

**LAPORAN PRAKTIKUM  
BASIS DATA**

**MODUL VII  
RELASI TABEL**



Disusun Oleh:

Septiandi Nugraha

21104060

SE05-B

**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK  
FAKULTAS INFORMATIKA  
INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

## **I. TUJUAN**

1. Mahasiswa dapat membuat relasi tabel
2. Mahasiswa mampu membuktikan integritas dengan tabel yang saling berhubungan (relasi)

## **II. DASAR TEORI**

Relasi Database dan RDMS, Relational Database sebenarnya adalah salah satu konsep penyimpanan data, sebelum konsep database relasional muncul sebenarnya sudah ada dua model database yaitu Network Database dan Hierarchie Database. Dalam database relasional, data disimpan dalam bentuk relasi atau tabel dua dimensi, dan antar tabel satu dengan tabel lainnya terdapat hubungan atau relationship, sehingga sering kita baca di berbagai literatur, database didefinisikan sebagai “ Kumpulan dari sejumlah tabel yang saling hubungan atau keterkaitan”. Nah, kumpulan dari data yang diorganisasikan sebagai tabel tadi disimpan dalam bentuk data elektronik di dalam hardisk komputer. Untuk membuat struktur tabel, mengisi data ke tabel, mengubah data jika diperlukan dan menghapus data dari tabel diperlukan software.

Software yang digunakan membuat tabel, isi data, ubah data dan hapus data disebut Relational Database Management System atau dikenal dengan singkatan DBMS sedangkan perintah yang digunakan untuk membuat tabel, isi, ubah dan hapus data disebut perintah SQL yang merupakan singkatan dari Structure Query Language. Jadi, setiap software RDMS pasti bisa digunakan untuk menjalankan perintah SQL.

Sebenarnya fungsi RDMS bukan cuman buat tabel, isi data, ubah dan hapus data, untuk manajemen data dalam skala besar dan agar bisa mendukung proses bisnis yang kontinyu dan real time. Suatu RDMS dituntut untuk mempunyai kemampuan manajemen user dan keamanan data, backup dan recovery data serta kemampuan lainnya yang berkaitan dengan kecepatan pemrosesan data (performance).

Primary key adalah suatu field atau kombinasi field yang secara unik mengidentifikasi setiap record di dalam tabel. Contoh NIM yang terdapat pada tabel mahasiswa.

Foreign key adalah primary key pada suatu tabel yang dimasukkan pada tabel lain dan dijadikan salah satu key pada tabel tersebut.

References menunjukkan bahwa foreign key di suatu tabel merupakan key pada tabel lain. Pada perubahan suatu record baik itu update atau delete, dapat diikuti dengan kata : cascade, restrict atau set null. Kita bisa atur sesuai kebutuhan.

Contoh :

- on update CASCADE on delete RESTRICT
- on update RESTRICT on delete SET NULL

### III. PRAKTIKUM

1. Buat sebuah database dengan nama akademik!

Input:

```
Execute  
CREATE DATABASE db_akademik21104060;
```

Output:

```
CREATE DATABASE db_akademik21104060
```

Free 1

Input to filter result

```
CREATE DATABASE db_akademik21104060
```

AffectedRows : 1

2. Buat tabel Mahasiswa dengan tipe innodb

Input:

```
Execute  
CREATE TABLE mahasiswa (  
    nim VARCHAR(20) NOT NULL,  
    nama VARCHAR(25),  
    PRIMARY KEY(nim)  
);  
Execute  
DESC mahasiswa;
```

Output:

```
CREATE TABLE mahasiswa (
  nim VARCHAR(20) NOT NULL,
  nama VARCHAR(25),
  PRIMARY KEY(nim)
)
```

Cost: 5s676ms

CREATE TABLE mahasiswa ( nim VARCHAR(20) NOT NULL, nama VARCHAR(25), PRIMARY KEY(nim) )

