

MODUL 1

PENGENALAN MYSQL

MODUL 1 PENGENALAN MYSQL

Tujuan :

Setelah menyelesaikan modul ini, mahasiswa diharapkan dapat :

1. Mengetahui lingkungan kerja MySQL
2. Mampu melakukan instalasi tools pemrograman Basis Data MySQL/XAMPP
3. Mengetahui perintah-perintah sederhana di SQL

Pengenalan MySQL

MySQL adalah program *database server* yang mampu menerima dan mengirimkan datanya dengan cepat, *multi user* serta menggunakan perintah standar SQL (*Structure Query Language*). MySQL dikembangkan pertama kali pada tahun 1994 oleh sebuah perusahaan pengembang *software* bernama TcX Data Konsult AB yang berada di Swedia. Tujuan awal dikembangkannya MySQL adalah untuk mengembangkan aplikasi berbasis web pada *client*. MySQL merupakan *database open source* yang populer di dunia. MySQL memiliki fitur-fitur yang sangat baik, sehingga sangat cocok untuk digunakan dalam implementasi aplikasi basis data, khususnya yang berbasis *web*. MySQL adalah *database* yang paling banyak dipakai karena relatif simpel dan mudah digunakan. MySQL dimiliki oleh Oracle Corporation sebuah perusahaan skala besar di bidang basis data (yang sebelumnya juga sudah memiliki DBMS yang paling populer yaitu Oracle).

Format Perintah

Berikut adalah ketentuan-ketentuan memberi perintah pada MySQL :

1. Setiap perintah harus diakhiri dengan tanda titik koma, kecuali untuk perintah tertentu, misal : *quit*.

2. Setiap perintah akan disimpan dalam *buffer* (memori sementara) untuk menyimpan histori perintah-perintah yang pernah diberikan.
3. Perintah dapat berupa perintah SQL atau perintah khusus MySQL.
4. Perintah-perintah dalam lingkungan MySQL tidak menerapkan aturan *case sensitive*. Aturan *case insensitive* yaitu perintah bisa dituliskan dalam huruf besar atau pun huruf kecil.
5. Aturan *case sensitive* diterapkan pada penamaan objek-objek dalam *database* seperti nama *database* atau nama *table*, namun aturan ini hanya ada dalam lingkungan Unix dan Linux.

Ada beberapa tanda yang sering muncul di prompt :

Prompt	Arti
mysql>	Siap menerima perintah baru
->	Menunggu baris berikut untuk perintah yang lebih dari satu baris
'>	Menunggu baris berikut, menunggu penutup string yang dimulai dengan tanda kutip satu (""")
">	Menunggu baris berikut, menunggu penutup string yang dimulai dengan tanda kutip dua (""")
`>	Menunggu baris berikutnya, menunggu penutup identifier yang dimulai dengan tanda backtick (""")

Perintah Pada SQL

SQL (*Structured Query Language*) adalah sebuah bahasa permintaan *database* yang terstruktur. Bahasa SQL ini dibuat sebagai bahasa yang dapat merelasikan beberapa tabel dalam *database* maupun merelasikan antar *database*. SQL terbagi ke dalam tiga bentuk query yaitu :

1. *Data Definition Language* (DDL)
2. *Data Manipulation Language* (DML)
3. *Data Control Language* (DCL)

Data Definition Language (DDL)

DDL adalah sebuah metode Query SQL yang berguna untuk mendefinisikan data pada sebuah *database*, Query yang dimiliki DDL adalah :

- CREATE : Digunakan untuk membuat *Database* dan Tabel
- Drop : Digunakan untuk menghapus *Database* dan Tabel

- Alter : Digunakan untuk melakukan perubahan struktur tabel yang telah dibuat, baik menambah *Field (Add)*, mengganti nama *Field (Change)* ataupun menamakannya kembali (*Rename*) dan menghapus *Field (Drop)*.

Data Manipulation Language (DML)

DML adalah sebuah metode Query yang dapat digunakan apabila DDL telah terjadi, sehingga fungsi dari Query DML ini untuk melakukan pemanipulasian *database* yang telah dibuat.

