

The background of the page is a repeating pattern of a watermark logo. The logo consists of a stylized gear or circular shape with three colored segments (blue, yellow, and grey) and the text 'POLITEKNIK NEGERI CILACAP' below it.

# **BAB II**

# **TINJAUAN PUSTAKA**

# **DAN LANDASAN TEORI**

## **BAB II**

### **TINJAUAN PUSTAKA DAN LANDASAN TEORI**

#### **2.1 Tinjauan Pustaka**

Dalam penelitian ini dicantumkan beberapa penelitian terdahulu untuk mendukung penelitian yang akan dijalankan. Penelitian sejenis dengan Metode *Multi Attribute Utility Theory* (MAUT) diimplementasikan pada Sistem Pendukung Keputusan dalam Menentukan Penerima Kredit. Pada penelitian ini diangkat masalah untuk menyelesaikan persoalan untuk mempermudah bagian *surveyor* dan *credit analyst* dalam melakukan penilaian penentuan penerima kredit (calon nasabah) serta meminimalisir kesalahan dalam penilaian penentuan penerima kredit. Hasil dari penelitian ini adalah membuat aplikasi sistem pendukung keputusan dengan menggunakan bahasa pemrograman PHP (*Hypertext Processor*) [5].

Penelitian selanjutnya mengenai sistem pendukung keputusan metode *Multi Attribute Utility Theory* (MAUT) pernah diteliti oleh Ratih HafSarah Maharrani, Abdul Rohman Supriyono dan Lutfi Syafirullah Dosen Politeknik Negeri Cilacap dengan judul “SIPGANG: Sistem Pendukung Keputusan Rekomendasi Magang Industri Berbasis *Multi Attribute Utility Theory* (MAUT)”. Sistem yang dibuat diharapkan dapat memberi kemudahan bagi para mahasiswa Politeknik Negeri Cilacap dalam memilih tempat magang sesuai dengan rekomendasi yang telah diberikan berdasarkan kriteria pada jam kerja, bobot tugas yang diberikan selama magang, kesesuaian tugas dengan keahlian, standar perusahaan, penerapan K3 di perusahaan, fasilitas dan peralatan praktik untuk peserta magang serta bidang keahlian. Hal tersebut dilakukan karena pemilihan magang sebelumnya beberapa mahasiswa mengeluhkan bahwa perusahaan tempat magang tidak sesuai dengan standar yang diinginkan, tidak sesuai bidang serta beban kerja yang berlebih sehingga dalam pelaksanaannya mahasiswa merasa tidak bisa optimal dalam menggunakan kemampuan yang dimiliki. Sistem ini menerapkan metode pengembangan system *Rapid Development Prototyping* (RAD). Hasil dari penelitian ini yaitu aplikasi pengelolaan magang yang dikembangkan dimulai dari proses pengajuan, bimbingan, penilaian seminar magang serta penilaian rekomendasi. Proses

rekomendasi dengan perhitungan metode MAUT itu sendiri akan dapat diproses apabila magang telah terlaksana oleh mahasiswa[6].

Penelitian berikutnya dilakukan oleh mahasiswa Fakultas Teknik Universitas Widyatama yang berjudul Sistem Pendukung Keputusan Pemilihan Karyawan Berprestasi Berdasarkan Kinerja Menggunakan Metode *Analityc Hierarchy Process*. Pada penelitian ini permasalahan yang terjadi yaitu dikarenakan kelonjakan jumlah karyawan yang semakin banyak sehingga keanekaragaman karyawan juga semakin kompleks maka sangat sulit memilih karyawan yang berprestasi menurut lembaga dan sulitnya menentukan prioritasnya. Hasil dari penelitian ini adalah dibuatkannya aplikasi sistem berbasis desktop untuk melakukan pemilihan karyawan berprestasi dengan menggunakan penilaian kinerja, *score TOEIC*, dan kedisiplinan kerja (kehadiran karyawan)[7].

