

## **BAB 2**

### **DATA DEFINITION LANGUAGE**

#### **2.1. Tujuan Pembelajaran**

1. Mengetahui dan Mengenal Bahasa SQL beserta pembagiannya
2. Menggunakan Bahasa SQL
3. Mengenal dan Menggunakan Sub Bahasa DDL (Data Definition Language)

#### **2.2. Dasar Teori**

##### **2.2.1. Mengelola Database**

Sebelum kita membuat suatu tabel yang digunakan untuk menyimpan data, maka harus terlebih dahulu dibuat database yang merupakan kumpulan atau berisi tabel - tabel yang saling berhubungan dengan menggunakan kunci - kunci yang ditentukan. Tapi bagaimana caranya memerintahkan mysql untuk membuat database, tabel, dan lainnya yang kita perlukan.

Untuk itu kita perlu untuk mempelajari bahasa yang akan digunakan untuk berkomunikasi dengan MySQL. Bahasa ini disebut dengan SQL ( Structured Query Language ) dan orang sering meng akronimnya dengan " sequel ". SQL merupakan bahasa standar untuk pengolahan database. Ini berarti bahwa DBMS yang lain juga mengenal bahasa ini. Walau ada beberapa istilah di salah satu DBMS yang tidak dikenal, tetapi dikenal di DBMS yang lain.

Di dalam bahasa SQL, perintah dibedakan menjadi 3 sub bahasa:

- a. DDL ( Data Definition Language )
- b. DML ( Data Manipulation Language )
- c. DCL ( Data Control Language )

##### **2.2.2. Data Definition Language**

Kelompok perintah ini bisa digunakan untuk melakukan pendefinisian database dan pendefinisian tabel. Sehingga dengan menggunakan perintah - perintah ini, kita bisa memerintahkan untuk membuat database, membuat tabel, mengubah strukturnya, menghapus tabel, membuat index tabel dan lain - lain yang berhubungan dengan pendefinisian database dan tabel.

#### **2.3. Software**

- XAMPP
- MySQL

## 2.4. Data Definition Language

### 2.4.1. Mengakses Database Melalui Command Prompt

Untuk mengakses nya kita bisa masuk ke folder direktori tempat kita menaruh xampp atau aplikasi sejenisnya, kalau di xampp kita masuk ke file C:\xampp\mysql\bin. Selanjutnya kita buka command prompt lalu arahkan ke direktori yang dituju, lalu ketikkan `mysql -u root`.

```
Microsoft Windows [Version 10.0.19044.2006]
(c) Microsoft Corporation. All rights reserved.

C:\xampp\mysql\bin>mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 210
Server version: 10.4.24-MariaDB mariadb.org binary distribution
```

Gambar 2.1 Masuk Ke SQL Melalui CMD

Maksud dari query diatas yaitu, kita ingin masuk dengan username bernama root dan password none atau tidak ada password, hal ini harus kita lakukan sebelum bisa mengetikkan query di SQL.

### 2.4.2. Membuat Database

Untuk mengetahui atau melihat database yang sudah ada, bisa digunakan perintah `show databases;`

Contoh:

```
MariaDB [(none)]> show databases;
```

Gambar 2.2 Sintak Untuk Melihat Database

Output:

```
+-----+
| Database |
+-----+
| ibik_api_academic |
| information_schema |
| mysql |
| performance_schema |
| phpmyadmin |
| test |
+-----+
6 rows in set (0.001 sec)
```

Gambar 2.3 List Database Yang Pernah Dibuat

Pada gambar diatas kita sudah berhasil melakukan query untuk melihat database. Maksud dari perintah diatas yaitu perlihatkan database yang ada di perangkat kita.

Dalam pembuatan database perlu perhatikan penulisan nama database tidak boleh menggunakan spasi dan karakter non standar. Bentuk penulisan perintah untuk membuat database baru adalah `create database nama_database;`

Contoh:

```
MariaDB [(none)]> create database perpustakaan;  
Query OK, 1 row affected (0.002 sec)
```

*Gambar 2.4 Membuat Database Perpustakaan*

Akhirnya kita sudah berhasil membuat database, tapi apa sih maksud query diatas? Maksudnya yaitu, kita memberi perintah untuk membuat database atau (create database). Untuk memastikan bahwa database yang kita buat sudah jadi, perintahkan `show databases` sehingga semua database yang ada di server dan diakses oleh user akan ditampilkan.

