

MODUL II

DDL (Data Definition Language)

2.1. TUJUAN

1. Memahami konsep dasar Data Definition Language (DDL)
2. Memahami implementasi beberapa perintah Create dari DDL
3. Memahami implementasi beberapa perintah Alter dari DDL

2.2. DASAR TEORI

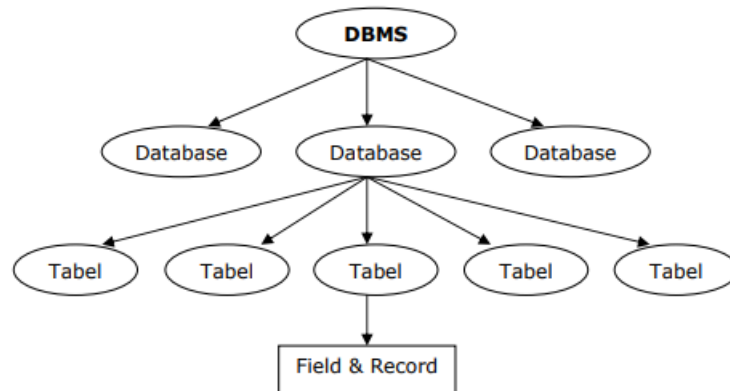
2.2.1. Pengertian DDL

Data Definition Language (DDL) adalah kumpulan perintah SQL yang berkaitan dengan pembuatan, perubahan, dan penghapusan database maupun objek-objek yang terdapat di dalam database. Salah satu bentuk bahasa basis data yaitu Data Definition Language (DDL) yang digunakan untuk membuat, mengubah, serta menghapus basis data dan objek-objek yang diperlukan dalam basis data. Metode yang digunakan untuk menerjemahkan kalimat perintah dalam bahasa Indonesia ke query DDL adalah metode Rule-Based. Proses pada sistem yang dibangun memiliki dua tahap utama yaitu Preprocessing dan Translasi. Tahap Preprocessing terdiri dari case folding, filtering, tokenizing kata, stemming, dan removing stopword. Tahap Translasi terdiri dari deteksi kata kunci, tokenizing perintah, identifikasi perintah DDL, identifikasi konten, dan penyusunan query

2.2.2. Pengertian SQL

SQL atau Structured Query Language merupakan suatu bahasa (language) yang digunakan untuk mengakses database. SQL sering disebut juga sebagai query. Bahasa ini secara de facto merupakan bahasa standar yang digunakan dalam manajemen basis data relasional. Saat ini hampir semua server basis data yang ada mendukung bahasa ini untuk melakukan manajemen datanya. Sejarah SQL dimulai dari artikel seorang peneliti dari IBM bernama EF Codd yang membahas tentang ide pembuatan basis data relasional pada bulan Juni 1 1970. Artikel ini juga membahas kemungkinan pembuatan bahasa standar untuk mengakses data dalam basis data tersebut. Bahasa tersebut kemudian diberi nama SEQUEL (Structured English Query Language). Setelah terbitnya artikel tersebut, IBM mengadakan proyek pembuatan basis data relasional berbasis bahasa SEQUEL. Akan tetapi, karena

permasalahan hukum mengenai penamaan SEQUEL, IBM pun mengubahnya menjadi SQL. Implementasi basis data relasional dikenal dengan System/R. Di akhir tahun 1970-an, muncul perusahaan bernama Oracle yang membuat server basis data populer yang bernama sama dengan nama perusahaannya. Dengan naiknya kepopuleran Oracle, maka SQL juga ikut populer sehingga saat ini menjadi standar de facto bahasa dalam manajemen basis data.



Gambar 2.1 Hierarki Database

2.2.3. Membuat Data Base dan menghapus data base

a. Membuat Data Base

Sintaks umum SQL untuk membuat suatu database adalah sebagai berikut :

```
CREATE DATABASE [IF NOT EXISTS]
nama_database;
```

Bentuk perintah di atas akan membuat sebuah database baru dengan nama nama_database. Aturan penamaan sebuah database sama seperti aturan penamaan sebuah variabel, dimana secara umum nama database boleh terdiri dari huruf, angka dan underscore (_). Jika database yang akan dibuat sudah ada, maka akan muncul pesan error. Namun jika ingin otomatis menghapus database yang lama jika sudah ada, aktifkan option IF NOT EXISTS.

