1. MODUL II

**MANIPULASI STRUKTUR TABEL**

* 1. **TUJUAN**
     1. Mahasiswa dapat membaca diagram ER.
     2. Mahasiswa dapat mengimplementasikan rancangan dalam struktur tabel.
     3. Mahasiswa dapat membuat tabel dengan SQL.
  2. **LANDASAN TEORI**
     1. **Implementasi ERD kedalam *database* SQL**

Secara umum, sebuah diagram ER akan direpresentasikan menjadi sebuah basis data secara fisik. Sedang komponen-komponen diagram ER yang berupa himpunan entitas dan himpunan relasi akan ditransformasikan menjadi tabel-tabel (*file-file* data) yang merupakan komponen utama pembentuk basis data [4]. Selanjutnya, atribut-atribut yang melekat pada masing-masing himpunan entitas dan himpunan relasi akan dinyatakan sebagai *field-field* dari tabel.

Berikut adalah aturan umum dalam pemetaan model data yang digambarkan dengan diagram ER menjadi basis data fisik atau menjadi dalam bentuk *database* SQL.

* 1. Setiap entitas akan diimplementasikan sebagai sebuah tabel [1].

|  |
| --- |
|  |

**Gambar 2.1** Entitas dokter

**Tabel 2.1** Tabel dokter

|  |  |  |
| --- | --- | --- |
| Id\_dokter | Nm\_dokter | Alamat |
|  |  |  |

* 1. Relasi dengan derajat relasi 1-1 (satu ke satu) yang menghubungkan 2 buah himpunan entitas akan direpresentasikan dalam bentuk penambahan atau penyetaraan atribut-atribut relasi ke tabel yang mewakili salah satu dari kedua himpunan entitas [1].

|  |
| --- |
|  |

**Gambar 2.2** Relasi 1-1

**Tabel 2.2** Tabel dokter

|  |  |  |
| --- | --- | --- |
| Id\_Dokter | Nm\_Dokter | Alamat |
|  |  |  |

**Tabel 2.3** Tabel ruang rumah sakit

|  |  |  |
| --- | --- | --- |
| Id\_Ruang | Nm\_Ruang | Id\_Dokter |
|  |  |  |
|  | | |
|  | Field yang ditambahkan dari entitas dokter. | |

* 1. Relasi dengan derajat relasi 1-N (satu ke banyak) yang menghubungkan 2 buah himpunan entitas, juga akan direpresentasikan dalam bentuk pemberian atau pencantuman atribut *key* dari himpunan entitas pertama (yang berderajat 1) ke tabel yang mewakili himpunan entitas kedua (yang berderajat N). Atribut *key* dari himpunan entitas pertama ini menjadi atribut tambahan bagi himpunan entitas kedua [1].

|  |
| --- |
|  |

**Gambar 2.3** Relasi 1-N

**Tabel 2.4** Tabel kamar

|  |  |  |
| --- | --- | --- |
| Id\_Kamar | Nm\_Kamar | Kapasitas |
|  |  |  |

**Tabel 2.5** Tabel pasien

|  |  |  |
| --- | --- | --- |
| Id\_Pasien | Nm\_Pasien | Id\_Kamar |
|  |  |  |
|  | | |
|  | Field yang ditambahkan dari entitas kamar. | |

* 1. Relasi dengan derajat relasi M-N (banyak ke banyak) yang menghubungkan 2 buah himpunan entitas, akan diwujudkan dalam bentuk tabel (*file* data) khusus yang memiliki *field* (tepatnya *foreign key*) yang berasal dari *key*-*key* dari himpunan entitas yang dihubungkannya [2].

|  |
| --- |
|  |

**Gambar 2.4** Relasi N-M

**Tabel 2.6** Tabel dokter

|  |  |  |
| --- | --- | --- |
| Id\_Dokter | Nm\_Dokter | Alamat |
|  |  |  |

**Tabel 2.7** Tabel pasien

|  |  |  |
| --- | --- | --- |
| Id\_Pasien | Nm\_Pasien | JK |
|  |  |  |

**Tabel 2.8** Tabel memeriksa

|  |  |  |
| --- | --- | --- |
| Id\_Dokter | Id\_Pasien | Tgl\_Periksa |
|  |  |  |
|  | | |
| Tabel khusus yang mewakili himpunan relasi. | | |

* 1. Implementasi himpunan entitas lemah dan sub entitas penggunaan himpunan entitas lemah dan sub entitas dalam diagram ER diimplementasi dalam bentuk tabel sebagaimana himpunan entitas kuat. Bedanya, jika himpunan entitas kuat sudah dapat langsung menjadi sebuah tabel utuh, walaupun hanya melihat relasinya dengan himpunan entitas yang lain, maka himpunan entitas lemah dan sub entitasnya hanya dapat ditransformasikan menjadi sebuah tabel dengan menyerahkan pula atribut *key* yang ada dihimpunan entitas kuat yang berelasi dengannya [3].

|  |
| --- |
|  |

**Gambar 2.5** Entitas lemah

**Tabel 2.9** Tabel dokter

|  |  |  |
| --- | --- | --- |
| id\_dokter | nm\_dokter | Alamat |
|  |  |  |

**Tabel 2.10** Tabel *skill*

|  |  |  |
| --- | --- | --- |
| id\_skill | Jenis | id\_dokter |
|  |  |  |

* + 1. **Membuat dan Menghapus Basisdata**

Di dalam operasi atau proses *database*, dikenal istilah SQL. SQL (*Structured Query Language*) merupakan perintah yang berbentuk satu perintah untuk satu proses atau operasi *database*. Perintah SQL terbagi menjadi beberapa bentuk, salah satunya yaitu DDL(*Data Definition Language*). DDL merupakan perintah SQL yang berhubungan dengan pendefinisian suatu struktur basis data [3].

Beberapa perintah dasar yang termasuk DDL antara lain CREATE untuk membuat objek baru dalam *database*, ALTER untuk mengubah struktur dari suatu objek di dalam *database*, RENAME untuk mengganti nama objek di dalam *database*, DROP untuk menghapus objek dan TRUNCATE untuk mengosongkan suatu objek.

1. **Membuat *Database***

Untuk membuat *database* digunakan perintah CREATE dengan format penulisan *query*-nya sebagai berikut:

|  |
| --- |
| CREATE DATABASE nama\_database; |

“nama\_database” adalah nama yang digunakan untuk database yang akan dibuat. Aturan penamaan sebuah *database* sama seperti aturan penamaan suatu variabel, dimana secara umum nama database boleh dari huruf, angka dan *underscore*(\_). Namun, dalam penamaan *database* hanya dapat diawali dengan huruf maupun *underscore*(\_) saja.

Contoh:

|  |
| --- |
| CREATE DATABASE praktikum\_sbd; |

1. **Menghapus *Database***

Jika *database* sudah tidak digunakan lagi, maka dapat dihapus dengan menggunakan perintah DROP. Proses penghapusan ini akan menghapus *database*, termasuk seluruh tabel dan isi dari tabel tersebut. *Database* yang sudah dihapus tidak dapat ditampilkan lagi. Format *query* untuk menghapus *database* adalah sebagai berikut:

|  |
| --- |
| DROP DATABASE nama\_database; |

“nama\_database” adalah nama *database* yang akan dicari untuk dihapus.

Contoh:

|  |
| --- |
| DROP DATABASE praktikum\_sbd; |

* + 1. **Membuat dan Menghapus Tabel**

Sebuah basis data terdiri dari sekumpulan tabel yang saling terhubung dan terintegrasi satu sama lain. Berikut adalah cara pembuatan dan penghapusan tabel dalam basis data.

* 1. **Membuat Tabel**

Untuk membuat tabel pada basis data digunakan perintah CREATE. Format *query* untuk membuat tabel yaitu:

|  |
| --- |
| CREATE TABLE nama\_tabel  ( nama\_kolom1 tipe\_data(size) nama\_konstrain,  nama\_kolom1 tipe\_data(size) nama\_konstrain,  ....  ); |

1. ”nama\_tabel” adalah nama dari tabel yang akan dibuat.
2. ”nama\_kolom1” adalah nama kolom yang akan dibuat.
3. ”tipe\_data” adalah *query* opsional dimana *query* ini dapat mendefinisikan tipe data untuk setiap atribut yang dibuat, misalnya VARCHAR, INT, TEXT, dan lain sebagainya.
4. Konstrain adalah batasan yang diberikan untuk tiap kolom, misalnya PRIMARY KEY, NOT NULL, dan sebagainya.