Output:

```
MariaDB [(none)]> show databases;  
+-----+  
| Database |  
+-----+  
| ibik_api_academic |  
| information_schema |  
| mysql |  
| performance_schema |  
| perpustakaan |  
| phpmyadmin |  
| test |  
+-----+  
7 rows in set (0.001 sec)
```

*Gambar 2.5 Sintak Kueri Untuk Melihat Database*

### 2.4.3. Membuat Tabel

Selanjutnya untuk memulai membuat tabel di dalam database perpustakaan, maka kita harus mengaktifkan terlebih dahulu database perpustakaan dengan menggunakan perintah `use nama_database;`

Contoh:

```
MariaDB [(none)]> use perpustakaan;  
Database changed
```

*Gambar 2.6 Sintak Kueri Menggunakan Database Perpustakaan*

Setelah database aktif, kita baru bisa memulai untuk membuat tabel yang kita perlukan. Pada database perpustakaan, akan disimpan data-data tentang buku, judul buku, kelompok buku, pengarang, dan penerbit. Selain itu juga untuk menyimpan data anggotabeserta transaksi yang dilakukan.

Pertama kali, kita membuat tabel buku yang fungsinya untuk menyimpan data tentang buku. Kita bisa mengetikkan query seperti ini.

```
MariaDB [perpustakaan]> create table buku(  
-> id_buku bigint(5) not null auto_increment primary key,  
-> index_buku varchar(15) not null default "",  
-> no_urut tinyint(3) default 0,  
-> referensi tinyint(1));  
Query OK, 0 rows affected (0.852 sec)
```

Gambar 2.7 Membuat Tabel Buku

Penjelasan dari query di atas yaitu, kita memberi perintah untuk membuat tabel buku dengan kolom pertama yaitu id buku bertipe data bigint dengan nilai sepanjang 5, maksudnya auto increment yaitu ketika kita memasukkan data misal 1, maka dia akan menambahkan otomatis pada baris yang baru jadi bernilai 2, dan yang terakhir yaitu primary key. Apa sih primary key? Mudahnya, primary key itu atribut kunci, nanti akan dibahas lebih jauh di pertemuan selanjutnya.

Kolom yang kedua bernama index buku dengan tipe data varchar dengan nilai sepanjang 15, maksudnya not null adalah index buku harus di isi datanya dan tidak bolehkosong, dan diberi nilai default yaitu string kosong atau tanda "".

Selanjutnya, ada kolom nomor urut dengan tipe data tinyint dengan nilai sepanjang 3 dan memiliki nilai default 0, kenapa 0? Karena tipe datanya tinyint, kalau tipe datanya bukan angka maka kita memasukkan tanda string kosong atau "".

Yang terakhir ada kolom referensi, dengan tipe data tinyint dengan nilai sepanjang

1. Untuk membuktikan bahwa kita berhasil, maka tabel yang ada database bisa kitatampilkan dengan cara show tables:

Contoh:

```
MariaDB [perpustakaan]> show tables;  
+-----+  
| Tables_in_perpustakaan |  
+-----+  
| buku                    |  
+-----+  
1 row in set (0.001 sec)
```

Gambar 2.8 Sintak Kueri Untuk Melihat Tabel

Sampai sini kita sudah berhasil membuat tabel buku yang ada di database perpustakaan, Perintah diatas digunakan saat kita ingin memastikan apakah tabel yang kita buat sudah berhasil atau belum.

Pada hasil perintah diatas ditampilkan, bahwa tabel yang sudah ada di database perpustakaan adalah tabel buku. Untuk menampilkan struktur dari tabel buku, perintahkan **desc buku**:

Contoh:

```
MariaDB [perpustakaan]> desc buku;
```

Gambar 2.9 Sintak Kueri Untuk Mendeskripsikan Tabel Buku

Output:

Field	Type	Null	Key	Default	Extra
id_buku	bigint(5)	NO	PRI	NULL	auto_increment
index_buku	varchar(15)	NO			
no_urut	tinyint(3)	YES		0	
referensi	tinyint(1)	YES		NULL	

4 rows in set (0.015 sec)

