

*Modul Praktikum*  
**BASIS DATA**

Menggunakan DBMS MySQL 5.0



SEKOLAH TINGGI INFORMATIKA & KOMPUTER INDONESIA  
2017

# Kegiatan Praktikum

## Mata Kuliah

Praktikum Basis Data

## PraSyarat dan KoSyarat

PraSyarat: -

KoSyarat: Basis Data

## Standar Kompetensi

Setelah menyelesaikan perkuliahan ini mahasiswa akan dapat membangun database (DDL & DML) dengan menggunakan MySQL.

## Bobot SKS, Semester

1 SKS, Semester 2

## Program Studi

Teknik Informatika dan Manajemen Informatika

## Jumlah Pertemuan

8 kali reguler, 2 kali responsi.

## Rencana Pertemuan

Pertemuan 1 – Modul 1

Pertemuan 2 – Modul 2

Pertemuan 3 – Modul 3

Pertemuan 4 – Modul 4

Pertemuan 5 – Responsi 1 (UTS)

Pertemuan 6 – Modul 5

Pertemuan 7 – Modul 6

Pertemuan 8 – Modul 7

Pertemuan 9 – Modul 8

Pertemuan 10 – Responsi 2 (UAS)

## Alokasi Waktu per Pertemuan

150 menit.

# Kata Pengantar

Perkembangan basis data yang sedemikian pesatnya telah merambah berbagai bidang pemrograman. Mulai dari *website content managemen system* yang pada halamannya menampilkan isi dari basis data, *software desktop client server* sampai dengan software pada *platform mobile* semua telah menggunakan basis data. Untuk itu kemampuan menguasai dasar-dasar basis data telah menjadi sebuah kemampuan yang harus dikuasai oleh setiap mahasiswa teknik informatika.

Salah satu basis data yang cukup terkenal karena kehandalannya dalam mengelola data serta integrasinya dengan berbagai sistem operasi adalah MySQL. Basis data MySQL berdiri di atas lisensi General Public License yang artinya dapat digunakan oleh siapapun. Nama untuk perusahaan yang membutuhkan dukungan dari pengembang MySQL, terdapat Commercial License yang akan memberikan dukungan penuh untuk pengguna produknya.

Pada awalnya MySQL dikembangkan dibawah bendera Sun Microsystem. Namun pada bulan Januari 2008, Oracle membeli Sun Microsystem termasuk didalamnya MySQL sebagai software basis data gratis yang paling populer. Maka sampai saat ini MySQL dimiliki oleh perusahaan Oracle yang merupakan salah satu raksasa di bidang basis data.

MySQL merupakan salah satu *Relation Database Managemen System* yang memberikan fasilitas sangat kompleks kepada penggunaanya secara gratis. Ditambah lagi MySQL dapat digunakan di hampir semua sistem operasi yang ada (*server*) membuatnya menjadi primadona di dunia basis data.

Pada modul ini penyusun mengajak praktikan memasuki dunia basis data dengan menggunakan software MySQL. Diharapkan setelah mengerti dasar – dasar basis data, praktikan dapat mengembangkan ilmunya dengan menggunakan *software* basis data lain.

Namun kami menyadari bahwa modul ini belum sempurna. Tinjauan dan saran yang bersifat membangun sangat diharapkan demi peningkatan kesempurnaan modul praktikan ini.

Malang, Maret 2017

Penyusun

# Daftar Isi

<b>KEGIATAN PRAKTIKUM.....</b>	<b>II</b>
<i>Mata Kuliah .....</i>	<i>ii</i>
<i>Standar Kompetensi .....</i>	<i>ii</i>
<i>Bobot SKS, Semester .....</i>	<i>ii</i>
<i>Program Studi .....</i>	<i>ii</i>
<i>Jumlah Pertemuan .....</i>	<i>ii</i>
<b>KATA PENGANTAR .....</b>	<b>III</b>
<b>DAFTAR ISI .....</b>	<b>IV</b>
<b>AKSES MYSQL .....</b>	<b>1</b>
<i>KEGIATAN INTI.....</i>	<i>1</i>
<i>Mengakses MySQL melalui DOS Prompt .....</i>	<i>1</i>
<i>Tambah user .....</i>	<i>3</i>
<i>memberi hak akses kepada user.....</i>	<i>3</i>
<i>Melihat Hak Akses .....</i>	<i>4</i>
<i>Mencabut Hak Akses.....</i>	<i>4</i>
<i>Menghapus User.....</i>	<i>4</i>
<i>Kesimpulan .....</i>	<i>5</i>
<i>PENUTUP .....</i>	<i>6</i>
<i>Tugas .....</i>	<i>6</i>
<b>DATA DEFINITION LANGUAGE.....</b>	<b>7</b>
<i>KEGIATAN INTI.....</i>	<i>7</i>
<i>Membuat ERD .....</i>	<i>7</i>
<i>Generate to PDM.....</i>	<i>13</i>
<i>Generate to Database .....</i>	<i>15</i>
<i>Kesimpulan .....</i>	<i>17</i>
<i>PENUTUP .....</i>	<i>18</i>
<i>Tugas .....</i>	<i>18</i>
<b>DATA DEFINITION LANGUAGES 2.....</b>	<b>19</b>
<i>KEGIATAN INTI.....</i>	<i>19</i>
<i>Create Database .....</i>	<i>19</i>
<i>Show Databases.....</i>	<i>19</i>
<i>Create Tabel .....</i>	<i>20</i>
<i>Backup dan Restore .....</i>	<i>22</i>
<i>Kesimpulan .....</i>	<i>22</i>
<i>PENUTUP .....</i>	<i>23</i>
<i>Tugas .....</i>	<i>23</i>
<b>DATA DEFINITION LANGUAGES 3 .....</b>	<b>24</b>
<i>KEGIATAN INTI.....</i>	<i>24</i>
<i>Mengubah Nama Tabel.....</i>	<i>24</i>
<i>Menambah Field atau kolom baru.....</i>	<i>25</i>
<i>Mengubah Nama Field atau kolom .....</i>	<i>25</i>
<i>Mengubah Tipe data Field atau kolom.....</i>	<i>25</i>
<i>Menambah dan Menghapus Primary Key .....</i>	<i>25</i>
<i>Menambah dan Menghapus Foreign Key.....</i>	<i>26</i>
<i>Menghapus Tabel .....</i>	<i>27</i>
<i>Menghaus Database .....</i>	<i>27</i>
<i>Kesimpulan .....</i>	<i>27</i>
<i>PENUTUP .....</i>	<i>28</i>
<i>Tugas .....</i>	<i>28</i>
<b>DATA MANIPULATION LANGUAGES .....</b>	<b>29</b>

<i>KEGIATAN INTI</i> .....	29
<i>Insert</i> .....	30
<i>Select</i> .....	31
<i>Update</i> .....	31
<i>Delete</i> .....	32
<i>Kesimpulan</i> .....	33
<i>PENUTUP</i> .....	35
<i>Tugas</i> .....	35
<b>DATA MANIPULATION LANGUAGES (SELECT) .....</b>	<b>36</b>
<i>KEGIATAN INTI</i> .....	37
<i>SELECT</i> .....	37
<i>WHERE</i> .....	37
<i>BETWEEN / NOT BETWEEN</i> .....	38
<i>LIKE / NOT LIKE</i> .....	39
<i>IN / NOT IN</i> .....	40
<i>IS NULL / IS NOT NULL</i> .....	41
<i>ORDER BY</i> .....	41
<i>GROUP BY</i> .....	42
<i>DISTINCT</i> .....	42
<i>SUM</i> .....	43
<i>AVG</i> .....	43
<i>MIN</i> .....	43
<i>MAX</i> .....	44
<i>COUNT</i> .....	44
<i>Kesimpulan</i> .....	44
<i>PENUTUP</i> .....	45
<i>Tugas</i> .....	45
<b>JOINING TABLE .....</b>	<b>46</b>
<i>KEGIATAN INTI</i> .....	46
<i>Join</i> .....	47
<i>Inner Join</i> .....	47
<i>Left Join</i> .....	48
<i>Right Join</i> .....	48
<i>Kesimpulan</i> .....	48
<i>PENUTUP</i> .....	50
<i>Tugas</i> .....	50
<b>VIEW .....</b>	<b>51</b>
<i>KEGIATAN INTI</i> .....	51
<i>Membuat VIEW</i> .....	51
<i>Menampilkan View yang telah dibuat</i> .....	52
<i>Menjalankan View</i> .....	52
<i>Menghapus View</i> .....	53
<i>Mengakses MySQL menggunakan phpMyAdmin (tambahan)</i> .....	53
<i>Kesimpulan</i> .....	56
<i>PENUTUP</i> .....	57

## Modul

## 1

## Akses MySQL

## Kompetensi Dasar

Mahasiswa dapat melakukan pengaksesan MySQL (XAMPP) serta melakukan konfigurasi user.

## Indikator

1. Mahasiswa mampu mengakses MySQL melalui Command Prompt
2. Mahasiswa mampu melakukan konfigurasi user pada MySQL
3. Mahasiswa mampu melakukan setting ijin akses database untuk user tertentu.

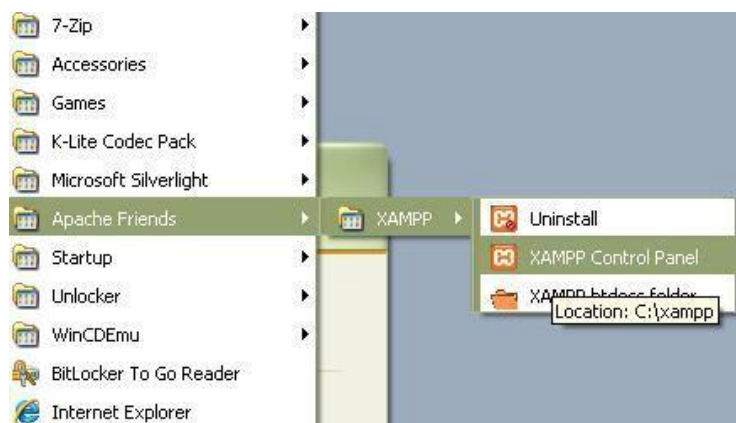
## Materi Pokok

1. Masuk ke dalam My SQL melalui DOS Prompt
2. Melakukan konfigurasi User

## Kegiatan Inti

## Mengakses MySQL melalui DOS Prompt

**Gambar 1.1 :**  
Membuka Control  
Panel secara manual



Untuk membuka XAMPP Control Panel secara manual, dapat di buka melalui START | All Programs | Apache Friends | XAMPP | XAMPP Control Panel

**Gambar 1.2 :**

XAMPP Control  
Panel



Untuk memulai service mysql, buka XAMPP control panel dan klik **start** pada baris MySQL hingga muncul tanda “Running”.

**Gambar 2.1 :**

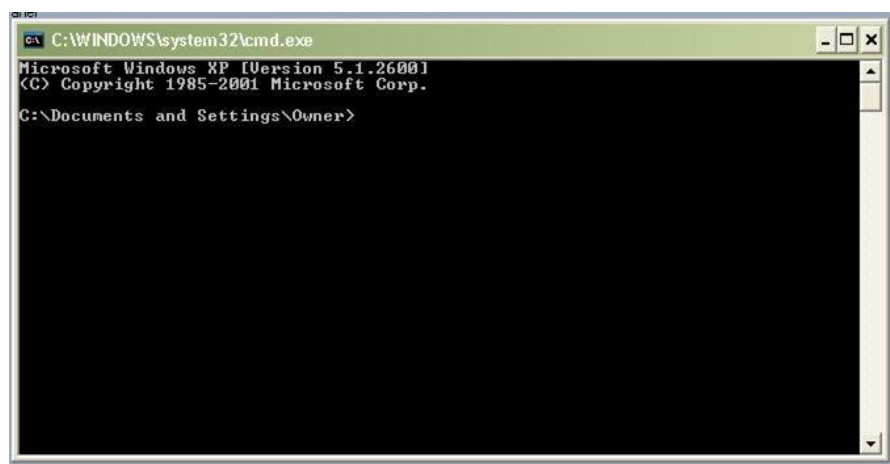
Memulai DOS Prompt



Untuk membuka DOS Prompt, klik Start | Run | cmd dan tekan **enter**.

**Gambar 2.2 :**

Program DDOS  
Prompt



**Gambar 2.3 :**  
Mengakses MySQL dari Prompt

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Owner>cd c:\xampp\mysql\bin
C:\xampp\mysql\bin>mysql -u root -p
Enter password:

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.5.16 MySQL Community Server (GPL)
Copyright (c) 2000, 2011, Oracle and/or its affiliates. All
rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or
its affiliates. Other names may be trademarks of their respective
owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current
input statement.

mysql>
```

Ketik Perintah : **cd c:\xampp\mysql\bin**, setelah berada di direktori MySQL, login dengan username root dengan mengetikkan command : **mysql -u root -p** , ketika diminta password, tekan enter (user root tidak memiliki password).

**Gambar 2.4 :**  
Membuat user baru

```
mysql> CREATE USER 'praktikum2013'@'localhost' IDENTIFIED BY
'stiki';
Query OK, 0 rows affected (0.00 sec)

mysql>
```

Untuk membuat user baru, masuk ke MySQL menggunakan user root dan ketikkan perintah :

**CREATE USER 'nama\_user'@'server' IDENTIFIED BY 'password';**

contoh:

**CREATE USER 'praktikum2015'@'localhost' IDENTIFIED BY 'stiki';**

**Gambar 2.5 :**  
Memberikan hak akses user

```
mysql> GRANT ALL ON test.* TO 'praktikan'@'localhost';
Query OK, 0 rows affected (0.00 sec)

mysql>
```

Untuk memberikan hak akses kepada user, masuk ke MySQL menggunakan user root dan ketikkan perintah

**GRANT [type of permission] ON [database].[table] TO '[user]'@'localhost';**

dengan type of permission sebagai berikut :

- ALL PRIVILEGE / ALL
- Create
- Drop
- Delete
- Insert
- Select
- Update
- Grant Option



Contoh Penerapan:

```
GRANT ALL on test.* to 'praktikan2015'@'localhost';  
GRANT Update on test.* to 'praktikan'@'localhost';  
GRANT Update, select, create on *.* to 'praktikan'@'localhost';
```

**Gambar 2.6 :**

Melihat hak akses user

```
mysql> Show grants for 'praktikan'@'localhost'; |
```

Untuk melihat user, masuk ke MySQL menggunakan user root dan ketikkan perintah :

```
Show grants for 'nama user'@'server';
```

Contoh Penerapan :

```
Show grants for 'praktikan'@'localhost';
```

**Gambar 2.7 :**

Mencabut Hak Akses

