

BAB II

TINJAUAN PUSTAKA

II.1. Sistem

Sistem merupakan kumpulan dari unsur atau elemen-elemen yang saling berkaitan/berinteraksi dan saling mempengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan tertentu (Asbon Hendra; 2012 : 157).

II.1.1.Sistem Pendukung Keputusan(*Decision Support System*)

DSS merupakan sistem informasi interaktif yang menyediakan informasi, pemodelan, dan pemanipulasian data. Sistem itu digunakan untuk membantu pengambilan keputusan dalam situasi yang semiterstruktur dan situasi yang tidak terstruktur, dimana tidak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat (Alter 2002). DSS biasanya dibangun untuk mendukung solusi atau suatu masalah atau untuk mengevaluasi suatu peluang. DSS yang seperti itu disebut aplikasi DSS. Aplikasi DSS digunakan dalam mengambil keputusan. Aplikasi DSS menggunakan CBIS (*Computer Based Information System*) yang fleksibel, interaktif, dan dapat diadaptasi, yang dikembangkan untuk mendukung solusi atas masalah manajemen spesifikasi yang tidak terstruktur.

Aplikasi DSS menggunakan data, memberikan antarmuka pengguna yang mudah, dan dapat menggabungkan pemikiran pengambilan keputusan. DSS

lebih ditujukan untuk mendukung manajemen dalam melakukan pekerjaan yang bersifat analisis dalam situasi yang kurang terstruktur dan dengan kriteria yang kurang jelas. DSS dimaksudkan untuk mengotomatisasikan pengambilan keputusan, tetapi memberikan perangkat interaktif yang memungkinkan pengambilan keputusan untuk melakukan berbagai analisis menggunakan model-model yang tersedia(Kusrini;2007:15).

II.1.2.AHP (*Analytical Hirarchi Proses*)

Ciri khas suatu DSS adalah digunakannya model yang salah satu fungsinya untuk penyederhanaan masalah. AHP yang dikembangkan oleh Thomas L saaty merupakan model hierarki fungsional dengan input utamanya persepsi manusia. Dengan adanya hierarki masalah yang kompleks atau tidak terstruktur dipecah dalam sub-sub masalah kemudian disusun menjadi suatu bentuk hierarki. AHP mempunyai kemampuan untuk memecah masalah multi-kriteria yang berdasarkan pada perbandingan preferensi dari setiap elemen dalam hierarki . Kriteria seleksi yang dimaksud dalam penelitian ini adalah kriteria penilaian yang digunakan oleh organisasi dalam menentukan seleksi tenaga kerja.

Adapun langkah-langkah dalam metode AHP sebagai berikut :

1. Mendefenisikan masalah dan menentukan solusi yang diinginkan
2. Membuat struktur hierarki, yang diawali dengan tujuan umum, dilanjutkan dengan sub-sub tujuan, kriteria dan kemungkinan *alternative* pada tingkatan kriteria yang paling bawah.

3. Membuat matriks perbandingan berpasangan yang menggambarkan kontribusi relatif pengaruh setiap elemen terhadap masing-masing tujuan kriteria yang setingkat di atasnya. Perbandingan berdasarkan “*judgment*” dari pengambilan keputusan dengan menilai tingkat kepentingan suatu elemen dibandingkan elemen lainnya.
4. Melakukan perbandingan berpasangan sehingga diperoleh *judgment* seluruhnya
5. Menghitung nilai eigen dan menguji konsisten maka pengambilan data diulangi
6. Mengulangi langkah 3,4 dan 5 untuk keseluruhan tingkat hierarki
7. Menghitung vektor eigen dari setiap matriks perbandingan berpasangan. Nilai vektor eigen merupakan bobot setiap elemen. Langkah ini untuk mensintesis *judgment* dalam penentuan prioritas elemen-elemen pada tingkat hirarki terendah sampai mencapai tujuan.
8. Memeriksa konsistensi hirarki. Jika nilainya lebih dari 10% maka penilainya data *judgment* harus diperbaiki.

