

# 6

## Bekerja Dengan Tabel

# Table

- Digunakan untuk menyimpan data.
- Terdiri dari kolom dan baris.
- Setiap kolom mendefinisikan data yang mempunyai tipe khusus. Kolom sering disebut juga sebagai field.
- Setiap baris berisi sebuah record/row.
- Dalam sebuah database biasanya berisi satu atau lebih table.

# Pembuatan Tabel

- Pembuatan tabel dibuat dengan menggunakan perintah CREATE TABLE
- Format Pembuatan Tabel adalah :

```
CREATE TABLE nama_tabel(  
    kolom1  tippedata(ukuran) Option_Kolom,  
    kolom2  tippedata(ukuran) Option_Kolom,  
    kolom3  tippedata(ukuran) ,  
    ... ,  
    kolomN  tippedata(ukuran)  
) ENGINE=NamaEngine;
```

## Pembuatan Tabel

- Option\_Kolom tidak wajib disertakan.
- Option\_Kolom yang diperbolehkan diantaranya :
  - NULL / NOT NULL : menentukan apakah kolom boleh kosong atau tidak.
  - PRIMARY KEY : menentukan apakah kolom adalah kolom primer key
  - AUTO\_INCREMENT : menentukan apakah kolom ini akan menjadi kolom auto increment.
  - UNIQUE : menentukan apakah kolom akan berisi data yang unik (tidak boleh ada yang sama)
  - DEFAULT nilai\_default : menentukan nilai yang akan diisikan secara otomatis jika kolom tersebut tidak diberi nilai secara eksplisit.
  - COMMENT 'komentar/keterangan' : memberikan keterangan mengenai kolom.

## Contoh Pembuatan Tabel

```
CREATE TABLE mhs (  
    nim char(8) NOT NULL PRIMARY KEY,  
    nama varchar(50) NOT NULL,  
    email varchar(100) UNIQUE,  
    tgllhr DATE COMMENT 'Tanggal Lahir',  
    kdjur varchar(2) DEFAULT 'IF'  
);
```

## Contoh Pembuatan Tabel

```
CREATE TABLE bukutamu (
    no_urut int AUTO_INCREMENT PRIMARY KEY,
    nama varchar(50) NOT NULL,
    email varchar(100) UNIQUE
) ENGINE=MyISAM;
```

- Secara default Engine akan berisi InnoDB jika database diset untuk mendukung transaction. Jika ingin membuat tabel dengan Engine yang bukan default, maka nama Engine harus disebutkan.
- Untuk melihat Engine yang dapat digunakan, gunakan perintah "SHOW ENGINES"
- Fitur-fitur setiap engine dapat dilihat pada MySQL Manual pada "Contents → Storage Engines".

# Contoh Perbandingan Storage Engine (MyISAM dan InnoDB)

Fitur	MyISAM	InnoDB
Storage limits	256TB	64TB
Transactions	Tidak	Ya
Locking granularity	Table	Row
Foreign key support	Tidak	Ya
Full-text search indexes	Ya	Tidak

## Melihat Definisi Tabel

- Untuk melihat struktur pendefinisian tabel, gunakan perintah “DESC nama\_tabel”
- Contoh :

```
mysql> DESC mhs;
```

Field	Type	Null	Key	Default	Extra
nim	char(8)	NO	PRI		
nama	varchar(50)	NO			
email	varchar(100)	YES	UNI	NULL	
tgllhr	date	YES		NULL	
kdjur	varchar(2)	YES		IF	

```
5 rows in set (0.10 sec)
```

# Melihat SQL Pembuatan Tabel

- Jika ingin melihat DDL pembuatan tabel, gunakan perintah “SHOW CREATE TABLE namatabel”

```
mysql> SHOW CREATE TABLE mhs\G
***** 1. row *****
      Table: mhs
Create Table: CREATE TABLE `mhs` (
  `nim` char(8) NOT NULL,
  `nama` varchar(50) NOT NULL,
  `email` varchar(100) default NULL,
  `tgllhr` date default NULL COMMENT 'Tanggal Lahir',
  `kdjur` varchar(2) default 'IF',
  PRIMARY KEY (`nim`),
  UNIQUE KEY `email` (`email`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
1 row in set (0.00 sec)
```

