\left. \begin{aligned} y_j^- &= \min_i y_{ij} \text{ jika } j \text{ adalah atribut keuntungan} \\ &\quad \max y_{ij} \text{ jika } j \text{ adalah atribut biaya} \end{aligned} \right\}$$

4. Menentukan jarak antara nilai setiap alternatif dengan matriks solusi ideal positif dan negatif.

$$D_i^+ = \sqrt{\sum_j^n = 1 (y_j^+ - y_{ij})^2} \dots \dots \dots (5)$$

$$D_i^- = \sqrt{\sum_j^n = 1 (y_{ij} - y_j^-)^2} \dots \dots \dots (6)$$

Menentukan nilai prefensi untuk setiap alternatif.

$$V_i = \frac{D_i^-}{D_i^- + D_i^+} \dots \dots \dots (7)$$

Keterangan:

D_i^+ adalah jarak antara nilai alternatif ke i dengan solusi ideal positif

D_i^- adalah jarak antara nilai alternatif ke i dengan solusi ideal negatif

2.2.3 PHP (*Hypertext Preprocessor*)

PHP (*Hypertext Preprocessor*) merupakan salah satu bahasa pemrograman yang berjalan dengan sebuah web server dan berfungsi sebagai pengolah data pada sebuah server. Data yang dikirim oleh user client akan diolah dan disimpan pada adatabase web server dan dapat ditampilkan kembali diakses. Agar program PHP dapat dijalankan, file

harus di *upload* kedalam *server*. *Upload* adalah proses mentransfer data atau file data atau file dari komputer client ke dalam *web server* ^[14].

Dengan membuat website yang dinamis dan mudah di update setiap saat dari browser, dibutuhkan sebuah program yang mampu mengolah data dari komputer client atau dari komputer *server* itu sendiri sehingga mudah dan nyaman digunakan di browser. Salah satu program yang dapat dijalankan di *server* dan cukup handal adalah PHP

2.2.4 Rekayasa Perangkat Lunak

Rekayasa Perangkat Lunak merupakan suatu bidang profesi menekuni cara-cara pengembangan perangkat lunak termasuk didalamnya pembuatan, pemeliharaan, manajemen, organisasi pengembangan perangkat lunak, dan sebagainya ^[15]. RPL juga dapat didefinisikan sebagai disiplin ilmu yang mengulas seluruh aspek produksi perangkat lunak mulai dari tahap spesifikasi hingga pemeliharaan sistem.

2.2.5 Metode Pengembangan Sistem

Model pengembangan sistem dalam pembangunan sistem ini menggunakan metode *waterfall*. Menurut Ian Sommerville, metode *waterfall* merupakan salah satu proses perangkat lunak yang mempunyai siklus hidup klasik (*classic life cycle*). Proses model *waterfall* harus menyelesaikan suatu tahap sampai selesai sebelum melanjutkan ke tahap berikutnya. Rangkaian aktivitas dalam metode ini disajikan dalam proses yang terpisah dengan 5 tahapan . Dalam proses pengembangannya dilakukan bertahap satu persatu sehingga meminimalisir kesalahan yang mungkin terjadi. Berikut adalah tahapan dari metode pengembangan sistem *waterfall* ^[16] :

- a. Analisa Kebutuhan (*Requirements Definition*).
Pada tahap pertama ini pengembangan perangkat lunak harus mengetahui kebutuhan dan informasi serta kegunaannya dalam perangkat lunak yang dikembangkan. Metode Analisis Kebutuhan ini dapat dilakukan dengan berbagai cara wawancara, observasi dan sebagainya. Kemudian dari data yang sudah diperoleh tersebut dikaji agar kebutuhan sistem untuk pengembangan perangkat lunak dapat dilakukan dengan baik.
- b. Desain Sistem (*System and Software Design*)
Pada tahap ini akan dipelajari kebutuhan dan spesifikasi dari tahap sebelumnya dan akan disiapkan desain sistemnya. Desain sistem