Ada dua cara dalam mendapatkan tingkat keyakinan dari sebuah metode AHP, yaitu:

1. Metode AHP’ Mencari nilai Consistency Index (CI) yaitu dengan Persamaan

$$CI = \frac{\lambda_{maks} - n}{n - 1}$$

Keterangan :

- a. CI = *Consistency Index*
 - b. $\lambda \text{ maks} - n$ = *Nilai rata-rata dari keseluruhan kriteria / subkriteria*
 - c. n = Jumlah matrik perbandingan suatu kriteria / subkriteria
2. Metode AHP' Mencari *Consistency Ratio (CR)* dengan mengacu pada nilai index random dengan persamaan :

$$CR = \frac{CI}{RI}$$

Keterangan :

- a. CR = *Consistency Ratio*
 - b. CI = *Consistency Index*
 - c. RI = *Random Indeks*
3. Metode AHP' Melakukan perhitungan nilai keseluruhan dari alternatif suatu subkriteria, yaitu dengan melakukan penggabungan antara metode AHP seperti pada persamaan :

$$Vi = \sum_i w_j \times ij$$

Keterangan :

- a. Vi = Nilai keseluruhan dari alternative pilihan suatu subkriteria
- b. Wj = TPV (Bobot prioritas) Subkriteria yang didapat dengan menggunakan metode AHP
- c. Xij = Nilai alternatif pilihan subkriteria
- d. I = Alternatif pilihan

e. j = Subkriteria

Kelebihan dari metode *AHP* adalah:

1. Struktur yang berhierarki sebagai konsekuensi dari kriteria yang dipilih sampai pada sub-sub kriteria yang paling dalam.
2. Memperhitungkan validitas sampai batas toleransi inkonsistensi sebagai kriteria dan alternatif yang dipilih oleh para pengambil keputusan.
3. Memperhitungkan daya tahan atau ketahanan output analisis sensitivitas pengambilan keputusan.

Sedangkan kekurangan metode *AHP* adalah:

1. Ketergantungan Model *AHP* pada input utamanya. Input utama ini berupa persepsi seorang ahli sehingga dalam hal ini melibatkan subjektivitas sang ahli selain itu juga model menjadi tidak berarti jika ahli tersebut memberikan penilaian yang keliru.
2. Metode *AHP* ini hanya metode matematis tanpa ada pengujian secara statistik sehingga tidak ada batas kepercayaan dari kebenaran model yang terbentuk (Jurnal Teknologi Informasi, Penerapan Metode *AHP*).

II.2.UML (*Unified Modeling Language*)

UML (*Unified Modeling Language*) adalah suatu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek (Munawar ; 2005 : 17). Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta

dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain. Adapun tipe diagram UML yang ada seperti pada tabel II.1.

Tabel II.1 Tipe Diagram UML

Diagram	Tujuan	Keterangan
Activity	Prilaku prosedural dan parallel	Sudah ada di UML 1
Class	Class, fitur dan relasinya	Sudah ada di UML 1
Communication	Interaksi diantara objek. Lebih menekankan kepada link	Di UML 1 disebut collaboration
Component	Struktur dan koneksi dari komponen	Sudah ada di UML 1
Composite Structure	Dekomposisi sebuah class saat runtime	Baru untuk UML 2
Deployment	Penyebaran/instalasi ke klien	Sudah ada di UML 1
Interaction Overview	Gabungan dari activity dan sequence diagram	Baru untuk UML 1
Object	Contoh konfigurasi instance	Tidak resmi ada di UML 1
Package	Struktur hierarki saat kompilasi	Tidak resmi ada di UML 1
Sequence	Interaksi antara objek. Lebih menekankan pada urutan.	Sudah ada di UML 1
State Machine	Bagaimana event mengubah sebuah objek	Sudah ada di UML 1
Timing	Interaksi antar objek. Lebih menekankan pada waktu	Sudah ada di UML 1
Use Case	Bagaimana user berinteraksi dengan sebuah sistem	Sudah ada di UML 1

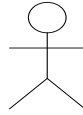
Sumber : Munawar (2005 : 23)

II.2.1. Notasi Dasar UML

1. Aktor

Aktor adalah abstraction dari orang dan system yang lain yang mengaktifkan fungsi dari target system. Orang atau system bisa muncul dalam

Beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan *use case*, tetapi tidak memiliki kontrol atas *use case*. Berikut notasi *aktor* dalam UML :