# Menghapus Tabel

- Penghapusan tabel dapat dilakukan dengan perintah “DROP TABLE [IF EXISTS] nama\_tabel”.
- IF EXISTS dapat digunakan untuk menghindari terjadinya error ketika tabel yang akan dihapus tidak ada.
- Jika sebuah tabel dihapus, maka tabel tidak bisa dikembalikan lagi.
- Contoh :

```
mysql> DROP TABLE bukutamu;  
Query OK, 0 rows affected (0.05 sec)
```

```
mysql> DROP TABLE IF EXISTS bukutamu;  
Query OK, 0 rows affected, 1 warning (0.00 sec)
```

# Mengganti Nama Tabel

- Mengganti nama tabel dapat dilakukan dengan menggunakan perintah “**RENAME TABLE** tabel\_lama TO tabel\_baru”.

```
mysql> SHOW TABLES;
```

```
+-----+
| Tables_in_test |
+-----+
| mhs             |
+-----+
1 row in set (0.00 sec)
```

```
mysql> RENAME TABLE mhs TO mahasiswa;
```

```
Query OK, 0 rows affected (0.11 sec)
```

```
mysql> SHOW TABLES;
```

```
+-----+
| Tables_in_test |
+-----+
| mahasiswa      |
+-----+
1 row in set (0.00 sec)
```

## Mengubah Struktur Tabel

- Mengubah tabel dapat dilakukan dengan perintah “ALTER TABLE nama\_table spesifikasi\_alter”.
- Spesifikasi\_alter yang diperbolehkan diantaranya adalah :
  - ADD COLUMN : menambah kolom baru
  - ADD PRIMARY KEY : menambah primary key
  - CHANGE COLUMN : mengganti nama kolom
  - MODIFY COLUMN : mengubah definisi kolom
  - DROP COLUMN : menghapus kolom
  - DROP PRIMARY KEY : menghapus primary key
- Keterangan lengkap mengenai mengubah struktur tabel dapat dilihat di MySQL Manual → Index → ALTER TABLE.

# Mengubah Struktur Tabel

- Contoh :

```
mysql> CREATE TABLE CONTOH(
    ->   f1 int,
    ->   f2 int
    -> );
```

Query OK, 0 rows affected (0.11 sec)

```
mysql> DESC contoh;
```

Field	Type	Null	Key	Default	Extra
f1	int(11)	YES		NULL	
f2	int(11)	YES		NULL	

2 rows in set (0.01 sec)

# Mengubah Struktur Tabel (Menambah Kolom Baru)

- Perintah untuk menambah kolom adalah “ADD COLUMN nama\_kolom typedata [FIRST|AFTER nama\_kolom]”

```
mysql> ALTER TABLE contoh ADD COLUMN f3 float;
Query OK, 0 rows affected (0.23 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> DESC contoh;
```

Field	Type	Null	Key	Default	Extra
f1	int(11)	YES		NULL	
f2	int(11)	YES		NULL	
f3	float	YES		NULL	

```
3 rows in set (0.01 sec)
```

# Mengubah Struktur Tabel (Menambah Kolom Baru)

```
mysql> ALTER TABLE contoh ADD COLUMN f0 INT FIRST;
Query OK, 0 rows affected (0.24 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> DESC contoh;
```

Field	Type	Null	Key	Default	Extra
f0	int(11)	YES		NULL	
f1	int(11)	YES		NULL	
f2	int(11)	YES		NULL	
f3	float	YES		NULL	

```
4 rows in set (0.01 sec)
```

# Mengubah Struktur Tabel (Menambah Kolom Baru)

```
mysql> ALTER TABLE contoh ADD COLUMN f11 float AFTER f1;
Query OK, 0 rows affected (0.25 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> DESC contoh;
```

Field	Type	Null	Key	Default	Extra
f0	int(11)	YES		NULL	
f1	int(11)	YES		NULL	
f11	float	YES		NULL	
f2	int(11)	YES		NULL	
f3	float	YES		NULL	

5 rows in set (0.01 sec)

# Mengubah Struktur Tabel (Menambah Primary Key)

- Perintah untuk menambah primary key adalah “ADD PRIMARY KEY(nama\_kolom\_PK)”

```
mysql> ALTER TABLE contoh ADD PRIMARY KEY(f0);
Query OK, 0 rows affected (0.20 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> DESC contoh;
```

