

Membuat database


Perintah dalam membuat database dengan cara “*create database nama_database ;*” dan ingat selalu diakhiri dengan titik koma “*;*”

Contoh membuat database latihan, lalu enter lalu akan muncul tulisan yang artinya database telah dibuat

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

Jika ingin melihat database sudah dibuat makanya ketik perintah “*show databases ;*” lalu enter.

```
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| kampus |  
| login |  
| mysql |  
| pegawai |  
| performance_schema |  
| sekolah |  
| test |  
+-----+  
8 rows in set (0.00 sec)
```



Menghapus database

Jika suatu saat database yang buat ingin dihapus maka ketik perintah “*drop database nama_database ;*”. **INGAT** dan **HATI-HATI** jika menghapus database karena sudah dihapus maka tidak bisa *restore* kembali. Apalagi

system pada Mysql tidak memberikan konfirmasi ataupun pertanyaan jika dihapus maka langsung hilang selamanya.

Contoh menghapus database latihan, maka perintahnya “*drop database nama_databases ;*” lalu enter muncul pemberitahuan database telah dihapus.

```
mysql> drop database pegawai;  
Query OK, 0 rows affected (0.01 sec)
```

Makanya database yang terhapus sebelumnya sudah tidak ada.

```
mysql> show databases;  
+-----+  
| Database  
+-----+  
| information_schema  
| kampus  
| login  
| mysql  
| performance_schema  
| sekolah  
| test  
+-----+  
7 rows in set (0.00 sec)
```

Memilih atau Menggunakan Database

Sebelum pembuatan table sebelumnya harus memilih database mana yang akan digunakan, tidak dipungkiri banyak sekali database dalam suatu aplikasi di database tersebut. Sebelumnya buat dahulu database yang baru dengan nama **karyawan**.

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

Setelah itu menggunakan database tersebut maka ketik perintah “*use nama_database ;*” lalu enter muncul pemberitahuan bahwa database tersebut siap digunakan.

```
mysql> use karyawan;  
Database changed
```

Membuat Table

Jika sudah digunakan database tersebut maka membuat tablenya. Perlu diingat bahwa kumpulan beberapa table disebut dengan database, yang pasti table yang berada di database tersebut masih kosong. Table merupakan kumpulan beberapa field, maka untuk membuat table maka harus tentukan field/kolom dalam table tersebut.

Sebagai contoh membuat table **sdm** pada database **karyawan**, makanya perintahnya ;

```
mysql> create table sdm  
-> (id char (5) primary key,  
-> nama varchar (15) not null,  
-> jenis_kelamin varchar (10) not null,  
-> alamat varchar (15) not null);  
Query OK, 0 rows affected (0.01 sec)
```

Pada perintah di atas jika dijabarkan :

- create table merupakan perintah di dalam pembuatan table

- sdm merupakan nama table
- id, merupakan field yang sebagai primary key yang artinya berbeda antara baris yang lain dan bersifat unik
- nama, jenis_kelamin, alamat merupakan nama field yang lain
- varchar dan char merupakan tipe data
- not null merupakan option bahwa data tidak boleh kosong
- 5, 25, 2, 25 merupakan panjang karakter dari suatu field

Melihat Table

Jika ingin melihat apakah table tersebut sudah ada di dalam database maka perintahnya “*show tables ;*” lalu enter maka terlihat satu baris table telah dibuat di dalam database karyawan.

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

Melihat Struktur Table

Jika lupa akan field yang telah dibuat sebelumnya beserta tipe dan panjang data pada table sdm, maka perintahnya “*desc nama_table ;*” lalu enter maka terlihat 4 baris nama field, type dan panjang data dan penjelasan yang lainnya.

```
mysql> desc sdm;
```

Field	Type	Null	Key	Default	Extra
id	char(5)	NO	PRI	NULL	
nama	varchar(15)	NO		NULL	
jenis_kelamin	varchar(10)	NO		NULL	
alamat	varchar(15)	NO		NULL	

4 rows in set (0.00 sec)

Menghapus Table

Jika ingin menghapus table perintahnya “*drop table nama_table ;*”. Sebagai contoh buat table **sdm1**, samakan saja struktur yang ada pada table sdm. Ingat kembali bahwa jika sudah terhapus maka tidak dapat diambil kembali dan tidak peringatan maupun keterangan jika akan dihapus, maka hati-hati jika ingin menghapus suatu table maupun database.