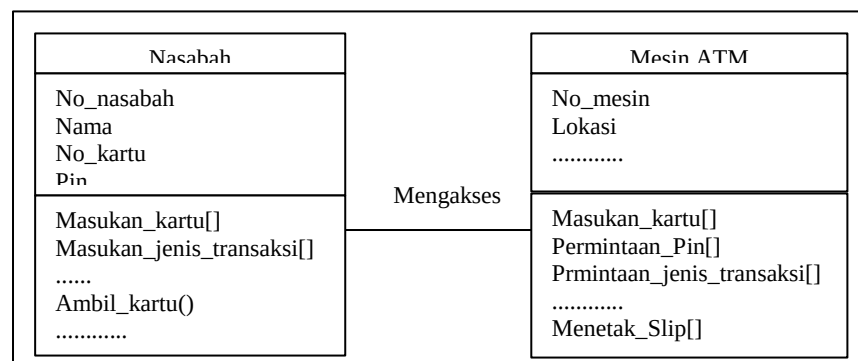


Gambar II.1 Notasi Actor pada UML

Sumber : Munawar (2005: 64)

2. Class Diagram

Class, dalam notasi UML digambarkan dengan kotak. Nama class menggunakan huruf besar diawal kalimatnya dan diletakkan diatas kotak. Bila *class* mempunyai nama yang terdiri dari 2 suku kata atau lebih, maka semua suku kata digabungkan tanpa spasi dengan huruf awal tiap suku kata menggunakan huruf besar. Notasi *class* dalam UML dapat dilihat pada gambar II.2 berikut :

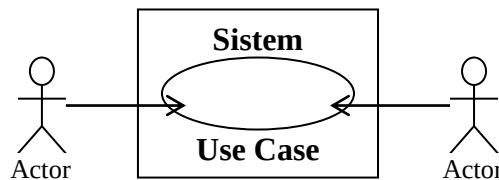


Gambar II.2 : Diagram Class Sederhana

Sumber : Adi Nugroho (2009 : 39)

3. Use case

Use case adalah alat bantu terbaik guna menstimulasi pengguna potensi untuk mengatakan tentang suatu system dari sudut pandangnya. Tidak selalu mudah bagi pengguna untuk menyatakan bagaimana mereka bermaksud menggunakan sebuah system. Karena system pengembangan tradisional sering ceroboh dalam melakukan analisis, akibatnya pengguna seringkali susah menjawabnya tatkala diminta masukan tentang sesuatu. Notasi use case dapat dilihat pada gambar II.3 :



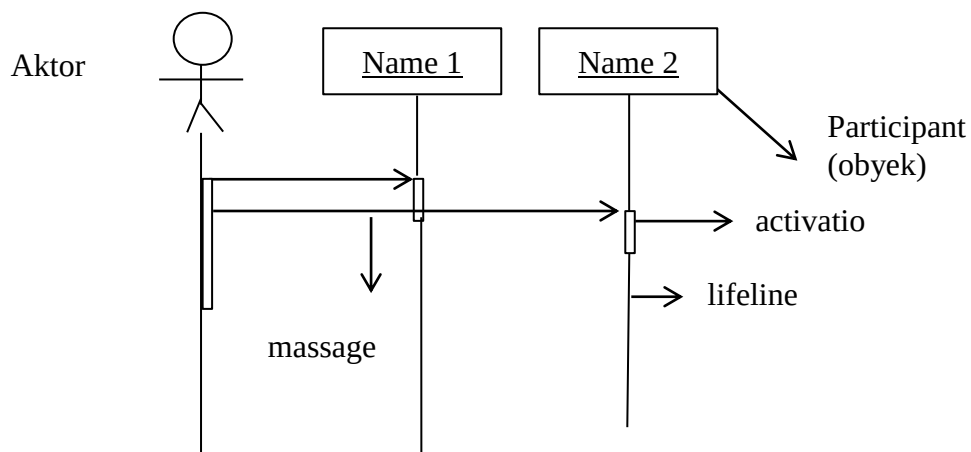
Gambar II.3 Notasi Use Case pada UML

Sumber : Munawar (2005 : 64)

4. Sequence Diagram

Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan jumlah contoh objek dan message (pesan) yang diletakkan diantara objek-objek ini dalam use case .

Komponen utama sequence diagram terdiri atas obyek yang dituliskan dengan kotak segiempat bernama. Message diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan progress vertical.