Query yang dimiliki DML adalah :

- INSERT : Digunakan untuk memasukkan data pada tabel *Database*
- UPDATE : Digunakan untuk pengubahan terhadap data yang ada pada tabel *Database*.
- DELETE : Digunakan untuk Penghapusan data pada tabel *Database*.

Data Control Language (DCL)

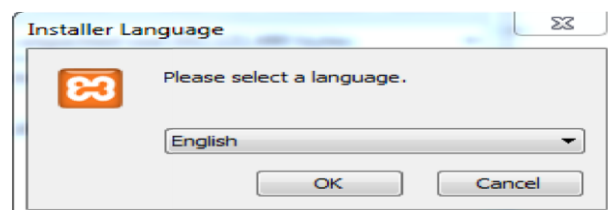
DCL adalah sebuah metode Query SQL yang digunakan untuk memberikan hak otorisasi mengakses *Database*, mengalokasikan *space*, pendefinisian *space* dan pengauditan penggunaan *database*. Query yang dimiliki DCL adalah :

- GRANT : Untuk mengizinkan *user* mengakses Tabel dalam *Database*.
- REVOKE : Untuk membatalkan izin hak *user*, yang ditetapkan oleh perintah GRANT
- COMMIT : Menetapkan penyimpanan *Database*
- ROLLBACK : Membatalkan penyimpanan *Database*

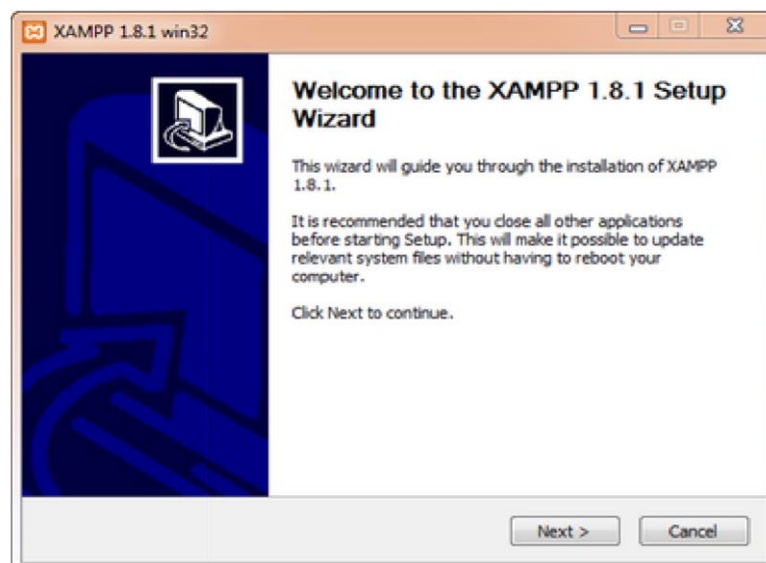
LANGKAH INSTALASI XAMPP

Cara Instalasi XAMPP

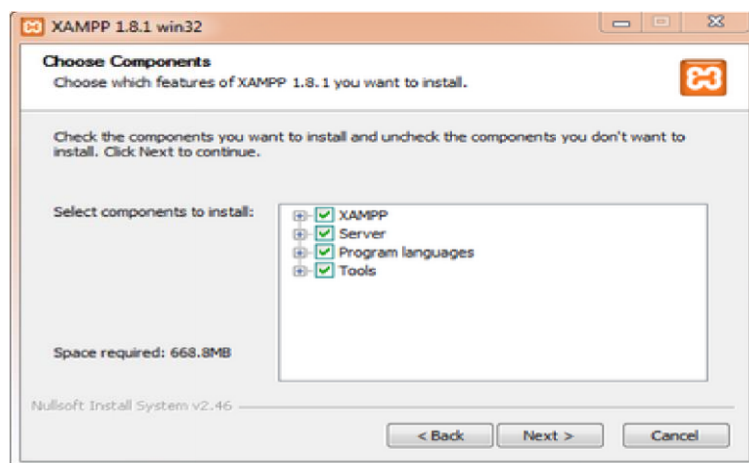
1. Buka *file* instalasi XAMPP yang sudah di *download*.
2. Dobel klik *file* XAMPP yang ada, selanjutnya akan muncul jendela ***installer language*** seperti di bawah ini :