Terlihat 2 baris table pada database **karyawan**.

```
mysql> show tables;
```

Tables_in_karyawan
sdm
sdm1

2 rows in set (0.00 sec)

Untuk menghapus table **sdm1** lalu muncul perintah benar.

```
mysql> drop table sdm1;  
Query OK, 0 rows affected (0.00 sec)
```

Terlihat bahwa table sdm1 telah terhapus dan hanya satu baris table sdm pada database karyawan.

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

Merubah Nama Table

Kadang penamaan suatu table membuat ribet dan ingin dibuat simple, untuk merubah nama table, maka perintahnya “*rename table nama_table yang diganti to nama_table yang baru ;*”. Sebagai contoh nama table **sdm** diganti menjadi **pegawai** lalu enter muncul keterangan perintah OK.

```
mysql> rename table sdm to pegawai;  
Query OK, 0 rows affected (0.00 sec)
```

Maka table sdm telah berubah menjadi pegawai pada database karyawan.

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

Menambah Field

Jika ingin menambahkan field dalam suatu table yang telah dibuat sebelumnya maka perintah “*alter table nama_table add nama_field tipe dan panjang data ;*”. Sebagai contoh ingin menambahkan field **status** dan **kota** pada table **pegawai**.

```
mysql> alter table pegawai
-> add status varchar (15) not null,
-> add kota varchar (15) not null;
Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Jika dilihat strukturnya maka bertambah field yang dibuat sebelumnya.

```
mysql> desc pegawai;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id    | char(5) | NO | PRI | NULL |       |
| name  | varchar(15) | NO |     | NULL |       |
| jenis_kelamin | varchar(10) | NO |     | NULL |       |
| alamat | varchar(15) | NO |     | NULL |       |
| status | varchar(15) | NO |     | NULL |       |
| kota  | varchar(15) | NO |     | NULL |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

Menambah Field Dengan Posisi Yang Diinginkan.

Mungkin ada pertanyaan bagaimana susunan struktur field berdasarkan yang diinginkan, biasanya memasukkan field baru pasti dibawah field yang lama, maka perintahnya “*alter table nama_table add nama_field tipe dan panjang after nama_filed yang akan disisipkan sebelumnya*”. Sebagai contoh masukkan field dengan nama tanggal_lahir tetapi field tersebut setelah jenis_kelamin.

```
mysql> alter table pegawai
-> add tanggal_lahir date after jenis_kelamin;
Query OK, 0 rows affected (0.00 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Terlihat hasilnya struktur table sudah bertambah dan posisi field tanggal_lahir setelah field jenis_kelamin.

```
mysql> desc pegawai;
```

Field	Type	Null	Key	Default	Extra
id	char(5)	NO	PRI	NULL	
nama	varchar(15)	NO		NULL	
jenis_kelamin	varchar(10)	NO		NULL	
tanggal_lahir	date	YES		NULL	
alamat	varchar(15)	NO		NULL	
status	varchar(15)	NO		NULL	
kota	varchar(15)	NO		NULL	

7 rows in set (0.00 sec)

Merubah Nama Field, Tipe Data dan Panjang Data.

Jika ingin merubah field dari nama hingga panjang karakter maka tambahkan perintah “*change nama_field nama_baru_field tipe (karakter) panjang_karakter ;*”. Sebagai contoh ingin merubah field jenis_kelamin menjadi jenkel dengan tipe char karakter (1) dan not null.

```
mysql> alter table pegawai
-> change jenis_kelamin jenkel char (1) not null;
Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Terlihat perubahan struktur pada field jenis_kelamin menjadi jenkel beserta tipe data dan panjang data.


```
mysql> desc pegawai;
```

Field	Type	Null	Key	Default	Extra
id	char(5)	NO	PRI	NULL	
nama	varchar(15)	NO		NULL	
jenkel	char(1)	NO		NULL	
tanggal_lahir	date	YES		NULL	
alamat	varchar(15)	NO		NULL	
status	varchar(15)	NO		NULL	
kota	varchar(15)	NO		NULL	

7 rows in set (0.02 sec)

Merubah Panjang Data Pada Field.