Gambar II.4 Contoh Sequence Diagram

Sumber : Munawar (2005 : 89)

5. Activity Diagram

Activity Diagram adalah teknik untuk mendiskripsikan logika prosedural. Proses bisnis dan aliran dalam banyak kasus. Activity Diagram mempunyai seperti halnya flowchart, akan tetapi perbedaannya dengan flowchart adalah activity diagram bisa mendukung perilaku paralel sedangkan flowchart tidak bisa. Simbol-simbol yang sering digunakan pada saat pembuatan activity diagram dapat dilihat pada tabel II.2 berikut :

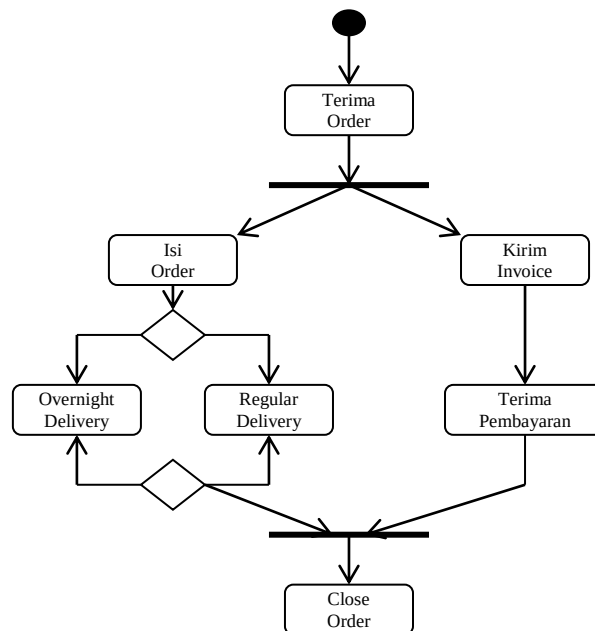
Tabel II.2 Simbol-simbol yang sering dipakai pada Activity Diagram

Simbol	Keterangan
●	Titik awal
●	Titik akhir
▭	Activity
◇	Pilihan untuk pengambilan keputusan
—	Fork; digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.

	Rake; menunjukkan adanya dekomposisi
	Tanda waktu
	Tanda pengiriman
	Tanda penerimaan
	Aliran akhir (Flow Final)

Sumber :Munawar (2005 : 109)

Adapun contoh dari *Activity Diagram* dapat di lihat pada Gambar II.5.



Gambar II.5 Contoh Activity Diagram Sederhana

Sumber :Munawar (2005: 111)

II.3. Pengertian Database

Database merupakan komponen terpenting dalam pembangunan SI, karena menjadi tempat untuk menampung dan mengorganisasikan seluruh data yang ada dalam sistem, sehingga dapat dieksplorasi untuk menyusun informasi-informasi dalam berbagai bentuk. Database tersebut diorganisasikan sedemikian rupa agar tidak terjadi duplikasi yang tidak perlu,

sehingga dapat diolah atau dieksporasi secara cepat dan mudah untuk menghasilkan informasi (Budi Sutedji Dharma Oetomo ; 2006 : 99).

II.3.1.Hierarki Data Dalam Database

Data dalam sebuah database disusun berdasarkan sistem hierarki yang unik yaitu :

1. Database, merupakan kumpulan file yang saling terkait satu sama lain, misalnya file data induk karyawan, file jabatan file penggajian dan lain sebagainya. Kumpulan file yang tidak saling terkait satu sama lain tidak dapat disebut database, misalnya file data induk karyawan, file tamu undangan perkawinan, file barang retail pasar swalayan.
2. File, yaitu kumpulan dari record yang saling terkait dan memiliki format *field* yang sama dan sejenis.
3. Record, yaitu kumpulan *field* yang menggambarkan suatu unit data individu tertentu.
4. Field, yaitu atribut dari *field* yang menunjukkan suatu item dari data, seperti nama,alamat, dan lain sebagainya.
5. Byte, yaitu atribut dari *field* yang berupa huruf yang membentuk nilai dari sebuah *field*. Huruf tersebut dapat berupa numerik maupun abjad atau karakter khusus.
6. Bit, yaitu bagian terkecil dari data secara keseluruhan, yaitu berupa karakter ASCII nol atau satu yang merupakan komponen pembentuk *byte* (Budi Sutedjo Dhrama Oetomo ; 2006 : 102).

