**MODUL I**

**DDL (*Data Definiton Language)***

**1.1** **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

**1.2** **DASAR TEORI**

**1.2.1 Pengertian *Structured Query Language* (SQL)**

*Structured Query Language* (SQL) merupakan komponen bahasa *relational database system*. SQL merupakan bahasa baku (ANSI/SQL), non prosedural, dan berorientasi himpunan (*set-oriented language*). SQL dapat digunakan baik secara interaktif atau ditempelkan (*embedded*) pada sebuah program aplikasi. SQL dapat digunakan untuk mendefinisikan struktur data, memodifikasi data pada basis data, menspesifikasi batasan kemanan (*security*), hingga pemeliharaan kinerja basis data. (Agung Baitul Hikmah, 2016)

**1.2.2**  **Pengertian *Data Definition Language* (DDL)**

*Data Definition Language* (DDL) merupakan perintah SQL yang digunakan untuk melakukan definisi awal suatu basis data dan tabel pada konsep RDBMS. Secara sederhana, penulisan perintah SQL pada kelompok ini terdiri dari *Create*, *Alter*, dan *Drop*. (Miftah Farid Adiwisastra,2016)

**1.2.3**  **Keyword pada DDL**

DDL adalah sebuah metode *Query* SQL yang berguna untuk mendefinisikan data pada sebuah *Database*, *Query* yang dimiliki DDL adalah :

1. *Create* : Digunakan untuk membuat *Database* dan Tabel
2. *Drop* : Digunakan untuk menghapus Tabel dan *Database*
3. *Alter* : Digunakan untuk melakukan perubahan struktur tabel yang telah dibuat, baik menambah *Field* (*Add*), mengganti nama *Field* (*Change*) ataupun menamakannya kembali (*Rename*), dan menghapus *Field* ( *Drop* ). (Haris Saputra, 2012)

**1.2.4 *Create Database***

Perintah SQL ini digunakan untuk membuat suatu basis data dan tabel pendukung didalam pembangunan basis data tersebut. Tabel merupakan objek yang sangat penting dalam suatu basis data. (Agung Baitul Hikmah, 2016)

1. Sintak untuk membuat database baru :

|  |
| --- |
| *CREATE DATABASE* databasename; |

1. Sintak untuk membuat tabel baru :

|  |
| --- |
| CREATE TABLE table\_name (  column1 datatype,  column2 datatype,  column3 datatype,  .... ); |

**1.2.5 *Create Table***

Tabel adalah entitas utama dalam model data relasional yang digunakan untuk menyimpan data. Tabel terdiri dari baris dan kolom, dengan setiap baris mewakili satu entitas atau objek, dan setiap kolom mewakili atribut atau ciri dari entitas tersebut.( Sri Nadriati, S.Kom., M.Kom, 2023)

**1.2.6 *Create Index***

Perintah *create index* digunakan untuk membuat indeks pada satu atau lebih kolom dalam tabel, yang dapat meningkatkan kinerja pencarian data. ( Sri Nadriati, S.Kom., M.Kom, 2023)

**1.2.7** ***Stored Procedure***

*Stored Procedure* merupakan serangkaian sub routine (baris program) yang dibuat dan disimpan dalam basis data. *Stored Procedure* memberikan banyak keuntungan bagi *developer* terutama dari sisi keamanan basis data dan pengembangan *software* yang *multiplatform* serta membutuhkan banyak *teamwork*.( Sri Nadriati, S.Kom., M.Kom, 2023)

Perintah *stored procedure* pada umumnya digunakan untuk proses *insert*, *update* dan *delete* pada sebuah DBMS. *Stored Procedure* memiliki variabel parameter yang dibedakan menjadi:

1. *IN* : Variabel parameter hanya dapat digunakan untuk menerima input saja. IN menjadi nilai *default* dari variabel parameter
2. *OUT* : variabel parameter hanya dapat digunakan untuk menyimpan hasil *output* saja.
3. *INOUT* : Variabel parameter yang digunakan untuk menerima input dan menyimpan hasil *output*.