Field	Type	Null	Key	Default	Extra
f0	int(11)	NO	PRI	0	
f1	int(11)	YES		NULL	
f11	float	YES		NULL	
f2	int(11)	YES		NULL	
f3	float	YES		NULL	

```
5 rows in set (0.01 sec)
```

# Mengubah Struktur Tabel (Mengganti Kolom)

- Perintah untuk mengganti kolom adalah “CHANGE COLUMN kolom\_lama kolom\_baru definisi\_kolom”

```
mysql> ALTER TABLE contoh CHANGE COLUMN f0 id int AUTO_INCREMENT;
Query OK, 0 rows affected (0.20 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> DESC contoh;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
f1	int(11)	YES		NULL	
f11	float	YES		NULL	
f2	int(11)	YES		NULL	
f3	float	YES		NULL	

```
5 rows in set (0.01 sec)
```

# Mengubah Struktur Tabel (Mengubah Definisi Kolom)

- Perintah untuk mengubah definisi kolom adalah “MODIFY COLUMN nama\_kolom definisi\_kolom”

```
mysql> ALTER TABLE contoh MODIFY COLUMN f11 varchar(20) AFTER f3,
->                                MODIFY COLUMN f3 DECIMAL(10,2);
Query OK, 0 rows affected (0.21 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> DESC contoh;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
f1	int(11)	YES		NULL	
f2	int(11)	YES		NULL	
f3	decimal(10,2)	YES		NULL	
f11	varchar(20)	YES		NULL	

5 rows in set (0.01 sec)

# Mengubah Struktur Tabel (Menghapus Kolom)

- Perintah untuk menghapus kolom adalah “DROP COLUMN nama\_kolom”

```
mysql> ALTER TABLE contoh DROP COLUMN f2;
Query OK, 0 rows affected (0.17 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> DESC contoh;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
f1	int(11)	YES		NULL	
f3	decimal(10,2)	YES		NULL	
f11	varchar(20)	YES		NULL	

4 rows in set (0.01 sec)

# Mengubah Struktur Tabel (Menghapus Primary Key)

- Perintah untuk menghapus primary key adalah “DROP PRIMARY KEY”
- Jika kolom primary key memiliki auto\_increment, maka auto\_increment harus dihapus.

```
mysql> ALTER TABLE contoh MODIFY id int, DROP PRIMARY KEY;
Query OK, 0 rows affected (0.21 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> DESC contoh;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	YES		NULL	
f1	int(11)	YES		NULL	
f3	decimal(10,2)	YES		NULL	
f11	varchar(20)	YES		NULL	

# Membuat Tabel Dari Hasil Select

```
CREATE TABLE nama_tabel AS  
SELECT {*|NamaKolom1,NamaKolom2} FROM table1  
WHERE kondisi;
```

```
CREATE TABLE negara AS  
SELECT * FROM world.country
```

- Membuat tabel dengan perintah di atas hanya akan membuat tabel dengan struktur yang sama, tetapi tidak ikut mengcopy Primary Key atau Constraint.

# Membuat Tabel Dari Hasil Select

```
CREATE TABLE negara AS  
SELECT * FROM world.country
```

- Membuat tabel “negara” berdasarkan SELECT semua data (baris dan kolom) dari tabel Country yang berada di database “World”.