Gambar 2.10 Tampilan Struktur Tabel Buku

- 1) id\_buku, menyimpan no id setiap buku walaupun judulnya sama, sifatnya unik dan semua field yang lain bergantung hanya kepada id buku. Karena itu field ini dijadikan primary key.
- 2) index\_buku, digunakan untuk menyimpan indeks setiap judul buku yang tiap judul bukunya tersimpan di dalam tabel judul buku yang akan kita buat berikutnya.
- 3) no\_urut, digunakan untuk menyimpan no urut buku no urut buku untuk judul yang sama.
- 4) referensi: digunakan untuk mengetahui apakah buku boleh dipinjam / dibawa keluar atau tidak, jika boleh maka nilainya 0 jika tidak maka nilainya 1. Seperti ini apabila tiap judul buku ada yang boleh dibawa keluar ada yang tidak. Seandainya setiap judul buku yang sama memiliki kondisi referensi yang sama, maka field ini dimasukkan kedalam tabel judul buku.

Berikutnya kita akan membuat tabel judul buku, kita bisa mengetikkan query seperti ini.

```
MariaDB [perpustakaan]> create table judul_buku(  
-> index_buku varchar(10) not null primary key,  
-> judul varchar(90),  
-> kd_pengarang varchar(5),  
-> kd_kelompok smallint(3),  
-> edisi tinyint(2),  
-> thn_terbit int(4));  
Query OK, 0 rows affected (0.290 sec)  
  
MariaDB [perpustakaan]>
```

Gambar 2.11 Membuat Tabel Judul Buku

- 1) index\_buku: buku dengan judul, pengarang, penerbit, dan edisi yang sama akan memiliki index yang sama.
- 2) judul: judul buku.
- 3) kd\_pengarang: kode pengarang yang nama pengarangnya ada di tabel pengarang.

- 4) kd\_kelompok: kode kelompok isi buku sesuai ilmu, yang domainnya dan tabelkelompok.
- 5) edisi: cetakan ke berapa untuk judul, penerbit, dan pengarang yang sama.
- 6) thn\_terbit: tahun terbit buku.

Untuk tabel - tabel yang selanjutnya, coba buatlah sendiri dengan struktur sebagai berikut.

#### Tabel Kelompok

Field	Type	Null	Key	Default	Extra
kd_kelompok	smallint(3)	NO	PRI	0	
kelompok	varchar(20)	YES		NULL	

Gambar 2.12 Struktur Tabel Kelompok

#### Tabel Penerbit

Field	Type	Null	Key	Default	Extra
id_penerbit	varchar(5)	NO	PRI	NULL	
penerbit	varchar(20)	YES		NULL	
kota	varchar(20)	YES		NULL	

Gambar 2.13 Struktur Tabel Penerbit

#### Tabel Pengarang

Field	Type	Null	Key	Default	Extra
id_pengarang	varchar(5)	NO	PRI	NULL	
nama	varchar(30)	YES		NULL	

Gambar 2.14 Struktur Tabel Pengarang

#### Tabel Anggota

Field	Type	Null	Key	Default	Extra
kd_anggota	varchar(7)	YES		NULL	
nama_anggota	varchar(50)	YES		NULL	
tgllahir	date	YES		NULL	
jklmn	tinyint(1)	YES		NULL	
alamat	varchar(50)	YES		NULL	
no_hp	varchar(12)	YES		NULL	
no_telp	varchar(15)	YES		NULL	
tmp_lahir	varchar(20)	YES		NULL	
sekolah_universitas	varchar(30)	YES		NULL	

Gambar 2.15 Struktur Tabel Anggota

## Tabel Peminjaman

Field	Type	Null	Key	Default	Extra
id_anggota	varchar(7)	YES		NULL	
id_buku	varchar(5)	YES		NULL	
tgl_pinjam	date	YES		NULL	
tgl_kembali	date	YES		NULL	

Gambar 2.16 Struktur Tabel Peminjaman

### 2.4.4. Merubah dan Memodifikasi Tabel

Terkadang ketika kita sedang membuat tabel, ternyata ada nama field yang kurang atau nama fieldnya susah dipahami, atau bisa juga tipe data serta ukurannya kurang sesuai. Sekarang coba kita perhatikan tabel yang sudah kita buat sebelumnya. Pada tabel buku terdapat field id\_buku dengan tipe bigint ukurannya 5, sedangkan pada tabel peminjam id\_buku tipenya varchar ukurannya 5. Ini tidak sama, bisa - bisa menimbulkan masalah kedepannya. Berarti tipe data di kedua tabel ini harus disamakan supaya kedua tabel bisa dihubungkan.