Jika ingin merubah panjang karakter field tersebut maka tambahkan “*alter table nama_table modify nama_field tipe_data panjang_karakter ;*”. Sebagai contoh merubah karakter dari field kota yang sebelumnya 15 menjadi 25.

```
mysql> alter table pegawai
-> modify kota varchar (25) not null;
Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Terlihat perubahan struktur field yang telah dirubah, maka terlihat panjang karakter pada field kota telah berganti.

```
mysql> desc pegawai;
```

Field	Type	Null	Key	Default	Extra
id	char(5)	NO	PRI	NULL	
nama	varchar(15)	NO		NULL	
jenkel	char(1)	NO		NULL	
tanggal_lahir	date	YES		NULL	
alamat	varchar(15)	NO		NULL	
status	varchar(15)	NO		NULL	
kota	varchar(25)	NO		NULL	

7 rows in set (0.01 sec)

Menghapus Field

Apabila salah satu field ingin dihapus maka perintahnya “*alter table nama_table drop nama_field*”, sebagai contoh menghilangkan field kota.

```
mysql> alter table pegawai
-> drop kota;
Query OK, 0 rows affected (0.00 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Untuk melihat struktur table pegawai field kota sudah terhapus.

```
mysql> desc pegawai;
```

Field	Type	Null	Key	Default	Extra
id	char(5)	NO	PRI	NULL	
nama	varchar(15)	NO		NULL	
jenkel	char(1)	NO		NULL	
tanggal_lahir	date	YES		NULL	
alamat	varchar(15)	NO		NULL	
status	varchar(15)	NO		NULL	

6 rows in set (0.00 sec)

Input Data / Record

Jika sudah membuat database dan table kemudian mengisi data ke dalam table, maka perintahnya “*insert into nama_table (nama_field, nama_fields,) values (“data yang diinput”, “data yang diinput”, “.....”);*”

```
mysql> insert into pegawai
-> (id, nama, jenkel, tanggal_lahir, alamat, status)
-> values
-> ("P001", "Burhan", "L", "1998-10-10", "Bekasi", "Menikah");
Query OK, 1 row affected (0.02 sec)
```

Penjelasannya ;

insert into pegawai = memasukkan data ke table pegawai

(id, nama, jenkel, tanggal_lahir, alamat, kota, status) = Nama field

values = nilainya

("P001", "Burhan", "L", "1998-10-10", "Bekasi", "Menikah") = Datanya yang diinput. Ingat harus ada tanda kutip untuk memasukkan data tersebut.

Jika ingin input data sekaligus pada values ketik “ , ” sesudah kurung tutup kemudian masukkan data yang lain dan jika sudah selesai mengisi data maka diakhiri dengan “ ; ” sebagai contoh membuat 5 data dibawah ini.

```
mysql> insert into pegawai
-> (id, nama, jenkel, tanggal_lahir, alamat, status)
-> values
-> ("P002", "Mirna", "P", "1999-05-09", "Tambun", "Belum Menikah"),
-> ("P003", "Joko Ardi", "L", "1997-04-25", "Cibitung", "Menikah"),
-> ("P004", "M. Sulaiman", "L", "1997-08-20", "Bekasi Selatan", "Menikah"),
-> ("P005", "Siska", "P", "1997-08-08", "Kerawang", "Belum Menikah");
Query OK, 4 rows affected (0.00 sec)
Records: 4 Duplicates: 0 Warnings: 0
```

Apabila ingin pengisian primary key berdasarkan no urut secara otomatis maka menggunakan perintah “*int unsigned auto_increment*”. Sebagai contoh buat table dengan **pegawai2** dengan field yang sama pada table **pegawai**.

```
mysql> create table pegawai2
-> (id int unsigned auto_increment primary key,
-> nama varchar (15) not null,
-> jenkel char (1) not null,
-> tanggal_lahir date not null,
-> alamat varchar (15) not null,
-> status varchar (15) not null);
Query OK, 0 rows affected (0.00 sec)
```

Penjelasannya

1. INT merupakan tipe yang berdasarkan angka 1 hingga 10 digit
2. UNSIGNED merupakan tanpa ketik ataupun diinput
3. AUTO_INCREMENT merupakan penomoran secara otomatis ketika input data.