```
mysql> REVOKE INSERT ON test.* from 'praktikum'@'localhost';
```

Untuk mencabut hak akses user, masuk ke MySQL menggunakan user root dan ketikkan perintah :

```
REVOKE [type of permission] ON [database name].[table name] FROM  
'[username]'@'localhost';
```

Contoh Penerapan :

```
REVOKE insert ON test.* FROM 'praktikan'@'localhost';
```

**Gambar 2.7 :**

Menghapus User

```
mysql> Drop user 'praktikan'@'localhost';
```

Untuk mencabut hak akses user, masuk ke MySQL menggunakan user root dan ketikkan perintah :

```
DROP USER 'nama user'@'server';
```

Contoh Penerapan :

```
DROP USER 'praktikan'@'localhost';
```

## Kesimpulan

### Soal 1.1

Sebuah website memiliki beberapa user dan setiap user memiliki hak akses berbeda terhadap databasenya, tentukan hak akses setiap user dengan memberikan tanda centang (✓) dan berikan alasannya.

User	Password	Jabatan	Hak Akses			
			Create	Read	Update	Delete
administrator	Stiki	Database Administrator				
Adi	adi	Kontributor				
Budi	Budu	Editor				
Rohman	Rohmin	Programmer				

Alasan

.....

.....

### Soal 1.2

Tuliskan perintah untuk membuat user beserta hak aksesnya berdasarkan tabel pada soal 1.1 yang telah anda tentukan, gunakan database "cdcol".

.....

.....

.....

.....

.....

.....

.....

.....

## Penutup

---

### Tugas

#### Tugas 1.1

1.2.1. jelaskan apa yang dimaksud dengan XAMPP dan MySQL

.....

.....

.....

.....

.....

1.2.2. sebut dan jelaskan beberapa macam database beserta kelebihan dan kekurangannya.

.....

.....

.....

.....

#### Tugas 1.2

Buatlah min. 5 user dengan hak akses berbeda dan berikan alasan pemberian hak akses tersebut. (tuliskan syntaxnya).

.....

.....

.....

.....

## Modul

## 2

## Data Definition Language 1

## Kompetensi Dasar

Setelah menyelesaikan perkuliahan ini mahasiswa akan dapat membangun database (DDL / Data Definition Language) dengan menggunakan MySQL.

## Indikator

1. Mahasiswa dapat menggunakan MySQL WorkBench untuk mendesain ERD.

## Materi Pokok

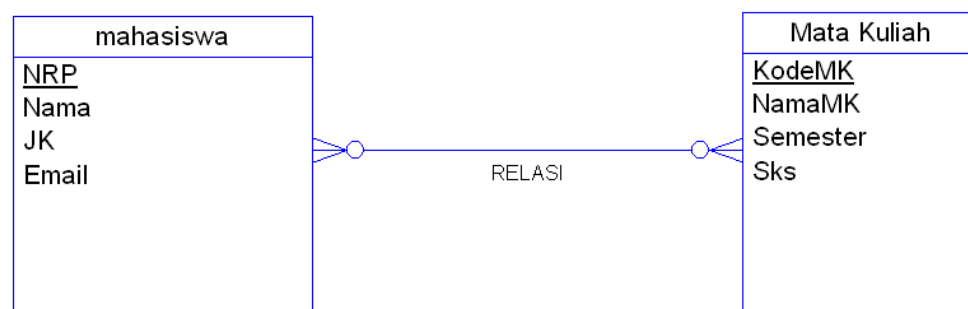
1. Pengenalan tampilan MySQL WorkBench Power Designer.
2. Menggunakan MySQL Workbench pada contoh kasus sederhana

## Kegiatan Inti

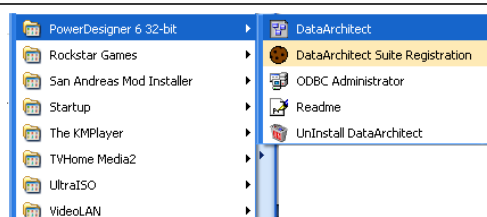
## Membuat ERD

Gambar berikut ini merupakan desain ERD yang akan dibuat dengan menggunakan Power Designer DataArchitect.

**Gambar 2.10:**  
Desain ERD.

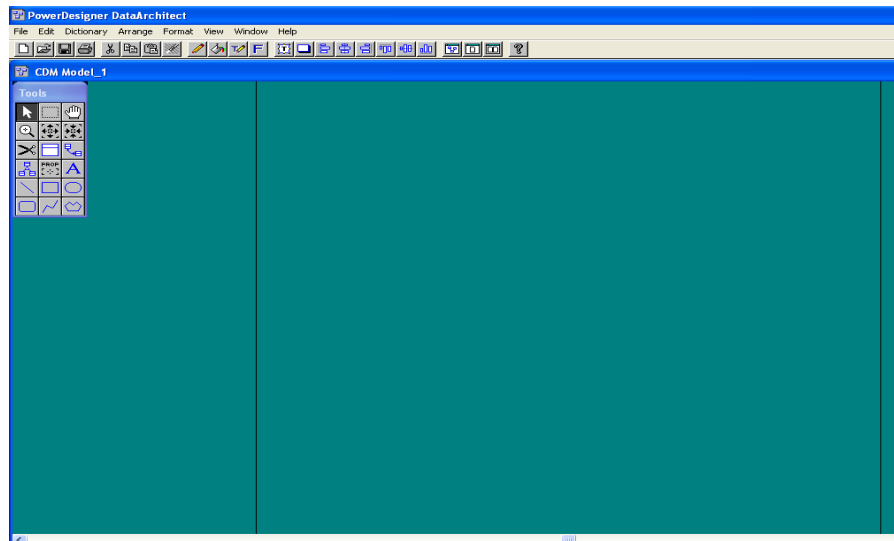


**Gambar 2.11:**  
DataArchitect Icon.



Buka Aplikasi PowerDesigner DataArchitect.


**Gambar 2.12:**  
DataArchitect Main  
Page.



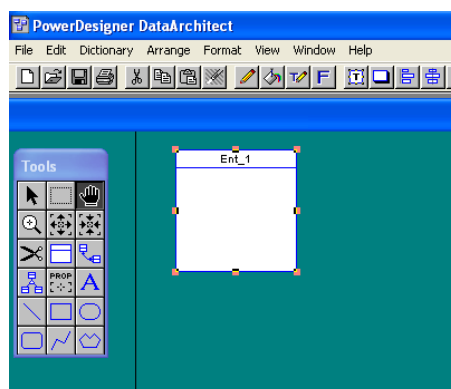
Gambar diatas merupakan tampilan utama aplikasi Power Designer DataArchitect.

**Gambar 2.13:**  
Tools Box



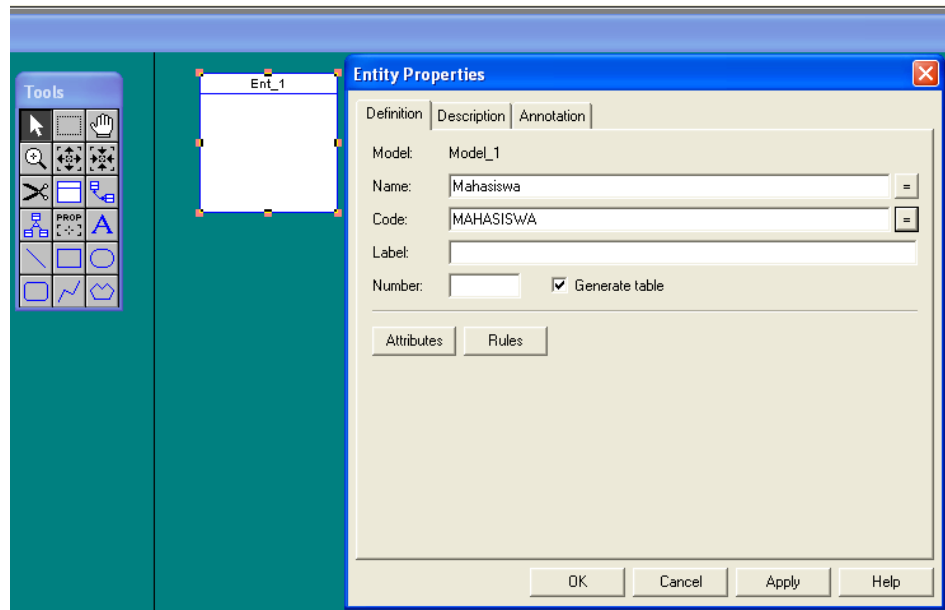
Untuk membuat Entitas, pilih tool Entity pada  Tools Box dan letakkan pada lembar kerja DataArchitect.


**Gambar 2.14:**  
Entity




Pilih tool Pointer pada Tools Box  dan klik dua kali pada Entitas yang telah dibuat untuk memunculkan halaman Entity Properties.

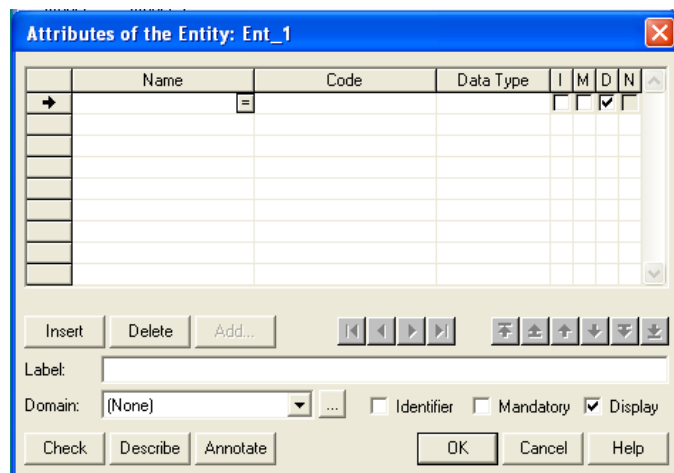
**Gambar 2.15:**  
Entity Properties





Ketikkan nama entitas yang dibuat pada kotak isian Name kemudian klik tombol  yang terdapat di sebelah kotak isian Code untuk memberikan Code secara otomatis.

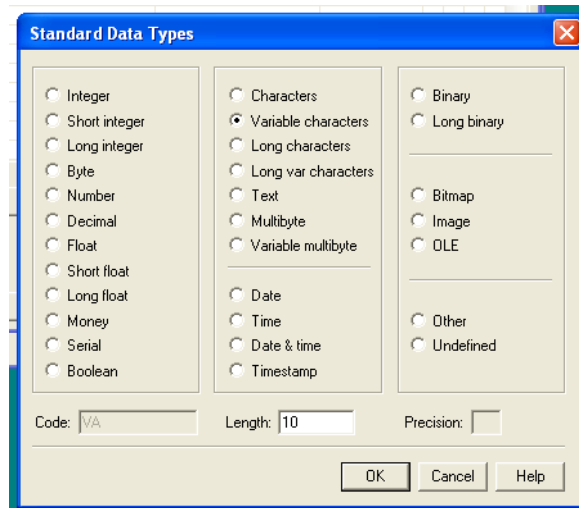
Klik tombol Attributes  untuk memberikan atribut pada entitas tersebut.

**Gambar 2.16:**  
Entity Attributes




Ketikkan Nama Atribut pada kolom Name, dan tekan  untuk memberikan Code secara otomatis dari nama yang diberikan. Kemudian klik tombol  yang terdapat pada kolom Data Type untuk memberikan tipe data.

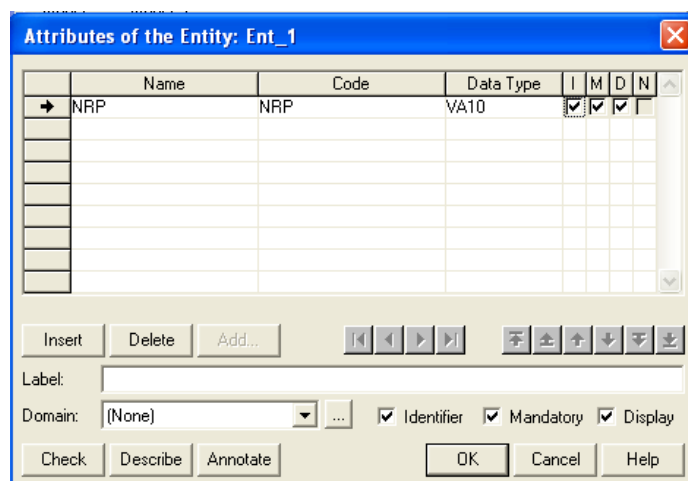
**Gambar 2.17:**  
Data Types




The 'Standard Data Types' dialog box shows various data type options. The 'Variable characters' option is selected under the 'Characters' section. The 'Length' field is set to 10. The 'Code' field is set to 'VA'. The 'Precision' field is empty. The 'OK', 'Cancel', and 'Help' buttons are at the bottom.


Pilih data type sesuai dengan kebutuhan dan masukkan panjang data type tersebut pada kotak isian Length. Pilih OK  untuk melanjutkan.

**Gambar 2.18:**  
Identifier, Mandatory



The 'Attributes of the Entity: Ent\_1' dialog box shows a table with columns: Name, Code, Data Type, I, M, D, N. The first row has 'NRP' in the Name and Code columns, and 'VA10' in the Data Type column. The 'I' (Identifier) and 'M' (Mandatory) checkboxes are checked. Below the table are buttons for 'Insert', 'Delete', 'Add...', and a set of navigation arrows. At the bottom, there are checkboxes for 'Identifier', 'Mandatory', and 'Display', all of which are checked. The 'Label' field is empty, and the 'Domain' is set to '(None)'. The 'Check', 'Describe', 'Annotate', 'OK', 'Cancel', and 'Help' buttons are at the bottom.

Centang pada kolom I  untuk menyatakan bahwa atribut tersebut merupakan sebuah Identifier atau sering disebut pula sebagai Primary Key (PK).

Centang pada kolom M  untuk menyatakan bahwa atribut tersebut merupakan Mandatory, yang artinya atribut tersebut harus diisi atau tidak boleh kosong.

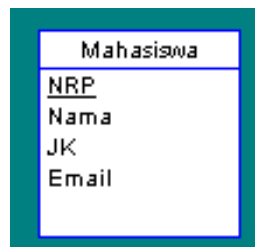
Lanjutkan langkah-langkah tersebut sesuai dengan desain ERD pada gambar 2.1, hingga dihasilkan Atribut seperti gambar dibawah ini

**Gambar 2.19:**  
Field Attributes

	Name	Code	Data Type	I	M	D	N
→	NRP	NRP	VA10	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2	Nama	NAMA	VA25	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	JK	JK	A1	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4	Email	EMAIL	VA25	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Pilih OK  untuk melanjutkan. Sehingga dihasilkan bentuk Entity seperti


**Gambar 2.20:**  
Entity



Lanjutkan langkah-langkah tersebut sesuai dengan desain ERD pada gambar 2.1 untuk semua table, hingga dihasilkan Atribut seperti gambar dibawah ini

**Gambar 2.21:**  
Entity

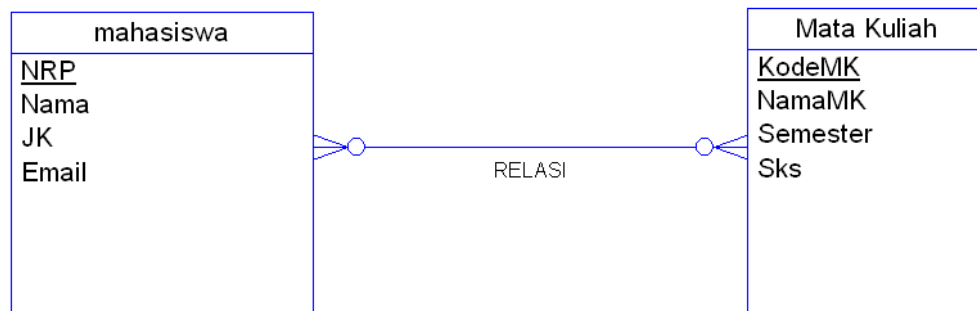