Sekarang, id\_buku pada tabel peminjam diubah menjadi bertipe bigint ( 5 ), caranya: `alter table nama_table modify nama_kolom;`

Contoh:

```
MariaDB [perpustakaan]> alter table peminjaman modify id_buku bigint(5);
Query OK, 0 rows affected (1.275 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Gambar 2.17 Mengganti Tipe Data Tabel Peminjaman

Output:

Field	Type	Null	Key	Default	Extra
id_anggota	varchar(7)	YES		NULL	
id_buku	bigint(5)	YES		NULL	
tgl_pinjam	date	YES		NULL	
tgl_kembali	date	YES		NULL	

Gambar 2.18 Struktur Tabel Peminjaman

Bisa dilihat bahwa query yang kita lakukan berjalan baik, maksud dari query yang kita ketikkan yaitu kita ingin melakukan modifikasi kolom dan tipe data yang ada di tabel peminjaman. Hal yang ingin kita ubah yaitu id\_buku yang semulanya bertipe data varchar(5) ingin diubah ke bigint(5) agar bisa terhubung dengan tabel yang lain. Kalau kalian belum menemukan perbedaannya coba lihat tipe id\_buku sebelum dan sesudah diubah.



Sebelum diubah

Field	Type	Null	Key	Default	Extra
id_anggota	varchar(7)	YES		NULL	
id_buku	varchar(5)	YES		NULL	
tgl_pinjam	date	YES		NULL	
tgl_kembali	date	YES		NULL	

Gambar 2.19 Struktur Tabel Peminjaman Sebelum Diubah

Sesudah diubah

Field	Type	Null	Key	Default	Extra
id_anggota	varchar(7)	YES		NULL	
id_buku	bigint(5)	YES		NULL	
tgl_pinjam	date	YES		NULL	
tgl_kembali	date	YES		NULL	

Gambar 2.20 Struktur Tabel Peminjaman Sesudah Diubah

#### 2.4.5. Merubah Nama Field

Pada tabel buku terdapat field `id_buku` dan `index_buku`, kalau melihat namanya, maka seolah - olah fungsinya sama, kenapa ada dua field `id` ( `index` ) buku pada satu tabel. Padahal fungsi kedua field tersebut berbeda. Field `index buku` fungsinya untuk menyimpan indeks judul buku sedangkan field `id_buku` untuk menyimpan indeks untuk setiap buku yang domainnya ada di tabel judul buku. Judul buku yang sama memiliki `index_buku` yang sama tapi `id_buku`-nya berbeda. Sebetulnya namanya tetap seperti itu juga tidak ada masalah, tapi akan membingungkan kedepannya.

Untuk lebih mempermudah pemahaman hanya dengan melihat namanya saja, maka sebaiknya namanya diganti dengan `id_judul`. Bagaimana caranya ?

```
MariaDB [perpustakaan]> alter table buku change index_buku id_judul varchar(15);
Query OK, 0 rows affected (0.906 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Gambar 2.21 Sintak Kueri Mengganti Nama Kolom

Query yang kita masukkan sudah berhasil, namun apa sih maksud dari query diatas? Maksudnya yaitu kita ingin melakukan perubahan nama kolom di tabel buku, dengan nama kolom sebelumnya yaitu `index_buku` diganti dengan nama kolom yang baru yaitu `id_judul` dengan tipe data `varchar` dan value sebesar 15. Perubahannya bisa kalian lihat dari tabel sebelum dan sesudah kita rubah.



Sebelum diubah

Field	Type	Null	Key	Default	Extra
id_buku	bigint(5)	NO	PRI	NULL	auto_increment
index_buku	varchar(15)	NO			
no_urut	tinyint(3)	YES		0	
referensi	tinyint(1)	YES		NULL	

Gambar 2.22 Struktur Tabel Judul Buku Sebelum Diubah

Sesudah diubah

Field	Type	Null	Key	Default	Extra
id_buku	bigint(5)	NO	PRI	NULL	auto_increment
id_judul	varchar(15)	YES		NULL	
no_urut	tinyint(3)	YES		0	
referensi	tinyint(1)	YES		NULL	

Gambar 2.23 Struktur Tabel Judul Buku Sesudah Diubah

Selanjutnya ubahlah field `index_buku` yang ada di tabel `judul_buku` dengan `id_judul`. Ubah juga field yang berawalan dengan `kd` dengan `id`, misalnya `kd_penerbit` menjadi `id_penerbit`. Ini hanya untuk melatih cara merubah nama field serta untuk menyeragamkan penamaan saja. Selain itu, ubah juga field sekolah universitas saja.