II.4. Entity Relationship Diagram – ERD

II.4.1. Model-model Data

Struktur yang mendasari suatu basisdata adalah model data yang merupakan kumpulan alat-alat konseptual untuk mendeskripsikan data, relasi data, data semantik, dan batasan konsistensi. Untuk menghitung konsep model data, berikut disajikan dua model data, yaitu *entity relationship model* dan *relational model*. Kedua model menyediakan cara mendeskripsikan rancangan basisdata pada tingkatan logis (Janner Simarmata & Imam prayudi ; 2006 : 59).

II.4.2. Entity Relationship Model

Entity Relationship (ER) data model didasarkan pada persepsi terhadap nyata yang tersusun atas kumpulan objek-objek dasar yang disebut entitas dan hubungannya antar objek. Entitas adalah sesuatu atau objek dalam dunia nyata yang dapat dibedakan dari objek lain. Sebagai contoh, masing-masing mahasiswa adalah entitas dan mata kuliah dapat pula dianggap sebagai entitas.

Entitas digambarkan dalam basisdata dengan kumpulan atribut. Misalnya atribut nim, nama, alamat dan kota bisa menggambarkan data mahasiswa tertentu dalam universitas. Atribut-atribut membentuk entitas mahasiswa. Demikian pula atribut kodeMK, namaMK, dan SKS mendeskripsikan entitas mata kuliah.

Atribut NIM digunakan untuk mengidentifikasi mahasiswa secara unik karena dimungkinkan terhadap dua mahasiswa dengan nama, alamat, dan

kota yang sama. Pengenal unik harus diberikan pada masing-masing mahasiswa.

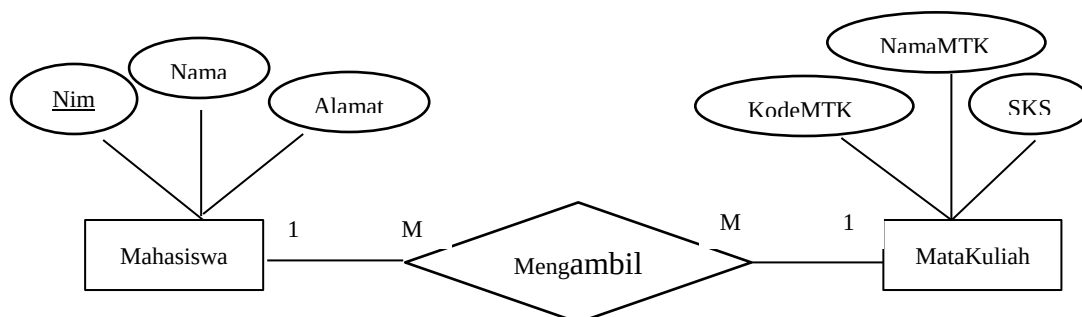
Relasi adalah hubungan antara beberapa entitas. Sebagai contoh, relasi menghubungkan mahasiswa dengan mata kuliah yang diambil. Kumpulan semua entitas bertipe sama disebut kumpulan entitas (*entity set*), sedangkan kumpulan semua relasi bertipe sama disebut kumpulan relasi (*relationship set*)

Struktur logis (skema *database*) dapat ditunjukkan secara grafis dengan diagram ER yang dibentuk dari komponen-komponen berikut :

Tabel II.3 Notasi ERD (*Entity Relationship Diagram*)

Entitas	Persegi panjang mewakili kumpulan entitas
Atribut	Elips mewakili atribut
Relasi	Belah ketupat mewakili relasi
—	Garis menghubungkan atribut dengan kumpulan entitas dan kumpulan entitas dengan relasi

Sumber : Janner Simarmata & Imam Prayudi (2006 : 60)



Gambar II.6 Diagram ER

Sumber : Janner Simarmata & Imab Prayudi(2006;60).

II.5. Kamus Data

Kamus data ikut berperan dalam perancangan dan pengembangan SI karena peralatan ini berfungsi untuk :

1. Menjelaskan arti aliran data dan penyimpanan dalam penggambaran dalam data flow diagram.
2. Mendeskripsikan komposisi paket data yang bergerak melalui aliran, misalnya data alamat diurai menjadi nama jalan, nomor, kota, negara dan kode pos.
3. Menjelaskan spesifikasi nilai dan satuan yang relevan terhadap data yang mengalir dalam sistem tersebut.