Pilih tool Relationship  untuk memberikan relasi antar Entitas yang telah dibuat. Klik pada salah satu Entitas dan drag ke Entitas lain yang akan direlasikan. Sehingga akan muncul garis relasi antar Entitas tersebut.



Gambar 2.22:


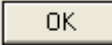
Relation



Gunakan tool Pointer  dan klik dua kali pada garis relasi untuk membuka halaman Relationship Properties

Gambar 2.23:

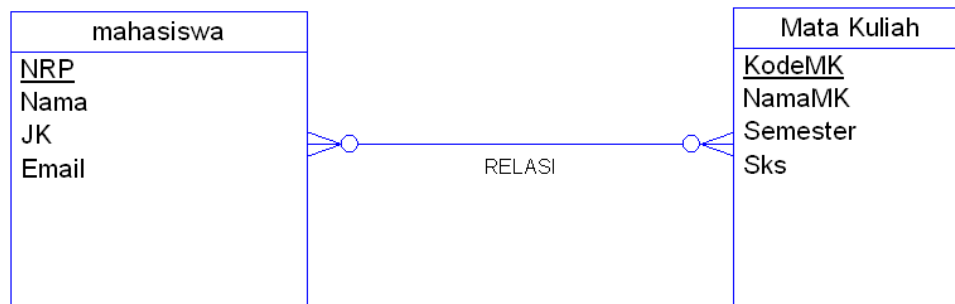
Relationship Properties

Isikan nama relasi pada kotak isian Name dan tekan tombol  untuk memberikan Code secara otomatis dari nama yang diberikan. Pada panel Cardinality pilih jenis kardinalitas sesuai dengan desain ERD. Klik OK  untuk menyimpan dan menutup jendela Relationship Properties.

Gunakan langkah-langkah diatas untuk membuat relasi antara entitas sesuai dengan desain ERD pada gambar 2.1. Hingga dihasilkan bentuk ERD seperti gambar berikut.

**Gambar 2.24:**

(CDM)  
Conceptual  
Data Model



Setelah Anda membuat skema seperti di atas, selanjutnya pilih tab menu Dictionary, lalu pilih Generate Physical Model. Lihat gambar di bawah ini.

**Gambar 2.25:**

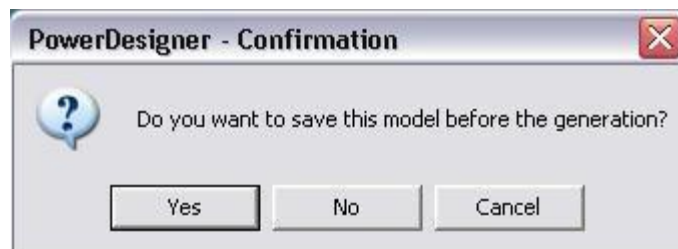
Langkah mengenerate logical data model menjadi physical data model.



Ketika muncul kotak dialog seperti di bawah ini, klik Yes.

**Gambar 2.26:**

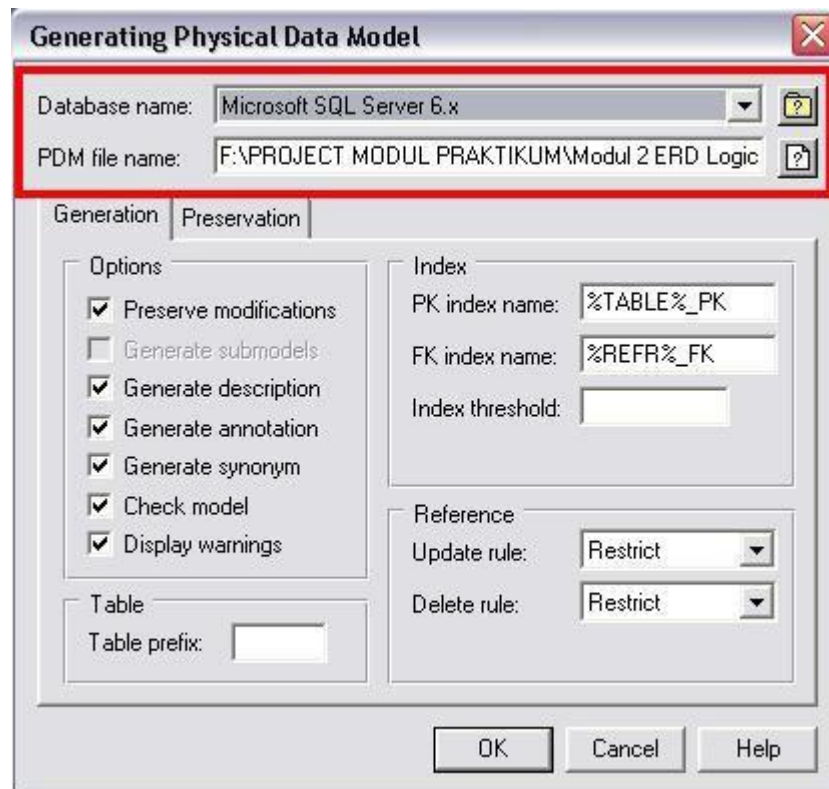
Dialog konfirmasi simpan dokumen.



Setelah itu muncul kotak dialog Generating Physical Data Model seperti gambar di bawah ini. Pada Database name pilih Microsoft SQL Server 6.x, pada PDM file name pilih folder yang berisi file .CDM yang telah Anda buat tadi, lalu ubah nama file .PDM nya, selanjutnya klik OK.

Gambar 2.27 :

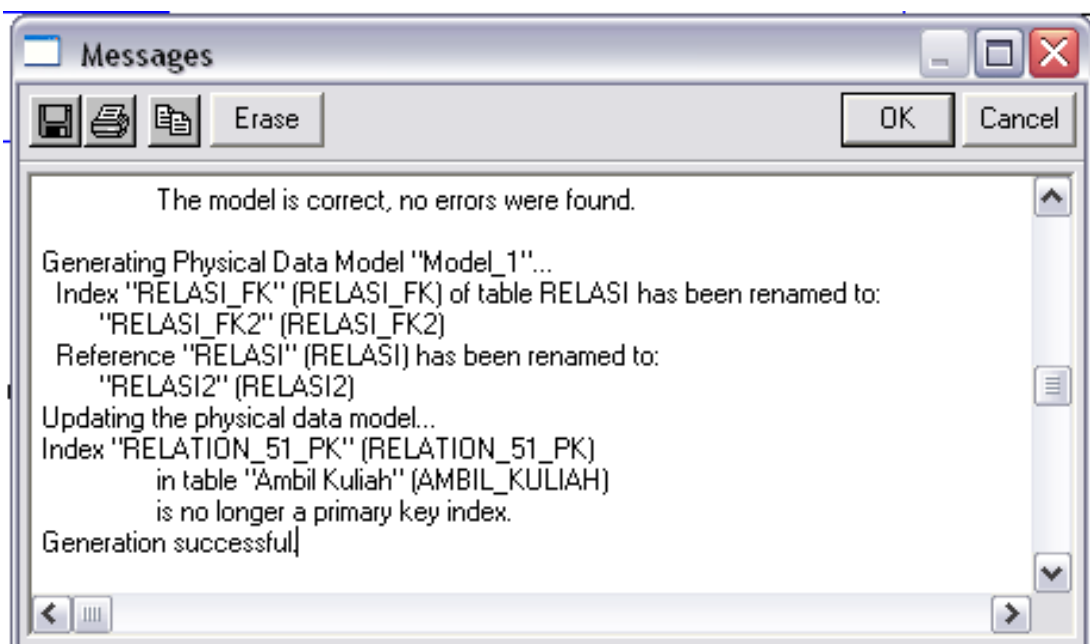
Kotak dialog untuk mengkonfigurasi Physical Data Model.



Apabila sudah dikonfigurasi kemudian klik OK. Maka akan muncul Message dialog yang berisi informasi dari database yang telah kita *generate* tadi, seperti gambar di bawah ini. Selanjutnya klik OK.

Gambar 2.28 :

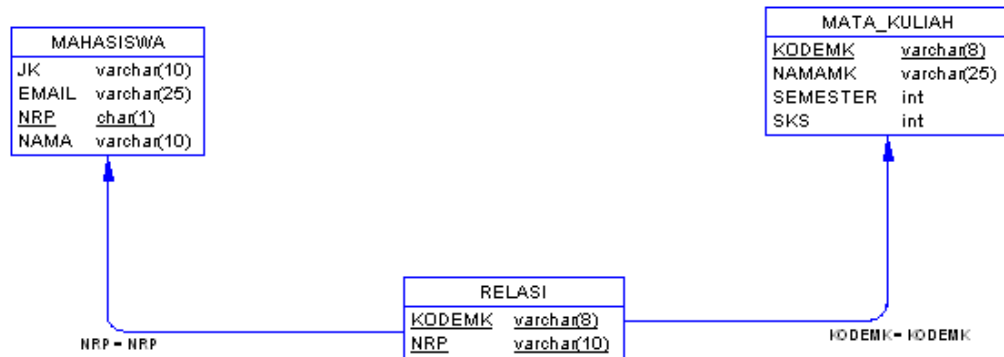
Message Dialog.



Apabila semua langkah telah dilakukan dengan baik, maka skema database akan menjadi Physical Model seperti ditunjukkan pada gambar di bawah ini.

**Gambar 2.29 :**

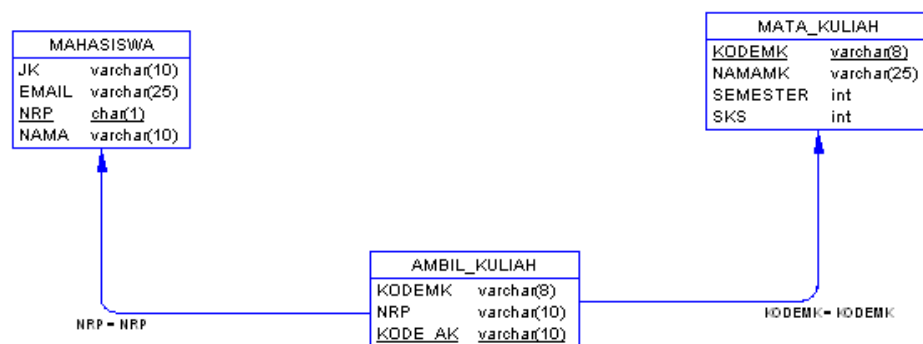
.Physical Data Model yang berhasil di buat.



Edit hasil Physical Data Model (PDM) di atas sehingga menjadi seperti di bawah ini

**Gambar 2.30 :**

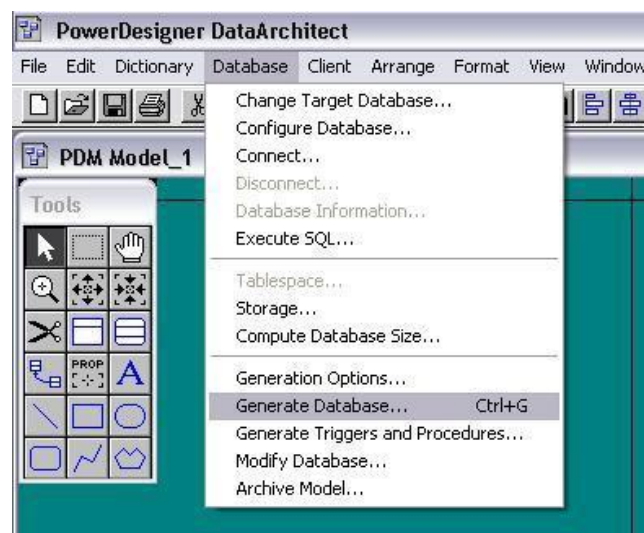
.Hasil Physical Data Model yang telah di Edit.



Setelah itu pilih tab menu Database, kemudian pilih Generate Database seperti gambar di bawah ini.

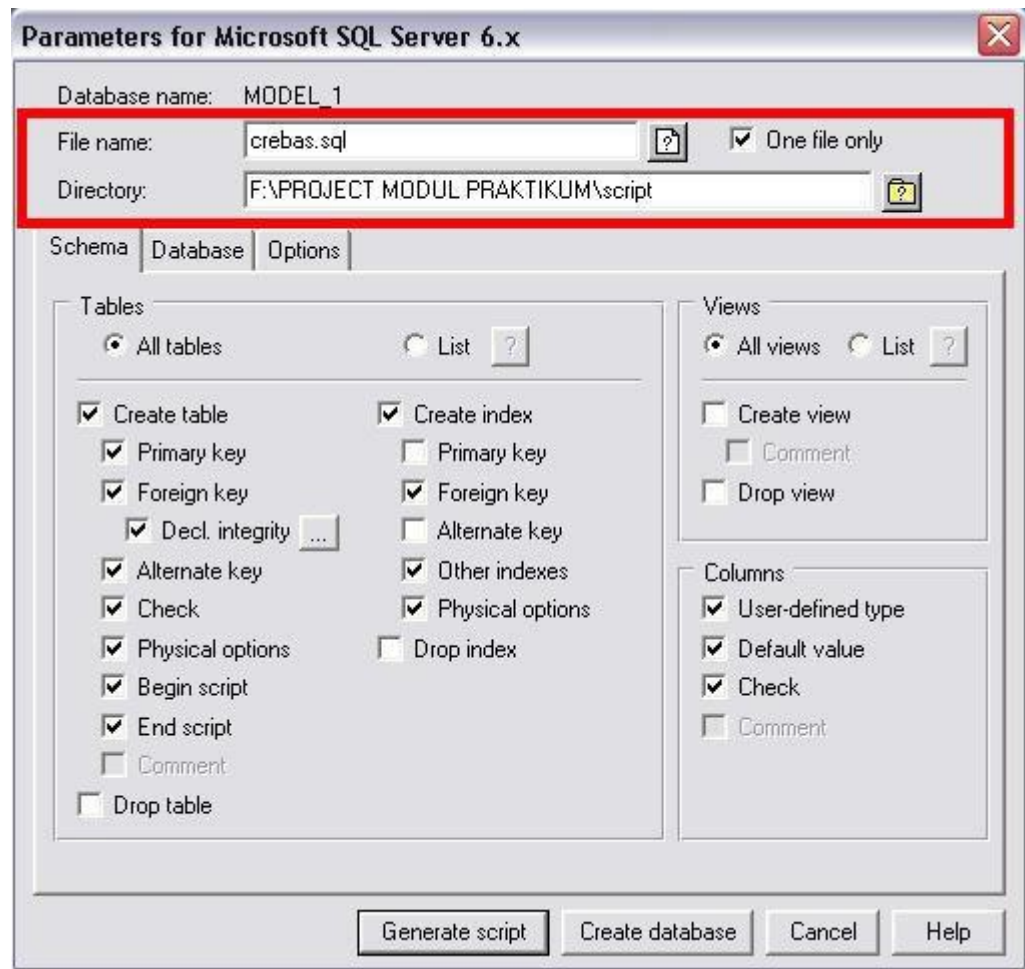
**Gambar 2.31 :**

Langkah-langkah mengenerate database.



Akan muncul kotak dialog seperti pada gambar di bawah ini.

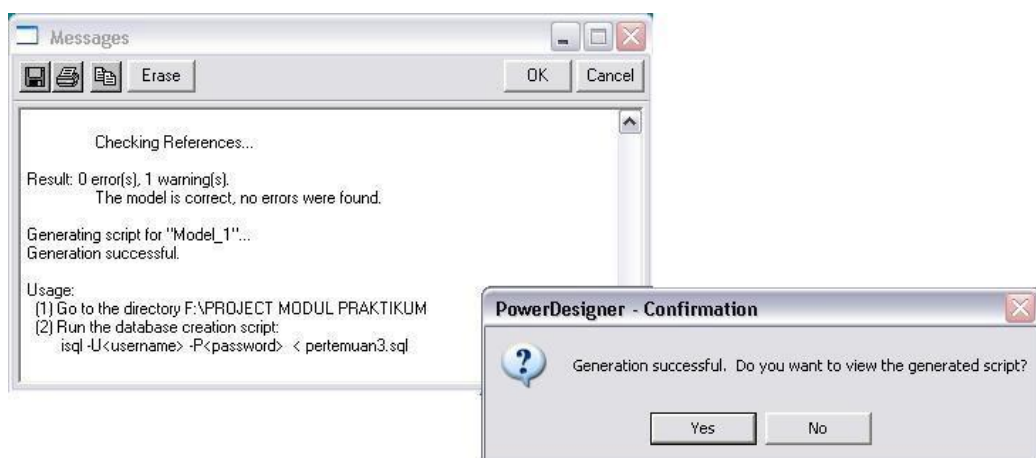
**Gambar 2.31 :**  
Konfigurasi untuk  
Microsoft SQL  
Server.



Pada File name, ubah nama filen SQLnya, pada Directory pilih folder untuk menyimpan file SQL tersebut.

Apabila semua langkah telah dilakukan dengan baik, akan muncul dua kotak dialog seperti pada gambar di bawah ini.

**Gambar 2.32 :**  
Konfirmasi untuk  
melihat isi file  
SQL.

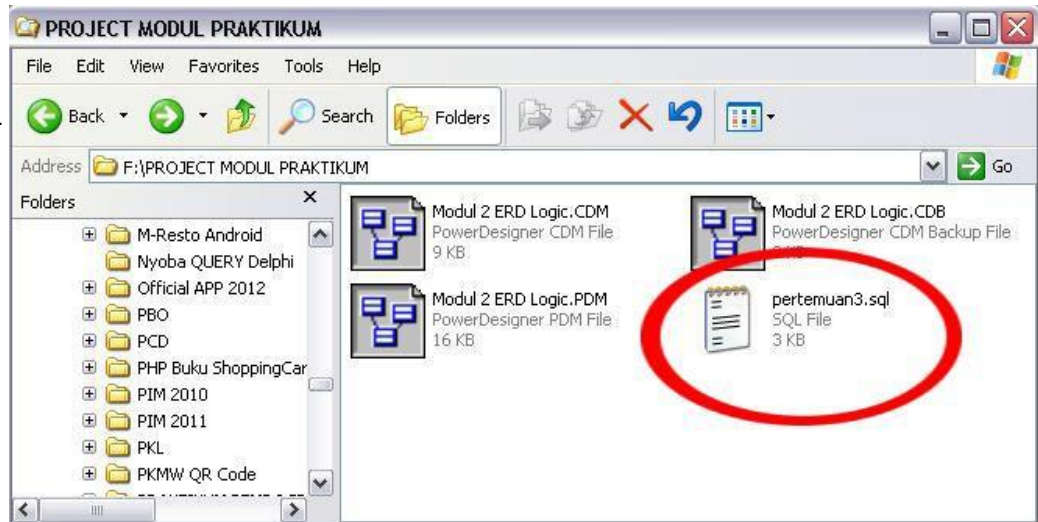


Klik OK pada kotak dialog Confirmation untuk melihat isi dari file SQL yang telah berhasil dibuat. Selanjutnya klik OK pada Message dialog untuk menutupnya.

Sampai pada tahap ini kita telah berhasil mengenerate Logical Data Model menjadi Physical Data Model kemudian menjadi file yang berisi kode-kode SQL untuk menjadikan Data Model tersebut menjadi database secara fisik. Kita dapat melihat file SQL yang berhasil kita generate tadi melalui Windows Explorer seperti pada gambar di bawah ini.

Gambar 2.33 :

File SQL dilihat dari windows explorer.



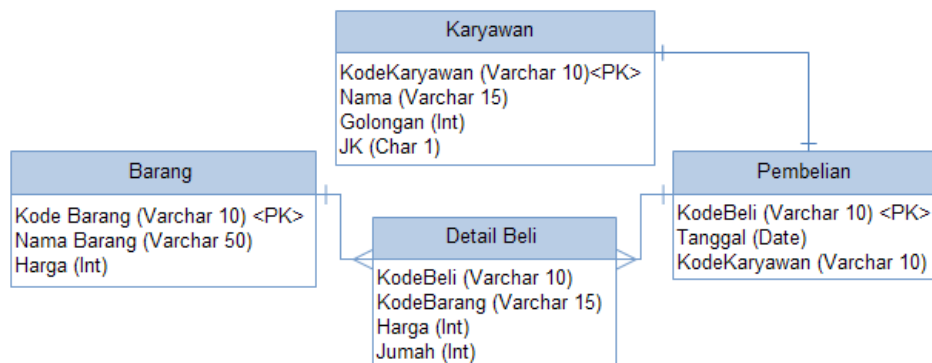
**Note :** Hasil generate database yang menghasilkan file berekstensi .sql seperti gambar di atas hanya dapat di implementasikan ke SQL Server, Karena waktu generate dari CDM ke PDM memilih Option Database name : Microsoft SQL Server 6.x

## Kesimpulan

### Soal 2.1

Gambar 2.35:

Desain ERD.



Buatlah ERD dari desain diatas dengan menggunakan Power Designer. Tunjukkan pada asisten kelas apabila sudah selesai.

### Soal 2.2

Dalam kardinalitas pada relasi antara entitas ada kemungkinan terjadinya relasi berupa Many to Many, dalam kenyataannya bentuk relasi tersebut tidak dapat terjadi. Jelaskan mengapa bentuk relasi Many to Many tidak dapat diterapkan dalam database, dan berikan solusi dari kasus tersebut.

.....  
 .....  
 .....

### Soal 2.3

Apa yang dimaksud dengan Logical Data Model dan Physical Data Model? Jelaskan pula perbedaan dari keduanya!

.....  
.....  
.....

### Soal 2.4

Mengapa desain database yang telah Anda buat menggunakan Logical Model harus di generate terlebih dahulu menjadi Physical Data Model sebelum di generate menjadi script SQL?

.....  
.....  
.....

## Penutup

---

### Tugas

#### Tugas 2.1

Buatlah desain database berupa ERD dengan menggunakan Power Designer seperti yang dibahas pada pertemuan kali ini dengan ketentuan minimal terdapat 4 tabel Master dan 2 tabel transaksi. Setiap Praktikan diharuskan membuat databasenya masing-masing dan tidak diperkenankan sama dengan praktikan lain. Simpan tugas dalam format Power Designer DataArchitect CDM Model (\*.cdm), PDM Model (\*.pdm), Skrip.sql dan tunjukkan pada asisten kelas pada pertemuan selanjutnya.

## 3

## Data Definition Languages 2

### Kompetensi Dasar

Mahasiswa mampu menggunakan perintah SQL (DDL) untuk membuat database, tabel serta relationship-nya

### Indikator

1. Mahasiswa mampu menggunakan perintah SQL (DDL) untuk pembuatan database
2. Mahasiswa mampu menggunakan perintah SQL (DDL) untuk pembuatan tabel
3. Mahasiswa mampu membuat relationship antara 2 tabel

### Materi Pokok

1. Create Database
2. Create Table
3. Relationship

## Kegiatan Inti

### Membuat Database

Gambar 3.1:

Membuat Database.