#### 2.4.6. Menambah Field

Kalau kita perhatikan tabel penerbit, dengan tabel manakah tabel ini berhubungan? Kalau tabel pengarang berhubungan dengan tabel judul buku menggunakan field `id_pengarang`. Sehingga untuk mengetahui nama pengarang dari suatu judul buku tinggal menghubungkan tabel judul buku dengan pengarang menggunakan field kunci `id_pengarang`. Harusnya tabel penerbit diperlukan juga oleh tabel judul buku untuk mengetahui nama penerbit dari suatu judul buku. Tapi jangan khawatir, SQL punya juga bahasa yang digunakan untuk menambah atau menyisipkan field ke dalam suatu tabel:

```
MariaDB [perpustakaan]> alter table judul_buku add id_penerbit varchar(5) after judul;  
Query OK, 0 rows affected (0.205 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

Gambar 2.24 Sintak Kueri Menambah Kolom

Query diatas digunakan untuk menyisipkan field `id_penerbit` bertipe data `varchar` dengan panjang 5 ke dalam tabel judul buku dan diletakkan setelah field `judul`. Untuk membuktikan, tampilkan saja struktur tabel judul buku.

Output setelah menambahkan query diatas.

Field	Type	Null	Key	Default	Extra
id_judul	varchar(10)	NO	PRI	NULL	
judul	varchar(90)	YES		NULL	
id_penerbit	varchar(5)	YES		NULL	
id_pengarang	varchar(5)	YES		NULL	
id_kelompok	smallint(3)	YES		NULL	
edisi	tinyint(2)	YES		NULL	
thn_terbit	int(4)	YES		NULL	

Gambar 2.25 Struktur Tabel Judul Buku

#### 2.4.7. Menghapus Field

Di dalam tabel anggota, kita bisa melihat di sana terdapat field no hp dan no telp. Perlukah keduanya disimpan? Tidak semua anggota punya no hp, dan tidak pula semuanya memiliki no telpon. Karena salah satunya saja sudah cukup, maka kita akan menggunakan no telpon saja dan akan menghapus kolom no hp, untuk menghapus field no hp gunakan perintah berikut:

```
MariaDB [perpustakaan]> alter table anggota drop no_hp;
Query OK, 0 rows affected (0.164 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Gambar 2.26 Menghapus Kolom No\_Hp

Query diatas sudah berhasil kita lakukan, maksud dari perintah diatas yaitu kita ingin menghapus kolom no\_hp pada tabel anggota. Hasil query diatas dapat kita lihat pada gambar dibawah ini:

Field	Type	Null	Key	Default	Extra
id_anggota	varchar(7)	YES		NULL	
nama_anggota	varchar(50)	YES		NULL	
tgllahir	date	YES		NULL	
jkln	tinyint(1)	YES		NULL	
alamat	varchar(50)	YES		NULL	
no_telp	varchar(15)	YES		NULL	
tmp_lahir	varchar(20)	YES		NULL	
sekolah	varchar(30)	YES		NULL	

Gambar 2.27 Struktur Tabel Anggota Setelah Kolom No\_Hp Dihapus

#### 2.4.8. Merubah Nama Tabel

Tabel peminjam berfungsi untuk menyimpan data anggota yang melakukan peminjaman dan pengembalian buku. Mungkin akan lebih jelasnya kalau nama peminjam diganti dengan transaksi. Untuk melakukannya gunakan perintah:

```
MariaDB [perpustakaan]> alter table peminjaman rename transaksi;
Query OK, 0 rows affected (0.276 sec)
```

Gambar 2.28 Sintak Kueri Mengganti Nama Tabel

Query diatas sudah berhasil kita lakukan, maksud dari perintah diatas yaitu kita ingin mengubah nama tabel atau rename table. Dari yang semula bernama tabel peminjaman sekarang kita ganti namanya menjadi tabel transaksi, kalau kita panggil nama tabel peminjaman pasti akan error.