Contoh pembuatan tabel dosen :

|  |
| --- |
| CREATE TABLE dosen  ( nip INT PRIMARY KEY,  nama VARCHAR(30),  alamat VARCHAR(50),  ); |

* 1. **Menghapus tabel**

Untuk menghapus sebuah tabel pada *database,* digunakan perintah DROP. Format *query* untuk menghapus tabel:

|  |
| --- |
| DROP TABLE nama\_tabel; |

Contoh:

|  |
| --- |
| DROP TABLE dosen; |

* + 1. **Merubah Struktur Tabel**

Berikut adalah bentuk manipulasi atau bentuk perubahan struktur tabel-tabel dalam basis data.

1. **ALTER TABLE**

ALTER TABLE adalah perintah SQL yang digunakan untuk merubah struktur kolom pada tabel.

1. **Mengubah Tipe Data Kolom MySQL**

*Query*:

|  |
| --- |
| ALTER TABLE nama\_tabel MODIFY nama\_kolom tipe\_data\_baru; |

1. **Mengubah Nama Kolom Tabel**

*Query*:

|  |
| --- |
| ALTER TABLE nama\_tabel CHANGE nama\_kolom nama\_kolom\_baru  tipe\_data; |

1. **Mengubah Nama Tabel**

*Query*:

|  |
| --- |
| ALTER TABLE nama\_tabel\_lama RENAME TO nama\_tabel\_ baru; |

1. **Menambahkan Kolom Baru Pada Tabel**

*Query*:

|  |
| --- |
| ALTER TABLE nama\_tabel ADD nama\_kolom\_baru tipe\_data; |

1. **Menghapus Kolom Tabel**

*Query*:

|  |
| --- |
| ALTER TABLE nama\_tabel DROP nama\_kolom; |

1. **Mengubah Tipe Data Tabel (*Table Engine*)**

*Query*:

|  |
| --- |
| ALTER TABLE nama\_tabel ENGINE=tipe\_tabel; |

1. **Menambah *Primary Key***

*Query*:

|  |
| --- |
| ALTER TABLE nama\_tabel ADD PRIMARY KEY (nama\_kolom); |

1. **Menghapus *Primary Key***

*Query* :

|  |
| --- |
| ALTER TABLE nama\_tabel DROP PRIMARY KEY; |

1. **Menambah *Foreign Key***

*Query* :

|  |
| --- |
| ALTER TABLE nama\_tabel ADD CONSTRAINT nama\_constraint FOREIGN KEY (nama\_kolom) REFERENCES nama\_tabel(nama\_kolom); |

1. **Menghapus *Foreign Key***

*Query* :

|  |
| --- |
| ALTER TABLE nama\_tabel DROP FOREIGN KEY; |

1. **Melihat informasi tabel**

Digunakan untuk melihat tabel apa saja yang terdapat dalam *database* tersebut.

|  |
| --- |
| SHOW TABLE FROM nama\_db; |

Untuk melihat deskripsi tabel tertentu:

|  |
| --- |
| DESCRIBEnama\_tabel; |

1. **Mendapatkan informasi tabel**

Digunakan untuk melihat seluruh isi tabel.

|  |
| --- |
| SELECT \*FROM nama\_tabel; |

Untuk menampilkan isi kolom – kolom tertentu:

|  |
| --- |
| SELECT kolom1, kolom2,…FROM nama\_tabel; |

Untuk menampilkan isi dari kolom dengan kondisi tertentu:

|  |
| --- |
| SELECT kolom1FROM nama\_tabel WHERE kolom1=isi\_kolom; |

1. **Modifikasi informasi dalam tabel**

Digunakan untuk menambah baris baru berupa informasi ke dalam tabel.

|  |
| --- |
| INSERT INTO nama\_tabel (nama\_kolom) VALUES(isi\_kolom); |

Untuk menambah atau merubah informasi suatu kolom dengan perubahan tertentu:

|  |
| --- |
| INSERT INTO nama\_tabel SET nama\_kolom=isi\_kolom; |

* 1. **KASUS KELOMPOK**

Sekolah Dasar Internasional Wakanda (SBIW) akan merancang sebuah sistem informasi. Basis data yang dibentuk berdasarkan data-data berbagai entitas. Guru akan didata nama, kontak dan alamat sehingga terbentuk NUPTK. Lalu guru dibedakan menjadi guru honorer dan PNS yang memiliki NIP. Semua guru dapat mengajar mata pelajaran, dan guru pun menguasai tidak hanya satu saja mata pelajaran, namun terkadang mereka harus mengajar mata pelajaran yang tidak mereka pelajari, sehingga mata pelajaran dapat dikuasai dan diajar oleh banyak guru. Mata pelajaran hanya didata namanya saja, sehingga dibentuk kode mata pelajaran. Setiap mata pelajaran yang diajar guru akan didata pegisian kelas mana yang diajar. Sehingga guru dapat mengajar mata pelajaran yang sama di kelas yang berbeda, begitupun dikelas yang sama dapat mengajar mata pelajaran lain. Kelaspun harus di data nama kelasnya sehinggaa terbentuk kode kelas. Sebuah kelas dapat diwalikan oleh seorang guru non-honorer. Sebuah kelaspun memiliki banyak siswa, dimana dari mereka terdapat satu yang menjadi ketua kelas. Setiap siswa akan didata nama, alamat, nama wali, kontak wali, dan rangking mereka hingga dibentuk NIS. Setiap siswa dapat mempelajari banyak mata pelajaran, begitu pula sebaliknya setiap mata pelajaran dapat dipelajari banyak siswa. Setiap kegiatan belajar mengajar akan dilakukan pengambilan nilai.

Berikut adalah rancangan diagram ER:

|  |
| --- |
|  |

**Gambar 2.6** Rancangan diagram ERD

* 1. **KASUS MODUL**

1. Setiap kelompok membagi jumlah tabel ke masing-masing anggota kelompok.
2. Untuk setiap tabel dilakukan perintah berikut.
3. Membuat *table*.
4. Menampilkan *table*.
5. Menambah struktur *table*.
6. Menambahkan *primary key*.
7. Menambah *field*.
8. Mengubah nama *field*.
9. Mengubah struktur *field*.
10. Mengubah nama tabel dengan perintah *rename*.
11. Mengubah nama tabel dengan perintah *alter*.
12. Menghapus *field*.
13. Menghapus *table*.
    1. **ANALISIS KASUS**
14. **Membuka XAMPP**

|  |
| --- |
| xampp.PNG |

**Gambar 2.7** Tampilan XAMPP

Pada **Gambar 2.7** merupakan sebuah *software* yang digunakan untuk mengakses MySQL, dimana pada *software* tersebut dapat digunakan untuk membuat suatu *database*, nantinya bila kita ingin membuka *command prompt* pada *software* tersebut, kita harus memilih *button* yang bertuliskan *shell*, maka secara langsung program akan mengarahkan kita menuju *command prompt* yang merupakan tempat untuk menuliskan *query* pada *database*.

1. **Membuka CMD**

|  |
| --- |
|  |

**Gambar 2.8** Tampilan CMD

Pada **Gambar 2.8** merupakan sebuah *query* yang digunakan untuk membuka halaman *command prompt*, dimana halaman tersebut digunakan untuk mengakses *database*. *Query* “mysql -u root” digunakan untuk mengakses MySQL melalui *command prompt*, sehingga *query – query* mysql dapat berjalan dan dibaca pada *command prompt*. Berikut merupakan *query* yang digunakan untuk mengakses MySQL pada *command prompt*.

1. **Membuat dan Menggunakan *Database***

|  |
| --- |
|  |

**Gambar 2.9** Membuat dan menggunakan *database*

*Query* “CREATE” merupakan sebuah perintah untuk membuat, *query* “DATABASE” menunjukkan nama yang dibuat pada *query* tersebut adalah *database*, dan “ririn” menunjukkan nama dari *database* yang dibuat. Sehingga *query* pada **Gambar 2.9** merupakan perintah yang digunakan untuk membuat sebuah *database* dengan nama “ririn”. “use ririn” adalah perintah untuk memilih database “ririn”.