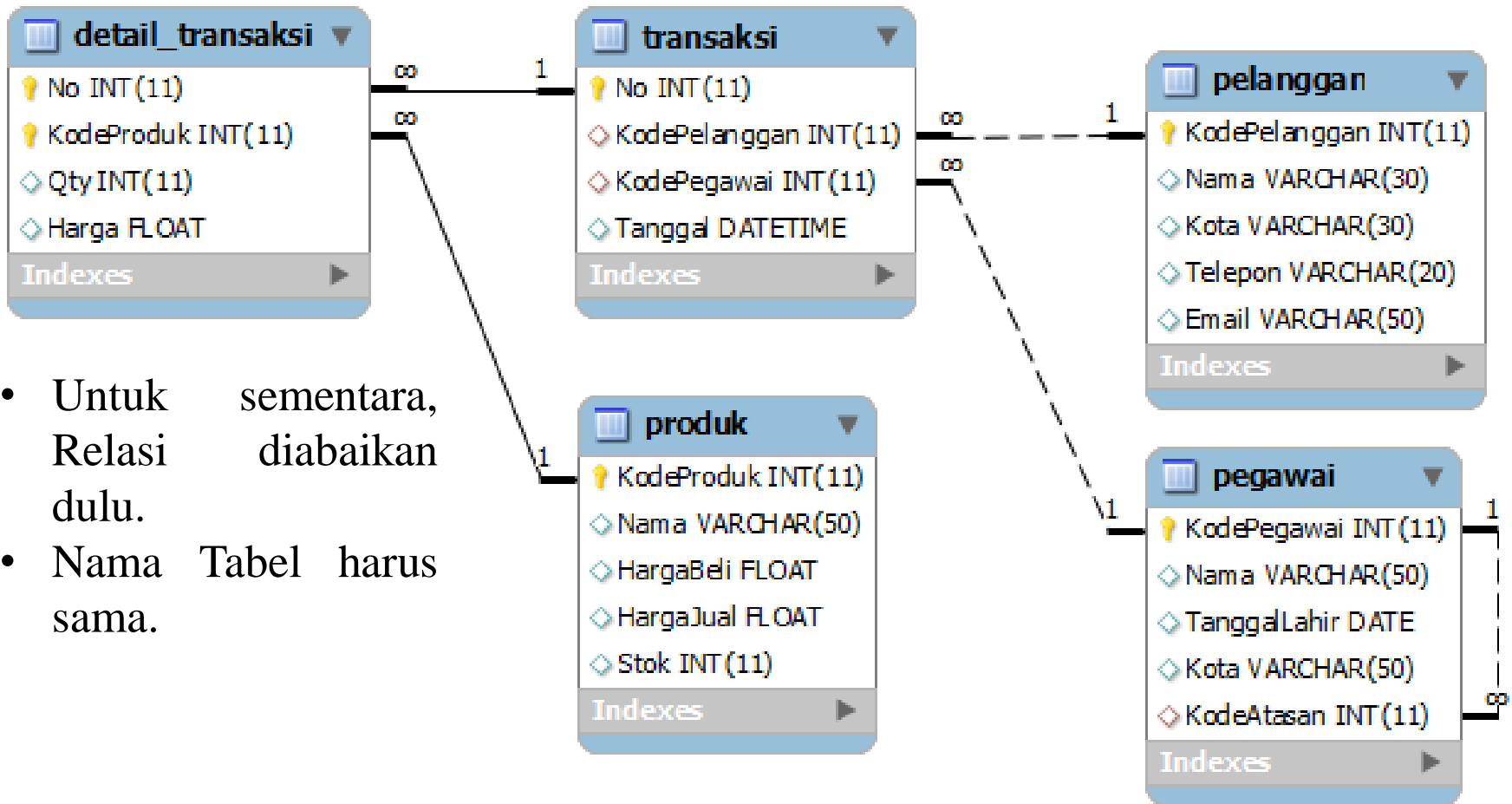
# Membuat Tabel Dari Hasil Select

```
CREATE TABLE negara_asia AS  
SELECT Code,Name FROM world.country  
WHERE Continent='Asia'
```

- Membuat tabel “negara\_asia” dari hasil SELECT yang hanya mengambil kolom Code dan Name dari tabel country yang negaranya berada di benua “Asia”

