

DASAR SQL

MI2143

MODUL PRAKTIKUM



giving and caring the world

Penyusun:
Wardani Muhamad, MT
wdm@politekniktelkom.ac.id

Program Studi Manajemen Informatika
Politeknik Telkom
2013

Daftar Isi

Daftar Isi	i
1 Bab I Data Definition Language (DDL).....	1
1.1 Identitas	1
1.2 Praktik	2
1.2.1 Menciptakan Table	2
1.2.2 Memodifikasi Table (Menambahkan column).....	4
1.2.3 Memodifikasi Table (mengubah tipe data column).....	5
1.2.4 Memodifikasi Table (mengubah nama column)	6
1.2.5 Memodifikasi Table (menghapus column)	6
1.2.6 Memodifikasi Table (menambahkan reference/foreign key)	6

1 BAB I DATA DEFINITION LANGUAGE (DDL)

1.1 IDENTITAS

Kajian

Penciptaan dan pengelolaan struktur basisdata

Topik

- Data Definition Language (DDL)

Referensi

1. Oracle Database 10g. SQL Fundamental 1 – Student Guide Volume 1. California: Oracle Publisher 2004.
2. Oracle Database 10g. SQL Reference 10g Release 2. California: Oracle Publisher 2005.

Kompetensi Utama

1. Mahasiswa mampu memilih tipe data dan *length* yang tepat
2. Mahasiswa mampu membuat dan memodifikasi tabel lengkap dengan Primary Key
3. Mahasiswa mampu membuat referential integrity

Lama Kegiatan Praktikum

1. Pertemuan Terbimbing : 1 x 120 menit
2. Kegiatan Mandiri : 2 x 120 menit

Parameter Penilaian

1. Jurnal (40%) : Jurnal Pengamatan
2. Tes Akhir (60%) : SQL Statement (DDL)

1.2 PRAKTIK

1.2.1 Menciptakan Table

Dalam menciptakan (*create*) sebuah table, perhatikanlah susunan table yang disajikan pada sebuah skema relasi. Susunan sebuah table terdiri atas nama table, column, dan constraint yang mungkin terdapat pada table. Penting untuk memprediksi record (data) yang akan disimpan pada tiap table, sehingga pada saat mendefinisikan tipe data pada column akan tepat.

1.2.1.1 Soal

Perhatikan skema relasi berikut:

mahasiswa	
PK	Nim
	nama
	tempat_lahir
	tgl_lahir
	gender
	is_aktif

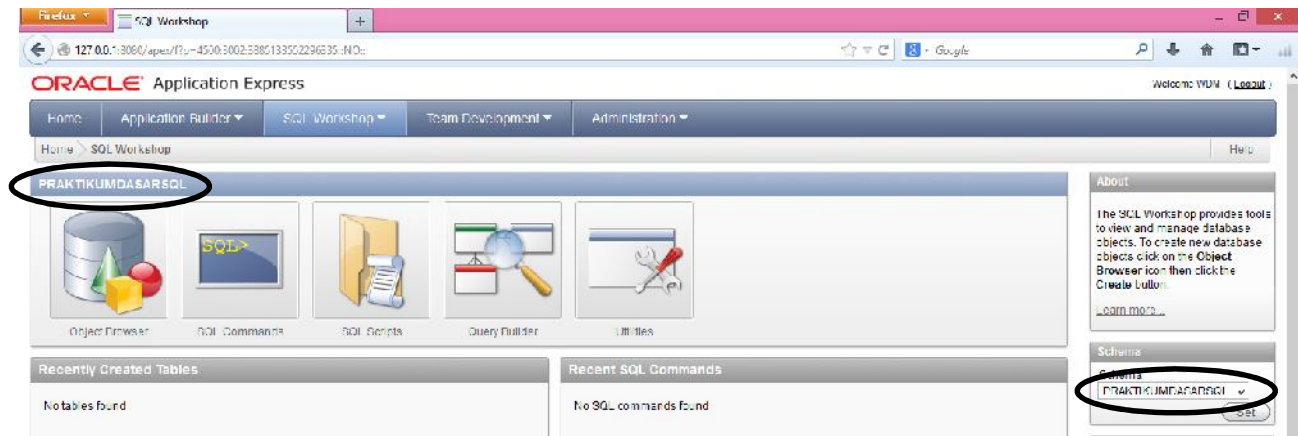
Sesuai dengan skema relasi diatas, buatlah table yang sesuai.