* + 1. **Tabel Guru**
    2. **Pembuatan Tabel**

|  |
| --- |
|  |

**Gambar 2.10** Membuat tabel guru

Pada **Gambar 2.10** *query* “create table”, merupakan sebuah perintah untuk membuat tabel dengan nama tabel “Guru”, di dalam tabel terdapat beberapa *field*, yaitu *field*  “alamat\_guru” yang bertipe data *varchar* yang memiliki batas 50 karakter, *field*  selanjutnya yaitu “kontak\_guru” bertipe data *integer*, *field* “nama\_guru” dengan tipe data *varchar* dengan batas karakter 30, dan untuk *field* “NUPTK” memiliki tipe data *varchar* dengan batas karakter sebanyak 10.

* + 1. **Menampilkan Tabel**

|  |
| --- |
|  |

**Gambar 2.11** Menampilkan tabel pada database

Pada **Gambar 2.11** *query* “show tables” merupakan perintah untuk menampilkan tabel yang terdapat pada *database* “ririn” tersebut. Di dalam *database* tersebut kita dapat melihat tabel yang baru saja dibuat.

1. **Menampilkan Struktur Tabel**

Berikut merupakan *query* yang digunakan dalam menampilkan struktur tabel:

|  |
| --- |
|  |

**Gambar 2.12** Menampilkan struktur tabel guru

Pada **Gambar 2.12** *query* yang digunakan untuk menampilkan struktur dari tabel guru yang terdapat dalam *database*. Sehingga akan muncul *field - field* yang terdapat di dalam tabel guru.

1. **Menambah *Primary Key***

Berikut merupakan *query* yang digunakan dalam menambahkan *primary key* pada tabel:

|  |
| --- |
|  |

**Gambar 2.13** Menambahkan *primary key* tabel guru

Pada **Gambar 2.13** terdapat *query* “add primary key” yang digunakan untuk menambahkan *constraint primary key* pada *field* “NUPTK” yang terdapat pada tabel guru.

1. **Menambah *Field***

Berikut merupakan *query* yang digunakan dalam menambahkan *field* pada tabel:

|  |
| --- |
|  |

**Gambar 2.14** Menambahkan *field* baru pada tabel guru

Pada **Gambar 2.14** terdapat *query* yang digunakan untuk menambahkan sebuah *field* ke dalam tabel guru dengan nama “status\_guru” dan dengan tipe data *varchar*  yang memiliki batas maksimal 10**.**

1. **Mengubah Nama *Field***

Berikut merupakan *query* yang digunakan dalam mengubah nama *field* pada tabel:

|  |
| --- |
|  |

**Gambar 2.15** Mengubah *field* pada tabel guru

Pada **Gambar 2.15** terdapat *query* “alter table guru change status\_guru status” yang berfungsi untuk mengubah nama dari *field* “status\_guru” menjadi “status” bertipe data *varchar*  yang memiliki batas maksimum 10.

1. **Mengubah Struktur *Field***

Berikut merupakan *query* yang digunakan dalam mengubah struktur *field* pada tabel:

|  |
| --- |
|  |

**Gambar 2.16** Mengubah struktur *field* pada tabel guru

Pada **Gambar 2.16** terdapat *query* “alter table Guru change status status int; ” yang digunakan untuk mengubah struktur dari *field* “status” yaitu dengan mengubah *size* *field* “status” menjadi bertipe data *int*.

1. **Mengubah Nama Tabel dengan Perintah *Rename***

Berikut merupakan *query* yang digunakan dalam mengubah nama tabel dengan perintah “Rename”.

|  |
| --- |
|  |

**Gambar 2.17** Mengubah nama tabel giri dengan perintah *rename*

Pada **Gambar 2.17**  terdapat *query* “rename table giri to guru” merupakan perintah untuk mengubah nama tabel dari “giri” menjadi “guru” dengan perintah “guru”.

1. **Mengubah Nama Tabel dengan Perintah *Alter***

Berikut merupakan *query* yang digunakan dalam mengubah nama tabel dengan perintah “ALTER”:

|  |
| --- |
|  |

**Gambar 2.18** Mengubah nama tabel guru dengan perintah *alter*

Pada **Gambar 2.18** terdapat *query* yang digunakan untuk mengubah nama tabel gurumenjadi giri.

1. **Menghapus *Field***

Berikut merupakan *query* yang digunakan dalam menghapus *field* dari tabel:

|  |
| --- |
|  |

**Gambar 2.19** Menghapus *field* dalam tabel guru

Pada **Gambar 2.19** terdapat *query* yang digunakan untuk menghapus *field* “status” di dalam tabel guru.

1. **Menghapus Tabel**

Berikut merupakan *query* yang digunakan dalam menghapus tabel:

|  |
| --- |
|  |

**Gambar 2.20** Menghapus tabel guru

Pada **Gambar 2.20** terdapat *query* yang digunakan untuk menghapus tabel guru dari sebuah *database*.

* + 1. **Tabel Mata\_Pelajaran**
    2. **Pembuatan Tabel**

Berikut merupakan *query* untuk membuat tabel:

|  |
| --- |
|  |

**Gambar 2.21** Membuat tabel Mata\_Pelajaran

Pada **Gambar 2.21** *query* “create table” merupakan sebuah perintah untuk membuat tabel dengan nama tabel “Mata\_Pelajaran”, di dalam tabel terdapat beberapa *field*, yaitu *field*  “kode\_mapel” yang bertipe data *int*, *field*  selanjutnya yaitu “nama\_mapel” bertipe data *varchar* dan memiliki batas maksimum 20.

* + 1. **Menampilkan Tabel**

Berikut merupakan query yang digunakan dalam penampilan tabel:

|  |
| --- |
|  |

**Gambar 2.22** Menampilkan tabel mata\_pelajaran

Pada **Gambar 2.22** *query* “show tables” merupakan perintah untuk menampilkan tabel yang terdapat pada *database* “ririn” tersebut. Di dalam *database* tersebut terdapat tabel “guru” dan “Mata\_Pelajaran” .

* + 1. **Menampilkan Struktur Tabel**

Berikut merupakan *query* yang digunakan dalam menampilkan struktur tabel:

|  |
| --- |
|  |

**Gambar 2.23** Menampilkan struktur tabel

Pada **Gambar 2.23** *query* yang digunakan untuk menampilkan struktur dari tabel Mata\_Pelajaran yang terdapat dalam *database*. Sehingga akan muncul *field - field* yang terdapat di dalam tabel Mata\_Pelajaran.

* + 1. **Menambah *Primary Key***

Berikut merupakan *query* yang digunakan dalam menambahkan *primary key* pada tabel:

|  |
| --- |
|  |

**Gambar 2.24** Menambahkan *primary key* tabel Mata\_Pelajaran

Pada **Gambar 2.24** terdapat *query* “add primary key” yang digunakan untuk menambahkan *constraint primary key* pada *field* “kode\_mapel” yang terdapat pada tabel “Mata\_Pelajaran”.

* + 1. **Menambah *Field***

Berikut merupakan *query* yang digunakan dalam menambahkan *field* pada tabel:

|  |
| --- |
|  |

**Gambar 2.25** Menambahkan *field* baru pada tabel Mata\_Pelajaran

Pada **Gambar 2.25** terdapat *query* yang digunakan untuk menambahkan sebuah *field* ke dalam tabel Mata\_Pelajaran dengan nama “guru\_mapel” dan dengan tipe data *varchar*  yang memiliki batas maksimal 20**.**

* + 1. **Mengubah Nama *Field***

Berikut merupakan *query* yang digunakan dalam mengubah nama *field* pada tabel:

|  |
| --- |
|  |

**Gambar 2.26** Mengubah *field* pada tabel Mata\_Pelajaran

. Pada **Gambar 2.26** terdapat *query* “alter table Mata\_Pelajaran change guru\_mapel gurumapel” yang berfungsi untuk mengubah nama dari *field* “guru\_mapel” menjadi “gurumapel” bertipe data *varchar*  yang memiliki batas maksimum 20.

* + 1. **Mengubah Struktur *Field***

Berikut merupakan *query* yang digunakan dalam mengubah struktur *field* pada tabel:

|  |
| --- |
|  |

**Gambar 2.27** Mengubah struktur *field* pada tabel Mata\_Pelajaran

Pada **Gambar 2.27** terdapat *query* “alter table Mata\_Pelajaran change gurumapel gurumapel int; ” yang digunakan untuk mengubah struktur dari *field* “gurumapel” yaitu dengan mengubah *size* *field* “gurumapel” menjadi bertipe data *int*.