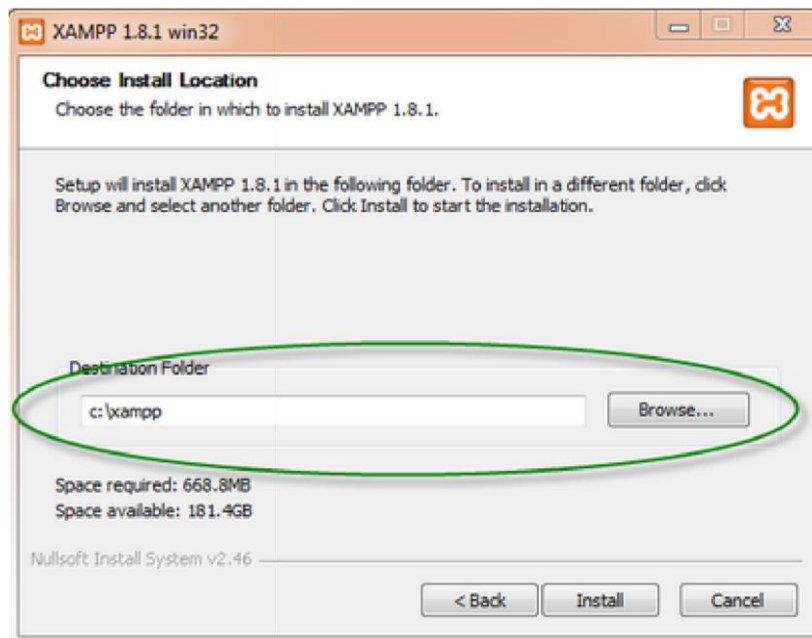
3. Selanjutnya pilih bahasa. Pilih yang Bahasa Inggris (English). Klik OK.
4. Kadang pada proses ini muncul pesan *error*. Jika ada, abaikan saja dan lanjutkan dengan klik OK dan YES.
5. Berikutnya akan muncul jendela yang isinya meminta *user* menutup semua aplikasi yang sedang berjalan. Jika semua aplikasi sudah ditutup, maka klik tombol **Next**.



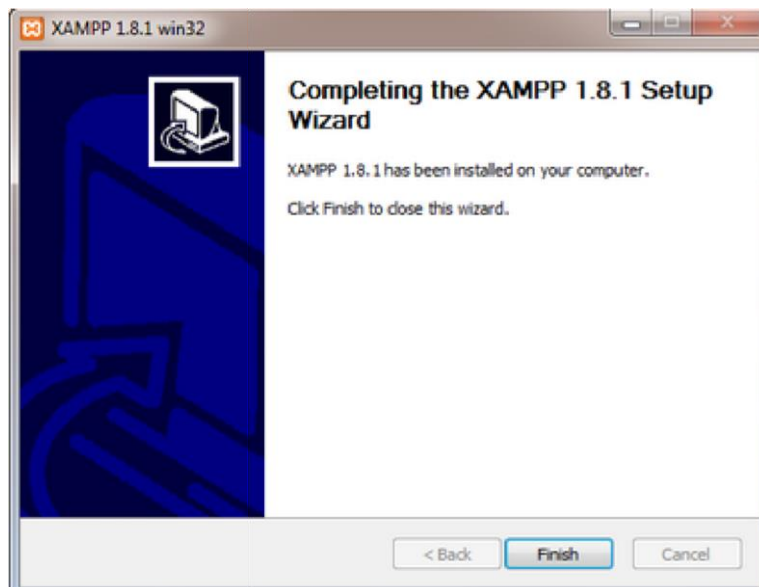
6. Selanjutnya akan diminta untuk memilih aplikasi yang mau diinstal. Centang semua pilihan dan klik tombol **Next**.