Jika kita mengakses tabel peminjaman, maka akan menampilkan error kalau tabel peminjaman sudah tidak ada.

```
MariaDB [perpustakaan]> desc peminjaman;
ERROR 1146 (42S02): Table 'perpustakaan.peminjaman' doesn't exist
MariaDB [perpustakaan]> desc transaksi;
```

Gambar 2.29 Sintak Kueri Mendeskripsikan Tabel Peminjaman

Coba kita ganti query nya menjadi tabel transaksi, maka seharusnya akan menampilkan hasil yang kita inginkan.

```
MariaDB [perpustakaan]> desc transaksi;
```

Field	Type	Null	Key	Default	Extra
id_anggota	varchar(7)	YES		NULL	
id_buku	bigint(5)	YES		NULL	
tgl_pinjam	date	YES		NULL	
tgl_kembali	date	YES		NULL	

Gambar 2.30 Struktur Tabel Transaksi

#### 2.4.9. Menghapus Tabel dan Database

Terkadang kita sudah membuat suatu tabel atau database, tapi ternyata tabel tersebutsebetulnya tidak diperlukan. Mungkin karena pembatasan masalah, ataupun mungkin karena diketahui setelahnya bahwa tabel tersebut tidaklah diperlukan, dan hanya memboroskan tempat penyimpanan saja, atau dengan tabel tersebut menjadikan tidak efektif dan efisien. Atau bisa saja tabel tersebut diciptakan hanya untuk keperluan sementara saja ( temporari ), maka kita harus atau perlu untuk menghapus tabel tersebut. Begitu juga dengan database yang sudah kita buat ternyata database tersebut sudah kita perlukan lagi. Perintah yang digunakan untuk menghapus adalah DROP

##### Menghapus tabel

Struktur perintah drop table nama\_table

Contoh:

```
MariaDB [perpustakaan]> drop table pengarang;  
Query OK, 0 rows affected (0.148 sec)
```

*Gambar 2.31 Menghapus Tabel Pengarang*

Query diatas digunakan untuk menghapus tabel pengarang, dan ketika kita ingin melihat tabel pengarang maka akan menampilkan error karena tabel pengarang sudah tidak ada.

```
MariaDB [perpustakaan]> desc pengarang;  
ERROR 1146 (42S02): Table 'perpustakaan.pengarang' doesn't exist  
MariaDB [perpustakaan]>
```

*Gambar 2.32 Mengecek Tabel Pengarang*

## Menghapus Database

Struktur perintah drop database

nama\_table

Contoh:

```
MariaDB [perpustakaan]> drop database pengarang;  
Query OK, 0 rows affected (0.002 sec)
```

*Gambar 2.33 Menghapus Database Pengarang*

Query diatas digunakan untuk menghapus database pengarang, dan ketika kita ingin melihat database yang ada di perangkat kita, maka database pengarang tidak akan ada karena sudah kita hapus. Untuk lebih jelasnya silahkan dilihat dari gambar dibawah ini.

```
+-----+  
| Database |  
+-----+  
| ibik_api_academic |  
| information_schema |  
| mysql |  
| pengarang |  
| performance_schema |  
| perpustakaan |  
| phpmyadmin |  
| test |  
+-----+
```

(1)

```
+-----+  
| Database |  
+-----+  
| ibik_api_academic |  
| information_schema |  
| mysql |  
| performance_schema |  
| perpustakaan |  
| phpmyadmin |  
| test |  
+-----+
```

(2)

*Gambar 2.34 (1) List Database Sebelum Dihapus, (2) List Database Sesudah Dihapus*

Akhirnya kita sudah belajar query dasar untuk membuat database, membuat tabel, modifikasi tabel, mengubah isi didalam tabel dan banyak yang lainnya. Di materi kedepannya kita akan belajar lebih dalam lagi tentang SQL, sebelum itu ada latihan yang harus dikerjakan untuk melatih pemahaman kalian tentang SQL.

## 2.5 Latihan

1. Ubahlah field index\_buku yang ada di tabel judul\_buku dengan id\_judul.
2. Ubah juga field yang berawalan dengan kd dengan id, misalnya kd\_penerbit menjadi id\_penerbit.