1.2.1.2 Langkah Penyelesaian

1. Perhatikan nama table dan column pada table. Pastikan nama yang akan digunakan memenuhi aturan penamaan table dan column.
2. Prediksi nilai yang akan tersimpan pada tiap column
 - a. "nim" akan menyimpan nomor induk mahasiswa, dengan contoh: 30109999
 - b. "nama" akan menyimpan nama lengkap mahasiswa, dengan contoh: Wardani Muhamad
 - c. "tempat_lahir" akan menyimpan nama kota kelahiran mahasiswa, dengan contoh: Jepara
 - d. "tgl_lahir" akan menyimpan tanggal lahir mahasiswa, dengan contoh: 17/08/1945
 - e. "gender" akan menyimpan jenis kelamin mahasiswa, dengan contoh: L
 - f. "is_aktif" akan menyimpan status aktif mahasiswa di semester terakhir, dengan contoh: TRUE
3. Prediksi nilai akan digunakan untuk menentukan tipe data yang tepat dan panjang karakter maksimal (*length*) yang akan tersimpan pada tiap column
4. Ciptakan (*create*) table tersebut dengan menggunakan perintah DDL: CREATE TABLE

1.2.1.3 Solusi Lengkap

Untuk menyusun table "mahasiswa", pastikan *schema* yang dipilih adalah "PRAKTIKUMDASARSQL"



Perintah DDL yang dapat digunakan untuk menyusun table “mahasiswa” adalah sebagai berikut:

```
CREATE TABLE mahasiswa (
  nim NUMBER(8),
  nama VARCHAR2(20),
  tempat_lahir VARCHAR2(20),
  tgl_lahir DATE,
  gender CHAR,
  is_aktif VARCHAR2(5),
  CONSTRAINT pk_mahasiswa PRIMARY KEY (nim)
)
```

Hint: pada Oracle 11G tidak terdapat tipe data BOOLEAN, sehingga untuk menyimpan data sejenis dapat digunakan VARCHAR2 atau NUMBER

Untuk menampilkan kembali struktur table yang sudah dibuat, dapat digunakan perintah DESC. Contoh, dengan menggunakan perintah **DESC mahasiswa** pada SQL Command, maka akan ditampilkan struktur table mahasiswa seperti berikut:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MAHASISWA	NIM	NUMBER	8		0	1			
	NAMA	VARCHAR2	20				✓		
	TEMPAT LAHIR	VARCHAR2	20				✓		
	TGL LAHIR	DATE	7				✓		
	GENDER	CHAR	1				✓		
	IS_AKTIF	VARCHAR2	5				✓		

1.2.1.4 Pengamatan

Melalui struktur table “mahasiswa” yang sudah terbentuk, prediksilah apakah nilai-nilai dibawah ini dapat tersimpan pada column yang bersesuaian? Kemudian tuliskan alasannya

No	Column	Nilai yang akan disimpan	Berhasil ? (Ya/Tidak)	Alasan
1	nama	Christiano Ronaldo		
2	nama	Ronaldo Luis Nazario de Lima		
3	nim	301103XX		
4	is_aktif	BENAR		
5	gender	LAKI-LAKI		

1.2.2 Memodifikasi Table (Menambahkan column)

Struktur table yang sudah dibuat dapat dimodifikasi sesuai dengan kebutuhan. Salah satu modifikasi yang dapat dilakukan adalah menambahkan column. Perlu diingat bahwa modifikasi hanya dapat dilakukan jika table sudah terbentuk.

1.2.2.1 Soal

Tambahkan sebuah column “daerah_asal” yang akan menyimpan kota asal / domisili mahasiswa. Nilai yang akan tersimpan pada column tersebut misalnya: Bandung.

1.2.2.2 Langkah Penyelesaian

1. Perhatikan nama table yang akan dimodifikasi. Pastikan table tersebut sudah terbentuk dengan cara menggunakan perintah DESC.
2. Prediksi nilai yang akan tersimpan pada column yang akan ditambahkan.
3. Modifikasi table dengan menggunakan perintah DDL: **ALTER TABLE ... ADD**

1.2.2.3 Solusi Lengkap

```
ALTER TABLE mahasiswa ADD (
    daerah_asal VARCHAR2 (20)
)
```

Untuk melihat perubahan struktur table setelah SQL Statement diatas dijalankan, dapat kembali menggunakan perintah DESC

Object Type: TABLE Object: MAHASISWA									
Table	Column	Data type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MAHASISWA	NIM	NUMBER	-	8	0	1	-	-	-
	NAMA	VARCHAR2	20	-	-	-	✓	-	-
	TEMPAT_LAHIR	VARCHAR2	20	-	-	-	✓	-	-
	TGL_LAHIR	DATE	7	-	-	-	✓	-	-
	STATUS	CHAR	1	-	-	-	✓	-	-
	PEKERJAAN	VARCHAR2	5	-	-	-	✓	-	-
	DAERAH ASAL	VARCHAR2	20	-	-	-	✓	-	-
									1 - 7