* + 1. **Mengubah Nama Tabel dengan Perintah *Rename***

Berikut merupakan *query* yang digunakan dalam mengubah nama tabel dengan perintah “Rename”:

|  |
| --- |
|  |

**Gambar 2.28** Mengubah nama tabel *admin* dengan perintah Mata\_Pelajaran

Pada **Gambar 2.28**  terdapat *query* “rename table Mata\_Pelajaran to MAPEL” merupakan perintah untuk mengubah nama tabel dari “Mata\_Pelajaran” menjadi “MAPEL” dengan perintah “rename”.

* + 1. **Mengubah Nama Tabel dengan Perintah *Alter***

Berikut merupakan *query* yang digunakan dalam mengubah nama tabel dengan perintah “ALTER”:

|  |
| --- |
|  |

**Gambar 2.29** Mengubah nama tabel *admin* dengan perintah Mata\_Pelajaran

Pada **Gambar 2.29** terdapat *query* yang digunakan untuk mengubah nama tabel MAPELmenjadi Mata\_Pelajaran.

* + 1. **Menghapus *Field***

Berikut merupakan *query* yang digunakan dalam menghapus *field* dari tabel:

|  |
| --- |
|  |

**Gambar 2.30** Menghapus *field* dalam tabel Mata\_Pelajaran

Pada **Gambar 2.30** terdapat *query* yang digunakan untuk menghapus *field* “gurumapel” di dalam tabel Mata\_Pelajaran.

* + 1. **Menghapus Tabel**

Berikut merupakan *query* yang digunakan dalam menghapus tabel:

|  |
| --- |
|  |

**Gambar 2.31** Menghapus tabel Mata\_Pelajaran

Pada **Gambar 2.31** terdapat *query* yang digunakan untuk menghapus tabel Mapel dari sebuah *database*.

* + 1. **Tabel PNS**
  1. **Pembuatan Tabel**

Berikut merupakan *query* untuk membuat tabel:

|  |
| --- |
|  |

**Gambar 2.32** Membuat tabel PNS

Pada **Gambar 2.32** *query* “create table” merupakan sebuah perintah untuk membuat tabel dengan nama tabel “PNS”, di dalam tabel terdapat beberapa *field*, yaitu *field*  “NIP” yang bertipe data *varchar* yang memiliki batas 10, *field*  selanjutnya yaitu “NUPTK” bertipe data *varchar* dan memiliki batas maksimum 20.

* 1. **Menampilkan Tabel**

Berikut merupakan query yang digunakan dalam penampilan tabel:

|  |
| --- |
|  |

**Gambar 2.32** Menampilkan tabel

Pada **Gambar 2.32** *query* “show tables” merupakan perintah untuk menampilkan tabel yang terdapat pada *database* “ririn” tersebut. Di dalam *database* tersebut terdapat tabel “guru , mata\_pelajaran , menguasai, pns”.

* 1. **Menampilkan Struktur Tabel**

Berikut merupakan *query* yang digunakan dalam menampilkan struktur tabel:

|  |
| --- |
|  |

**Gambar 2.33** Menampilkan struktur tabel PNS

Pada **Gambar 2.33** *query* yang digunakan untuk menampilkan struktur dari tabel PNS yang terdapat dalam *database*. Sehingga akan muncul *field - field* yang terdapat di dalam tabel PNS.

* 1. **Menambah *Primary Key***

Berikut merupakan *query* yang digunakan dalam menambahkan *primary key* pada tabel:

|  |
| --- |
|  |

**Gambar 2.34** Menambahkan *primary key* tabel PNS

Pada **Gambar 2.34** terdapat *query* “add primary key” yang digunakan untuk menambahkan *constraint primary key* pada *field* “NUPTK” yang terdapat pada tabel mempelajari.

* 1. **Menambah *Field***

Berikut merupakan *query* yang digunakan dalam menambahkan *field* pada tabel:

|  |
| --- |
|  |

**Gambar 2.35** Menambahkan *field* baru pada tabel *PNS*

Pada **Gambar 2.35** terdapat *query* yang digunakan untuk menambahkan sebuah *field* ke dalam tabel mempelajari dengan nama “guru” dan dengan tipe data *varchar*  yang memiliki batas maksimal 10**.**

* 1. **Mengubah Nama *Field***

Berikut merupakan *query* yang digunakan dalam mengubah nama *field* pada tabel:

|  |
| --- |
|  |

**Gambar 2.36** Mengubah *field* pada tabel PNS

Pada **Gambar 2.36** terdapat *query* “alter table PNS change guru namaguru” yang berfungsi untuk mengubah nama dari *field* “guru” menjadi “namaguru” bertipe data *varchar*  yang memiliki batas maksimum 10.

* 1. **Mengubah Struktur *Field***

Berikut merupakan *query* yang digunakan dalam mengubah struktur *field* pada tabel:

|  |
| --- |
|  |

**Gambar 2.37** Mengubah struktur *field* pada tabel PNS

Pada **Gambar 2.37** terdapat *query* “alter table PNS change namaguru namaguru int; ” yang digunakan untuk mengubah struktur dari *field* “namaguru” yaitu dengan mengubah *size* *field* “namaguru” menjadi bertipe data *int*.

* 1. **Mengubah Nama Tabel dengan Perintah *Rename***

Berikut merupakan *query* yang digunakan dalam mengubah nama tabel dengan perintah “Rename”:

|  |
| --- |
|  |

**Gambar 2.38** Mengubah nama tabel PNSdengan perintah *rename*

Pada **Gambar 2.38**  terdapat *query* “rename table PNS to pegawainegerisipil” merupakan perintah untuk mengubah nama tabel dari “PNS” menjadi “pegawainegerisipil” dengan perintah “rename”.

* 1. **Mengubah Nama Tabel dengan Perintah *Alter***

Berikut merupakan *query* yang digunakan dalam mengubah nama tabel dengan perintah “ALTER”:

|  |
| --- |
|  |

**Gambar 2.39** Mengubah nama tabel PNS dengan perintah *alter*

Pada **Gambar 2.39** terdapat *query* yang digunakan untuk mengubah nama tabel pegawainegerisipilmenjadi PNS.

* 1. **Menghapus *Field***

Berikut merupakan *query* yang digunakan dalam menghapus *field* dari tabel:

|  |
| --- |
|  |

**Gambar 2.40** Menghapus *field* dalam tabel PNS

Pada **Gambar 2.40** terdapat *query* yang digunakan untuk menghapus *field* “namaguru” di dalam tabel PNS.

* 1. **Menghapus Tabel**

Berikut merupakan *query* yang digunakan dalam menghapus tabel:

|  |
| --- |
|  |

**Gambar 2.41** Menghapus tabel PNS

Pada **Gambar 2.41** terdapat *query* yang digunakan untuk menghapus tabel PNS dari sebuah *database*.

* + 1. **Tabel Honorer**

1. **Pembuatan Tabel**

Berikut merupakan *query* untuk membuat tabel:

|  |
| --- |
|  |

**Gambar 2.42** Membuat tabel mempelajari

Pada **Gambar 2.42** *query* “create table” merupakan sebuah perintah untuk membuat tabel dengan nama tabel “Honorer”, di dalam tabel terdapat beberapa *field*, yaitu *field*  “NUPTK” yang bertipe data *varchar* yang memiliki batas 20.

1. **Menampilkan Tabel**

Berikut merupakan query yang digunakan dalam penampilan tabel:

|  |
| --- |
|  |

**Gambar 2.43** Menampilkan tabel mempelajari

Pada **Gambar 2.43** *query* “show tables” merupakan perintah untuk menampilkan tabel yang terdapat pada *database* “ririn” tersebut. Di dalam *database* tersebut terdapat tabel “guru, honorer, mata\_pelajaran, menguasai , pns”.

1. **Menampilkan Struktur Tabel**

Berikut merupakan *query* yang digunakan dalam menampilkan struktur tabel:

|  |
| --- |
|  |

**Gambar 2.44** Menampilkan struktur tabel mempelajari

Pada **Gambar 2.44** *query* yang digunakan untuk menampilkan struktur dari tabel honorer yang terdapat dalam *database*. Sehingga akan muncul *field - field* yang terdapat di dalam tabel honorer.

1. **Menambah *Primary Key***

Berikut merupakan *query* yang digunakan dalam menambahkan *primary key* pada tabel:

|  |
| --- |
|  |

**Gambar 2.45** Menambahkan *primary key* tabel mempelajari

Pada **Gambar 2.45** terdapat *query* “add primary key” yang digunakan untuk menambahkan *constraint primary key* pada *field* “NUPTK” yang terdapat pada tabel Honorer.