AffectedRows : 0

		Field varchar	Type blob	Null varchar	Key varchar	Default text	Extra varchar
1		nim	varchar(20)	NO	PRI	(NULL)	
2		nama	varchar(25)	YES		(NULL)	

3. Buat tabel dosen pembimbing dengan tipe innodb dan relasikan ke tabel mahasiswa dengan mengambil key NIM dari tabel mahasiswa sebagai foreign key di tabel dosen\_pembimbing

Input:

```
CREATE TABLE dosen_pembimbing (
  idx INT AUTO_INCREMENT PRIMARY KEY,
  nim VARCHAR(20),
  nama_dosen VARCHAR(255),
  FOREIGN KEY (nim) REFERENCE mahasiswa(nim))ENGINE=InnoDB;

DESC dosen_pembimbing;
```

Output:

Cost: 567ms

CREATE TABLE dosen\_pembimbing ( idx INT AUTO\_INCREMENT PRIMARY KEY, nim VARCHAR(20), nama\_dosen VARCHAR(255) )

AffectedRows : 0

		Field varchar	Type blob	Null varchar	Key varchar	Default text	Extra varchar
1		idx	int(11)	NO	PRI	(NULL)	auto_increment
2		nim	varchar(20)	YES		(NULL)	
3		nama_dosen	varchar(255)	YES		(NULL)	

4. Isi data di tabel seperti data berikut :

- Tabel Mahasiswa

Input:

```

Execute
INSERT INTO mahasiswa VALUES
('001', 'Ferry Gustiwa'),
('002', 'Zarkasih'),
('003', 'Aris'),
('004', 'Obet');

Execute
SELECT * FROM mahasiswa;

```

Output:

```

INSERT INTO mahasiswa VALUES
('001', 'Ferry Gustiwa'),
('002', 'Zarkasih'),
('003', 'Aris'),
('004', 'Obet')

```

Cost: 194ms < >

INSERT INTO mahasiswa VALUES ('001', 'Ferry Gustiwa'), ('002', 'Zarkasih'), ('003', 'Aris'), ('004', 'Obet')

AffectedRows : 4

<

- Tabel dosen\_pembimbing

Input:

```

Execute
INSERT INTO dosen_pembimbing VALUES
(NULL, 'D001', 'Deris', '001'),
(NULL, 'D002', 'Dian', '002'),
(NULL, 'D003', 'Deris', '003');

Execute
SELECT * FROM dosen_pembimbing;

```

		* idx int	kode_dosen varchar(5)	nama_dosen varchar(20)	nim varchar(20)
	1	1	D001	Deris	001
	2	2	D002	Dian	002
	3	3	D003	Deris	003

Output:

Note : Untuk mengisi data yang fieldnya AUTO\_INCREMENT maka sebutkan NULL untuk field tersebut

5. Kemudian update satu record di tabel mahasiswa :

Input:

```

Execute
UPDATE mahasiswa
SET nama = 'Bonar' WHERE nim = '003';

Execute
SELECT * FROM mahasiswa;

```

Output:

		* nim varchar(20)	nama varchar(25)
	1	001	Ferry Gustiawa
	2	002	Zarkasih
	3	003	Bonar
	4	004	Obet

6. Hapus sebuah record di tabel mahasiswa :

Input:

```

Execute
DELETE FROM mahasiswa WHERE nim='004';

Execute
SELECT * FROM mahasiswa;

```

Output:

		* nim varchar(20)		nama varchar(25)	
	1	001		Ferry Gustiawa	
	2	002	+	Zarkasih	
	3	003		Bonar	

#### IV. EVALUASI DAN PERTANYAAN

Tuliskan perintah -perintah SQL percobaan di atas beserta outputnya.

1. Tulis semua perintah - perintah SQL percobaan di atas beserta outputnya!
2. Berikan Kesimpulan Anda

Kesimpulan:

Kesimpulan yang bisa diambil adalah bahwa relasi tabel yaitu suatu penyimpanan data yang memiliki fitur dapat diubah dan dihapus dengan menggunakan innodb, yang dimana berguna untuk menjaga keamanan data.