**1.2.8 *Alter Table***

Perintah SQL ini digunakan untuk mengubah struktur tabel yang terdapat didalam basis data. Hal ini dapat saja terjadi, jika ingin melakukan penambahan atau penghapusan suatu *field* atau atribut *key* (*Primary/Foreign*), tanpa mendefinisikan struktur baru pada tabel. Melalui perintah SQL ini jika sudah terisikan *record* pada tabel tersebut, maka mengenai hilangnya *record* yang terdapat pada suatu tabel tidak akan terjadi. Berikut ini merupakan aturan umum penulisan yang digunakan pada *alter table*. (Agung Baitul Hikmah, 2016)

1. Sintak menambah kolom baru:

|  |
| --- |
| *ALTER TABLE* table\_name ADD column\_name datatype; |

1. Sintak menghapus kolom:

|  |
| --- |
| *ALTER TABLE* table\_name DROP COLUMN column\_name; |

1. Sintak merubah kolom:

|  |
| --- |
| *ALTER TABLE* table\_name ALTER COLUMN column\_name datatype; |

1. Contoh penerapan dan penjelasan:

Menambahkan kolom alamat pada tabel mahasiswa dengan tipe data *varchar* panjang *field* 25

|  |
| --- |
| *ALTER TABLE* mahasiswa ADD alamat varchar(25); |

1. Menghapus kolom alamat pada tabel mahasiswa:

|  |
| --- |
| *ALTER TABLE* mahasiswa DROP COLUMN alamat; |

**1.2.9 *Drop***

Perintah SQL ini digunakan untuk menghapus tabel yang terdapat didalam basis data. Berikut ini merupakan aturan umum penulisan yang digunakan pada *drop table*. (Agung Baitul Hikmah, 2016)

1. Sintak menghapus *database*:

|  |
| --- |
| *DROP DATABASE* databasename; |

1. Sintak menghapus tabel:

|  |
| --- |
| *DROP TABLE* table\_name; |

1. Contoh menghapus *database* testDB:

|  |
| --- |
| *DROP DATABASE* testDB; |

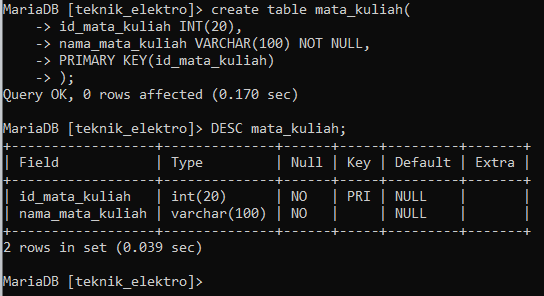
1. Contoh menghapus tabel mahasiswa:

|  |
| --- |
| *DROP TABLE mahasiswa;* |

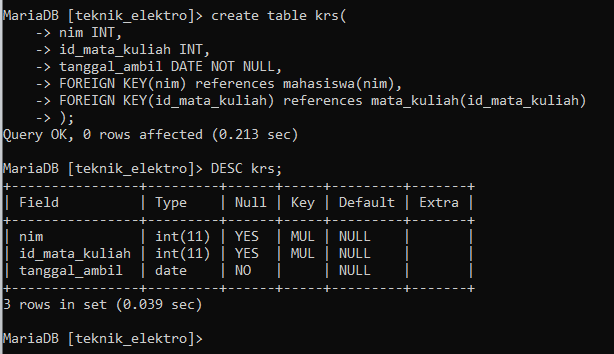
* 1. **Tugas Mandiri**
     1. Buatlah tabel mata\_kuliah dengan struktur berikut:
        + id\_mata\_kuliah (*INT*, *PRIMARY KEY*)
        + nama\_mata\_kuliah (*VARCHAR*(100), *NOT NULL*)
     2. Buat relasi antara tabel mahasiswa dan mata\_kuliah dengan membuat tabel krs yang berisi:
        + id\_mahasiswa (*INT*, *FOREIGN KEY* dari tabel mahasiswa)
        + id\_mata\_kuliah (*INT*, *FOREIGN KEY* dari tabel mata\_kuliah)
        + tanggal\_ambil (*DATE*, *NOT NULL*)