7. Kemudian akan diminta untuk menentukan lokasi *folder* penyimpanan *file-file* dan *folder* XAMPP. Secara *default* akan diarahkan ke lokasi **c:\xampp**. Namun jika ingin menyimpannya di *folder* lain, klik **browse** dan tentukan secara manual *folder* yang ingin digunakan. Jika sudah selesai, lanjutkan dan klik tombol **Install**.



8. Tunggu beberapa menit hingga proses instalasi selesai. Jika sudah muncul jendela seperti di bawah ini, klik tombol **Finish** untuk menyelesaikannya.

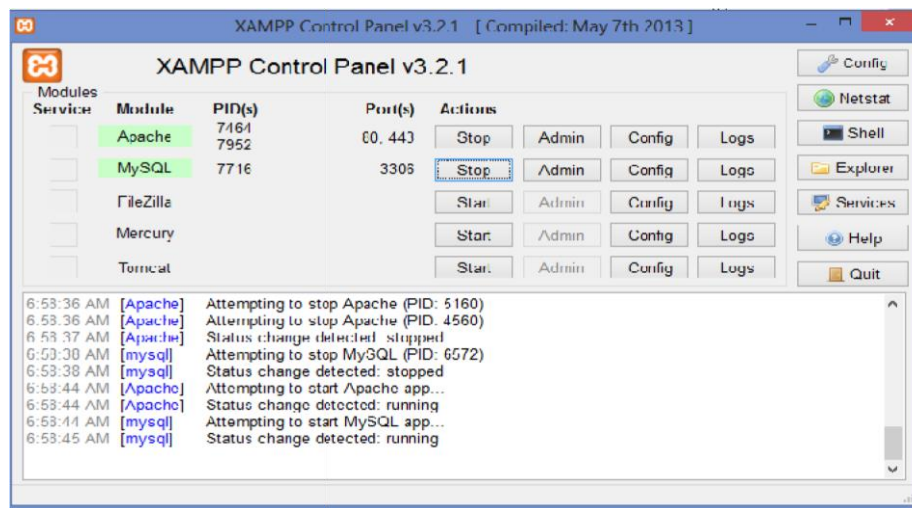


9. Berikutnya, akan muncul jendela dialog seperti gambar di bawah ini yang menanyakan apakah mau langsung menjalankan aplikasi XAMPP atau tidak. Jika ya, maka klik YES.



Cara Menjalankan Aplikasi XAMPP

1. Bukalah aplikasi XAMPP, bisa melalui **Start Menu** atau **Desktop**, dan klik icon XAMPP. Aktifkan Xampp Control Panel Application, klik start apache dan mysql.



2. Kemudian aktifkan *command prompt*, lalu ketik perintah `cd c://xampp/mysql/bin` sehingga hasilnya seperti gambar berikut:

```
Microsoft Windows [Version 10.0.17134.1130]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\lenovo>cd c://xampp/mysql/bin

c:\xampp\mysql\bin>mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 33
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> _
```

3. Sedangkan untuk keluar dari MySQL dapat menggunakan perintah : \q, exit dan quit.

```
Microsoft Windows [Version 10.0.17134.1130]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\lenovo>cd c://xampp/mysql/bin

c:\xampp\mysql\bin>mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 33
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.

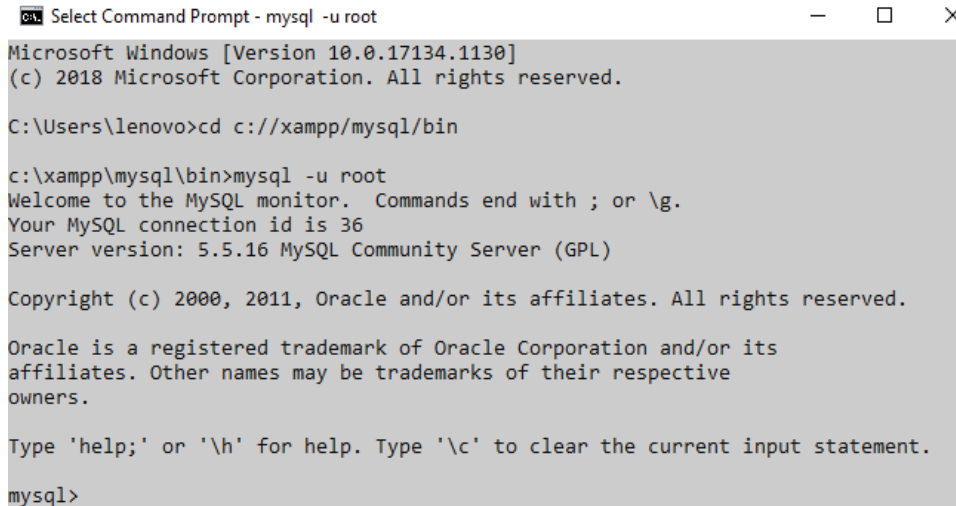
mysql> \q
Bye

c:\xampp\mysql\bin>
```