Perbedaan penelitian yang dilakukan dengan penelitian yang telah ada sebelumnya penulis mengembangkan sistem rekam periode setiap bulan. Hasil pemilihan akan disimpan pada menu periode dan dapat dilihat pada saat pemilihan selanjutnya. Rekaman pemilihan ini juga bisa menjadi acuan untuk para staff karyawan sebagai pengingat agar dapat meningkatkan potensi yang lebih baik kedepannya. Adapun penelitian yang akan dikembangkan berjudul “Sistem Pendukung Keputusan Pemilihan Karyawan Terbaik Menggunakan Metode *Multi Attribute Utility Theory* Pada PT KAI Logistik *Express Cilacap*”. Selain itu metode pengembangan yang digunakan yaitu metode *Prototype*.

## **2.2 Landasan Teori**

### **2.2.1 Karyawan**

Karyawan merupakan seseorang yang bekerja pada suatu lembaga, instansi, atau perusahaan[8]. Karyawan terbaik sendiri merupakan seseorang yang memiliki potensi kualitas kinerja yang baik dan dapat memberikan hasil yang memuaskan sesuai dengan keinginan instansi/ lembaga[9].

### **2.2.2 Sistem Pendukung Keputusan**

Sistem merupakan suatu kumpulan jaringan kerja dari beberapa prosedur yang bertujuan untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu permasalahan. Sistem juga dapat diartikan sebagai

kumpulan jaringan atau elemen yang saling berhubungan guna memproses suatu masukan (*input*) yang kemudian diolah oleh sistem dan menghasilkan keluaran (*output*). Prosedur sendiri merupakan urutan operasi tulis menulis yang melibatkan beberapa orang dalam pengoperasiannya[8].

Keputusan merupakan suatu tindakan memilih strategi untuk memecah suatu masalah. Pengambilan keputusan merupakan tindakan yang diyakini akan memberikan solusi terbaik atas sesuatu itu. Tujuannya adalah untuk mencapai target atau aksi tertentu yang harus dilakukan[5].

SPK atau Sistem Pendukung Keputusan merupakan sistem yang menghasilkan informasi yang terdiri dari beberapa alternatif keputusan oleh manager untuk membantu menyelesaikan suatu masalah dengan mengumpulkan data dan model[10].

### 2.2.3 Metode Multi Attribute Utility Theory (MAUT)

Metode MAUT merupakan metode rancangan evaluasi yang perhitungannya menggunakan bobot dan nilai terkait dengan alternatif. Dapat juga didefinisikan sebagai algoritma suatu skema pendekatan evaluasi akhir  $v(x)$  pada sebuah nilai bobot dalam dimensinya atau yang biasa disebut dengan utilitas[11].

Untuk menentukan nilai bobot alternatif dapat dilakukan menggunakan persamaan berikut,

$$v(x) = \sum_{i=1}^n w_i v_i(x) \dots\dots\dots (1)$$

- $v(x)$  = evaluasi total alternatif ke -x
- $w_i$  = bobot relatif kriteria ke -i
- $v_i(x)$  = hasil evaluasi atribut (kriteria) ke -i untuk alternatif ke -x
- $i$  = indeks untuk menunjukkan kriteria
- $n$  = jumlah kriteria

Lalu untuk menentukan  $v_i(x)$  menjadi skala 0-1 dibutuhkan rumus  $U(x)$ , dengan rumus sebagai berikut:

$$U(x) = \frac{x - x_i^-}{x_i^+ - x_i^-} \dots\dots\dots (2)$$

- $U(x)$  = Nilai utilitas dari setiap kriteria alternative ke -x
- $x_i^-$  = Nilai terkecil dari kriteria ke -i di semua alternatif
- $x_i^+$  = Nilai terbesar dari kriteria ke -i di semua alternatif

Lalu untuk menghitung bobot ( $w_i$ ), diperlukan rumus sebagai berikut:

$$\sum_{i=1}^n w_i = 1 \dots\dots\dots (3)$$

#### 2.2.4 PHP (Hypertext Preprocessor)

Pemrograman ini banyak digunakan untuk membuat program web situs dinamis dan merupakan suatu bahasa pemrograman script yang ramai digunakan di era sekarang. Meskipun tidak menutup bahwa Bahasa pemrograman ini bisa dipakai untuk kegunaan lain. *Rumus Lerdorf* adalah seorang pencipta PHP pertama di tahun 1995, ketika itu memiliki nama FI (*Form Interpreted*). Dan merupakan sekelompok *script* yang dapat dipakai saat pengolahan data *form* dari *web*[12].