1. **Menambah *Field***

Berikut merupakan *query* yang digunakan dalam menambahkan *field* pada tabel:

|  |
| --- |
|  |

**Gambar 2.46** Menambahkan *field* baru pada tabel mempelajari

Pada **Gambar 2.46** terdapat *query* yang digunakan untuk menambahkan sebuah *field* ke dalam tabel mempelajari dengan nama “guru” dan dengan tipe data *varchar*  yang memiliki batas maksimal 30**.**

1. **Mengubah Nama *Field***

Berikut merupakan *query* yang digunakan dalam mengubah nama *field* pada tabel:

|  |
| --- |
|  |

**Gambar 2.47** Mengubah *field* pada tabel mempelajari

Pada **Gambar 2.47** terdapat *query* “alter table mempelajari change guru namaguru” yang berfungsi untuk mengubah nama dari *field* “guru” menjadi “namaguru” bertipe data *varchar*  yang memiliki batas maksimum 30.

1. **Mengubah Struktur *Field***

Berikut merupakan *query* yang digunakan dalam mengubah struktur *field* pada tabel:

|  |
| --- |
|  |

**Gambar 2.48** Mengubah struktur *field* pada tabel Honorer

Pada **Gambar 2.48** terdapat *query* “alter table Honorer change namaguru namaguru int; ” yang digunakan untuk mengubah struktur dari *field* “namaguru” yaitu dengan mengubah *size* *field* “namaguru” menjadi bertipe data *int*.

1. **Mengubah Nama Tabel dengan Perintah *Rename***

Berikut merupakan *query* yang digunakan dalam mengubah nama tabel dengan perintah “Rename”:

|  |
| --- |
|  |

**Gambar 2.49** Mengubah nama tabel Honorerdengan perintah *rename*

Pada **Gambar 2.49**  terdapat *query* “rename table Honorer to Honor” merupakan perintah untuk mengubah nama tabel dari “Honorer” menjadi “Honor” dengan perintah “rename”.

1. **Mengubah Nama Tabel dengan Perintah *Alter***

Berikut merupakan *query* yang digunakan dalam mengubah nama tabel dengan perintah “ALTER”:

|  |
| --- |
|  |

**Gambar 2.50** Mengubah nama tabel Honor dengan perintah *alter*

Pada **Gambar 2.50** terdapat *query* yang digunakan untuk mengubah nama tabel Honormenjadi Honorer.

1. **Menghapus *Field***

Berikut merupakan *query* yang digunakan dalam menghapus *field* dari tabel:

|  |
| --- |
|  |

**Gambar 2.51** Menghapus *field* dalam tabel Honorer

Pada **Gambar 2.51** terdapat *query* yang digunakan untuk menghapus *field* “namaguru” di dalam tabel Honorer.

1. **Menghapus Tabel**

Berikut merupakan *query* yang digunakan dalam menghapus tabel:

|  |
| --- |
|  |

**Gambar 2.52** Menghapus tabel Honor

Pada **Gambar 2.52** terdapat *query* yang digunakan untuk menghapus tabel Honor dari sebuah *database*.

* + 1. **Tabel Siswa**
    2. **Pembuatan Tabel**

Berikut merupakan *query* untuk membuat tabel:

|  |
| --- |
|  |

**Gambar 2.53** Membuat tabel Siswa

Pada **Gambar 2.53** *query* “create table” merupakan sebuah perintah untuk membuat tabel dengan nama tabel “siswa”, di dalam tabel terdapat beberapa *field*, yaitu *field*  “NIS” yang bertipe data *varchar* yang memiliki batas 20, *field*  selanjutnya yaitu “nama\_siswa” bertipe data *varchar* dan memiliki batas maksimum 30, *field* “alamat\_siswa” dengan tipe data *varchar* dengan batas karakter 50, *field*  “nama\_wali” yang bertipe data *varchar* yang memiliki batas 20, *field*  selanjutnya yaitu “kontak\_wali” bertipe data *varchar* dan memiliki batas maksimum 20, *field* “rank” dengan tipe data *varchar* dengan batas karakter 20, *field* “kode\_kelas” dengan tipe data *varchar* dengan batas karakter 5.

* 1. **Menampilkan Tabel**

Berikut merupakan query yang digunakan dalam penampilan tabel:

|  |
| --- |
|  |

**Gambar 2.54** Menampilkan tabel Siswa

Pada **Gambar 2.54** *query* “show tables” merupakan perintah untuk menampilkan tabel yang terdapat pada *database* “ririn” tersebut. Di dalam *database* tersebut terdapat tabel “guru, honorer, mata\_pelajaran, menguasai , pns, siswa”.

* 1. **Menampilkan Struktur Tabel**

Berikut merupakan *query* yang digunakan dalam menampilkan struktur tabel:

|  |
| --- |
|  |

**Gambar 2.55** Menampilkan struktur tabel Siswa

Pada **Gambar 2.55** *query* yang digunakan untuk menampilkan struktur dari tabel siswa yang terdapat dalam *database*. Sehingga akan muncul *field - field* yang terdapat di dalam tabel mempelajari.

* 1. **Menambah *Primary Key***

Berikut merupakan *query* yang digunakan dalam menambahkan *primary key* pada tabel:

|  |
| --- |
|  |

**Gambar 2.56** Menambahkan *primary key* tabel Siswa

. Pada **Gambar 2.56** terdapat *query* “add primary key” yang digunakan untuk menambahkan *constraint primary key* pada *field* “NIS” yang terdapat pada tabel Siswa.

* 1. **Menambah *Field***

Berikut merupakan *query* yang digunakan dalam menambahkan *field* pada tabel:

|  |
| --- |
|  |

**Gambar 2.57** Menambahkan *field* baru pada tabel Siswa

Pada **Gambar 2.57** terdapat *query* yang digunakan untuk menambahkan sebuah *field* ke dalam tabel Siswa dengan nama “guru” dan dengan tipe data *varchar*  yang memiliki batas maksimal 20**.**

* 1. **Mengubah Nama *Field***

Berikut merupakan *query* yang digunakan dalam mengubah nama *field* pada tabel:

|  |
| --- |
|  |

**Gambar 2.58** Mengubah *field* pada tabel Siswa

Pada **Gambar 2.58** terdapat *query* “alter table Siswa change guru namaguru” yang berfungsi untuk mengubah nama dari *field* “guru” menjadi “namaguru” bertipe data *varchar* yang memiliki batas maksimum 20.

* 1. **Mengubah Struktur *Field***

Berikut merupakan *query* yang digunakan dalam mengubah struktur *field* pada tabel:

|  |
| --- |
|  |

**Gambar 2.59** Mengubah struktur *field* pada tabel siswa

Pada **Gambar 2.59** terdapat *query* “alter table Siswa change namaguru namaguru int; ” yang digunakan untuk mengubah struktur dari *field* “namaguru” yaitu dengan mengubah *size* *field* “namaguru” menjadi bertipe data *int*.

* 1. **Mengubah Nama Tabel dengan Perintah *Rename***

Berikut merupakan *query* yang digunakan dalam mengubah nama tabel dengan perintah “Rename”:

|  |
| --- |
|  |

**Gambar 2.60** Mengubah nama tabel siswa dengan perintah *rename*

Pada **Gambar 2.60**  terdapat *query* “rename table Siswa to Siswi” merupakan perintah untuk mengubah nama tabel dari “Siswa” menjadi “Siswi” dengan perintah “rename”.

* 1. **Mengubah Nama Tabel dengan Perintah *Alter***

Berikut merupakan *query* yang digunakan dalam mengubah nama tabel dengan perintah “ALTER”:

|  |
| --- |
|  |

**Gambar 2.61** Mengubah nama tabel siswi dengan perintah *alter*

Pada **Gambar 2.61** terdapat *query* yang digunakan untuk mengubah nama tabel Siswimenjadi Siswa.

* 1. **Menghapus *Field***

Berikut merupakan *query* yang digunakan dalam menghapus *field* dari tabel:

|  |
| --- |
|  |

**Gambar 2.62** Menghapus *field* dalam tabel siswa

Pada **Gambar 2.62** terdapat *query* yang digunakan untuk menghapus *field* “namaguru” di dalam tabel Siswa.

* 1. **Menghapus Tabel**

Berikut merupakan *query* yang digunakan dalam menghapus tabel:

|  |
| --- |
|  |

**Gambar 2.63** Menghapus tabel siswi

Pada **Gambar 2.63** terdapat *query* yang digunakan untuk menghapus tabel Siswi dari sebuah *database*.