1.2.2.5 Pengamatan

Tentukan tipe data yang tepat pada column “daerah_asal” jika nilai yang akan disimpan pada column tersebut adalah:

No	Nilai yang akan disimpan	Tipe data	Length
6	01		
7	1		
8	JKT		

1.2.3 Memodifikasi Table (mengubah tipe data column)

Pada saat mendefinisikan tipe data dan *length* pada sebuah table, haruslah memperhatikan nilai yang akan tersimpan pada column tersebut. Akan tetapi, pada kondisi tertentu, perubahan kebutuhan akan nilai yang disimpan bisa jadi berubah setelah table sudah terbentuk.

1.2.3.1 Soal

Pada saat akan menyimpan sebuah nilai pada column “daerah_asal” ternyata ditemukan adanya masalah. Nilai “Karang Asih Jaya Sentosa” tidak dapat tersimpan ke dalam column “daerah_asal” karena panjang (*length*) karakter yang akan disimpan melebihi 20 karakter. Untuk itu, modifikasilah struktur table “mahasiswa”, khususnya pada column “daerah_asal” sehingga bisa menyimpan nilai yang panjang karakternya lebih dari 20.

1.2.3.2 Langkah Penyelesaian

Untuk mengubah tipe data dan length pada sebuah column, digunakan perintah DDL: **ALTER TABLE ... MODIFY**

```
ALTER TABLE mahasiswa MODIFY(
    daerah_asal VARCHAR2 (30)
)
```

Table	Column	Data type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MAHASISWA	NIM	NUMERIC	-	8	0	1	-	-	-
	NAMA	VARCHAR2	20	-	-	-	✓	-	-
	TEMPAT LAHIR	VARCHAR2	20	-	-	-	✓	-	-
	TGL LAHIR	DATE	7	-	-	-	✓	-	-
	GENDE	CHAR	1	-	-	-	✓	-	-
	DIKIR	VARCHAR2	5	-	-	-	✓	-	-
	DAERAH ASAL	VARCHAR2	30	-	-	-	✓	-	-

Perlu diingat bahwa perintah DDL diatas tidak digunakan untuk merubah nama column.

1.2.3.3 Pengamatan

Lakukan pengubahan tipe data pada column “daerah_asal” dengan mempertimbangkan nilai yang akan disimpan sesuai pada sub bab 1.2.2.5

No	Perintah DDL untuk memodifikasi
9	
10	
11	

1.2.4 Memodifikasi Table (mengubah nama column)

Selain mengubah tipe data dan *length* pada sebuah column, DDL juga dapat digunakan untuk mengubah nama sebuah *column*. Pengubahan nama column tidak akan mengubah tipe data dan *length* pada column tersebut.

1.2.4.1 Soal

Ubahlah nama column “daerah_asal” menjadi domisili.

1.2.4.2 Langkah Penyelesaian

Untuk mengubah nama sebuah column, digunakan perintah DDL: **ALTER TABLE ... RENAME COLUMN... TO**

```
ALTER TABLE mahasiswa RENAME COLUMN daerah_asal TO domisili
```

Object Type: TABLE Object: MAHASISWA

Table	Column	Data type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MAHASISWA	NIM	NUMBER	-	0	0	1	-	-	-
	NAMA	VARCHAR2	20	-	-	-	✓	-	-
	TEMPAT_LAHIR	VARCHAR2	20	-	-	-	✓	-	-
	TGL_LAHIR	DATE	7	-	-	-	✓	-	-
	SLIPULIR	CHAR	1	-	-	-	✓	-	-
	IS_AKHIR	VARCHAR2	5	-	-	-	✓	-	-
	DOMISILI	VARCHAR2	30	-	-	-	✓	-	-

1 - 7

1.2.5 Memodifikasi Table (menghapus column)

Selain menambahkan column pada sebuah table, modifikasi table juga memungkinkan perintah untuk menghapus column yang sudah tidak sesuai dengan kebutuhan. Penghapusan column akan mengakibatkan terhapusnya semua nilai yang tersimpan pada column tersebut.

Ketika melakukan penghapusan pada sebuah column harus memperhatikan constraint yang dikenakan pada column tersebut.

1.2.5.1 Soal

Hapuslah column “domisili”.