```
mysql > create database stiki;
```

Untuk membuat database baru, masuk ke MySQL menggunakan user root dan ketikkan perintah

```
CREATE DATABASE nama_database;
```

contoh

```
CREATE DATABASE stiki_malang ;
```

Gambar 3.2:

Melihat database.

```
mysql > show databases;
```

Untuk melihat database, masuk ke MySQL menggunakan user root dan ketikkan perintah

```
SHOW DATABASES;
```



## Membuat Tabel

**Gambar 3.3:**

Perintah Membuat tabel.

```
CREATE TABLE nama tabel(
    kolom1 typedata(ukuran) atribut ,
    kolom2 typedata(ukuran) atribut,
    kolom3 typedata(ukuran)
);
```

Berikut contoh untuk membuat tabel :

```
CREATE TABLE Mahasiswa (
    NRP varchar (10) not null,
    Nama varchar (25) not null,
    JK char (1) not null,
    Email varchar (25) null,
    PRIMARY KEY (NRP)
);
```

## Latihan

Keterangan: Praktikum ini menggunakan console.

Buatlah sebuah database dengan nama akademik\_db, dengan perintah seperti di bawah ini:

-> create database akademik\_db;

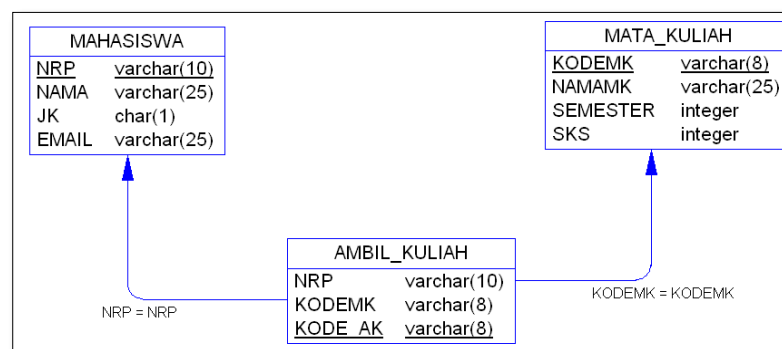
Setelah database telah berhasil dibuat, sekarang Anda aktifkan/gunakan database tersebut dengan perintah:

-> use akademik\_db;

Sekarang Anda buat 3 buah tabel dengan ketentuan sebagai berikut:

**Gambar 3.4:**

Skema Tabel.  
Desain (PDM)



### a. Nama Tabel: Mahasiswa

- Field 1:
  - Nama: NRP (Primary Key, not null)
  - Tipe Data: varchar (10)
- Field 2:
  - Nama: Nama (not null)
  - Tipe Data: varchar (25)

- Field 3:
  - Nama: JK (not null)
  - Tipe Data: char (1)
- Field 4:
  - Nama: Email (null)
  - Tipe Data: varchar (25)

**b. Nama Tabel: MataKuliah**

- Field 1:
  - Nama: KodeMK (Primary Key, not null)
  - Tipe Data: varchar (8)
- Field 2:
  - Nama: NamaMK (not null)
  - Tipe Data: varchar (25)
- Field 3:
  - Nama: Semester (not null)
  - Tipe Data: integer
- Field 4:
  - Nama: SKS (not null)
  - Tipe Data: integer

**c. Nama Tabel: AmbilKuliah**

- Field 1:
  - Nama: Kode\_AK (Primary Key, not null)
  - Tipe Data: varchar (8)
- Field 2:
  - Nama: KodeMK (not null)
  - Tipe Data: varchar (8)
- Field 3:
  - Nama: NRP (not null)
  - Tipe Data: varchar (10)

Setelah itu ketikkan script SQL di bawah ini untuk membuat tabel tersebut:

```
CREATE TABLE Mahasiswa (  
    NRP varchar (10) not null,  
    Nama varchar (25) not null,  
    JK char (1) not null,  
    Email varchar (25) null,  
    PRIMARY KEY (NRP)  
);  
  
CREATE TABLE MataKuliah (  
    KodeMK varchar (8) not null,  
    NamaMK varchar (25) not null,  
    Semester int not null,  
    Sks int not null,  
    PRIMARY KEY (KodeMK)  
);  
  
CREATE TABLE AmbilKuliah(  
    kode_ak VARCHAR (8) not null,  
    NRP varchar (10) not null,  
    KodeMK varchar (8) not null,  
    PRIMARY KEY (kode_ak),  
    FOREIGN KEY (KodeMK) REFERENCES MataKuliah(KodeMK),  
    FOREIGN KEY (NRP) REFERENCES Mahasiswa(NRP)  
);
```

Keterangan :

Perintah untuk membuat relasi / foreign Key yaitu :

**FOREIGN KEY (nama field) REFERENCES nama tabel referensi (nama field pada tabel referensi)**

contoh :

**FOREIGN KEY (NRP) REFERENCES Mahasiswa(NRP)**

## Back up and Restore

Untuk melakukan Back up and Restore pada Mysql digunakan fasilitas MySql Dump.

Untuk melakukan Back Up database digunakan perintah