* + 1. **Tabel Kelas**
    2. **Pembuatan Tabel**

Berikut merupakan *query* untuk membuat tabel:

|  |
| --- |
|  |

**Gambar 2.64** Membuat tabel kelas

Pada **Gambar 2.64** *query* “create table” merupakan sebuah perintah untuk membuat tabel dengan nama tabel “kelas”, di dalam tabel terdapat beberapa *field*, yaitu *field*  “kode\_kelas” yang bertipe data *varchar* yang memiliki batas 20, *field*  selanjutnya yaitu “nama\_kelas” bertipe data *varchar* dan memiliki batas maksimum 20, *field* “NIS\_ketua” dengan tipe data *varchar* dengan batas karakter 10, *field* “NUPTK\_walikelas” dengan tipe data *varchar* dengan batas karakter 20.

* + 1. **Menampilkan Tabel**

Berikut merupakan query yang digunakan dalam penampilan tabel:

|  |
| --- |
|  |

**Gambar 2.65** Menampilkan tabel kelas

Pada **Gambar 2.65** *query* “show tables” merupakan perintah untuk menampilkan tabel yang terdapat pada *database* “ririn” tersebut. Di dalam *database* tersebut terdapat tabel “guru, honorer, kelas, mata\_pelajaran, menguasai, pns, siswa”.

**3. Menampilkan Struktur Tabel**

Berikut merupakan *query* yang digunakan dalam menampilkan struktur tabel:

|  |
| --- |
|  |

**Gambar 2.66** Menampilkan struktur tabel kelas

Pada **Gambar 2.66** *query* yang digunakan untuk menampilkan struktur dari tabel Kelas yang terdapat dalam *database*. Sehingga akan muncul *field - field* yang terdapat di dalam tabel Kelas.

* + 1. **Menambah *Primary Key***

Berikut merupakan *query* yang digunakan dalam menambahkan *primary key* pada tabel:

|  |
| --- |
|  |

**Gambar 2.67** Menambahkan *primary key* tabel kelas

Pada **Gambar 2.67** terdapat *query* “add primary key” yang digunakan untuk menambahkan *constraint primary key* pada *field* “kode\_kelas” yang terdapat pada tabel Kelas.

* 1. **Menambah *Field***

Berikut merupakan *query* yang digunakan dalam menambahkan *field* pada tabel:

|  |
| --- |
|  |

**Gambar 2.68** Menambahkan *field* baru pada tabel kelas

Pada **Gambar 2.68** terdapat *query* yang digunakan untuk menambahkan sebuah *field* ke dalam tabel Kelas dengan nama “guru” dan dengan tipe data *varchar*  yang memiliki batas maksimal 20**.**

* 1. **Mengubah Nama *Field***

Berikut merupakan *query* yang digunakan dalam mengubah nama *field* pada tabel:

|  |
| --- |
|  |

**Gambar 2.69** Mengubah *field* pada tabel kelas

Pada **Gambar 2.69** terdapat *query* “alter table Kelas change guru namaguru” yang berfungsi untuk mengubah nama dari *field* “guru” menjadi “namaguru” bertipe data *varchar*  yang memiliki batas maksimum 20.

* 1. **Mengubah Struktur *Field***

Berikut merupakan *query* yang digunakan dalam mengubah struktur *field* pada tabel:

|  |
| --- |
|  |

**Gambar 2.70** Mengubah struktur *field* pada tabel kelas

Pada **Gambar 2.70** terdapat *query* “alter table Kelas change namaguru namaguru int; ” yang digunakan untuk mengubah struktur dari *field* “namaguru” yaitu dengan mengubah *size* *field* “namaguru” menjadi bertipe data *int*.

* 1. **Mengubah Nama Tabel dengan Perintah *Rename***

Berikut merupakan *query* yang digunakan dalam mengubah nama tabel dengan perintah “Rename”:

|  |
| --- |
|  |

**Gambar 2.71** Mengubah nama tabel kelasdengan perintah *rename*

Pada **Gambar 2.71**  terdapat *query* “rename table Kelas to Ruangan” merupakan perintah untuk mengubah nama tabel dari “Kelas” menjadi “Ruangan” dengan perintah “rename”.

* 1. **Mengubah Nama Tabel dengan Perintah *Alter***

Berikut merupakan *query* yang digunakan dalam mengubah nama tabel dengan perintah “ALTER”:

|  |
| --- |
|  |

**Gambar 2.72** Mengubah nama tabel ruangan dengan perintah *alter*

Pada **Gambar 2.72** terdapat *query* yang digunakan untuk mengubah nama tabel Ruanganmenjadi Kelas.

* 1. **Menghapus *Field***

Berikut merupakan *query* yang digunakan dalam menghapus *field* dari tabel:

|  |
| --- |
|  |

**Gambar 2.73** Menghapus *field* dalam tabel kelas

Pada **Gambar 2.73** terdapat *query* yang digunakan untuk menghapus *field* “namaguru” di dalam tabel Kelas.

* 1. **Menghapus Tabel**

Berikut merupakan *query* yang digunakan dalam menghapus tabel:

|  |
| --- |
|  |

**Gambar 2.74** Menghapus tabel ruangan

Pada **Gambar 2.74** terdapat *query* yang digunakan untuk menghapus tabel Ruangan dari sebuah *database*.

* + 1. **Tabel Menguasai**
  1. **Pembuatan Tabel**

Berikut merupakan *query* untuk membuat tabel:

|  |
| --- |
|  |

**Gambar 2.75** Membuat tabel menguasai

Pada **Gambar 2.75** *query* “create table” merupakan sebuah perintah untuk membuat tabel dengan nama tabel “Menguasai”, di dalam tabel terdapat beberapa *field*, yaitu *field*  “NUPTK” yang bertipe data *varchar* yang memiliki batas 10, *field*  selanjutnya yaitu “kode\_mapel” bertipe data *varchar* dan memiliki batas maksimum 10.

1. **Menampilkan Tabel**

Berikut merupakan query yang digunakan dalam penampilan tabel:

|  |
| --- |
|  |

**Gambar 2.76** Menampilkan tabel menguasai

Pada **Gambar 2.76** *query* “show tables” merupakan perintah untuk menampilkan tabel yang terdapat pada *database* “teater\_kegelapan” tersebut. Di dalam *database* tersebut terdapat tabel “guru, honorer, kelas, mata\_pelajaran, menguasai, pns, siswa”.

1. **Menampilkan Struktur Tabel**

Berikut merupakan *query* yang digunakan dalam menampilkan struktur tabel:

|  |
| --- |
|  |

**Gambar 2.77** Menampilkan struktur tabel menguasai

Pada **Gambar 2.77** *query* yang digunakan untuk menampilkan struktur dari tabel Menguasai yang terdapat dalam *database*. Sehingga akan muncul *field - field* yang terdapat di dalam tabel Menguasai.

1. **Menambah *Field***

Berikut merupakan *query* yang digunakan dalam menambahkan *field* pada tabel:

|  |
| --- |
|  |

**Gambar 2.78** Menambahkan *primary key* tabel menguasai

Pada **Gambar 2.78** terdapat *query* yang digunakan untuk menambahkan sebuah *field* ke dalam tabel Menguasai dengan nama “status” dan dengan tipe data *varchar*  yang memiliki batas maksimal 20**.**

1. **Mengubah Nama *Field***

Berikut merupakan *query* yang digunakan dalam mengubah nama *field* pada tabel:

|  |
| --- |
|  |

**Gambar 2.79** Menambahkan *field* baru pada tabel menguasai

Pada **Gambar 2.79** terdapat *query* “alter table change change status statussiswa” yang berfungsi untuk mengubah nama dari *field* “status” menjadi “statussiswa” bertipe data *varchar*  yang memiliki batas maksimum 20.

1. **Mengubah Struktur *Field***

Berikut merupakan *query* yang digunakan dalam mengubah struktur *field* pada tabel:

|  |
| --- |
|  |

**Gambar 2.80** Mengubah *field* pada tabel menguasai

Pada **Gambar 2.80** terdapat *query* “alter table Menguasai change statussiswa statussiswa int; ” yang digunakan untuk mengubah struktur dari *field* “statussiswa” yaitu dengan mengubah *size* *field* “statussiswa” menjadi bertipe data *int*.