Jumlah simbol yang digunakan dalam penggambaran kamus data ditunjukkan dalam tabel.II.3.

Tabel II.4 Notasi Kamus Data

Notasi	Arti
=	Terdiri atas,mendefinisikan, diuraikan menjadi, artinya contoh : nama=sebutan+nama1+nama+gelar1+gelar2
+	Dan
()	Optional (pilihan boleh ada atau boleh tidak) Contoh : alamat=alamat rumah+(alamat surat)
{ }	Pengulangan Contoh : nama1={karakter_valid}
[]	Memilih salah satu dari sejumlah alternatif,seleksi Contoh : sebutan =[Bapak Ibu yang ,mulia]
**	Komentar Contoh :*seminar yang akan diikuti*
	Pemisah sejumlah alternatif pilihan antara simbol []

Sumber : Budi Sutedjo Dharma Oetomo (2006:118).

II.6. Normalisasi

Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basis data relasional. Pada dasarnya, normalisasi adalah proses dua langkah yang meletakkan data dalam bentuk tabulasi dengan menghilangkan kelompok berulang lalu menghilangkan data yang terduplikasi dari tabel relational.

Teori normalisasi didasarkan pada konsep bentuk normal. Sebuah tabel relasional dikatakan berada pada bentuk normal tertentu jika tabel memenuhi himpunan batasan tertentu. Ada lima bentuk normal yang telah ditemukan (Janner Simarmata & Imam Prayudi ; 2006: 76) .

1.Bentuk Normal Pertama (1NF/First Normal Form), bentuk normal pertama adalah suatu bentuk relasi dimana atribut bernilai banyak (*multivalues attribute*) telah dihilangkan sehingga kita akan menjumpai nilai tunggal (mungkin saja nilai *null*) pada perpotongan setiap baris dan kolom satu nilai untuk irisan baris dan kolom pada tabel.

2.Bentuk Normal Kedua (2NF/Second Normal Form), semua kebergantungan fungsional (*functional dependency*) yang bersifat sebagian (*partial functional dependency*) telah dihilangkan.

3.Bentuk Normal Ketiga (3NF/Third Normal Form), semua kebergantungan transitif (*transitive dependency*) telah dihilangkan.

4.Boyce-Codd Normal Form (BCNF/Boyce-Codd Normal Form), semua anomali yang tersisa dari hasil penyempurnaan kebergantungan fungsional (*functional dependency*) diatas telah dihilangkan.

5. Bentuk Normal Keempat (4NF/Fifth Normal Form), semua anomali yang berasal dari kebergantungan banyak-nilai (*multivalued dependency*) telah dihilangkan (Adi Nugroho ; 2010: 34).

Tujuan normalisasi adalah membuat kumpulan tabel relasional yang bebas dari data berulang yang dapat dimodifikasi secara benar dan konsisten. Ini berarti bahwa semua tabel pada basisdata relasional harus berada pada bentuk normal ketiga (3NF). Sebuah tabel relasional berada pada 3NF jika dan hanya jika semua kolom bukan kunci adalah (a) saling independen dan (b) sepenuhnya tergantung pada kunci utama. Saling independen berarti bahwa tidak ada kolom bukan kunci yang tergantung pada senbarang kombinasi kolom lainnya. Dua bentuk normal pertama adalah langkah antara untuk mencapai tujuan, yaitu mempunyai semua tabel dalam 3NF (Stephens and Plew, 2000) (Janner Simarmata & Imam Prayudi ; 2006 : 77).

II.7. PHP (*Hypertext Preprocessor*)

Sebelum kita memasuki Bab pemrograman PHP, ada baiknya kita mengenal pengertian dari PHP. Agar kita lebih memantapkan langkah belajar bahasa pemrograman PHP.

Pengertian PHP atau kependekatan dari Hypertext Preprocessor adalah salah satu bahasa pemrograman open source yang sangat cocok atau dikhususkan untuk pengembangan Web dan dapat ditanamkan pada sebuah skrip HTML. PHP diciptakan untuk mempermudah pengembang web dalam menulis halaman web dinamis dengan cepat, bahkan lebih dari itu kita dapat mengeksplorasi hal-hal yang luar biasa dengan PHP. Sehingga dengan demikian

PHP sangat untuk/bagi para pemula, menengah maupun expert sekalipun(A.M Hirin & Virgi;2011:25).