#### 2.2.5 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak (*software engineering*) merupakan pembangunan dengan menggunakan prinsip atau konsep rekayasa dengan tujuan menghasilkan perangkat lunak yang bernilai ekonomi yang dipercaya dan bekerja secara efisien menggunakan mesin[13].

##### 2.2.5.1 Metode Pengembangan Sistem

Dalam merancang dan membangun Sistem Pendukung Keputusan Pemilihan Karyawan Terbaik Menggunakan Metode *Multi Attribute Utility Theory* Berbasis Web ini menggunakan metode *prototype*. Untuk tahapan pengembangannya sebagai berikut :

1. Analisa Kebutuhan  
Proses pengumpulan kebutuhan dilakukan secara intensif untuk mespesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*.
2. Membangun *Prototyping*  
Membuat rencana *prototype* sistem pendukung keputusan menggunakan pemodelan UML dengan langkah-langkah sebagai berikut :
  - a. Menentukan perencanaan awal  
Pada tahap ini dibuat perencanaan mengenai kegiatan apa saja yang akan dilakukan beserta waktu yang dibutuhkan untuk masing-masing kegiatan.

- b. Melakukan analisa proses bisnis  
Pada tahap ini dilakukan analisis terhadap proses penilaian kinerja karyawan pada PT KALOG Cilacap.
  - c. Memodelkan sistem pendukung keputusan dengan menggunakan UML  
Pada tahap ini dibuat pemodelan kebutuhan sistem pendukung keputusan pemilihan karyawan terbaik dengan menggunakan UML yaitu *use case diagram*, *sequence diagram*, dan *class diagram*.
  - d. Membangun *prototype* sistem  
Pada tahap ini dibuat *prototype* sistem berupa *user interface*.
3. Evaluasi *Prototyping*  
Pada tahap ini, bertujuan sebagai dasar pertimbangan *prototype* yang telah dirancang untuk kebutuhan yang akan dibutuhkan suatu sistem. Jika kebutuhan sistem sudah sesuai maka dapat dilanjut ke tahapan selanjutnya. Tetapi jika kebutuhan sistem belum sesuai maka harus mengulangi ke tahap sebelumnya.
  4. Mengkodekan Sistem  
Pada tahap ini hasil pembuatan *prototype* yang sudah sesuai dengan yang diinginkan akan diterjemahkan ke dalam bahasa pemrograman.
  5. Menguji Sistem  
Tahap ini dilakukan untuk menguji sistem yang telah dibuat.
  6. Evaluasi Sistem  
Setelah tahap pengujian telah dilakukan. Dilakukannya tahap evaluasi guna mengetahui apakah sistem sesuai dengan yang diharapkan. Jika sudah, maka dapat dilanjutkan ke tahap akhir. Jika belum, maka harus mengulang ke tahap 4 dan 5.
  7. Menggunakan Sistem  
Pada tahap terakhir ini, sistem yang telah diuji dan disetujui tandanya sistem sudah siap dan layak untuk digunakan.

### 2.2.5.2 Metode Pengujian Sistem

Terkait dengan pengujian program, pengujian yang akan digunakan adalah pengujian *black-box*. Pengujian *black-box* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian *black-box* memungkinkan rekayasa perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian *black-box*

bukan merupakan alternatif dari teknik *white-box*, tetapi merupakan pendekatan komplementer yang memungkinkan besar mampu mengungkap kelas kesalahan daripada metode *white-box*[14]. Pengujian *black-box* berusaha menemukan kesalahan dalam kategori sebagai berikut :

1. Fungsi-fungsi yang tidak benar atau hilang,
2. Kesalahan *interface*,
3. Kesalahan dalam struktur data atau akses *database* eksternal,
4. Kesalahan kinerja,
5. Inisialisasi dan kesalahan terminal.