1. **Mengubah Nama Tabel dengan Perintah *Rename***

Berikut merupakan *query* yang digunakan dalam mengubah nama tabel dengan perintah “Rename”:

|  |
| --- |
|  |

**Gambar 2.81** Mengubah struktur *field* pada tabel menguasai

Pada **Gambar 2.81**  terdapat *query* “rename table Menguasai to Kuasai” merupakan perintah untuk mengubah nama tabel dari “Menguasai” menjadi “Menguasai” dengan perintah “Kuasai”.

1. **Mengubah Nama Tabel dengan Perintah *Alter***

Berikut merupakan *query* yang digunakan dalam mengubah nama tabel dengan perintah “ALTER”:

|  |
| --- |
|  |

**Gambar 2.82** Mengubah nama tabel kuasaidengan perintah *rename*

Pada **Gambar 2.82** terdapat *query* yang digunakan untuk mengubah nama tabel Kuasaimenjadi Menguasai.

1. **Menghapus *Field***

Berikut merupakan *query* yang digunakan dalam menghapus *field* dari tabel:

|  |
| --- |
|  |

**Gambar 2.83** Menghapus *field* dalam tabel menguasai

Pada **Gambar 2.83** terdapat *query* yang digunakan untuk menghapus *field* “statussiswa” di dalam tabel Menguasai.

1. **Menghapus Tabel**

Berikut merupakan *query* yang digunakan dalam menghapus tabel:

|  |
| --- |
|  |

**Gambar 2.84** Menghapus *field* dalam tabel menguasai

Pada **Gambar 2.84** terdapat *query* yang digunakan untuk menghapus tabel menguasai dari sebuah *database*.

* + 1. **Tabel Mengajar**
  1. **Pembuatan Tabel**

Berikut merupakan *query* untuk membuat tabel:

|  |
| --- |
|  |

**Gambar 2.85** Membuat tabel menguasai

Pada **Gambar 2.85** *query* “create table” merupakan sebuah perintah untuk membuat tabel dengan nama tabel “Menguasai”, di dalam tabel terdapat beberapa *field*, yaitu *field*  “NUPTK” yang bertipe data *varchar* yang memiliki batas 10, *field*  selanjutnya yaitu “kode\_mapel” bertipe data *varchar* dan memiliki batas maksimum 10.

* 1. **Menampilkan Tabel**

Berikut merupakan query yang digunakan dalam penampilan tabel:

|  |
| --- |
|  |

**Gambar 2.86** Menampilkan tabel menguasai

Pada **Gambar 2.86** *query* “show tables” merupakan perintah untuk menampilkan tabel yang terdapat pada *database* “teater\_kegelapan” tersebut. Di dalam *database* tersebut terdapat tabel “guru, honorer, kelas, mata\_pelajaran, mengajar, menguasai, pns, siswa”.

* 1. **Menampilkan Struktur Tabel**

Berikut merupakan *query* yang digunakan dalam menampilkan struktur tabel:

|  |
| --- |
|  |

**Gambar 2.87** Menampilkan struktur tabel mengajar

Pada **Gambar 2.87** *query* yang digunakan untuk menampilkan struktur dari tabel Mengajar yang terdapat dalam *database*. Sehingga akan muncul *field - field* yang terdapat di dalam tabel Mengajar.

* 1. **Menambah *Field***

Berikut merupakan *query* yang digunakan dalam menambahkan *field* pada tabel:

|  |
| --- |
|  |

**Gambar 2.88** Menambahkan *primary key* tabel mengajar

Pada **Gambar 2.88** terdapat *query* yang digunakan untuk menambahkan sebuah *field* ke dalam tabel mempelajari dengan nama “status” dan dengan tipe data *varchar*  yang memiliki batas maksimal 10**.**

* 1. **Mengubah Nama *Field***

Berikut merupakan *query* yang digunakan dalam mengubah nama *field* pada tabel:

|  |
| --- |
|  |

**Gambar 2.89** Menambahkan *field* baru pada tabel mengajar

Pada **Gambar 2.89** terdapat *query* “alter table mengajar change status statussiswa” yang berfungsi untuk mengubah nama dari *field* “status” menjadi “statussiswa” bertipe data *varchar*  yang memiliki batas maksimum 10.

* 1. **Mengubah Struktur *Field***

Berikut merupakan *query* yang digunakan dalam mengubah struktur *field* pada tabel:

|  |
| --- |
|  |

**Gambar 2.90** Mengubah *field* pada tabel mengajar

Pada **Gambar 2.90** terdapat *query* “alter table mempelajari change teshehe teshehe date; ” yang digunakan untuk mengubah struktur dari *field* “teshehe” yaitu dengan mengubah *size* *field* “teshehe” menjadi bertipe data *date*.

* 1. **Mengubah Nama Tabel dengan Perintah *Rename***

Berikut merupakan *query* yang digunakan dalam mengubah nama tabel dengan perintah “Rename”:

|  |
| --- |
|  |

**Gambar 2.91** Mengubah struktur *field* pada tabel mengajar

Pada **Gambar 2.91**  terdapat *query* “rename table mengajar to ajar” merupakan perintah untuk mengubah nama tabel dari “mengajar” menjadi “ajar” dengan perintah “rename”.

* 1. **Mengubah Nama Tabel dengan Perintah *Alter***

Berikut merupakan *query* yang digunakan dalam mengubah nama tabel dengan perintah “ALTER”:

|  |
| --- |
|  |

**Gambar 2.92** Mengubah nama tabel ajardengan perintah *rename*

Pada **Gambar 2.92** terdapat *query* yang digunakan untuk mengubah nama tabel ajarmenjadi mengajar.

* 1. **Menghapus *Field***

Berikut merupakan *query* yang digunakan dalam menghapus *field* dari tabel:

|  |
| --- |
|  |

**Gambar 2.93** Mengubah nama tabel mengajar dengan perintah *alter*

Pada **Gambar 2.93** terdapat *query* yang digunakan untuk menghapus *field* “teshaha” di dalam tabel *locations*.

* 1. **Menghapus Tabel**

Berikut merupakan *query* yang digunakan dalam menghapus tabel:

|  |
| --- |
|  |

**Gambar 2.94** Menghapus *field* dalam tabel ajar

Pada **Gambar 2.94** terdapat *query* yang digunakan untuk menghapus tabel ajar dari sebuah *database*.

* + 1. **Tabel Menghadiri**
       - 1. **Pembuatan Tabel**

Berikut merupakan *query* untuk membuat tabel:

|  |
| --- |
|  |

**Gambar 2.95** Membuat tabel menguasai

Pada **Gambar 2.95** *query* “create table” merupakan sebuah perintah untuk membuat tabel dengan nama tabel “menghadiri”, di dalam tabel terdapat beberapa *field*, yaitu *field*  “NUPTK” yang bertipe data *varchar* yang memiliki batas 10, *field*  selanjutnya yaitu “kode\_mapel” bertipe data *varchar* dan memiliki batas maksimum 10, *field* “kode\_kelas” dengan tipe data *varchar*  dengan batas karakter 10.

* + - * 1. **Menampilkan Tabel**

Berikut merupakan query yang digunakan dalam penampilan tabel:

|  |
| --- |
|  |

**Gambar 2.96** Menampilkan tabel

Pada **Gambar 2.96** *query* “show tables” merupakan perintah untuk menampilkan tabel yang terdapat pada *database* “teater\_kegelapan” tersebut. Di dalam *database* tersebut terdapat tabel “guru, honorer, kelas, mata\_pelajaran, mengajar, menghadiri, menguasai, pns, siswa”.

* 1. **Menambah *Field***

Berikut merupakan *query* yang digunakan dalam menambahkan *field* pada tabel:

|  |
| --- |
|  |

**Gambar 2.98** Menambahkan *primary key* tabel menghadiri

Pada **Gambar 2.98** terdapat *query* yang digunakan untuk menambahkan sebuah *field* ke dalam tabel menghadiri dengan nama “status” dan dengan tipe data *varchar*  yang memiliki batas maksimal 10**.**

* 1. **Mengubah Nama *Field***

Berikut merupakan *query* yang digunakan dalam mengubah nama *field* pada tabel:

|  |
| --- |
|  |

**Gambar 2.99** Menambahkan *field* baru pada tabel menghadiri

Pada **Gambar 2.99** terdapat *query* “alter table menghadiri change status statussiswa” yang berfungsi untuk mengubah nama dari *field* “status” menjadi “statussiswa” bertipe data *varchar*  yang memiliki batas maksimum 10.