## LATIHAN

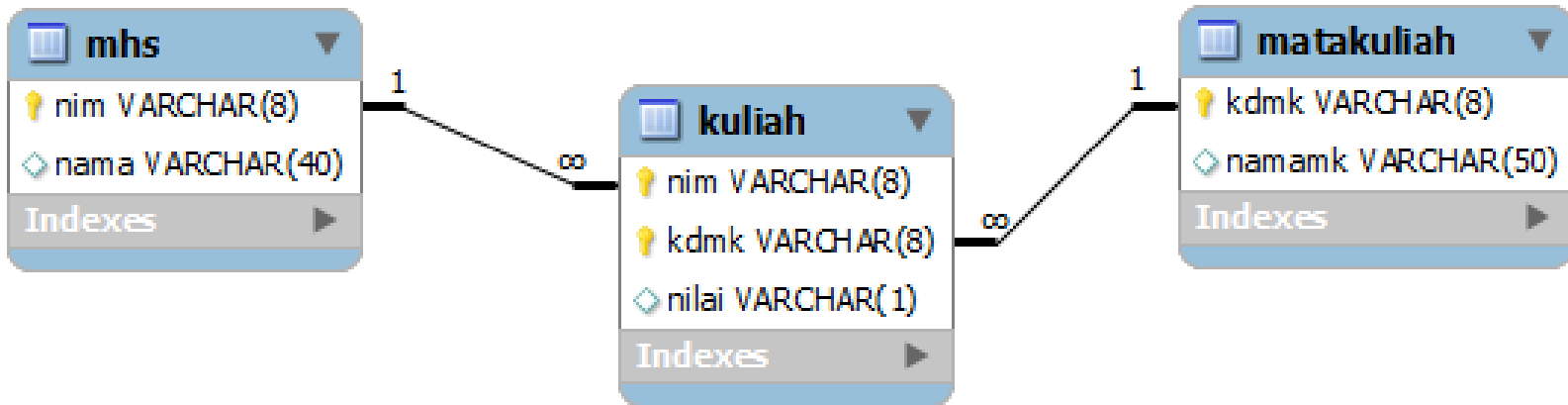
### BUAT TABEL SEPERTI SKEMA RELASI BERIKUT



- Untuk sementara, Relasi diabaikan dulu.
- Nama Tabel harus sama.

# Foreign Key

- Perhatikan skema Relasi di bawah ini.

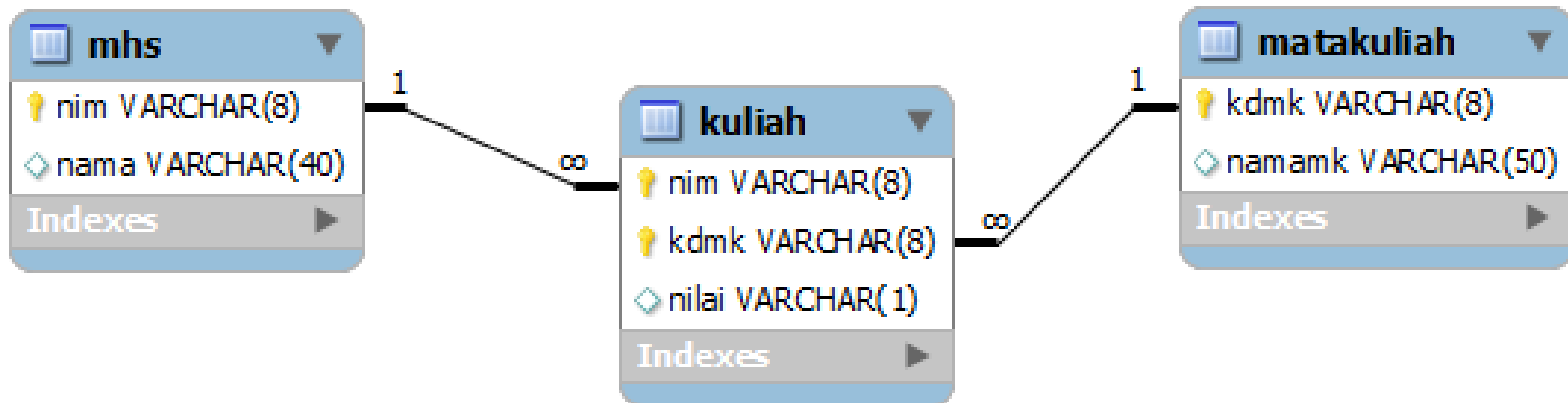


- Ada 2 relasi yang terdapat di gambar tersebut yaitu :
  - Relasi tabel kuliah dengan mahasiswa berdasarkan nim (Tabel Mahasiswa direference oleh Tabel Kuliah berdasarkan NIM)
  - Relasi tabel kuliah dengan matakuliah berdasarkan kdmk (Tabel Matakuliah direference oleh Tabel Kuliah berdasarkan kdmk)

## Kemampuan Foreign Key

- Digunakan untuk mengimplementasikan konsep Relasi dalam database
- Digunakan untuk penggunaan Referential Integrity. Dengan ini, data antara tabel akan konsisten (baik ketika ada pengupdatean atau penghapusan).