LANGKAH PRAKTIKUM

Memulai Koneksi

Saat DBMS MySQL pertama kali di instalasi, akan ada seorang *user* bernama **root** yang bertindak sebagai *superuser/administrator*. Koneksi dilakukan dari *prompt shell*. Sebagai contoh jika menggunakan sistem operasi Windows maka menggunakan **command prompt**.



```
Select Command Prompt - mysql -u root
Microsoft Windows [Version 10.0.17134.1130]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\lenovo>cd c://xampp/mysql/bin

c:\xampp\mysql\bin>mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 36
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>
```

Pembuatan Basis Data

Untuk melihat basis data yang sudah ada, gunakan perintah **show databases**:

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| cdcol      |
| mysql      |
| performance_schema |
| phpmyadmin |
| test       |
| webauth    |
+-----+
7 rows in set (0.00 sec)
```

Basis data dapat dibuat dengan perintah **create database**. Misal akan dibuat basis data kepegawaian, perintahnya dapat ditulis seperti tampilan berikut ini :

```
mysql> create database kepegawaian;
Query OK, 1 row affected (0.00 sec)
```


Untuk melihat apakah basis data kepegawaian berhasil dibuat, gunakan lagi perintah **show databases**:

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| cdcol |
| kepegawaian |
| mysql |
| performance_schema |
| phpmyadmin |
| test |
| webauth |
+-----+
8 rows in set (0.00 sec)
```

Jika basis data sudah berhasil dibuat, basis data tersebut tidak otomatis langsung terbuka. Untuk membuka basis data tersebut gunakan perintah **use**.

Untuk melihat tabel apakah sudah ada dalam basis data, gunakan perintah **show tables**. Perintah *show tables* akan menampilkan tabel hanya dari basis data yang aktif saja yaitu, basis data kepegawaian dan hasilnya tabel masih belum ada.

```
mysql> use kepegawaian;
Database changed
mysql> show tables;
Empty set (0.00 sec)
```

Membuat Tabel

Buat tabel pertama dengan nama tabel **divisi**. Tabel divisi memiliki dua *field* yaitu **id_div** dan **nama_divisi**. *Field* id_div sebagai *primary key* dengan tipe char sebanyak 3 karakter. sedangkan *field* nama_divisi dengan tipe data varchar dan memiliki maksimal 10 karakter.

```
mysql> create table divisi
-> (id_div char(3) primary key,
-> nama_divisi varchar(20));
Query OK, 0 rows affected (0.01 sec)
```

Untuk melihat apakah tabel divisi sudah ada didaftar tabel, gunakan perintah **show tables**:

```
mysql> show tables;
+-----+
| Tables_in_kepegawaian |
+-----+
| divisi |
+-----+
1 row in set (0.00 sec)
```

Tabel divisi telah ada didalam basis data kepegawaian. Untuk melihat struktur tabel divisi, gunakan perintah **desc**:

```
mysql> desc divisi;
+-----+-----+-----+-----+-----+-----+
| Field      | Type        | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_div     | char(3)     | NO   | PRI | NULL    |       |
| nama_divisi | varchar(20) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.03 sec)
```