Berikut ini contoh perintah untuk membuat database baru dengan nama “penjualan” :

```
CREATE DATABASE penjualan;
```

Jika query di atas berhasil dieksekusi dan database berhasil dibuat, maka akan ditampilkan pesan kurang lebih sebagai berikut :

Query OK, 1 row affected (0.02 sec)

b. Menghapus Data Base

Untuk menghapus suatu database, sintaks umumnya adalah sbb :

DROP DATABASE [IF EXISTS] nama_database;
--

Bentuk perintah di atas akan menghapus database dengan nama nama_database. Jika databasenya ada maka database dan juga seluruh tabel di dalamnya akan dihapus. Jadi berhati-hatilah dengan perintah ini! Jika nama database yang akan dihapus tidak ditemukan, maka akan ditampilkan pesan error. Aktifkan option IF EXISTS untuk memastikan bahwa suatu database benar-benar ada

2.2.4. Stored Procedure

Stored Procedure merupakan suatu kumpulan perintah atau statement yang disimpan dan dieksekusi di server database MySQL. Dengan SP (Stored Procedure), kita dapat menyusun program sederhana berbasis sintaks SQL untuk menjalankan fungsi tertentu. Hal ini menjadikan aplikasi yang kita buat lebih efektif dan efisien.

Stored Procedure dari segi bentuk dan sifatnya terbagi menjadi 2 (dua), yaitu FUNCTION dan PROCEDURE. Perbedaan utama antara function dan Procedure adalah terletak pada nilai yang dikembalikannya (di-return). Function memiliki suatu nilai yang dikembalikan (di-return), sedangkan procedure tidak. Umumnya suatu procedure hanya berisi suatu kumpulan proses yang tidak menghasilkan value, biasanya hanya menampilkan saja.

2.2.5. Membuat dan Menghapus Tabel

a. Membuat Tabel

Bentuk umum SQL untuk membuat suatu table secara sederhana sebagai berikut :

CREATE TABLE nama_tabel (field1 tipe(panjang), field2 tipe(panjang), fieldn tipe(panjang), PRIMARY KEY (field_key));

Bentuk umum di atas merupakan bentuk umum pembuatan tabel yang sudah disederhanakan. Penamaan tabel dan field memiliki aturan yang sama dengan penamaan database.

b. Menghapus table

Untu`k menghapus sebuah tabel, bentuk umum dari perintah SQL adalah sebagai berikut :

DROP TABLE nama_tabel;

2.2.6. Merubah Struktur Tabel

Untuk mengubah struktur suatu tabel, bentuk umum perintah SQL-nya sebagai berikut :

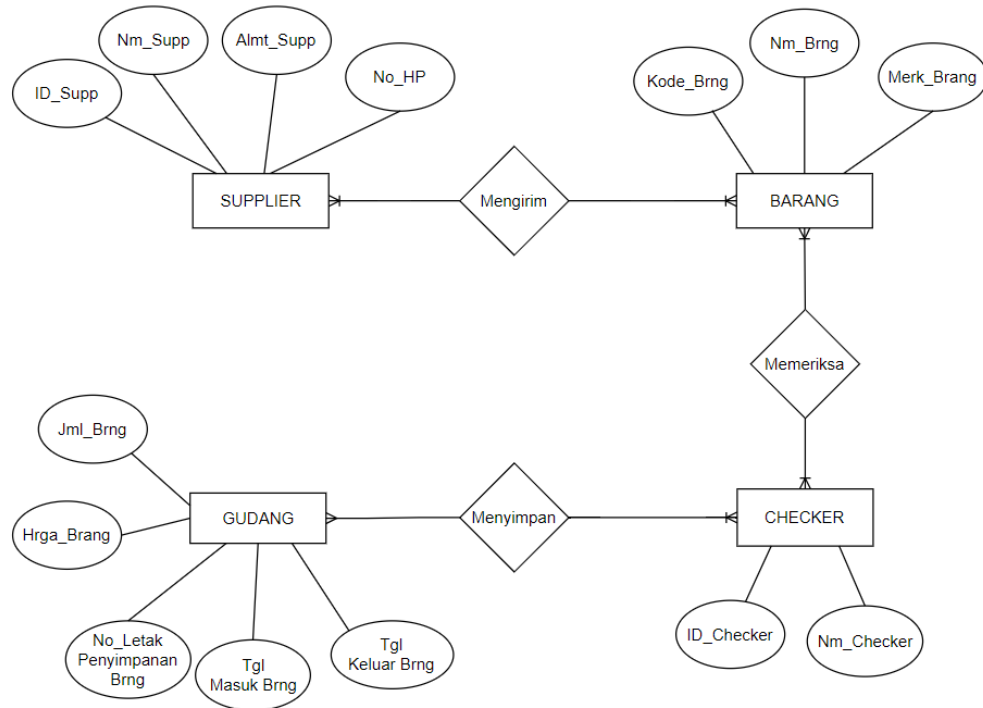
ALTER TABLE nama_tabel alter_options;

dimana :