1.2.5.2 Langkah Penyelesaian

Untuk menghapus column, digunakan perintah DDL: **ALTER TABLE ... DROP COLUMN**

```
ALTER TABLE mahasiswa DROP COLUMN domisili
```

Object Type: TABLE Object: MAHASISWA

Table	Column	Data type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MAHASISWA	NIM	NUMBER	-	0	0	1	-	-	-
	NAMA	VARCHAR2	20	-	-	-	✓	-	-
	TEMPAT_LAHIR	VARCHAR2	20	-	-	-	✓	-	-
	TGL_LAHIR	DATE	7	-	-	-	✓	-	-
	SLIPULIR	CHAR	1	-	-	-	✓	-	-
	IS_AKHIR	VARCHAR2	5	-	-	-	✓	-	-

1 - 6

1.2.6 Memodifikasi Table (menambahkan reference/foreign key)

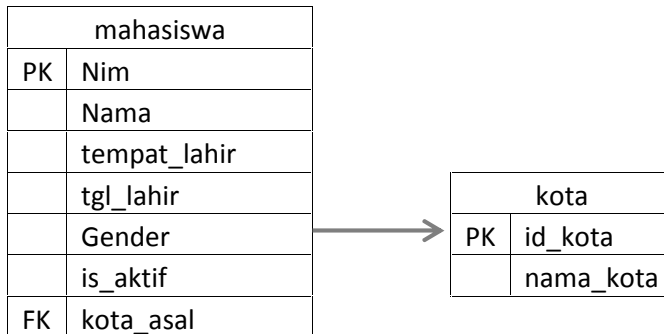
Dalam sebuah basisdata relasional terdapat keterhubungan antara table yang satu dengan table yang lain. Keterhubungan tersebut ditunjukkan dengan adanya column yang merelasikan kedua table melalui hubungan referential (Primary Key – Foreign Key).

Jika sebuah table yang seharusnya memiliki rujukan ke table lain terlanjur dibentuk tanpa mendefinisikan foreign key-nya, maka DDL dapat digunakan untuk mengubah table tersebut dan menambahkan foreign key yang sesuai.

Penting untuk memperhatikan tipe data pada column yang menjadi rujukan dan urutan penciptaan table.

1.2.6.1 Soal

Perhatikan diagram relationship berikut:



Pada diagram relationship diatas, column “kota_asal” pada table “mahasiswa” memiliki rujukan ke column “id_kota” yang terdapat pada table “kota”

1.2.6.2 Langkah Penyelesaian

Untuk menyelesaikan permasalahan (soal) diatas, dapat digunakan langkah-langkah sebagai berikut:

1. Bentuk (create) table “kota”
2. Modifikasi table “mahasiswa” dengan menambahkan column “kota_asal” yang memiliki tipe data sama seperti tipe data pada column rujukan sekaligus set column tersebut memiliki constraint foreign key

1.2.6.3 Solusi Lengkap

Untuk membentuk table “kota”, dapat menggunakan perintah DDL sebagai berikut:

```
CREATE TABLE kota(
    id_kota number(8),
    nama_kota VARCHAR2(20),
    CONSTRAINT pk_kota PRIMARY KEY (id_kota)
)
```

Object Type: TABLE Object: KOTA									
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
KOTA	ID_KOTA	NUMBER	-	8	-	1	-	-	-
	NAMA_KOTA	VARCHAR2	20	-	-	-	✓	-	-
						1 2			

Untuk memodifikasi table “mahasiswa” dengan menambahkan column “kota_asal” sekaligus menunjuk column tersebut menjadi *foreign key* dapat menggunakan perintah DDL sebagai berikut:

```
ALTER TABLE mahasiswa ADD kota_asal number(8);
ALTER TABLE mahasiswa ADD CONSTRAINT fk_mahasiswa_kota FOREIGN KEY
(kota_asal) REFERENCES kota(id_kota);
```

Object Type: TABLE Object: MAHASISWA

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MAHASISWA	NIM	NUMBER	-	8	0	1	-	-	-
	NAMA	VARCHAR2	20				✓	-	-
	EMAIL	VARCHAR2	20				✓	-	-
	ISL_EMAIL	DATE	1				✓	-	-
	GENDER	CHAR	1				✓	-	-
	IS_AKTIF	VARCHAR2	5				✓	-	-
	KOTA_ASL	NUMBER	8		0		✓	-	-

1-7

1.2.6.4 Pengamatan

Lakukan pengubahan tipe data pada column “kota_asal”, kemudian tuliskan penjelasan hasil pengamatan tersebut.

No	Tipe data baru	Penjelasan hasil pengamatan
12	number (9)	
13	number (8,2)	
14	varchar2(8)	
15	number (7)	