PHP dan Internet sebenarnya saat kita berinternet menggunakan browser seperti Mozilla, Internet Explorer, Opera, dan Safari, halaman yang muncul di depan layar computer kita adalah halaman yang diperoleh dari proses pemanggilan dengan menuliskan alamat sesuai nama file yang terdapat pada web server. Web server adalah aplikasi yang berfungsi untuk melayani permintaan pemanggilan alamat dari pengguna melalui web browser, dimana web server mengirimkan kembali informasi yang diminta tersebut melalui HTML (HiperText Transfer Protocol) untuk ditampilkan ke layar monitor computer kita. Agar kita dapat mengubah isi dari website yang dibuat, kita membutuhkan program PHP. Script-script PHP tersebut yang berfungsi membuat halaman website menjadi dinamis. Dinamis artinya pengunjung web dapat memberikan komentar saran/masukan pada website kita. Website yang kita buat menjadi lebih hidup karena ada komunikasi antara pengunjung diatas sebagai webmastermya(Anhar;2010:4-5) .

II.8. Database MySQL (*phpMyAdmin*)

PhpMyAdmin adalah aplikasi berbasis web yang dibuat dari pemograman *PHP*. *PhpMyAdmin* juga dapat disebut sebagai tools yang berguna untuk mengakses database MySQL Server dalam bentuk tampilan *web*. dengan adanya *phpMyAdmin*, semua pekerjaan menjadi mudah, karena tanpa harus

mengerti perintah-perintah dasar *SQL* namun sudah dapat memanajemen database dan data yang ada didalamnya (Bunafit Nugroho ; 2009: 13).

Struktur *SQL* penggunaannya, struktur *sql* dikategorikan menjadi 3 sub perintah yaitu :

1. Data Definition Language (DDL) merupakan sub bahasa *SQL* yang digunakan untuk membangun kerangka database. Ada 3 perintah yang termasuk DDL yaitu :

- a. CREATE : perintah ini digunakan untuk membuat, termasuk diantaranya membuat database baru, table baru, view baru dan kolom.
- b. ALTER : perintah ini digunakan untuk mengubah struktur table, menambah kolom mengubah kolom, menghapus kolom, maupun memberikan atribut pada kolom.
- c. DROP : perintah ini digunakan untuk menghapus database dan table

2. Data Manipulation Language (DML)

Data Manipulation Language merupakan sub bahasa *SQL* yang digunakan untuk memanipulasi data dalam database yang telah dibuat. Perintah yang digunakan diantaranya yaitu :

- a. INSERT : perintah ini digunakan untuk menyisipkan atau memasukkan data baru ke dalam table. Penggunaannya setelah database dan table selesai dibuat.
- b. SELECT : perintah ini digunakan untuk mengambil data atau menampilkan data dari suatu table atau beberapa table dalam relasi. Data

yang diambil dapat kita tampilkan dalam layar Promp MySQL secara langsung maupun ditampilkan pada tampilan aplikasi.

c. UPDATE : perintah ini digunakan untuk memperbarui data lama menjadi data terkini. Jika anda memiliki data yang salah atau kurang *up to date* dengan kondisi sekarang, maka dapat diubah isi datanya menggunakan perintah UPDATE.

d. DELETE : perintah ini digunakan untuk menghapus data dari table. Biasanya data yang dihapus merupakan data yang sudah tidak diperlukan lagi. Pada saat menghapus data perintah yang dijalankan tidak dapat digagalkan, sehingga data yang telah hilang tidak dapat dikembalikan.

3. Data Control Language (DCL)

Data Control Language merupakan sub bahasa SQL yang digunakan untuk melakukan pengontrolan data dan server databasenya. Perintah DCL diantaranya :

a. DRAN : perintah ini digunakan untuk memberikan hak/izin akses oleh administrator (pemilik utama) server kepada user (penguna biasa). Hak akses tersebut berupa hak membuat (CREATE), mengambil (SELECT), menghapus (DELETE), mengubah (UPDATE), dan hak khusus berkenaan dengan system databasenya.

b. REVOKE : perintah ini memiliki kegunaan terbaik dengan GRAND, yaitu untuk menghilangkan atau mencabut hak akses yang telah diberikan kepada user oleh administrator.