```
Mysqldump -u [username] -p [password] [nama database] > [namafile.sql]
```

Hasil MySql Dump adalah file dengan format \*.sql. dan biasanya setelah proses dump selesai akan muncul pada direktory ... \xampp\mysql\bin\

Untuk melakukan Restore database dari file dump yang telah dibuat sebelumnya digunakan perintah berikut

```
Mysql -u [username] -p [password] [nama database] < [namafile.sql]
```

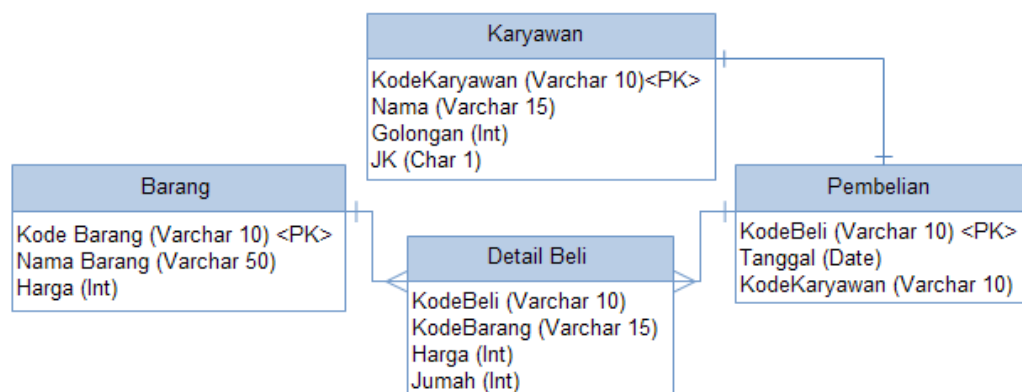
Biasanya untuk beberapa versi Mysql lama proses dump database proses pembuatan database tidak disertakan, sehingga sebelum melakukan restore file \*.sql terlebih dahulu harus menyiapkan database yang akan direstore.

## Kesimpulan

Pada bagian ini, berikan pertanyaan pada praktikan untuk dijawab.

### Soal 3.1

**Gambar 3.5:**  
Desain ERD.



Buatlah sebuah database dengan acuan gambar diatas ! Tunjukkan kepada asisten apabila sudah selesai

### Soal 3.2

Apa yang terjadi jika dalam satu tabel terdapat dua foreign key?  
Jelaskan.

## Penutup

---

### Tugas

#### Tugas 3.1

Pada pertemuan sebelumnya praktikan sudah ditugaskan untuk membuat desain databasenya masing-masing. Tugas untuk pertemuan ini adalah implementasikan desain atau rancangan database yang telah dibuat pada pertemuan sebelumnya menjadi sebuah database sebenarnya. Dump database yang telah selesai dibuat dan tunjukkan kepada asisten pada pertemuan selanjutnya.

## 4

## Data Definition Language 3

### Kompetensi Dasar

Mahasiswa mampu menggunakan perintah SQL (DDL) untuk modifikasi tabel serta relationship-nya.

### Indikator

1. Mahasiswa mampu menggunakan perintah SQL (DDL) untuk memodifikasi Tabel.

### Materi Pokok

1. Mengubah Nama Tabel
2. Menambah Field baru
3. Mengubah nama field
4. Mengubah tipe data pada field
5. Menambah Primary Key
6. Menghapus Primary Key
7. Menambah dan menghapus Foreign Key

## Kegiatan Inti

### Mengubah Nama Tabel

**Gambar 4.1:**  
Mengubah  
Nama Tabel

```
ALTER TABLE 'nama_tabel' RENAME TO 'nama_tabel_baru';
```

Contoh Penerapan :

```
ALTER TABLE Ambil_Kuliah RENAME TO Ambil_Mata_Kuliah;
```

```
ALTER TABLE mahasiswa RENAME TO mahasiswa_stiki;
```

## Menambah Field / Kolom Baru

---

**Gambar 4.2:**  
Menambah Field /  
Kolom Baru

```
ALTER TABLE 'Nama_Tabel' ADD 'Kolom_Baru' 'Attribute Field';
```

Contoh Penerapan :

```
ALTER TABLE AmbilKuliah ADD AmbilNRPMahasiswa int not null;
```

```
ALTER TABLE AmbilKuliah ADD AmbilKodeMK varchar (8) not null;
```

```
ALTER TABLE AmbilKuliah ADD Nilai float;
```

```
ALTER TABLE Mahasiswa ADD Kota varchar (15);
```

## Mengubah Nama Field / Kolom

---

**Gambar 4.3:**  
Mengubah Nama  
Field / Kolom

```
ALTER TABLE 'Nama Tabel' CHANGE 'Nama Field_Lama'  
'Nama_field_baru' 'Attribute Field';
```

Contoh Penerapan :

```
ALTER TABLE Mahasiswa CHANGE JK Jenis_Kelamin varchar (10);
```

```
ALTER TABLE AmbilKuliah CHANGE AmbilNRPMahasiswa A_NRP varchar (10);
```

## Mengubah Type Data pada Field / Kolom

---

**Gambar 4.5:**  
Mengubah Type  
Data Field

```
ALTER TABLE 'Nama Tabel' MODIFY 'Nama_Field' 'Attribute  
Perubahan';
```

Contoh Penerapan :

```
ALTER TABLE Mahasiswa MODIFY JK varchar (15);
```

```
ALTER TABLE MataKuliah MODIFY NamaMK varchar (35);
```

## Menambah Primary Key pada Tabel

---

**Gambar 4.6:**  
Menambah  
Primary Key

```
ALTER TABLE 'Nama Tabel' ADD PRIMARY KEY (Nama Field);
```

Contoh Penerapan :

```
ALTER TABLE Mahasiswa ADD PRIMARY KEY (NRP);
```

## Menghapus Primary Key

---

**Gambar 4.7:**  
Menghapus  
Primary Key

```
ALTER TABLE 'Nama Tabel' DROP PRIMARY KEY;
```

Contoh Penerapan :

```
ALTER TABLE Mahasiswa DROP PRIMARY KEY;
```

```
ALTER TABLE MataKuliah DROP PRIMARY KEY;
```

Catatan :

Setiap Tabel hanya boleh mempunyai 1 Primary Key.

## Menambah Foreign Key

---

**Gambar 4.8:**  
Menambah  
Foreign Key

```
ALTER TABLE 'Nama Tabel' ADD CONSTRAINT 'Nama_Foreign Key'  
FOREIGN KEY (Nama_Field) REFERENCES  
'Nama_tabel_acuan'(Nama_field_acuan)
```

Contoh Penerapan :

```
ALTER TABLE AmbilKuliah ADD CONSTRAINT fk_AmbilNRP FOREIGN KEY  
(NRP) REFERENCES Mahasiswa(NRP);
```

```
ALTER TABLE AmbilKuliah ADD CONSTRAINT fk_AmbilKodeMK FOREIGN  
KEY (KodeMK) REFERENCES MataKuliah(KodeMK);
```

## Menghapus Foreign Key

---

**Gambar 4.9:**  
Menghapus  
Foreign Key

```
ALTER TABLE 'Nama Tabel' DROP FOREIGN KEY 'Nama_Foreign_Key';
```

Contoh Penerapan :

```
ALTER TABLE AmbilKuliah DROP FOREIGN KEY fk_AmbilNRP;
```

```
ALTER TABLE AmbilKuliah DROP FOREIGN KEY fk_AmbilKodeMK;
```

## Menghapus Tabel pada Database

**Gambar 4.10:**  
Menghapus  
Tabel

```
DROP TABLE 'Nama_Tabel';
```

Contoh Penerapan :

```
DROP TABLE Mahasiswa;
```

```
DROP TABLE MataKuliah;
```

## Melihat Tabel pada Database

**Gambar 4.10:**  
Melihat Tabel  
pada Database

```
SHOW TABLES;
```

Contoh Penerapan :

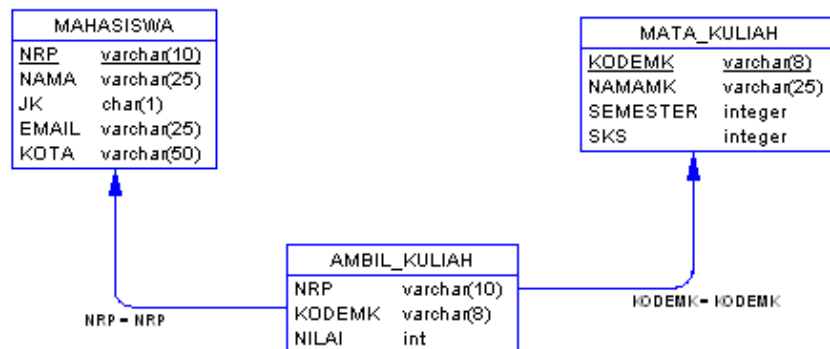
```
USE 'nama database';
```

```
SHOW TABLES;
```

## Kesimpulan

### Soal 4.1

**Gambar 4.x:**  
Desain Database.



Modifikasi database yang telah anda buat pada pertemuan 3, Sehingga Susunan Tabel dan Field/kolom sesuai dengan gambar di atas.

Berikut Penjelasannya :

1. Mahasiswa
  - NRP            varchar (10)
  - NAMA        varchar (25)
  - JK            char (1)
  - EMAIL       varchar (25)
  - KOTA        varchar (50)
2. Mata\_Kuliah
  - KodeMK      varchar (8)
  - NamaMK      varchar (25)
  - SEMESTER   integer
  - SKS          integer



3. Ambil\_Kuliah
  - NRP                varchar (10)
  - KodeMK        varchar (8)
  - Nilai            int

## Penutup

---

### Tugas

#### Tugas 4.1

Pada pertemuan sebelumnya praktikan sudah ditugaskan untuk membuat desain databasenya masing-masing. Tugas untuk pertemuan ini adalah modifikasi masing tabel. Tuliskan bagaimana perintah untuk membuat relasinya (Foreign Key). Tulis perintah modifikasi tersebut. (Minimal 10 perintah)

## Data Manipulation Languages 1

### Kompetensi Dasar

---

Mahasiswa mampu menggunakan perintah SQL (DML) untuk memanipulasi data pada 1 tabel.

### Indikator

---

1. Mahasiswa mampu menggunakan perintah SQL (DML) untuk memanipulasi data dalam tabel.
2. Mahasiswa mampu memasukkan data kedalam tabel.
3. Mahasiswa mampu melakukan perubahan data pada tabel.
4. Mahasiswa mampu melakukan penghapusan data pada tabel.
5. Mahasiswa mampu menampilkan data yang ada didalam tabel

### Materi Pokok

---

1. Data Manipulation Languages (insert, update, delete).
2. Insert into.
3. Update where.
4. Delete where.
5. Select \* from.

### Kegiatan Inti

---

Pada pertemuan kali ini akan belajar bagaimana memasukkan data kedalam sebuah tabel, mengubah data dalam tabel dan menghapus data pada tabel, serta menampilkan isi data dari tabel. Pertemuan ini menggunakan database yang sudah dibuat bersama pada pertemuan empat. Dengan menggunakan data berikut

**Gambar 5.1:**  
Data Mahasiswa &  
Mata Kuliah

Mahasiswa				
NRP	Nama	JK	Email	Kota
111110393	Hilman N.	L	hilman@stiki.ac.id	Malang
101110249	Teguh H.S.	L	Teguh_hs@stiki.ac.id	Blitar
102210055	Chandra Eka	L	chandra.eka@stiki.ac.id	Pasuruan
101110254	Adrian	L	rian@stiki.ac.id	Kediri
101110211	Eka Widya Sari	P	eka.widya@stiki.ac.id	Malang
Mata Kuliah				
KodeMK	NamaMK	Semester	SKS	
TI09KB21	Basis Data	2	3	
TI09KB22	Praktikum Basis Data	2	1	
TI09KK22	Matematika Diskrit	2	3	
TI13KB23	Pemrograman Dasar 1	2	3	
TI13KK23	Algoritma & Struktur Data	2	3	
TI09KB41	Analisa Sistem Informasi	4	3	
TI09KB44	Pemrograman Web	4	3	
TI09KB42	Interaksi Manusia & Komputer	4	2	

## Insert

Untuk melakukan insert data pada MySql digunakan perintah

```
Insert into <Nama Table> (<Nama Field>) values (<Isi Data>)
```

Atau bisa pula dengan menggunakan

```
Insert into <Nama Table> values (<Isi Data>)
```

Untuk melakukan insert beberapa data sekaligus dapat menggunakan

```
Insert into <Nama Table> (<Nama Field>) values (<Isi Data>), (<Isi Data>), (<Isi Data>), ...
```

Atau

```
Insert into <Nama Table> values (<Isi Data>), (<Isi Data>), (<Isi Data>), ...
```

Contoh :

Dengan menggunakan data berikut.

Mahasiswa				
NRP	Nama	JK	Email	Kota
111110393	Hilman N.	L	hilman@stiki.ac.id	Malang

Untuk melakukan insert data berikut kedalam Database maka digunakan perintah

```
Insert into Mahasiswa(NRP>Nama,JK>Email) Values („081110021“,„Roly Segara“, „L“, „roly@stiki.ac.id“);
```

Sehingga apabila berhasil, dalam halaman konsole akan tampak seperti gambar berikut

Gambar 5.2:

Inserting

```

C:\WINDOWS\system32\cmd.exe - mysql -uroot -p

C:\xampp\mysql\bin>mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 5.1.41 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use akademik_db;
Database changed
mysql> insert into Mahasiswa(NRP,Nama,JK,Email) Values ('081110021','Roly Sigara', 'L','roly@stiki.ac.id');
Query OK, 1 row affected (0.00 sec)

mysql> _

```

Dengan menggunakan langkah-langkah diatas lakukan entry data sesuai dengan data yang terdapat pada gambar 5.1 untuk tabel Mahasiswa dan table Mata Kuliah.

## Select

Perintah select digunakan untuk menampilkan isi tabel kedalam layar monitor. Cara menggunakan perintah select adalah sebagai berikut.

Select \* from <Nama Table>

Tanda Asterisk (\*) digunakan untuk menampilkan semua field yang terdapat dalam table.

Contoh :

Untuk menampilkan data pada tabel Mahasiswa perintah yang digunakan adalah

Select \* from Mahasiswa

Sehingga tampilan konsole akan menjadi seperti gambar dibawah ini

Gambar 5.3:

Select