**Penyelesaian:**

1. Membuat tabel mata\_kuliah dimana terdiri dari kolom id\_mata\_kuliah sebagai *Primary Key* dan nama\_mata\_kuliah dan berikut tampilan dari tabel tersebut:



1. Membuat tabel krs dengan menyambungkan nim dari tabel mahasiswa dan id\_mata\_kuliah dari tabel mata\_kuliah menggunakan *Foreign Key* dan berikut tampilan tabel berikut:



* 1. **Jobsheet**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Latihan** | | | **Query dan Hasil** |
| **1** | ***Database* Baru** | **1** | **Membuat *database* baru** | Create database pemesanan lapangan futsal |
| **2** | **Melihat *database*** | Show databases; |
| **3** | **Menggunakan *database* yang sudah dibuat** | Use Pemesanan\_lapangan\_futsal; |
| **2** | **Membuat tabel** | **1** | **Membuat tabel Pemesanan Lapangan Futsal** | create table pemesanan\_lapangan\_futsaL(  -> nama\_pelanggan char(100),  -> alamat\_pelanggan char(100),  -> nomor\_pelanggan char(100)); |
| **2** | **Membuat tabel pemesanan Lapangan Futsal** | create table pemesanan\_lapangan\_futsaL(  -> nama\_pelanggan char(100),  -> alamat\_pelanggan char(100),  -> nomor\_pelanggan char(100)); |
| **3** | **Membuat tabel pelanggan** | create table identitas\_pemesan(  -> nama\_pelanggan char(100),  -> nomor\_pelanggan int(15),  -> alamat\_pelanggan char(20)  -> ); |
| **4** | **Membuat tabel**  **lapangan** | create table lapangan  -> (  -> nomor\_lapangan int(5),  -> alamat\_lapangan char(20),  -> jenis\_lapangan char(20)  -> ); |
| **3** | **Membuat *index*** | **1** | **Membuat *index* baru** | create index futsal on pemesanan\_lapangan\_futsal(  -> nama\_pelanggan,  -> alamat\_pelanggan,  -> nomor\_pelanggan  -> ); |
| **4** | **Membuat *stored procedure*** | **1** | **Membuat *stored procedure*** | delimiter $$  create procedure pemesanan\_lapangan\_futsal()  -> begin  -> select \*from pemesanan\_lapangan\_futsal;  -> end $$ |
| **5** | **Melihat tabel** | **1** | **Melihat tabel yang sudah dibuat** | SHOW tables;  **C:\Users\Ardy Pratama\Documents\IMG-20240911-WA0015.jpg** |
| **6** | **Melihat deskripsi tabel** | **1** | **Melihat deskripsi tabel Pelanggan** | DESC pemesanan\_lapangan\_futsal; |
|  |  | **2** | **Melihat deskripsi tabel** | DESC pemesanan\_lapangan\_futsal; |
| **3** | **Melihat deskripsi tabel pemesan** | DESC IDENTIAS\_PEMESAN; |
| **4** | **Melihat deskripsi tabel lapangan** | DESC LAPANGAN; |
|
| **7** | **Merubah tabel** | **1** | **Merubah nama tabel** | ALTER TABLE PEMESANAN\_LAPANGAN\_FUTSAL TO ARENA\_FUTSAL; |
| **2** | **Membuat kolom baru** | ALTER TABLE IDENTITAS\_PEMESAN ADD JENIS\_KELAMIN; |
| **3** | **Menghapus kolom pada tabel** | ALTER TABLE IDENTITAS\_PEMESAN DROP JENIS\_KELAMIN; |
| **8** | **Menghapus tabel dan *database*** | **1** | **Menghapus tabel** | DROP table identitas\_pemesan; |
| **2** | **Menghapus *database*** | DROP database pemesanan\_lapangan\_futsal; |