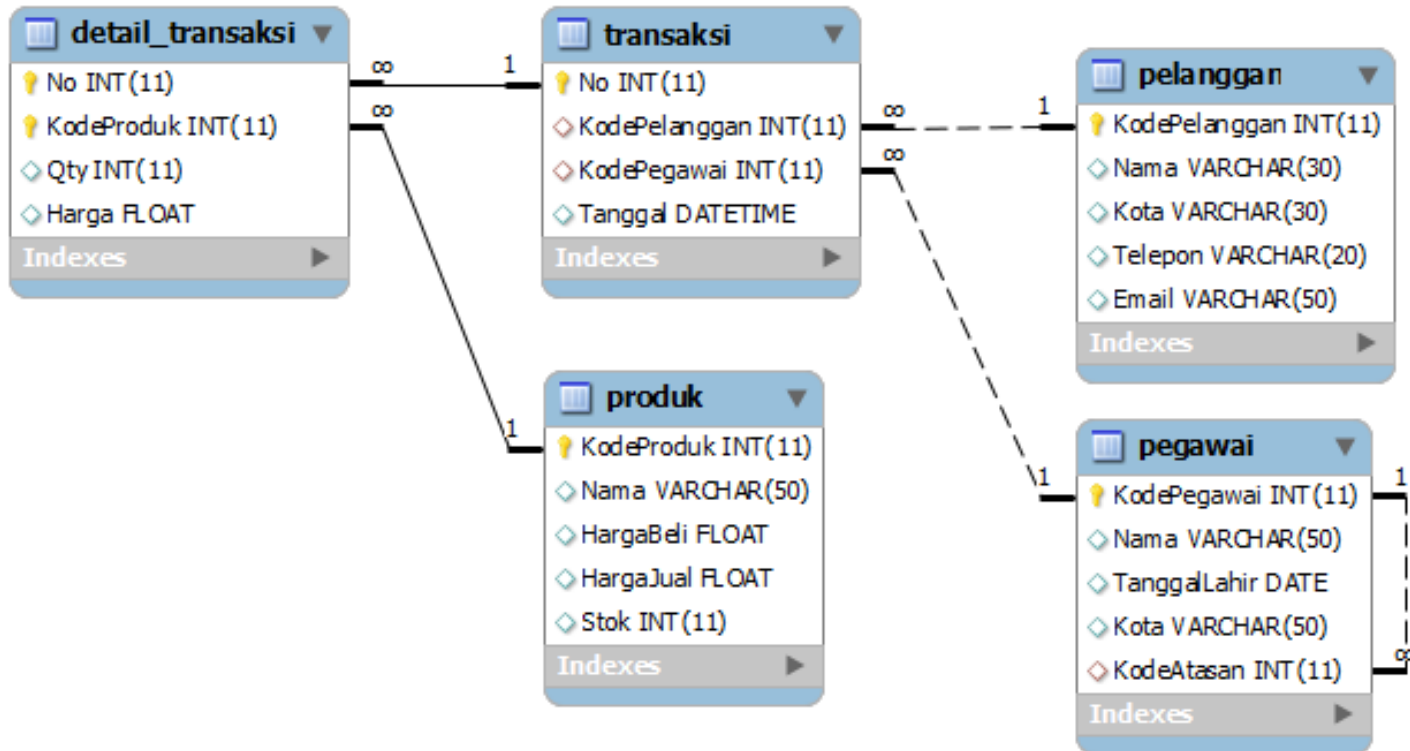
# Contoh Kasus Referential Integrity



- Dampak foreign key pada relasi kuliah → mhs
  - NIM di tabel Kuliah harus sudah terdaftar di mhs
  - Apa yang terjadi ketika ada data di tabel Mhs dihapus/diupdate?
    - Apakah data di tabel Kuliah juga akan dihapus/diupdate
    - Apakah penghapusan /pengupdaten tidak boleh dilaksanakan/dibatalkan?