## 2.2.6 Tools/ Alat Bantu Penelitian

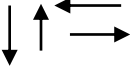
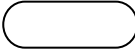
### 2.2.6.1 Flowchart


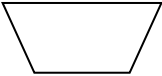

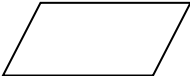

*Flowchart* adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam sebuah program atau prosedur sistem secara logika. Bagan alir (*flowchart*) digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi. Ada beberapa jenis-jenis *flowchart* diantaranya :

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>	Yaitu simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain.
2		<i>Terminator Symbol</i>	Yaitu simbol untuk permulaan ( <i>start</i> ) atau akhir ( <i>end</i> ) dari suatu kegiatan.

No	Simbol	Nama	Keterangan
3		<i>Processing Symbol</i>	Yaitu simbol yang menunjukkan pengolahan yang dilakukan oleh komputer.
4		<i>Manual Operation Symbol</i>	Yaitu simbol yang menunjukkan pengolahan data yang tidak dilakukan oleh komputer.
5		<i>Decision Symbol</i>	Yaitu simbol untuk pemilihan proses berdasarkan kondisi yang ada.
5		<i>Input-Output Symbol</i>	Yaitu simbol yang menyatakan proses input dan output tanpa tergantung dari jenis peralatannya.
6		<i>Document Symbol</i>	Yaitu simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas.

### 2.2.6.2 Unified Modeling Language (UML)

*Unified Modeling Language* (UML) adalah hasil kerja dari konsorium berbagai organisasi yang berhasil dijadikan sebagai standar baku dalam *Object Oriented Analysis* dan *Design* (OOAD). Macam-macam dari *Unified Modeling Language* (UML) antara lain : *use case diagram*, *sequence diagram* dan *class diagram*.

#### a. Use Case Diagram




*Use Case* adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use Case* bekerja dengan cara mendeskripsikan



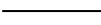






tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem tersebut *scenario* sedangkan pengguna disebut *actor*. *Actor* adalah sebuah peran yang biasa dimainkan 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*.

Berdasarkan definisi diatas maka dapat disimpulkan bahwa *Use Case* adalah kontruks untuk mendeskripsikan bagaimana sistem akan terlihat dimata pengguna potensial yang terdiri dari sekumpulan *scenario* dan *actor*. Sedangkan *use case* diagram memfasilitasi komunikasi diantara analis dan pengguna serta analis dan klien. Simbol *Use Case* dapat dilihat pada Table 2.2.

**Tabel 2. 2** Simbol *Use Case Diagram*

No	Simbol	Nama	Keterangan
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri ( <i>not independent</i> ).
3		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).


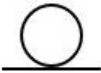
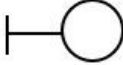


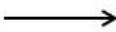
No	Simbol	Nama	Keterangan
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association,</i>	Yang menghubungkan antara objek satu dengan objek lainnya
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor.
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerjasama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

### b. *Sequence Diagram*

*Sequence Diagram* digunakan untuk menggambarkan perilaku pada sebuah *scenario*. Diagram ini menunjukkan sejumlah obyek dan *message* yang diletakan antara obyek-obyek didalam *use case*.

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-simbol yang dipakai dalam pembuatan.

**Tabel 2. 3** Simbol *Sequence Diagram*

No	Simbol	Nama	Keterangan
1		<i>Actor</i>	Menggambar 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 &amp; A Life Line</i>	Menggambarkan tempat mulai dan berakhirnya <i>massage</i> .
6		<i>A Message</i>	Menggambarkan Pengiriman Pesan.







### c. *Class Diagram*

*Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan *variable-variabel* yang dimiliki oleh suatu kelas. Sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Diagram kelas dibuat agar *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron.