- ALTER TABLE merupakan perintah dasar untuk mengubah tabel.
- nama_tabel merupakan nama tabel yang akan diubah strukturnya.
- alter_options merupakan pilihan perubahan tabel. Option yang bisa digunakan, beberapa di antaranya sebagai berikut :
 - ADD definisi_field_baru
Option ini digunakan untuk menambahkan field baru dengan “definisi_field_baru” (nama field, tipe dan option lain).
 - ADD INDEX nama_index
Option ini digunakan untuk menambahkan index dengan nama “nama_index” pada tabel.
 - ADD PRIMARY KEY (field_kunci)
Option untuk menambahkan primary key pada tabel
 - CHANGE field_yang_diubah definisi_field_baru Option untuk mengubah field_yang_diubah menjadi definisi_field_baru
 - MODIFY definisi_field
Option untuk mengubah suatu field menjadi definisi_field
 - DROP nama_field
Option untuk menghapus field nama_field
 - RENAME TO nama_tabel_baru
Option untuk mengganti nama tabel

2.3. DATA HASIL

2.3.1. Entity Relationship Diagram



2.3.2. Jobsheet

No	Latihan		Query dan Hasil
1	Database Baru	Membuat Database baru	MariaDB [(none)]> create database dataGudang; Query OK, 1 row affected (0.002 sec)
		Melihat database	MariaDB [(none)]> show databases; +-----+ Database +-----+ datagudang information_schema mysql performance_schema phpmyadmin test +-----+ 6 rows in set (0.002 sec)
		Menggunakan database yang telah dibuat	MariaDB [(none)]> use dataGudang; Database changed MariaDB [dataGudang]> █
2	Membuat tabel	Membuat table	MariaDB [dataGudang]> create table Supplier(idSupplier int, namaSupplier varchar(50), alamatSupplier varchar (50), noHP varchar(13), primary key(idSupplier)); Query OK, 0 rows affected (0.014 sec)
		Membuat tabel	MariaDB [dataGudang]> create table Barang(kodeBarang int, namaBarang varchar(50), merekBarang varchar (50), jumlahBarang int, hargaBarang int, noTataLetak int, tanggalMasuk int, tanggalKeluar int, primary key(kodeBarang)); Query OK, 0 rows affected (0.014 sec)
		Membuat tabel	MariaDB [dataGudang]> create table Checker(idChecker int, namaChecker varchar(50), primary key(idChecker)); Query OK, 0 rows affected (0.014 sec)
3	Membuat Index	Membuat Index	MariaDB [dataGudang]> create index `namaBarang` on `Barang`(`namaBarang`); Query OK, 0 rows affected (0.009 sec) Records: 0 Duplicates: 0 Warnings: 0 MariaDB [dataGudang]> desc Barang; +-----+-----+-----+-----+-----+-----+ Field Type Null Key Default Extra +-----+-----+-----+-----+-----+-----+ kodeBarang int(11) NO PRI NULL namaBarang varchar(50) YES MUL NULL merekBarang varchar(50) YES NULL jumlahBarang int(11) YES NULL hargaBarang int(11) YES NULL noTataLetak int(11) YES NULL tanggalMasuk int(11) YES NULL tanggalKeluar int(11) YES NULL +-----+-----+-----+-----+-----+-----+ 8 rows in set (0.006 sec)