# Membuat Foreign Key

- Buatlah Foreign Key sesuai Skema Relasi berikut :



## Membuat Foreign Key

- Relasi / Foreign Key yang harus dibuat
  1. Transaksi (KodePelanggan) → Pelanggan (KodePelanggan)
  2. Transaksi (KodePegawai) → Pegawai (KodePegawai)
  3. Pegawai (KodeAtasan) → Pegawai (KodePegawai)
  4. DetailTransaksi (No) → Transaksi (No)
  5. DetailTransaksi (KodeProduk) → Produk (KodeProduk)

# Membuat Foreign Key

Ada 2 cara pembuatan Foreign key

- Bersamaan dengan pembuatan Tabel
- Setelah tabel dibuat

# Membuat Foreign Key

- Bersamaan dengan membuat Tabel

```
CREATE TABLE nama_tabel(
    kolom1 tippedata(ukuran) Option_Kolom,
    kolom2 tippedata(ukuran) Option_Kolom,
    kolom3 tippedata(ukuran) ,
    . . . ,
    [CONSTRAINT nama_FK]
    FOREIGN KEY (KolomReferensi)
        REFERENCES NamaTabelYgDireference(KolomYgDireference)
        [ON DELETE {CASCADE | NO ACTION | RESTRICT | SET NULL}]
        [ON UPDATE {CASCADE | NO ACTION | RESTRICT | SET NULL}]

) ENGINE=NamaEngine;
```

# Membuat Foreign Key

- Setelah tabel dibuat

```
ALTER TABLE nama_tabel
  ADD [CONSTRAINT nama_FK]
  FOREIGN KEY (KolomReferensi)
    REFERENCES NamaTabelYgDireference (KolomYgDireference)
    [ON DELETE {CASCADE | NO ACTION | RESTRICT | SET NULL}]
    [ON UPDATE {CASCADE | NO ACTION | RESTRICT | SET NULL}]
```

# Membuat Foreign Key

Keterangan :

1. Nama\_FK : Nama Constraints/FK. Jika tidak diberi, maka akan diberi secara otomatis.
2. KolomReferensi : Kolom yang mereference (kolom di tabel child)
3. NamaTabelYgDireference : Nama Tabel yang direference (parent)
4. KolomYgDireference : Nama Kolom di tabel parent yang direference oleh KolomReference.
5. ON DELETE : Aksi apa yang akan dilakukan ketika terjadi penghapusan baris di tabel yang direference. Opsi yang bisa dipilih adalah :
  - CASCADE : Baris di tabel child ikut terhapus.
  - NO ACTION / RESTRICT : Jika memiliki baris di table child, maka penghapusan dibatalkan.
  - SET NULL : Baris di table child di-null-kan.
- ON UPDATE : Aksi apa yang akan dilakukan ketika terjadi update baris di table yang direference. Opsinya sama dengan opsi ON DELETE.

## Membuat Foreign Key

Transaksi (KodePelanggan) → Pelanggan (KodePelanggan)

```
ALTER TABLE transaksi
  ADD CONSTRAINT FK_transaksi_pelanggan
    FOREIGN KEY (KodePelanggan)
      REFERENCES pelanggan(KodePelanggan) ;
```

# Membuat Foreign Key

Transaksi (KodePegawai) → Pegawai (KodePegawai)

```
ALTER TABLE transaksi  
  ADD CONSTRAINT FK_transaksi_pegawai  
    FOREIGN KEY (KodePegawai)  
    REFERENCES pegawai (KodePegawai) ;
```

# Membuat Foreign Key

Pegawai (KodeAtasan) → Pegawai (KodePegawai)

```
ALTER TABLE pegawai  
  ADD CONSTRAINT FK_pegawai_atasan  
    FOREIGN KEY (KodeAtasan)  
    REFERENCES pegawai (KodePegawai) ;
```

# Membuat Foreign Key

DetailTransaksi (No) → Transaksi (No)

```
ALTER TABLE detail_transaksi  
  ADD CONSTRAINT FK_detail_transaksi_transaksi  
    FOREIGN KEY (No)  
    REFERENCES transaksi (No) ;
```

## Membuat Foreign Key

DetailTransaksi (KodeProduk) → Produk (KodeProduk)

```
ALTER TABLE detail_transaksi  
  ADD CONSTRAINT FK_detail_transaksi_produk  
    FOREIGN KEY (KodeProduk)  
    REFERENCES Produk (KodeProduk) ;
```