* 1. **Mengubah Struktur *Field***

Berikut merupakan *query* yang digunakan dalam mengubah struktur *field* pada tabel:

|  |
| --- |
|  |

**Gambar 2.100** Mengubah *field* pada tabel mempelajari

Pada **Gambar 2.100** terdapat *query* “alter table mempelajari change teshehe teshehe date; ” yang digunakan untuk mengubah struktur dari *field* “teshehe” yaitu dengan mengubah *size* *field* “teshehe” menjadi bertipe data *date*.

* 1. **Mengubah Nama Tabel dengan Perintah *Rename***

Berikut merupakan *query* yang digunakan dalam mengubah nama tabel dengan perintah “Rename”:

|  |
| --- |
|  |

**Gambar 2.101** Mengubah struktur *field* pada tabel menguasai

Pada **Gambar 2.101**  terdapat *query* “rename table menghadiri to hadir” merupakan perintah untuk mengubah nama tabel dari “menghadiri” menjadi “hadir” dengan perintah “rename”.

* 1. **Mengubah Nama Tabel dengan Perintah *Alter***

Berikut merupakan *query* yang digunakan dalam mengubah nama tabel dengan perintah “ALTER”:

|  |
| --- |
|  |

**Gambar 2.102** Mengubah nama tabel menguasaidengan perintah *rename*

Pada **Gambar 2.102** terdapat *query* yang digunakan untuk mengubah nama tabel Hadirmenjadi Menghadiri.

* 1. **Menghapus *Field***

Berikut merupakan *query* yang digunakan dalam menghapus *field* dari tabel:

|  |
| --- |
|  |

**Gambar 2.103** Mengubah nama tabel menguasai dengan perintah *alter*

Pada **Gambar 2.103** terdapat *query* yang digunakan untuk menghapus *field* “statussiswa” di dalam tabel *locations*.

* 1. **Menghapus Tabel**

Berikut merupakan *query* yang digunakan dalam menghapus tabel:

|  |
| --- |
|  |

**Gambar 2.104** Menghapus *field* dalam tabel menguasai

Pada **Gambar 2.104** terdapat *query* yang digunakan untuk menghapus tabel hadir dari sebuah *database*.

* + 1. **Tabel Mempelajari**

1. **Pembuatan Tabel**

Berikut merupakan *query* untuk membuat tabel:

|  |
| --- |
|  |

**Gambar 2.105** Membuat tabel mempelajari

Pada **Gambar 2.105** *query* “create table” merupakan sebuah perintah untuk membuat tabel dengan nama tabel “Mempelajari”, di dalam tabel terdapat beberapa *field*, yaitu *field*  “nilai” yang bertipe data *integer* yang memiliki batas yang tidak diketahui jumlahnya, *field*  selanjutnya yaitu “kode\_kelas” bertipe data *varchar* dan memiliki batas maksimum 10, *field* “kode\_mapel” dengan tipe data *varchar* dengan batas karakter 10.

1. **Menampilkan Tabel**

Berikut merupakan query yang digunakan dalam penampilan tabel:

|  |
| --- |
|  |

**Gambar 2.106** Menampilkan tabel mempelajari

Pada **Gambar 2.106** *query* “show tables” merupakan perintah untuk menampilkan tabel yang terdapat pada *database* “teater\_kegelapan” tersebut. Di dalam *database* tersebut terdapat tabel “guru, honorer, kelas, mata\_pelajaran, mempelajari, mengajar, menguasai, pns, siswa”.

1. **Menampilkan Struktur Tabel**

Berikut merupakan *query* yang digunakan dalam menampilkan struktur tabel:

|  |
| --- |
|  |

**Gambar 2.107** Menampilkan struktur tabel mempelajari

Pada **Gambar 2.107** *query* yang digunakan untuk menampilkan struktur dari tabel mempelajari yang terdapat dalam *database*. Sehingga akan muncul *field - field* yang terdapat di dalam tabel mempelajari.

1. **Menambah *Field***

Berikut merupakan *query* yang digunakan dalam menambahkan *field* pada tabel:

|  |
| --- |
|  |

**Gambar 2.108** Menambahkan *primary key* tabel menguasai

Pada **Gambar 2.108** terdapat *query* yang digunakan untuk menambahkan sebuah *field* ke dalam tabel mempelajari dengan nama “pelajar” dan dengan tipe data *varchar*  yang memiliki batas maksimal 10**.**

1. **Mengubah Nama *Field***

Berikut merupakan *query* yang digunakan dalam mengubah nama *field* pada tabel:

|  |
| --- |
|  |

**Gambar 2.109** Menambahkan *field* baru pada tabel menguasai

Pada **Gambar 2.109** terdapat *query* “alter table mempelajari change pelajar ajar varchar(10);” yang berfungsi untuk mengubah nama dari *field* “pelajar” menjadi “ajar” bertipe data *varchar*  yang memiliki batas maksimum 10.

1. **Mengubah Struktur *Field***

Berikut merupakan *query* yang digunakan dalam mengubah struktur *field* pada tabel:

|  |
| --- |
|  |

**Gambar 2.110** Mengubah *field* pada tabel mempelajari

Pada **Gambar 2.110** terdapat *query* “alter table mempelajari change ajar ajar int; ” yang digunakan untuk mengubah struktur dari *field* “ajar” yaitu dengan mengubah *size* *field* “ajar” menjadi bertipe data *int*.

1. **Mengubah Nama Tabel dengan Perintah *Rename***

Berikut merupakan *query* yang digunakan dalam mengubah nama tabel dengan perintah “Rename”:

|  |
| --- |
|  |

**Gambar 2.111** Mengubah struktur *field* pada tabel menguasai

Pada **Gambar 2.111**  terdapat *query* “rename table mempelajari to ajar” merupakan perintah untuk mengubah nama tabel dari “mempelajari” menjadi “ajar” dengan perintah “rename”.

1. **Mengubah Nama Tabel dengan Perintah *Alter***

Berikut merupakan *query* yang digunakan dalam mengubah nama tabel dengan perintah “ALTER”:

|  |
| --- |
|  |

**Gambar 2.112** Mengubah nama tabel menguasaidengan perintah *rename*

Pada **Gambar 2.112** terdapat *query* yang digunakan untuk mengubah nama tabel ajarmenjadi mempelajari.

1. **Menghapus *Field***

Berikut merupakan *query* yang digunakan dalam menghapus *field* dari tabel:

|  |
| --- |
|  |

**Gambar 2.113** Menghapus *field* ajar pada tabel mempelajari

Pada **Gambar 2.113** terdapat *query* yang digunakan untuk menghapus *field* “ajar” di dalam tabel mempelajari.

1. **Menghapus Tabel**

Berikut merupakan *query* yang digunakan dalam menghapus tabel:

|  |
| --- |
|  |

**Gambar 2.114** Menghapus tabel ajar

Pada **Gambar 2.114** terdapat *query* yang digunakan untuk menghapus tabel ajar dari sebuah *database*.

* 1. **KESIMPULAN**

Berdasarkan praktikum yang telah dilakukan pada Modul 2 Manipulasi Struktur Tabel, dapat disimpulkan bahwa:

1. Untuk mengimplementasikan diagram ER ke dalam bentuk tabel, maka yang harus diperhatikan yaitu relasi dan juga hubungan antar entitas tersebut (relasi).
2. Terdapat beberapa bentuk implementasi diagram ER yaitu :

Relasi tunggal 1-1, maka salah satu atribut dapat dititipkan ke mana saja sebagai *foreign-key*.

Relasi tunggal 1-M atau M-1, maka atribut pada entitas tunggal berderajat 1 dititipkan ke entitas tunggal berderajat banyak sebagai *foreign-key*.

Relasi tunggal M-M, maka setiap entitas membuat sebuah tabel dan ada sebuah tabel lagi sebagai tabel relasi yang berisi *primary-key* dari masing – masing entitas yang berelasi.

1. Dalam pengimplementasian sebuah tabel, tidak boleh terdapat 2 *primary-key* dalam suatu tabel tetapi boleh terdapat 2 *foreign-key* di dalam suatu tabel. Terdapat perbedaan antara perintah atau query “CHANGE” dengan “MODIFY”, perintah “CHANGE” digunakan untuk merubah sebuah kolom termasuk nama dan tipe datanya. Sedangkan query “MODIFY” digunakan untuk merubah tipe data dari suatu kolom tertentu.