Dengan cara yang sama tambahkan tabel kedua bernama tabel pegawai. Tabel pegawai memiliki *field* nip, nama dan alamat.

```
mysql> create table pegawai
-> (nip char(5) primary key not null,
-> nama varchar(20),
-> alamat varchar(50));
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> desc pegawai;
+-----+-----+-----+-----+-----+-----+
| Field      | Type        | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nip        | char(5)     | NO   | PRI | NULL    |       |
| nama       | varchar(20) | YES  |     | NULL    |       |
| alamat     | varchar(50) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
```

Buat tabel yang ketiga dan beri nama tabel biaya. Tabel biaya memiliki *field* nip, kda, tglupdate dan rupiah.

```
mysql> create table biaya
-> (nip char(5),
-> kda int,
-> tglupdate datetime,
-> rupiah bigint);
Query OK, 0 rows affected (0.03 sec)
```

Mengisi Data Tabel

Ada tiga tabel yang sudah dibuat yaitu tabel divisi, tabel pegawai dan tabel biaya. Tabel tersebut masih kosong. Untuk mengisi tabel gunakan perintah **insert** :

```
mysql> insert into pegawai values
-> ('11001','Ajeng Putri','Jl. Pancasila No.100 Pontianak');
Query OK, 1 row affected (0.00 sec)
```

Perintah di atas berarti menambahkan sebuah *record* baru dengan isi dari *field* berturut-turut adalah '11001', 'Ajeng Putri', 'Jl. Pancasila No. 100 Pontianak'. Sesuai dengan urutan deskripsi tabel pegawai yaitu nip, nama dan alamat.

Kemudian isi *record* dengan data seperti berikut :

```
mysql> insert into pegawai values
-> ('11002','Deri Darmawan','Jl. Merdeka No.8 Pontianak');
Query OK, 1 row affected (0.01 sec)

mysql> insert into pegawai values
-> ('11003','Ira Kartika','Jl. Perjuangan No.10 Pontianak');
Query OK, 1 row affected (0.01 sec)
```

Melihat Isi Tabel

Untuk melihat isi tabel pegawai, gunakan perintah **select** :

```
mysql> select * from Pegawai;
+-----+-----+-----+
| nip   | nama           | alamat                                     |
+-----+-----+-----+
| 11001 | Ajeng Putri    | Jl. Pancasila No.100 Pontianak          |
| 11002 | Deri Darmawan  | Jl. Merdeka No.8 Pontianak              |
| 11003 | Ira Kartika    | Jl. Perjuangan No.10 Pontianak          |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

Perintah diatas akan memunculkan semua data yang ada ditabel pegawai, tanpa syarat, tanpa memilih kolom dan tanpa urutan. Karakter '*' (bintang) menunjukkan semua *field* dibaca untuk dalam kasus ini ditampilkan secara tabular.

Membatasi Jumlah Record yang Dibaca

Untuk membatasi *record* yang muncul atau untuk mencari *record* dengan kriteria tertentu, dapat digunakan klausa **where**. Misal untuk melihat pegawai dengan nip='11002':

```
mysql> select * from pegawai where nip='11002';
+-----+-----+-----+
| nip   | nama           | alamat                                     |
+-----+-----+-----+
| 11002 | Deri Darmawan  | Jl. Merdeka No.8 Pontianak              |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Perintah untuk melihat pegawai yang beralamat diPontianak:

```
mysql> select * from pegawai where alamat like '%Pontianak';
```

nip	nama	alamat
11001	Ajeng Putri	Jl. Pancasila No.100 Pontianak
11002	Deri Darmawan	Jl. Merdeka No.8 Pontianak
11003	Ira Kartika	Jl. Perjuangan No.10 Pontianak

```
3 rows in set (0.00 sec)
```

Membatasi Jumlah Field yang Dibaca

Perintah untuk melihat nip dan nama pegawai:

```
mysql> select nip, nama from pegawai;
```

nip	nama
11001	Ajeng Putri
11002	Deri Darmawan
11003	Ira Kartika

```
3 rows in set (0.00 sec)
```