- membantu menentukan perangkat keras dan persyaratan sistem dan juga membantu menentukan arsitektur sistem secara keseluruhan.
- c. **Penulisan Kode Program (*Implementation and Unit Testing*)**
Pada tahap ini, sistem pertama kali dikembangkan dalam program kecil yang disebut unit, dan kemudian diintegrasikan. Setiap unit dikembangkan dan diuji fungsionalitasnya, yang disebut unit testing.
 - d. **Pengujian Program (*Integration and System Testing*)**
Pada tahap ini merupakan tahap pengujian pada semua fungsi sistem perangkat lunak yang dihasilkan telah sesuai dengan keinginan pengguna dan perancangan sistem. Pengujian dilakukan juga dilakukan untuk memeriksa fungsi sistem berjalan dengan baik tanpa ditemukan kesalahan.
 - e. **Penerapan Program dan Pemeliharaan (*Operation and Maintenance*)**
Pada tahap ini, perangkat lunak sudah jadi lalu dapat dioperasionalkan pengguna dan melakukan pemeliharaan. Pemeliharaan dilakukan untuk memeriksa setiap malfungsi atau kesalahan yang tidak terdeteksi pada tahap sebelumnya. Pemeliharaan meliputi perbaikan kesalahan, perbaikan implementasi unit sistem, peningkatan dan penyesuaian sistem sesuai dengan kebutuhan.

2.2.6 Metode Pengujian Sistem

Pentingnya pengujian sangat dibutuhkan dalam penilaian kualitas perangkat lunak supaya fungsi yang ada didalamnya dapat bekerja sesuai dengan harapan pengguna. Metode yang digunakan dalam pengujian ini adalah metode *black-box*. Metode *black-box* merupakan metode yang bertujuan untuk melihat apakah program tersebut sudah sesuai dengan fungsionalitas yang diharapkan pengguna tanpa mengetahui kode program yang dipakai. Pengujian *black-box* pada perangkat lunak bertujuan untuk mengetahui kesalahan-kesalahan pada fungsi yang tidak benar atau hilang, *interface*, struktur data atau pengaksesan *database*, kesalahan kinerja, inisialisasi dan kesalahan terminasi^[17].

2.2.7 Tools atau Alat Bantu Penelitian

a. Flowchart

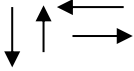


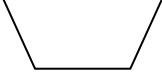
Flowchart merupakan penggambaran secara grafik dari langkah-langkah dan urutan prosedur suatu program. *Flowchart* biasanya memiliki bagan (*chart*) yang menunjukkan alir (*flow*) di dalam sebuah program




atau prosedur sistem secara logika. Bagan alir (*flowchart*) digunakan untuk alat bantu komunikasi dan dokumentasi [18]. Ada beberapa jenis jenis flowchart diantaranya:

1. Bagan alir sistem (*system flowchart*)
2. Bagan alir dokumen (*document flowchart*)
3. Bagan alir skematik (*schematic flowchart*)
4. Bagan alir program (*program flowchart*)
5. Bagan alir proses (*process flowchart*)

Simbol-simbol *flowchart* dapat dilihat pada Tabel 2.1.

Tabel 2. 1 Simbol *Flowchart*

No	Simbol	Nama	Keterangan
1		<i>Flow Direction Symbol</i>	Simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain.
2		<i>Terminator Symbol</i>	Simbol untuk permulaan (<i>start</i>) atau akhir (<i>end</i>) dari suatu kegiatan.
3		<i>Processing Symbol</i>	Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer.
4		<i>Manual Operation Symbol</i>	Simbol yang menunjukkan pengolahan data yang tidak dilakukan oleh komputer.

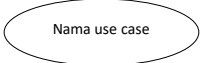


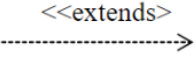

5		<i>Decision Symbol</i>	Simbol untuk pemilihan proses berdasarkan kondisi yang ada.
5		<i>Input-Output Symbol</i>	Simbol yang menyatakan proses input dan output tanpa tergantung dari jenis peralatannya.
6		<i>Document Symbol</i>	Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas.