**1.5 Analisa**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Latihan** | **Query dan Hasil** | **ANALISA** |
| **1.** | **Membuat Database Baru** | CREATE DATABASE PEMESANAN\_LAPANGAN\_FUTSAL; | Berdasarkan gambar di samping, untuk membuat sebuah *database* pemesanan lapangan futsal maka digunakan perintahcreate database pemesanan\_lapangan \_futsal; |
| **Melihat Database** | Show databases; | Berdasarkan gambar di samping, perintah show databases; digunakan untuk menampilkan hasil yang terdiri dari beberapa *database* yang sudah pernah dibuat. |
| **Menggunakan Database yang sudah di buat** | use pemesanan\_lapangan\_futsal; | Berdasarkan gambar di samping, perintah use pemesa nan\_lapangan\_futsal; digunakan untuk memilih basis data yang telah kita buat, berupa “*Database Changed*” yang menandakan bahwa *database* berhasil digunakan. |
| **2.** | **Membuat Tabel** | create table pemesanan\_lapangan\_futsal(  nama\_pelanggan char(100),  alamat\_pelanggan char(100),  nomor\_pelanggan char(100)  ); | Berdasarkan gambar di samping, untuk membuat tabel dalam *database* dapat menggunakan perintah create table, dari gambar disamping tabel diberikan nama pemesanan\_lapangan\_futsal. Di dalam tabel terdapat beberapa kolom diantaranya nama\_pelanggan, alamat \_pelanggan, dan nomor\_pelanggan. |
| **Membuat Tabel pemesanan lapangan futsal** | create table pemesanan\_lapangan\_futsal(  nama\_pelanggan char(100),  alamat\_pelanggan char(100),  nomor\_pelanggan char(100)  ); | Berdasarkan gambar di samping, perintah create table diberikan nama pemesanan\_lapangan\_futsal. Di dalam tabel terdapat kolom. nama\_pelanggan, alamat\_pelanggan, dan nomor\_pelanggan. Dimana ketiga kolom tersebut memiliki tipe data char. |
| **Membuat Tabel pelanggan** | create table identitas\_pemesan  -> (  -> nama\_pelanggan char(20),  -> alamat\_pelanggan char(20),  -> nomor\_pelanggan int(15)  -> ); | Berdasarkan gambar di samping, perintah create table diberikan nama identitas\_pemesan. Dalam tabel tersebut terdapat kolom nama\_pelanggan, alamat\_pelanggan, dan nomor\_pelanggan.yang memiliki tipe data char dan int. |
|  | **Membuat Tabel lapangan** | create table lapangan  -> (  -> nomor\_lapangan int(5),  -> alamat\_lapangan char(20),  -> jenis\_lapangan char(20,  -> PRIMARY KEY(nomor\_lapangan)  -> ); | Berdasarkan gambar di samping, perintah create table diberikan nama lapangan. Dalam tabel ini terdapat kolom nomor\_lapangan, alamat\_lapangan, dan jenis\_lapangan. yang memiliki tipe data *char* dan *int.* |
| **3.** | **Membuat Index Baru** | create table pemesanan\_lapangan\_futsaL(  -> nama\_pelanggan char(100),  -> alamat\_pelanggan char(100),  -> nomor\_pelanggan char(100)  -> ); | Berdasarkan gambar di samping, perintah create table diberikan nama pemesanan\_lapangan\_futsal. Dalam tabel ini terdapat kolom nama\_pelanggan, alamat\_pelanggan, dan nomor\_pelanggan.  Index digunakan untuk mempercepat proses pencarian data dalam *database*. |
| **4.** | **Membuat Store Procedure** | delimiter $$  create procedure pemesanan\_lapangan\_futsal()  -> begin  -> select \*from pemesanan\_lapangan\_futsal;  -> end $$ | Berdasarkan gambar di samping untuk membuat sebuah *store procedure* baru, dapat menggunakan *query* delimiter $$. *Store procedure* berfungsi untuk mengumpulkan pernyataan-pernyataan yang ada di dalam *database* sehingga dapat dieksekusi. |
| **5.** | **Melihat Tabel Yang Sudah Dibuat** | Show tables;  **C:\Users\Ardy Pratama\Documents\IMG-20240911-WA0015.jpg** | Berdasarkan gambar di samping untuk menampilkan tabel-tabel yang ada dalam suatu *database* dapat menggunakan *query* show tables; |
| **6.** | **Melihat Deskripsi Tabel** | desc pemesanam\_lapangan\_futsal; | Berdasarkan gambar di samping untuk menampilkan deskripsi tabel dapat menggunakan perintah desc pemesanam\_lapangan\_futsal; Hasil yang akan ditampilkan berupa nama dan tipe data kolom tersebut serta spesifikasi lainnya. |
| **Melihat Deskripsi Tabel pemesan** | desc identitas\_pemesan; | Berdasarkan gambar di samping untuk menampilkan deskripsi tabel identitas pemesan, perintah yang digunakan adalah desc identitas\_pemesan; Hasil yang akan ditampilkan berupa nama dan tipe data kolom tersebut serta spesifikasi lainnya. |
| **Melihat Deskripsi Tabel Lapangan** | desc lapangan; | Berdasarkan gambar di samping untuk menampilkan deskripsi tabel lapangan, perintah yang digunakan adalah desc lapangan; hasil yang akan ditampilkan yaitu nama kolom yang ada dalam tabel serta tipe datanya dan spesifikasi lainnya. |
| **7.** | **Merubah Nama Tabel** | alter table pemesanan\_lapangan\_futsal rename to arena futsal; | Berdasarkan gambar di samping jika ingin merubah nama dari sebuah tabel maka dapat menggunakan perintah alter table pemesanan \_lapangan\_futsal rename to arena\_futsal; |
| **Membuat Kolom Baru** | alter table identitas\_pemesan add jenis\_kelamin char(20); | Berdasarkan gambar di samping untuk menambahkan kolom baru pada tabel yang telah ada dapat menggunakan perintah alter table identitas \_pemesan add jenis\_kelamin char(20); |
| **Menghapus Kolom Pada Tabel** | alter table identitas\_pemesan drop jenis\_kelamin; | Berdasarkan gambar di samping jika ingin menghapus kolom pada sebuah tabel dapat menggunakan perintah alter table identitas \_pemesan drop jenis\_kelamin; |
| **8.** | **Menghapus Tabel** | drop table identitas \_pemesan; | Berdasarkan gambar di samping untuk menghapus tabel, dapat menggunakan perintah drop table identitas \_pemesan; |
| **Menghapus Database** | drop database pemesanan \_lapangan\_futsal; | Berdasarkan gambar di samping untuk menghapus data termasuk semua data yang ada di dalamnya, dapat menggunakan perintah drop database pemesanan \_lapangan\_futsal; |