Untuk menampilkan nip dan nama pegawai dengan nip tertentu dapat digunakan perintah seperti tampilan berikut:

```
mysql> select nip, nama from pegawai where nip='11003';
```

nip	nama
11003	Ira Kartika

```
1 row in set (0.00 sec)
```

Mengubah Isi Data Tabel

Untuk melakukan pengubahan pada isi data tabel yang telah dibuat, digunakan perintah **update**. Perintah *update* harus dilakukan secara hati-hati karena untuk kondisi tertentu dapat mengubah isi keseluruhan *record* tabel. Biasanya perintah ini dikombinasikan dengan *filter record* (menggunakan klausa **where**). Misal dilakukan perubahan alamat pada pegawai dengan nip=11001

```
mysql> update pegawai set alamat='Jl. Ayani No. 5 Pontianak'
-> where nip='11001';
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> select * from pegawai;
```

nip	nama	alamat
11001	Ajeng Putri	Jl. Ayani No. 5 Pontianak
11002	Deri Darmawan	Jl. Merdeka No.8 Pontianak
11003	Ira Kartika	Jl. Perjuangan No.10 Pontianak

```
3 rows in set (0.00 sec)
```

Menghapus Data Tabel

Untuk menghapus data pada *record* tertentu menggunakan perintah **delete from**. Perintah ini sering dikombinasi dengan klausa **where**. Pemilihan kondisi where harus dipastikan hanya berefek pada *record* yang memang memenuhi kriteria untuk dihapus.

```
mysql> delete from pegawai where nip='11003';
Query OK, 1 row affected (0.00 sec)
```

```
mysql> select * from pegawai;
```

nip	nama	alamat
11001	Ajeng Putri	Jl. Ayani No. 5 Pontianak
11002	Deri Darmawan	Jl. Merdeka No.8 Pontianak

```
2 rows in set (0.00 sec)
```

Mengubah Struktur Tabel

Pengubahan struktur tabel meliputi penambahan *field*, pengurangan *field*, pengubahan tipe data atau ukuran *field*, pergantian nama *field* dan sebagainya. Perintah untuk pengubahan adalah **alter table**. Misal pada tabel pegawai ditambahkan *field* baru bernama **id_div** dengan tipe data char berukuran 3 karakter.

```
mysql> alter table pegawai add column id_div char(3);
Query OK, 2 rows affected (0.05 sec)
Records: 2 Duplicates: 0 Warnings: 0
```

Secara *default field* baru selalu diletakkan terakhir, sehingga pada struktur diatas *field id_div* ada diposisi setelah alamat.

```
mysql> desc pegawai;
```

Field	Type	Null	Key	Default	Extra
nip	char(5)	NO	PRI	NULL	
nama	varchar(20)	YES		NULL	
alamat	varchar(50)	YES		NULL	
id_div	char(3)	YES		NULL	

```
4 rows in set (0.05 sec)
```

Karena *field* baru **id_div** belum diisi maka tabel pegawai hasilnya seperti tampilan berikut yaitu *field* id_div bernilai null:

```
mysql> select * from pegawai;
```

nip	nama	alamat	id_div
11001	Ajeng Putri	Jl. Ayani No. 5 Pontianak	NULL
11002	Deri Darmawan	Jl. Merdeka No.8 Pontianak	NULL

```
2 rows in set (0.00 sec)
```

Menghapus Tabel

Penghapusan tabel berbeda dengan penghapusan isi tabel. Pada operasi ini tabel dihapus dari daftar objek yang ada di DBMS. Perintah yang digunakan adalah **drop table**:

```
mysql> show tables;
```

Tables_in_kepegawaian
biaya
divisi
pegawai

```
3 rows in set (0.00 sec)
```

```
mysql> drop table pegawai;
```

Query OK, 0 rows affected (0.01 sec)

Tabel pegawai telah berhasil dihapus, tabel yang ada di basis data kepegawaian adalah tabel biaya dan divisi.

```
mysql> select * from pegawai;
ERROR 1146 (42S02): Table 'kepegawaian.pegawai' doesn't exist
mysql> show tables;
+-----+
| Tables_in_kepegawaian |
+-----+
| biaya                  |
| divisi                 |
+-----+
2 rows in set (0.00 sec)
```