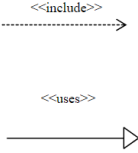
b. *Unified Modeling Language (UML)*

1. *Use Case Diagram*

Use Case Diagram merupakan pemodelan yang menggambarkan perilaku sebuah sistem yang dapat digunakan sebagai interaksi antara sistem dengan pengguna. *Use Case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem tersebut *scenario* sedangkan pengguna disebut *actor*. *Actor* adalah sebuah peran yang biasa dimainkan oleh pengguna dalam interaksinya dengan sistem. Model *use case* adalah bagian dari model *requirement*. Definisi lain *use case* adalah abstraksi dari interaksi antara sistem dan *actor*. *Use case* dibuat berdasarkan keperluan *actor* ^[19]. Simbol *Use Case* dapat dilihat pada Table 2.2.

Tabel 2. 2 Simbol *Use Case Diagram*

No.	Simbol	Nama	Fungsi
1.		<i>Use case</i>	Fungsionalitas yang tersedia pada sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.
2.		Aktor	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.
3.		Asosiasi	Penghubung komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> .
4.		Ekstensi	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan tersebut.
5.		Generalisasi	Hubungan generalisasi dan spesialisasi antara dua buah <i>use case</i> .



6.		<i>Include</i> atau <i>uses</i>	<p><i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan.</p> <p><i>Use case</i> tambahan akan mengecek apakah <i>use case</i> yang ditambahkan sudah dijalankan sebelum menjalankan <i>use case</i> tambahan.</p>
----	---	---	---

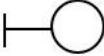



2. Sequence Diagram

Sequence Diagram merupakan diagram yang menggambarkan perilaku pada sebuah *scenario*. Diagram ini menunjukkan sejumlah obyek dan *message* yang diletakan antara obyek-obyek didalam *use case* ^[20].

Komponen utama *sequence diagram* terdiri atas objek yang dituliskan dengan kotak segiempat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan *progress vertical*. *Sequence Diagram* menambahkan dimensi waktu pada interaksi diantara obyek. Simbol *Sequence Diagram* dapat dilihat pada Table 2.3.

Tabel 2. 3 Simbol *Sequence diagram*

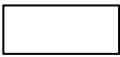
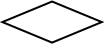


No	Simbol	Nama	Keterangan
1		<i>Actor</i>	Menggambarkan orang yang sedang berinteraksi dengan sistem.
2		<i>Entity Class</i>	Menggambarkan hubungan yang akan dilakukan.

3		<i>Boundary Class</i>	Menggambarkan sebuah gambaran dari <i>foem</i> .
4		<i>Control Class</i>	Menggambarkan penghubung antara boundary dengan tabel.
5		<i>A Focus of Control & A Life Line</i>	Menggambarkan tempat mulai dan berakhirnya <i>massage</i> .
6		<i>A Message</i>	Menggambarkan Pengiriman Pesan.

3. ERD

ERD (*Entity Relationship Diagram*) merupakan suatu model yang digunakan untuk mendesain database dengan tujuan menggambarkan data yang berelasi pada sebuah database ^[21]. Simbol ERD dapat dilihat pada Table 2.4.

Tabel 2. 4 Simbol ERD (*Entity Relationship Diagram*)

No	Simbol	Nama	Keterangan
1		<i>Entitas</i>	Suatu objek yang dapat diidentifikasi dalam lingkungan pemakai.
2		Relasi	Menunjukkan adanya hubungan diantara sejumlah entitas yang berbeda.
3		<i>Association</i>	Penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut.
4		Atribut	Mendesripsikan karakter entitas (atribut yang berfungsi sebagai key yang diberi garis bawah).

2.2.8 Pemrograman Berorientasi Objek (PBO)

Pemrograman berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis [22]. Syarat sebuah bahasa pemrograman bisa digolongkan sebagai berorientasi objek adalah bila bahasa pemrograman tersebut memiliki fitur untuk mengimplementasikan ke 4 (empat) konsep berorientasi objek yaitu *abstraksi*, *encapsulation*, *polymorphison*, dan *inheritance*. Metode berorientasi objek adalah sebuah sistem yang komponennya dibungkus menjadi kelompok data dan fungsi.