4	Membuat stored procude	Membuat	<pre> MariaDB [dataGudang]> delimiter // MariaDB [dataGudang]> create procedure lihat() -> begin -> select*from Barang; -> end // Query OK, 0 rows affected (0.010 sec) MariaDB [dataGudang]> delimiter; MariaDB [dataGudang]> call lihat(); +-----+-----+-----+-----+-----+-----+-----+ kodeBarang namaBarang merekBarang jumlahBarang hargaBarang noTataLetak tanggalMasuk tanggalKeluar +-----+-----+-----+-----+-----+-----+-----+ 1 Stop Kontak Broco 100 40000 5 220522 220522 2 Stop Kontak mitsubishi 100 40000 5 220522 220522 +-----+-----+-----+-----+-----+-----+-----+ 2 rows in set (0.001 sec) Query OK, 0 rows affected (0.020 sec) MariaDB [dataGudang]> create procedure lihatBarang -> (IN kodeInput int) -> begin -> select*from Barang where kodeBarang=kodeInput; -> end // Query OK, 0 rows affected (0.007 sec) MariaDB [dataGudang]> call lihatBarang(2)// +-----+-----+-----+-----+-----+-----+-----+ kodeBarang namaBarang merekBarang jumlahBarang hargaBarang noTataLetak tanggalMasuk tanggalKeluar +-----+-----+-----+-----+-----+-----+-----+ 2 Stop Kontak mitsubishi 100 40000 5 220522 220522 +-----+-----+-----+-----+-----+-----+-----+ 1 row in set (0.003 sec) Query OK, 0 rows affected (0.015 sec) MariaDB [dataGudang]> delimiter // MariaDB [dataGudang]> create procedure inputBarang -> (IN inputKode int, IN inputNama varchar(50), IN inputMerek varchar(20), IN inputJumlah int, IN inputHarga int, IN inputLetak int, IN inputMasuk int, IN inputKeluar int) -> begin -> insert into Barang values(inputKode, inputNama, inputMerek, inputJumlah, inputHarga, inputLetak, inputMasuk, inputKeluar); -> end // Query OK, 0 rows affected (0.008 sec) MariaDB [dataGudang]> delimiter ; MariaDB [dataGudang]> call inputBarang(3, "Stop Kontak", "Panasonic", 500, 45000, 3, 250622, " "); Query OK, 1 row affected, 1 warning (0.005 sec) MariaDB [dataGudang]> call lihat(); +-----+-----+-----+-----+-----+-----+-----+ kodeBarang namaBarang merekBarang jumlahBarang hargaBarang noTataLetak tanggalMasuk tanggalKeluar +-----+-----+-----+-----+-----+-----+-----+ 1 Stop Kontak Broco 100 40000 5 220522 220522 2 Stop Kontak mitsubishi 100 40000 5 220522 220522 3 Stop Kontak Panasonic 500 45000 3 250622 250622 +-----+-----+-----+-----+-----+-----+-----+ 3 rows in set (0.001 sec) Query OK, 0 rows affected (0.015 sec) </pre>
5	Melihat tabel	Melihat table yang sudah di buat	<pre> MariaDB [dataGudang]> show tables; +-----+ Tables_in_datagudang +-----+ barang checker supplier +-----+ 3 rows in set (0.000 sec) </pre>
6	Melihat deskripsi tabel	Melihat deskripsi dari tabel	<pre> MariaDB [dataGudang]> select*from Supplier; +-----+-----+-----+-----+ idSupplier namaSupplier alamatSupplier noHP +-----+-----+-----+-----+ 1111 Ahmad Satriadi Kopang 087728728561 1112 Wilad Surya Mandala Narmada 087745655657 +-----+-----+-----+-----+ 2 rows in set (0.001 sec) </pre>
	Melihat deskripsi dari tabel	Melihat deskripsi dari tabel	<pre> MariaDB [dataGudang]> select*from Barang; +-----+-----+-----+-----+-----+-----+-----+ kodeBarang namaBarang merekBarang jumlahBarang hargaBarang noTataLetak tanggalMasuk tanggalKeluar +-----+-----+-----+-----+-----+-----+-----+ 1 Stop Kontak Broco 100 40000 5 220522 220522 2 Stop Kontak mitsubishi 100 40000 5 220522 220522 +-----+-----+-----+-----+-----+-----+-----+ 2 rows in set (0.000 sec) </pre>

		Melihat deskripsi dari tabel	<pre> MariaDB [dataGudang]> select*from Checker; +-----+-----+ idChecker namaChecker +-----+-----+ 1000 Otebbb 1001 Sarsana +-----+-----+ 2 rows in set (0.001 sec) </pre>
7	Merubah tabel	Merubah nama tabel	<pre> MariaDB [dataGudang]> alter table Supplier -> rename to Penyuplai; Query OK, 0 rows affected (0.011 sec) </pre>
		Membuat kolom baru	<pre> MariaDB [dataGudang]> alter table Penyuplai -> add namaSupplier varchar(100); Query OK, 0 rows affected (0.007 sec) Records: 0 Duplicates: 0 Warnings: 0 </pre>
		Menghapus kolom pada tabel	<pre> MariaDB [dataGudang]> alter table Penyuplai -> drop namaSupplier; Query OK, 0 rows affected (0.009 sec) Records: 0 Duplicates: 0 Warnings: 0 </pre>

2.4 ANALISA DATA HASIL

No.	Latihan	Query dan Hasil	ANALISA

2.5 KESIMPULAN

DAFTAR PUSTAKA

- Anwar, Purnamasari. *PENERJEMAHAN TEKS BAHASA
INDONESIAMENJADI DATA DEFINITION LANGUAGE (DDL)
DENGAN PENANGANAN KALIMAT MAJEMUK*. Teknik
Informatika Universitas Komputer Indonesia.
- Achmad Solichin. 2010 . *MySQL 5 Dari Pemula hingga Mahir*. Universitas
Budi Luhur , Jakarta
- Muhammad Denny Prayoga. *Pengertian dan Komponen SQL*. Fakultas Komputer