```

Administrator: C:\Windows\system32\cmd.exe - mysql -u root -p

mysql> select * from mahasiswa;
+----+-----+----+-----+-----+
| NRP | Nama      | JK | Email                      | Kota  |
+----+-----+----+-----+-----+
| 101110211 | Eka Widya Sari | P  | eka.widya@stiki.ac.id    | Malang |
| 101110249 | Teguh H.S.    | L  | Teguh_hs@stiki.ac.id    | Blitar |
| 101110254 | Adrian       | L  | rian@stiki.ac.id        | Kediri |
| 102210055 | Chandra Eka  | L  | chandra.eka@stiki.ac.id  | Pasuruan |
| 111110393 | Hilman N.    | L  | hilman@stiki.ac.id      | Malang |
+----+-----+----+-----+-----+
5 rows in set (0.00 sec)

mysql>

```

## Update

Untuk melakukan update (merubah) data pada tabel, digunakan perintah

Update <Nama Table> Set <Nama Field> = (Nilai baru) where (Kondisi)

Contoh :

Apabila hendak merubah data dari yang semula

Mahasiswa				
NRP	Nama	JK	Email	Kota
111110393	Hilman N.	L	hilman@stiki.ac.id	Malang

Menjadi

Mahasiswa				
NRP	Nama	JK	Email	Kota
111110393	Hilman N.H.	L	<u>hilman_nh@stiki.ac.id</u>	<u>Malang</u>

Maka perintah yang digunakan adalah

Update Mahasiswa set Email="hilman\_nh@stiki.ac.id" where NRP = "111110393"

Ketika dijalankan pada layar konsole akan nampak seperti gambar berikut.

Gambar 5.4:

Update

```

Administrator: C:\Windows\system32\cmd.exe - mysql -u root -p

mysql> Update Mahasiswa set Email='hilman_nh@stiki.ac.id' where NRP='111110393';
Query OK, 1 row affected (0.07 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql>
  
```

Guanakan perintah Select untuk melihat apakah data pada table sudah berhasil terupdate.

## Delete

Perintah delete digunakan untuk menghapus data dari dalam sebuah tabel. Yang perlu diingat bahwa perintah delete berbeda dengan perintah drop. Delete hanya menghapus data pada tabel, sedangkan drop akan menghapus skema dan isi dari tabel tersebut.

Perintah dideklarasikan seperti berikut

Delete from <Nama Tabel> where (Kondisi)

Contoh :

Apabila hendak menghapus data berikut dari dalam table

Mahasiswa				
NRP	Nama	J	Email	Kota
102210055	Chandra Eka	L	<u>Chandra.eka@stiki.ac.id</u>	<u>Malang</u>

Perintah yang digunakan adalah

Delete from Mahasiswa where NRP=" 102210055"

Setelah dijalankan pada layar konsole akan nampak seperti gambar berikut

Gambar 5.5:

Delete

```

Administrator: C:\Windows\system32\cmd.exe - mysql -u root -p

mysql> delete from mahasiswa where nrp='10220055';
Query OK, 0 rows affected (0.00 sec)

mysql>
  
```

Lakukan perintah Select untuk melihat apakah data sudah berhasil terhapus.

## Kesimpulan

### Soal 5.1

Lakukan insert pada database akademik\_db untuk tabel AmbilMataKuliah jika diberikan data Kartu Hasil Studi masing-masing mahasiswa adalah sebagai berikut.

**Gambar 5.6:**

Insert Detail

Hilman N.		Teguh H.S.	
Basis Data	79	Basis Data	70
P. Basis Data	82	P. Basis Data	74
Matematika Diskrit	83	Analisa Sistem Informasi	81
Algoritma & Struktur Data	75	Interaksi Manusia & Komputer	85
Chandra Eka		Adrian	
Basis Data	76	Basis Data	85
P. Basis Data	78	P. Basis Data	90
Pemrograman Web	81	Analisa Sistem Informasi	87
		Pemrograman Web	85
		Pemrograman Dasar 1	100
Eka Widya			
Basis Data	100		
P. Basis Data	100		
Pemrograman Dasar 1	76		
Algoritma & Struktur Data	79		

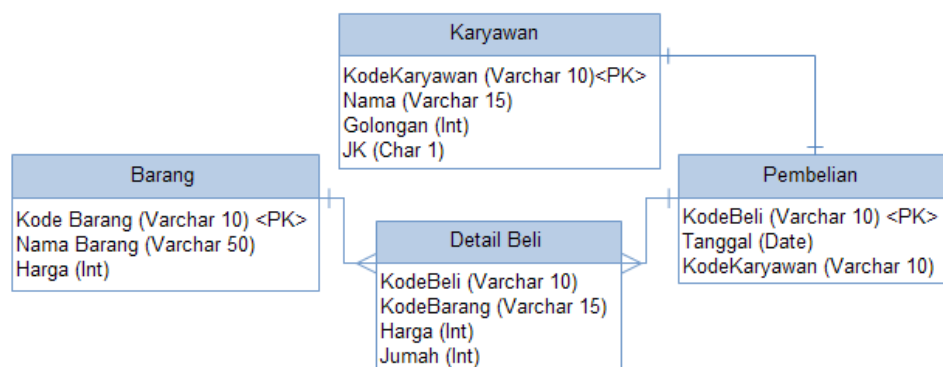
Tunjukkan kepada asisten apabila sudah selesai.

### Soal 5.2

Gunakan skema database yang telah dibuat pada latihan Soal 2.1 di pertemuan 2.

**Gambar 5.7:**

Skema Database



**Gambar 5.8:**  
Latihan Insert Data

Barang			
Kode Barang	Nama Barang	Harga	
KD001	Tango Wafer 104gr	5200	
KD002	Mintz Chewy Bag 125gr	3600	
KD003	Mr. Potato 60gr	5400	
KD004	Vita Pudding 108gr	2550	
KD005	Top Kopi 20x25gr	13900	
KD006	Sari Wangi Green Tea 25x25gr	5900	
KD007	Sari Bunga Madu 650 ml	50000	
KD008	Nusantara Madu Murni	63000	
KD009	The Pucuk Harum480ml	3250	
KD010	Coolant Botol 350ml	4200	
Karyawan			
Kode Karyawan	Nama	Golongan	JK
KS001	Abi Chandra	1	L
MG001	Adrian	2	L
KS002	Ko Liem	1	L
KS003	Sari Eka	3	P
KS004	Wahyu Deddy	2	L
KS005	Daniel Budiman	1	L

Lakukan insert data diatas kedalam database yang sudah dibuat pada latihan pertemuan 2. Tunjukkan kepada Asisten apabila sudah selesai.

### Soal 5.3

Bolehkah suatu field yang merupakan primary key diubah (Update) ? Jelaskan pendapat anda disertai alasan dan analisa anda !

.....

.....

.....

.....

### Soal 5.4

Berdasarkan Skema database tersebut pada tabel detail terdapat field KodeBarang yang merupakan foreign key dari field KodeBarang pada tabel Barang. Apakah yang terjadi apabila salah satu data pada Tabel barang dihapus ? Jelaskan pendapat anda !

.....

.....

.....

## Penutup

---

### Tugas

#### Tugas 5.1

Pada pertemuan sebelumnya sudah diberikan tugas membuat desain database untuk masing-masing praktikan. Pada pertemuan kali ini lanjutkan database yang telah dibuat tersebut dengan mengisi data pada setiap table. Dengan aturan minimal terdapat 15 data pada setiap tabel master dan 25 data pada setiap tabel transaksi. Tugas dikerjakan di rumah dan bawa file Dump dari database tersebut pada pertemuan selanjutnya.



## Data Manipulation Languages 2 (SELECT)

### Kompetensi Dasar

---

Mahasiswa mampu menggunakan perintah SQL untuk menampilkan data serta menggabungkannya dengan fungsi – fungsi standard

### Indikator

---

1. Mahasiswa dapat menggunakan perintah select
2. Mahasiswa dapat menampilkan data dalam berbagai kondisi yang diinginkan
3. Mahasiswa dapat menggunakan beberapa fungsi SQL standard

### Materi Pokok

---

1. Data Manipulation Languages (select)
2. Select \* from where
3. Select \* from where between / not
4. Select \* from where like / not
5. Select \* from where in / not
6. Select \* from where is null
7. Select \* from where order by
8. Select \* from where group by
9. Select distinct from
10. Select avg / max / min / count / sum

## Kegiatan Inti

### SELECT

Perintah ini digunakan untuk men-seleksi / menampilkan record pada 1 tabel maupun lebih.

Syntax : **SELECT** <nama\_field> **FROM** <nama\_tabel>;

Contoh :

```
mysql> SELECT nrp, nama, kota FROM mahasiswa;
+-----+-----+-----+
| nrp      | nama          | kota      |
+-----+-----+-----+
| 111110393 | Hilman N.     | Malang   |
+-----+-----+-----+
| 101110249 | Teguh H.S.    | Blitar   |
+-----+-----+-----+
| 102210055 | Chandra Eka   | Pasuruan |
+-----+-----+-----+
| 101110254 | Adrian        | Kediri   |
+-----+-----+-----+
| 101110211 | Eka Widya     | Malang   |
+-----+-----+-----+
5 row in set (0.00 sec)
```

### WHERE

Digunakan untuk memberikan filter pen-seleksian record berdasarkan kondisi yang telah ditentukan oleh user.

Syntax: **SELECT** <nama\_field> **FROM** <nama\_tabel> **WHERE** <kondisi>;

Contoh :

```
mysql> SELECT nrp, nama, kota FROM mahasiswa WHERE nrp='111110393';
+-----+-----+-----+
| nrp      | nama          | kota      |
+-----+-----+-----+
| 111110393 | Hilman N.     | Pasuruan  |
+-----+-----+-----+
```

## BETWEEN / NOT BETWEEN

Digunakan untuk memberikan filter pen-seleksian record berdasarkan range atau diluar range dengan value tertentu.

Syntax :

```
SELECT <nama_field> FROM <nama_table> WHERE <nama_field> BETWEEN  
<value_1> AND <value_2>
```

atau

```
SELECT nama_field FROM nama_table WHERE nama_field NOT BETWEEN value_1  
AND value_2
```

Contoh 1 :

```
mysql> SELECT nrp, nama, kota FROM mahasiswa WHERE nrp BETWEEN  
'101110211' AND '101110254';  
+-----+-----+-----+  
| nrp      | nama                | kota  |  
+-----+-----+-----+  
| 101110254 | Adrian              | Kediri|  
+-----+-----+-----+  
| 101110211 | Eka Widya           | Malang|  
+-----+-----+-----+  
2 rows in set (0.05 sec)
```

Contoh 2 :

```
mysql> SELECT nrp, nama, kota FROM mahasiswa WHERE nrp NOT BETWEEN  
'101110211' AND '101110254';  
+-----+-----+-----+  
| nrp      | nama                | kota  |  
+-----+-----+-----+  
| 111110393 | Hilman N.           | Malang|  
+-----+-----+-----+  
| 101110249 | Teguh H.S.          | Blitar|  
+-----+-----+-----+  
| 102210055 | Chandra Eka         | Pasuruan|  
+-----+-----+-----+  
3 row in set (0.00 sec)
```

---

## LIKE / NOT LIKE

Digunakan untuk memberikan filter pen-seleksian record berdasarkan beberapa karakter pada field tertentu atau tidak mengandung karakter yang telah ditentukan.

Syntax :

```
SELECT <nama_field> FROM <nama_tabel> WHERE <nama_field> LIKE
<'%karakter / value%'>;
```

atau

```
SELECT <nama_field> FROM <nama_tabel> WHERE <nama_field> NOT LIKE
<'%karakter / value%'>;
```

Contoh 1 :

```
mysql> SELECT nrp, nama, kota FROM mahasiswa WHERE nama LIKE '%Wid%';
+-----+-----+-----+
| nrp      | nama              | kota  |
+-----+-----+-----+
| 101110211|Eka Widya          | Malang|
+-----+-----+-----+
1 row in set (0.00 sec)
```

Contoh 2 :

```
mysql> SELECT nrp, nama, kota FROM mahasiswa WHERE nama NOT LIKE
'%Wid%';
+-----+-----+-----+
| nrp      | nama              | kota  |
+-----+-----+-----+
| 111110393 | Hilman N.         | Malang|
+-----+-----+-----+
| 101110249 | Teguh H.S         | Blitar|
+-----+-----+-----+
| 102210055 | Chandra Eka       | Pasuruan|
+-----+-----+-----+
| 101110254 | Adrian           | Kediri|
+-----+-----+-----+
4 row in set (0.00 sec)
```

## IN / NOT IN

Memiliki fungsi hampir sama dengan BETWEEN dan NOT BETWEEN, tetapi difungsi ini hanya menggunakan 2 parameter.

Syntax :

```
SELECT <nama_field> FROM <nama_tabel> WHERE <nama_field> IN
('value_1','value_2');
```

atau

```
SELECT <nama_field> FROM <nama_tabel> WHERE <nama_field> NOT IN
('value_1','value_2');
```

Contoh 1 :

```
mysql> SELECT nrp, nama, kota FROM mahasiswa WHERE nrp IN (111110393,
'101110249');
+-----+-----+-----+
| nrp      | nama                | kota  |
+-----+-----+-----+
| 111110393 | Hilman N.           | Malang|
+-----+-----+-----+
| 101110249 | Teguh H.S.          | Blitar|
+-----+-----+-----+
2 row in set (0.00 sec)
```

Contoh 2 :

```
mysql> SELECT nrp, nama, kota FROM mahasiswa WHERE nrp IN
('111110393',' 101110249');
+-----+-----+-----+
| nrp      | nama                | kota  |
+-----+-----+-----+
| 102210055 | Chandra Eka         | Pasuruan|
+-----+-----+-----+
| 101110254 | Adrian              | Kediri |
+-----+-----+-----+
| 101110211 | Eka Widya            | Malang|
+-----+-----+-----+
3 row in set (0.00 sec)
```

## IS NULL / IS NOT NULL

Memiliki fungsi menseleksi / menampilkan record yang memiliki field kosong dan sebaliknya.

Syntax :

```
SELECT <nama_field> FROM <nama_tabel> WHERE <nama_field> IS NULL;
```

atau

```
SELECT <nama_field> FROM <nama_tabel> WHERE <nama_field> IS NOT NULL;
```

Contoh :

```
mysql> SELECT nrp, nama, kota FROM mahasiswa WHERE kota IS NOT NULL;
+-----+-----+-----+
| nrp      | nama          | kota  |
+-----+-----+-----+
| 111110393 | Hilman N.     | Malang|
+-----+-----+-----+
| 101110249 | Teguh H.S.    | Blitar|
+-----+-----+-----+
| 102210055 | Chandra Eka   | Pasuruan|
+-----+-----+-----+
| 101110254 | Adrian        | Kediri|
+-----+-----+-----+
| 101110211 | Eka Widya     | Malang|
+-----+-----+-----+
5 row in set (0.00 sec)
```

## ORDER BY

Memiliki fungsi untuk menseleksi / menampilkan record secaraurut berdasarkan field tertentu secara ascending (ASC) atau descending (DESC).

Syntax :