Sistem berorientasi objek merupakan sebuah sistem yang dibangun dengan berdasarkan metode berorientasi objek. Metode berorientasi objek adalah sebuah sistem yang komponennya dibungkus menjadi kelompok data dan fungsi. Setiap komponen dalam sistem tersebut dapat mewarisi atribut, sifat, komponen lainnya, dan dapat berinteraksi satu sama lain.

2.2.9 Basis Data

Basis Data (database) merupakan kumpulan dari suatu data yang terhubung dan disimpan secara bersama pada suatu media, diorganisasikan berdasarkan skema atau struktur tertentu, serta dapat dilakukan manipulasi untuk kegunaan tertentu dengan bantuan software. Basis data juga dapat didefinisikan sebagai kumpulan data yang disusun dalam beberapa tabel yang saling memiliki relasi satu sama lain atau berdiri sendiri [23].

DBMS (Database Management System) merupakan software yang didesain agar user dapat mendefinisikan, membuat, memelihara dan menyediakan akses terkontrol pada data [24]. Salah satu server yang menyediakan layanan sistem manajemen basis data adalah MySQL. *MySQL* adalah sebuah program database *server* yang mampu menerima dan mengirimkan datanya dengan sangat cepat, multi *user* serta menggunakan perintah standar *Structured Query Language* (SQL). *MySQL* memiliki dua bentuk lisensi, yaitu *Free Software* dan *Shareware*. *MySQL* yang biasa kita gunakan adalah *MySQL Free SoftwareI* yang berada dibawah Lisensi *General Public License* (GPL) [25]. Bahasa dalam basis data dibedakan menjadi 2 yaitu :

a. *Data Definition Language* (DDL)

Data Definition Language (DDL) merupakan suatu perintah yang berfungsi untuk mendefinisikan atribut atribut basis data, tabel, atribut serta hubungan antar tabel. DDL berfungsi lebih ke dalam memanipulasi struktur *database*. DDL digunakan untuk membuat tabel atau menghapus tabel, membuat *key* atau *indeks*, membuat relasi antar tabel. Berikut sintaks yang ada didalam DDL:

1. *Create*

Perintah *create* digunakan untuk membuat objek baru, baik berupa *database*, tabel, indeks atau prosedur yang tersimpan.

2. *Alter*

Perintah *alter* digunakan untuk memodifikasi onjek pada *database*, seperti *indeks*, dan lokasi.

3. *Drop*

Perintah *drop* digunakan untuk menghilangkan atau menghapus objek pada *database*.

b. *Data Manipulation Language* (DML)

DML adalah *sub* perintah dari bahasa SQL yang digunakan untuk memanipulasi data dalam *database* yang telah dibuat. Terdapat 4 (Empat) perintah penting dalam DML, yaitu :

- a. *INSERT*: perintah ini digunakan untuk memasukkan data baru ke dalam sebuah tabel. Perintah ini tentu saja bisa dijalankan ketika *database* dan tabel sudah dibuat. Contoh: *INSERT INTO* nama_tabel *VALUES* (data1, data2, dst...);
- b. *SELECT*: perintah ini digunakan untuk mengambil dan menampilkan data dari tabel atau bahkan dari beberapa tabel dengan penggunaan relasi. Contoh: *SELECT* nama_kolom1, nama_kolom2 *FROM* nama_tabel;
- c. *UPDATE*: perintah *update* digunakan untuk memperbaharui data pada sebuah tabel. Contoh: *UPDATE* nama_tabel *SET* kolom1=data1, kolom2=data2,... *WHERE* kolom=data;
- d. *DELETE*: perintah *delete* digunakan untuk menghapus data dari sebuah tabel. Contoh: *DELETE FROM* nama_tabel *WHERE* kolom=data;