**1.8 KESIMPULAN**

1. Konsep DDL yaitu sekumpulan perintah yang ada pada SQL dimana menggambarkan desain dari *database* secara menyeluruh, selain itu data *definition* *language* ini juga digunakan untuk membuat, merubah maupun menghapus struktur atau definisi tipe data dari obyek yang ada pada *database*.
2. Perintah CREATE dalam DDL memungkinkan pembuatan berbagai elemen penting dalam basis data, seperti tabel dan indeks. Selain itu, perintah ini juga membantu memahami langkah-langkah serta aturan dasar yang dibutuhkan untuk membangun struktur basis data yang sesuai dengan kebutuhan.
3. Perintah ALTER dalam DDL memungkinkan perubahan pada struktur tabel *database*, termasuk menambah, menghapus, atau mengubah kolom serta mengatur kunci dan *index*. Ini membantu menyesuaikan kebutuhan data.

**DAFTAR PUSTAKA**

Anonim. 2024. Laporan Praktikum Basis Data. Laboratorium Jaringan dan Komputer. Jurusan Teknik Elektro. Fakultas Teknik. Universitas Mataram.

Adityo Permana W. 2022. "Modul Perancangan Basis Data". Universitas Teknologi Yogyakarta.

Agung Baitul Hikmah. 2016. "Modul Pembelajaran Aplikasi Basis Data Structured Query Language (SQL) ". AMIK BSI Tasikmalaya.

Haris Saputro. 2012."Modul Pembelajaran Praktek Basis Data (MySQL)".

Santoso, Joseph Teguh. 2021. "SQL Structured Query Language". Universitas STEKOM. Semarang.

Sri Nadriati. 2023."Sistem Basis Data". PT Penamuda Media. Yogyakarta.