```
SELECT <nama_field> FROM <nama_table> ORDER BY <nama_field>;
```

atau

```
SELECT <nama_field> FROM <nama_table> ORDER BY <nama_field> DESC;
```

Contoh :

```
mysql> SELECT nrp, nama, kota FROM mahasiswa ORDER BY nrp;
+-----+-----+-----+
| nrp      | nama          | kota  |
+-----+-----+-----+
| 121110598| Dimas Aditya   | Malang|
+-----+-----+-----+
| 121110532| Erwin Eka A    | Malang|
+-----+-----+-----+
| 101110211| Eka Widya Sari | Blitar|
```

```

+-----+-----+-----+
| 101110254| Adrian          | Kediri|
+-----+-----+-----+
| 102210055| Chandra Eka      | Pasuruan|
+-----+-----+-----+
5 row in set (0.00 sec)

```

## GROUP BY

Syntax : `SELECT <nama_field> FROM <nama_tabel> GROUP BY <nama_field>;`

Contoh :

```

mysql> SELECT * FROM nilai GROUP BY nrp;
+-----+-----+-----+
| kode_mk | nrp          | nilai |
+-----+-----+-----+
| MK001   | 102210055    | 90    |
| MK001   | 101110254    | 90    |
+-----+-----+-----+
2 rows in set (0.01 sec)

```

## DISTINCT

Syntax : `SELECT DISTINCT <nama_field> FROM <nama_tabel>;`

Contoh :

```

mysql> SELECT DISTINCT kode_mk FROM ambilmatakuliah;
+-----+
| kode_mk |
+-----+
| MK001   |
| MK002   |
+-----+
2 rows in set (0.00 sec)

```

## SUM

Memiliki fungsi untuk menjumlahkan nilai / value field di suatu tabel.

Syntax: **SELECT SUM(nama\_field) FROM <nama\_tabel>;**

Contoh :

```
mysql> SELECT SUM(nilai) FROM ambilmatakuliah;
+-----+
| SUM(nilai) |
+-----+
|      350   |
+-----+
1 row in set (0.02 sec)
```

## AVG

Memiliki fungsi untuk mencari nilai rata-rata sebuah field di suatu tabel.

Syntax: **SELECT AVG(nama\_field) FROM <nama\_tabel>;**

Contoh :

```
mysql> SELECT AVG(nilai) FROM ambilmatakuliah;
+-----+
| AVG(nilai) |
+-----+
|  87.5000   |
+-----+
1 row in set (0.00 sec)
```

## MIN

Memiliki fungsi untuk mencari nilai terendah sebuah field di suatu tabel.

Syntax: **SELECT MIN(nama\_field) FROM <nama\_table>;**

Contoh :

```
mysql> SELECT MIN(nilai) FROM ambilmatakuliah;
+-----+
| MIN(nilai) |
+-----+
|      80    |
+-----+
1 row in set (0.00 sec)
```



## MAX

Memiliki fungsi untuk mencari nilai tertinggi di sebuah field di suatu tabel.

Syntax: **MAX(nama\_field) FROM <nama\_tabel>;**

Contoh :

```
mysql> SELECT MAX(nilai) FROM ambilmatakuliah;
+-----+
| MAX(nilai) |
+-----+
|          90 |
+-----+
1 row in set (0.02 sec)
```

## COUNT

Digunakan untuk menghitung jumlah record di suatu tabel.

Syntax: **COUNT(nama\_field) FROM <nama\_tabel>;**

Contoh :

```
mysql> SELECT COUNT(*) FROM mahasiswa;
+-----+
| COUNT(*) |
+-----+
|          5 |
+-----+
1 row in set (0.00 sec)
```

## Kesimpulan

Gunakan database latihan pada modul 5 untuk mengerjakan soal dibawah ini.

### Soal 6.1

1. Tulislah perintah untuk menampilkan data karyawan yang berjenis kelamin laki-laki.  
.....
2. Tulislah perintah untuk menampilkan nama karyawan yang bernama depan „daniel“.  
.....
3. Tulislah perintah untuk menampilkan jumlah barang yang memiliki harga diatas 50.000.  
.....  
.....
4. Tulislah perintah untuk menampilkan harga barang tertinggi,terendah dan rata-rata.  
.....  
.....  
.....

5. Tulislah perintah untuk menampilkan barang seraca berurutan berdasarkan nama barang, dimana harga barang tersebut diantara 10.000 – 50.000 dan memiliki nama depan yang mengandung karakter „S“.

.....

.....

.....

## Penutup

---

### Tugas

#### Tugas 6.1

Terapkanlah perintah-perintah pada modul 6 pada database yang telah anda buat pada modul sebelumnya, tulis syntaxnya.

#### Tugas 6.2

1. Bagaimana cara menampilkan semua field dari sebuah table.  
.....  
.....
2. Bagaimana cara merubah nama field ketika ditampilkan tanpa merubah nama field tersebut didatabase.  
.....  
.....

## Joining Table

### Kompetensi Dasar

Mahasiswa dapat menggunakan perintah SQL untuk menampilkan data dari beberapa tabel.

### Indikator

1. Mahasiswa dapat menggunakan perintah SQL join.
2. Mahasiswa dapat menampilkan data lebih dari 1 tabel.

### Materi Pokok

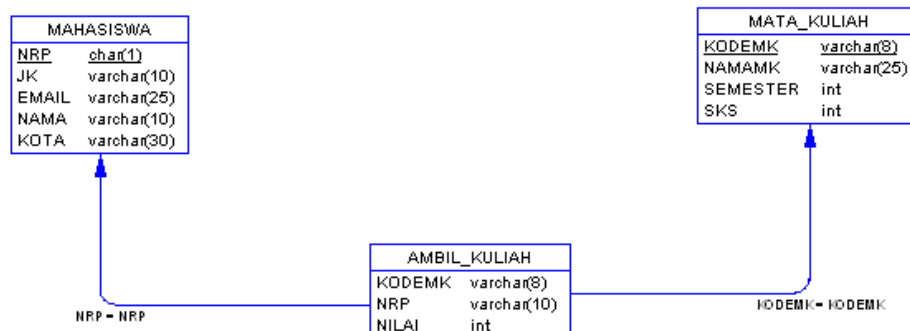
1. Join.
2. Inner Join.
3. Left Join(outter)
4. (outter) Right Join

## Kegiatan Inti

Pada kegiatan kali ini akan menggunakan database akademik\_db yang telah dibuat dan diisi datanya pada pertemuan-pertemuan sebelumnya. Pertemuan kali ini akan membahas bagaimana menampilkan data dari beberapa tabel sekaligus. Dengan skema database seperti gambar berikut

**Gambar 7.1:**

Desain ERD.



## Join

Untuk menampilkan data dari berapa table sekaligus dapat dilakukan dengan menggunakan operasi join. Perintah yang digunakan adalah sebagai berikut

```
Select <Nama Field1, Nama Field2, Nama Field3...> from <Nama Table1, Nama Table2, Nama Table3....> where <Kondisi>
```

Contoh :

Misalkan untuk menghasilkan tabel dalam bentuk seperti ini

NRP	Nama	KodeMK	Nilai

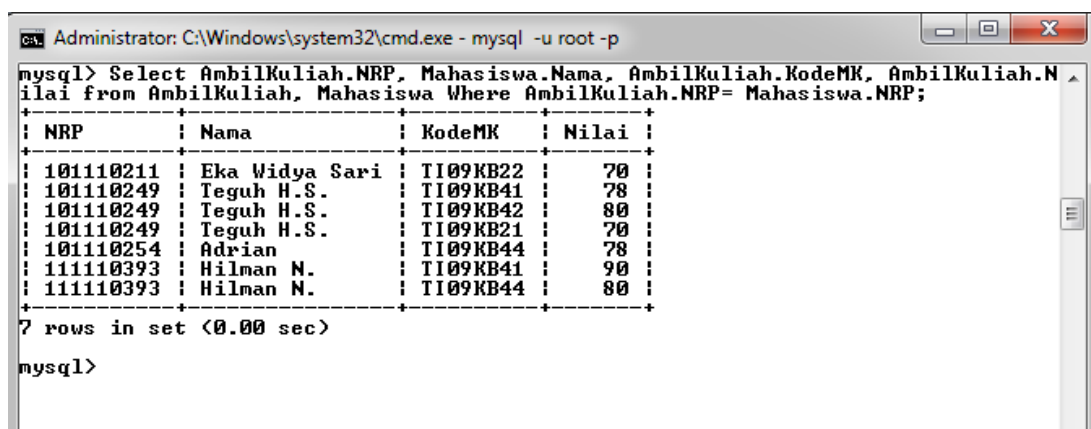
Dimana field Nama diambil dari Tabel Mahasiswa. Untuk menampilkan data dengan format tersebut perintah yang digunakan adalah

```
Select AmbilKuliah.NRP, Mahasiswa>Nama, AmbilKuliah.KodeMK, AmbilKuliah.Nilai from AmbilKuliah, Mahasiswa Where AmbilKuliah.NRP= Mahasiswa.NRP
```

Sehingga setelah perintah tersebut dijalankan MySql akan menampilkan seperti gambar berikut

Gambar 7.x:

Join



## Inner Join

Dengan menggunakan inner join perintah yang digunakan adalah sebagai berikut

```
Select <Nama Field1, Nama Field2, Nama Field2> from <Nama Table1> Inner Join <Nama Table2> On <Kondisi>
```

Contoh :

Untuk menampilkan data dengan bentuk seperti pada contoh Join diatas, dapat pula dilakukan dengan menggunakan Inner Join, perintah yang digunakan adalah

```
Select AmbilKuliah.NRP, Mahasiswa>Nama, AmbilKuliah.KodeMK, AmbilKuliah.Nilai from AmbilKuliah inner join Mahasiswa on AmbilKuliah.NRP= Mahasiswa.NRP;
```

Perhatikan tampilan yang muncul pada layar konsol dan bandingkan dengan contoh join biasa di atas.

## Left Join

Dengan menggunakan left join perintah yang digunakan adalah sebagai berikut

```
Select <Nama Field1, Nama Field2, Nama Field2> from <Nama Table1> left Join <Nama Table2> On <Kondisi>
```

Contoh :

Untuk menampilkan data dengan bentuk seperti pada contoh Join diatas, dapat pula dilakukan dengan menggunakan Inner Join, perintah yang digunakan adalah

```
Select AmbilKuliah.NRP, Mahasiswa>Nama, AmbilKuliah.KodeMK, AmbilKuliah.Nilai from AmbilKuliah left join Mahasiswa on AmbilKuliah.NRP= Mahasiswa.NRP;
```

Perhatikan tampilan yang muncul pada layar konsole dan bandingkan dengan contoh join biasa di atas.

## Right Join

Dengan menggunakan right join perintah yang digunakan adalah sebagai berikut

```
Select <Nama Field1, Nama Field2, Nama Field2> from <Nama Table1> right Join <Nama Table2> On <Kondisi>
```

Contoh :

Untuk menampilkan data dengan bentuk seperti pada contoh Join diatas, dapat pula dilakukan dengan menggunakan Inner Join, perintah yang digunakan adalah

```
Select AmbilKuliah.NRP, Mahasiswa>Nama, AmbilKuliah.KodeMK, AmbilKuliah.Nilai from AmbilKuliah right join Mahasiswa on AmbilKuliah.NRP= Mahasiswa.NRP;
```

Perhatikan tampilan yang muncul pada layar konsole dan bandingkan dengan contoh join biasa di atas.

## Kesimpulan

### Soal 7.1

Tuliskan perintah untuk menampilkan data dalam format seperti berikut

NRP	KodeMK	NamaMK	Nilai

.....

.....

.....

### Soal 7.2

Tuliskan perintah untuk menampilkan data dalam format seperti berikut

NRP	Nama	NamaMK	Nilai

.....

.....

.....

### Soal 7.3

Tuliskan perintah untuk menampilkan data dalam format seperti berikut, Hanya untuk Mahasiswa dengan nama „Adrian”

NRP	Nama	NamaMK	Nilai

.....

.....

.....

### Soal 7.4

Tuliskan perintah untuk menampilkan siapa saja yang mengambil mata kuliah „Matematika Diskrit”

.....

.....

.....

### Soal 7.5

Tuliskan perintah untuk menampilkan nilai rata-rata dari semua mahasiswa

.....

.....

.....

### Soal 7.6

Tuliskan perintah untuk menampilkan nilai rata-rata mahasiswa mahasiswa untuk setiap mata kuliah

.....

.....

.....

## Penutup

---

### Tugas

#### Tugas 7.1

Jelaskan perbedaan antara Full Join, Inner Join, Left Join dan Right Join

.....

.....

.....

.....

.....

.....

#### Tugas 7.2

Cobalah menggunakan perintah-perintah join pada database yang anda buat sendiri, tuliskan perintahnya disertai hasil output tampilan konsole. Cetak tugas dan kumpulkan kepada asisten masing-masing pada pertemuan selanjutnya.

---

## Kompetensi Dasar

Mahasiswa dapat menggunakan perintah SQL untuk membuat view.

---

## Indikator

1. Mahasiswa dapat membuat view
2. Mahasiswa dapat menghapus view

---

## Materi Pokok

1. View
2. Create view
3. Drop view

---

## Kegiatan Inti

---

### Membuat VIEW

---

Syntax :

```
CREATE VIEW <nama_view> AS <isi_perintah_view / action>;
```

Contoh :

```
mysql> CREATE VIEW tampil_nilai AS SELECT n.nrp, m.nama, n.kodemk,
mk.namamk, n.nilai FROM ambilmatakuliah n JOIN mahasiswa m ON m.nrp =
n.nrp JOIN matakuliah mk ON mk.kodemk = n.kodemk;
Query OK, 0 rows affected (0.07 sec)
```



## Menampilkan View yang telah dibuat

Syntax :

```
mysql> SELECT TABLE_NAME AS VIEWS FROM information_schema.VIEWS WHERE
TABLE_SCHEMA='<nama_tabel>';
```

Contoh :

```
mysql> SELECT TABLE_NAME AS VIEWS FROM information_schema.VIEWS WHERE
TABLE_SCHEMA='praktikum2015';
+-----+
| VIEWS          |
+-----+
| tampil_nilai   |
+-----+
1 row in set (0.00 sec)
```

## Menjalankan View

Syntax :

```
mysql> SELECT * FROM <nama_view>;
```

Contoh :