*Class diagram* membantu dalam visualisasi struktur kelas – kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak. *Class diagram* memperlihatkan hubungan antar kelas dan penjelasan detail tiap – tiap kelas di dalam model desain (dalam logical view) dari suatu sistem. Selama proses analisis, *class diagram* memperlihatkan aturan – aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Selama proses analisis, *class diagram* memperlihatkan aturan – aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Selama tahap decian, *class diagram* berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat

Berikut beberapa simbol dari *class diagram* :

**Tabel 2. 4** Simbol *Class Diagram*

No	Simbol	Nama	Keterangan
1		<i>Class</i>	Kelas pada struktur sistem.
2		<i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
3		<i>Directed association</i>	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain.
4		<i>Generalization</i>	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus).
5		<i>Dependency</i>	Relasi antarkelas dengan makna kebergantungan antarkelas.
6		<i>Aggregation</i>	Relasi antarkelas dengan makna semua-bagian ( <i>whole-part</i> ).

### 2.2.7 Pemrograman Berorientasi Objek

Metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan proyek yang berisi data dan operasi yang diperlakukan terhadapnya. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis.

Sistem berorientasi objek merupakan sebuah sistem yang dibangun dengan berdasarkan 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.

### 2.2.8 Basis Data (*Database*)

Basis data adalah kumpulan file-file yang mempunyai kaitan antara satu file dengan file lain sehingga membentuk satu bangun data untuk penginformasikan suatu perusahaan instansi, dalam bahasan tertentu.

*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 Software* yang berada dibawah Lisensi *General Public License (GPL)*[15]. *MySQL* juga dapat didefinisikan sebagai sebuah database server, dapat juga berpersan sebagai *client* sehingga sering disebut *database client / server* yang *open source* dengan kemampuan dapat berjalan baik di Operasi Sistem maupun dengan *Platform Windows* maupun *Linux*.

*MySQL* dikembangkan oleh sebuah perusahaan Swedia bernama *MySQL AB*, yang pada saat itu bernama *TcX Data Konsult AB* sekitar tahun 1994-1995. *MySQL* sudah ada sejak 1979. *MySQL* termasuk jenis *Relation Database Management System (RDBMS)* digunakan oleh banyak portal-portal internet sebagai basis data dari informasi yang ditampilkan pada situs *web*. Kepopuleran *MySQL* dimungkinkan karena kemudahannya untuk digunakan, cepat secara kinerja *query*, dan mencukupi untuk kebutuhan basis data perusahaan-perusahaan skala menengah dan kecil. Istilah seperti tabel, baris, dan kolom tetap

digunakan dalam *MySQL*. Sebuah basis data yang terdapat pada *MySQL* mengandung satu atau beberapa tabel yang terdiri dari sejumlah baris dan kolom.

*MySQL* sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah suatu bahasa (*language*) yang digunakan untuk mengakses data di dalam sebuah *database* relasional. SQL sering juga disebut dengan istilah *query*, dan bahasa SQL secara praktiknya digunakan sebagai bahasa standar untuk manajemen database relasional. Hingga saat ini hampir seluruh *server database* atau *software database* mengenal dan mengerti bahasa SQL. Dalam penggunaan SQL terdapat beberapa perintah yang berguna untuk mengakses dan memanajemen data yang terdapat dalam database. Secara garis besar, *SQL Server* mempunyai 3 (Tiga) jenis perintah SQL yaitu :

### 1. *Data Definition Language (DDL)*

DDL adalah sub perintah dari bahasa SQL yang digunakan untuk membangun kerangka sebuah database, dalam hal ini database dan table. Terdapat tiga perintah penting dalam DDL, yaitu :

- a. *CREATE*: perintah ini digunakan untuk membuat, termasuk di dalamnya membuat database baru, tabel baru *view* baru, dan kolom baru. Contoh: *CREATE DATABASE* nama\_ *database*.
- b. *ALTER*: perintah *ALTER* berfungsi untuk mengubah struktur tabel yang telah dibuat. Mencakup di dalamnya mengubah nama tabel, menambah kolom, mengubah kolom, menghapus kolom, dan memberikan atribut pada kolom. Contoh: *ALTER TABLE* nama\_ *tabel* *ADD* nama\_ *kolom* *datatype*.
- c. *DROP*: perintah *DROP* berfungsi untuk menghapus *database* atau tabel. Contoh: *DROP DATABASE* nama\_ *database*.

### 2. *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;