```
mysql> SELECT * FROM tampil_nilai;
+-----+-----+-----+-----+-----+
--+
| nrp      | nama          | kodemk | namamk          |
nilai |
+-----+-----+-----+-----+-----+
--+
| 102210055 | M. Chandra    | MK001  | Praktikum Basis Data |
90 |
| 102210055 | M. Chandra    | MK002  | Praktikum Sistem Operasi |
80 |
| 101110254 | Adrian       | MK001  | Praktikum Basis Data |
90 |
| 101110254 | Adrian       | MK002  | Praktikum Sistem Operasi |
90 |
+-----+-----+-----+-----+-----+
--+
4 rows in set (0.00 sec)
```

## Menghapus View

Syntax :

```
mysql> DROP VIEW <nama_tabel>;
```

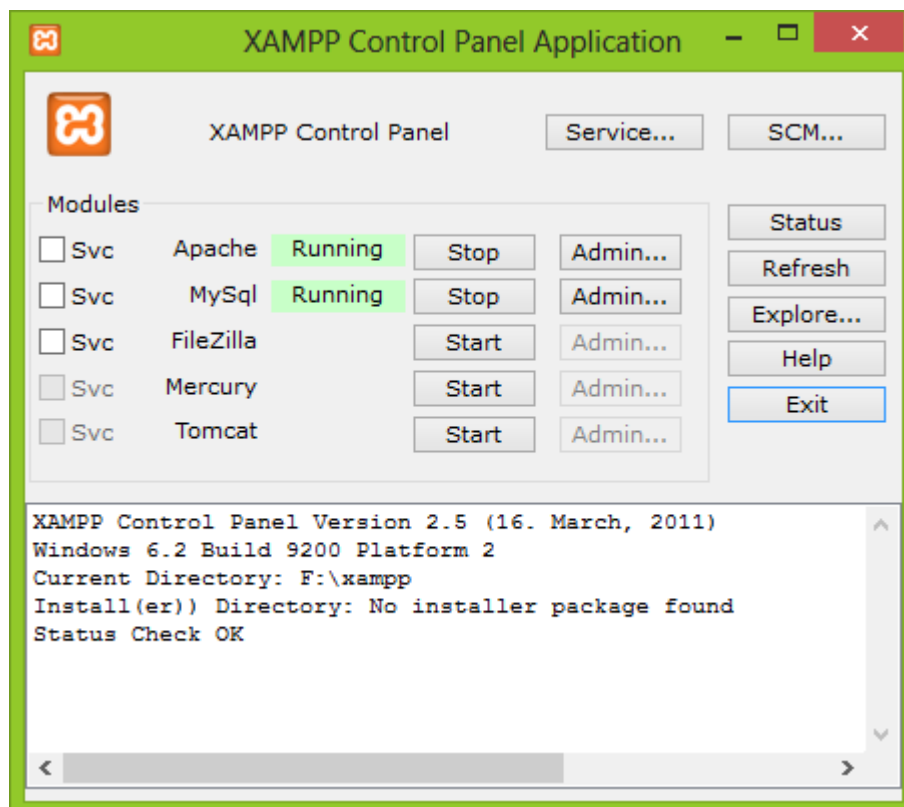
Contoh :

```
mysql> DROP VIEW tampil_nilai;  
Query OK, 0 rows affected (0.00 sec)
```

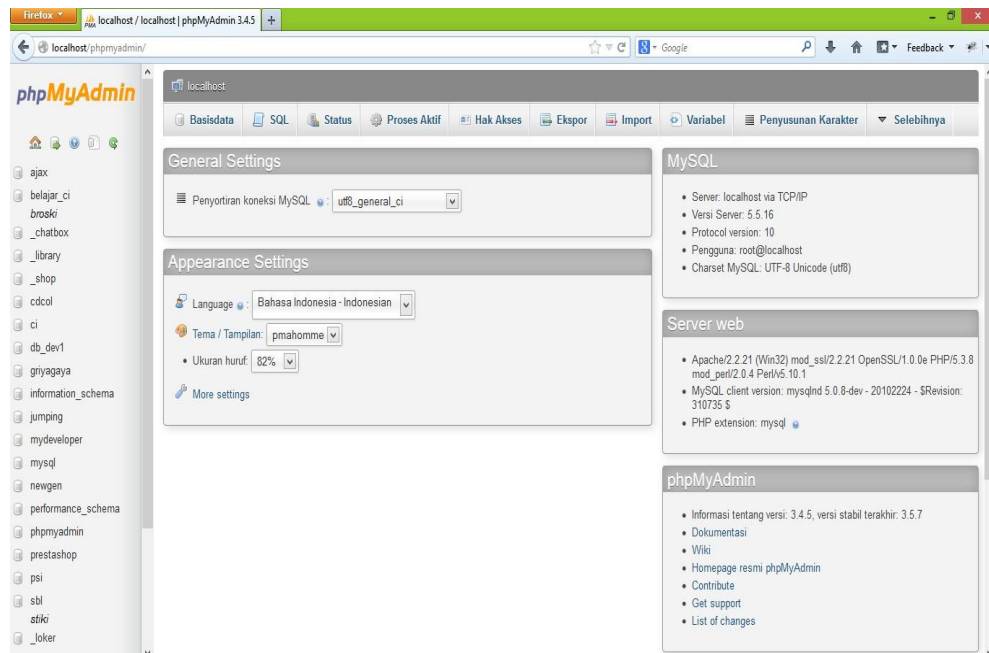
## Mengakses MySQL menggunakan phpMyAdmin (tambahan)

Untuk dapat mengakses MySQL menggunakan phpMyAdmin, pastikan service Apache telah running, untuk melihat apakah service Apache berjalan dapat kita cek melalui XAMPP Control Panel.

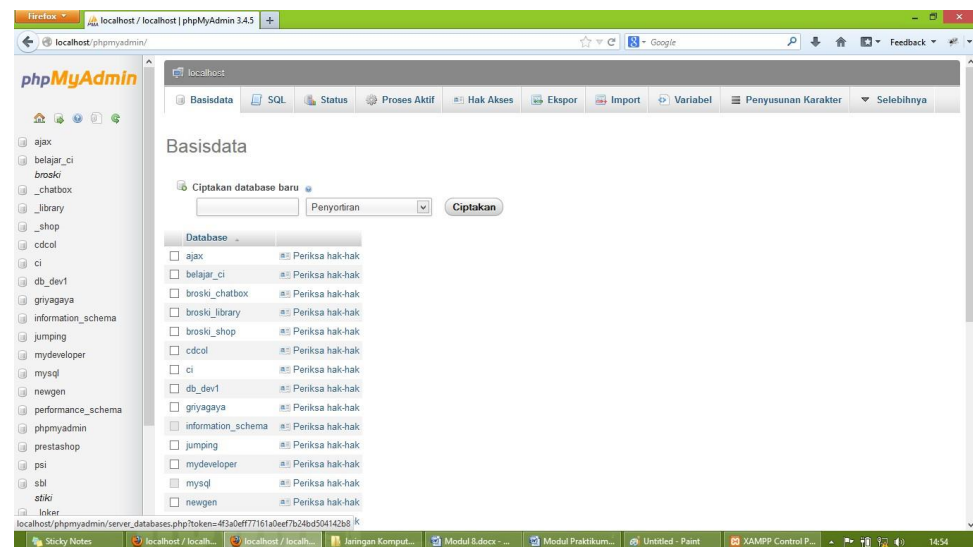
**Gambar 8.1:**  
XAMPP Control Panel



Untuk menjalankan service Apache, klik tombol **start** pada XAMPP Control Panel.

**Gambar 8.2:**Tampilan  
phpMyAdmin

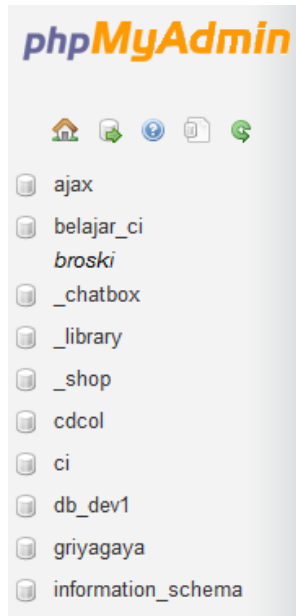
Buka Aplikasi Browser, pada address bar masukkan <http://localhost/phpmyadmin/>, tekan **enter/GO**.

**Gambar 8.3:**Membuat Database  
di phpMyAdmin

Untuk membuat database menggunakan phpMyAdmin, klik database / basisdata | masukkan pada nama database yang akan dibuat pada field yg disediakan | klik create / ciptakan.

**Gambar 8.4:**

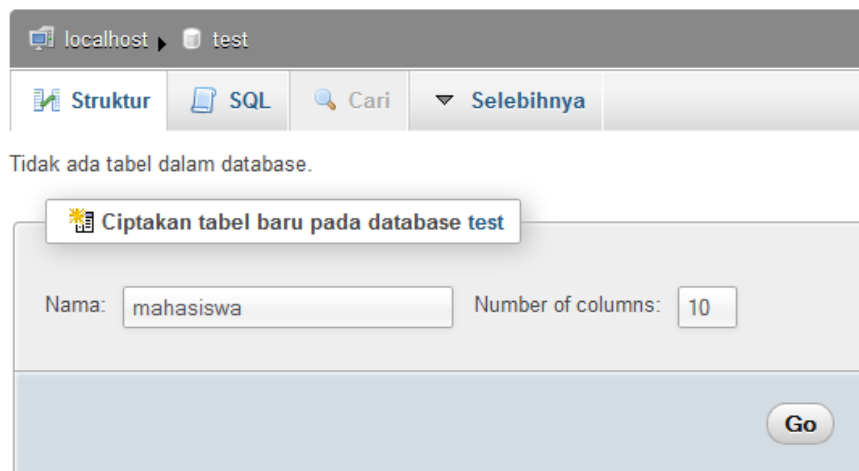
List menu database di phpMyAdmin



Untuk membuat table menggunakan phpMyAdmin, klik nama database yang terletak pada menu sebelah kiri phpMyAdmin .

**Gambar 8.5:**

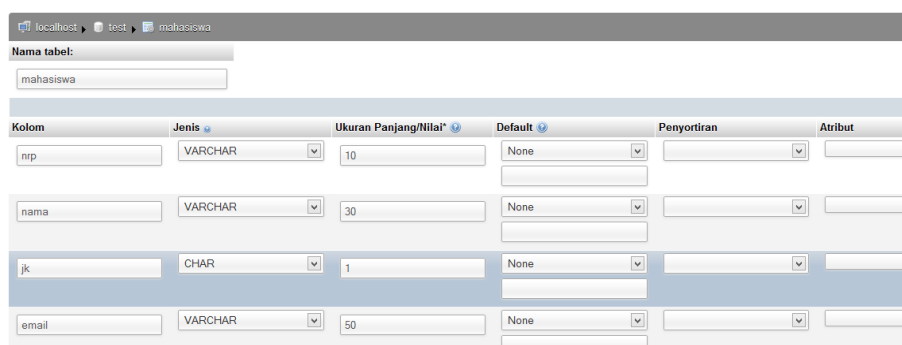
Membuat table di phpMyAdmin



Masukkan nama tabel dan jumlah coloum pada field yang disediakan da n klik **go**.

**Gambar 8.6:**

Membuat field dan menentukan tipe data di phpMyAdmin



Setelah menentukan nama table, maka akan muncul sebuah interface yang berisi field- field untuk menentukan field, tipe data, dan ukuran field yang akan dibuat. Setelah selesai, klik tombol simpan yang terletak di kanan bawah untuk menyimpan field dan konfigurasi.

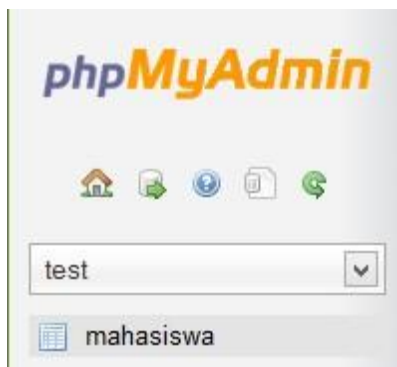
**Gambar 8.7:**

Tombol untuk menyimpan tabel



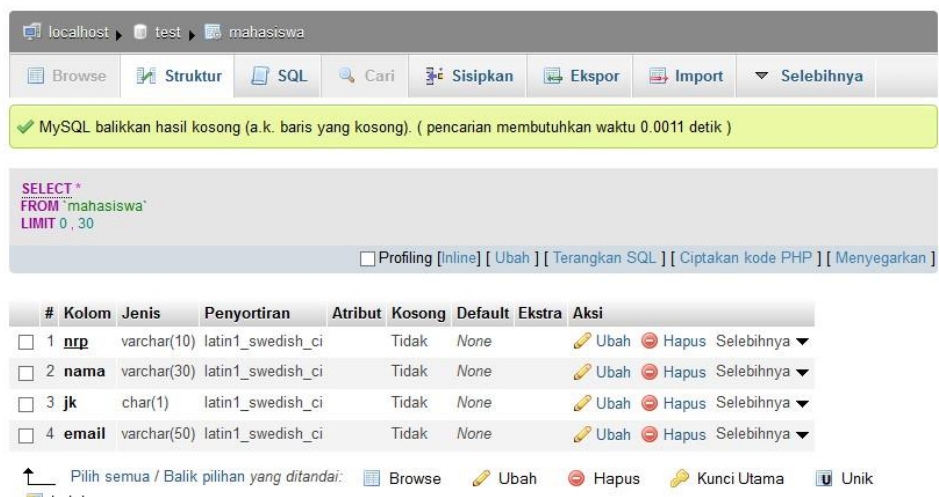
**Gambar 8.8:**

List tabel yang telah dibuat



**Gambar 8.9:**

Struktur table



Untuk melihat / membuka kembali tabel yang baru dibuat, klik nama table yang terletak di menu sisi kiri phpMyAdmin.

## Kesimpulan

Gunakan database transaksi pada modul sebelumnya untuk mengerjakan soal dibawah ini.

### Soal 8.1

1. Tulislah perintah untuk membuat view untuk menampilkan jumlah dan total harga semua barang.

.....

.....

.....

.....

2. Tulislah perintah untuk membuat view untuk menampilkan nama karyawan disetiap nota pembelian.

.....

.....

.....

.....

## Penutup

---

Gunakan database yang telah anda buat pada tugas modul sebelumnya untuk mengerjakan tugas dibawah ini.

## Tugas

### Tugas 8.1

Buatlah view dengan menggunakan tabel-tabel yang telah anda buat dan jelaskan fungsi dari view yang telah anda buat.

.....

.....

.....

.....