Jika ingin melihat struktur data tersebut.

```
mysql> desc pegawai2;
```

Field	Type	Null	Key	Default	Extra
id	int(10) unsigned	NO	PRI	NULL	auto_increment
nama	varchar(15)	NO		NULL	
jenkel	char(1)	NO		NULL	
tanggal_lahir	date	NO		NULL	
alamat	varchar(15)	NO		NULL	
status	varchar(15)	NO		NULL	

6 rows in set (0.00 sec)



Sebagai contoh membuat 5 data dibawah ini.

```
mysql> insert into pegawai2
-> (nama, jenkel, tanggal_lahir, alamat, status)
-> values
-> ("Rufman", "L", "1998-12-12", "Bekasi Timur", "Menikah"),
-> ("Makmum", "L", "1998-12-02", "Bekasi", "Tidak Menikah"),
-> ("Mumun", "P", "1997-02-02", "Tambun", "Menikah"),
-> ("Mimin", "P", "1997-10-02", "Tambun", "Tidak Menikah"),
-> ("Mirza", "L", "1997-10-02", "Cibatu", "Tidak Menikah");
Query OK, 5 rows affected (0.00 sec)
Records: 5 Duplicates: 0 Warnings: 0
```

Mungkin ada pertanyaan kenapa field id tidak ditulis, dikarenakan penulisan secara otomatis ketika ditampilkan.

Sebagai latihan buatlah data dibawah ini sebagai bahan berikutnya pada table pegawai.

```
mysql> insert into pegawai
-> (id, nama, jenkel, tanggal_lahir, alamat, status)
-> values
-> ("P001", "Burhan", "L", "1998-10-10", "Bekasi", "Menikah"),
-> ("P002", "Mirna", "P", "1999-05-09", "Tambun", "Menikah"),
-> ("P003", "Joko Ardi", "L", "1997-04-25", "Cibitung", "Menikah"),
-> ("P004", "M.Sulaiman", "L", "1997-08-20", "Bekasi Selatan", "Menikah"),
-> ("P005", "Siska", "P", "1997-08-08", "Kerawang", "Belum Menikah"),
-> ("P006", "Budinan", "L", "1999-08-08", "Bekasi", "Menikah"),
-> ("P007", "Riska Nurni", "P", "2000-08-08", "Cibitung", "Belum Menikah"),
-> ("P008", "Ardinan", "L", "1999-10-08", "Jakarta", "Menikah"),
-> ("P009", "Rudi Harahap", "L", "1998-10-08", "Bekasi", "Menikah"),
-> ("P010", "Rina Narni", "P", "1998-10-22", "Jakarta", "Belum Menikah"),
-> ("P011", "Ridwan", "L", "1990-10-25", "Jakarta", "Menikah"),
-> ("P012", "Susan", "P", "1992-08-03", "Bekasi", "Belum Menikah"),
-> ("P013", "Intan", "P", "1999-08-03", "Kerawang", "Belum Menikah"),
-> ("P014", "Edy", "L", "1995-10-03", "Jakarta", "Menikah"),
-> ("P015", "Risan", "L", "1994-10-25", "Cibitung", "Menikah");
Query OK, 15 rows affected (0.00 sec)
Records: 15 Duplicates: 0 Warnings: 0
```

Melihat Data / Record

Jika ingin melihat data yang telah diinput sebelumnya, maka perintahnya
*“select * from nama_table ; “*

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

id	nama	jenkel	tanggal_lahir	alamat	status
P001	Burhan	L	1998-10-10	Bekasi	Menikah
P002	Mirna	P	1999-05-09	Tambun	Menikah
P003	Joko Ardi	L	1997-04-25	Cibitung	Menikah
P004	M.Sulaiman	L	1997-08-20	Bekasi Selatan	Menikah
P005	Siska	P	1997-08-08	Kerawang	Belum Menikah
P006	Budinan	L	1999-08-08	Bekasi	Menikah
P007	Riska Nurni	P	2000-08-08	Cibitung	Belum Menikah
P008	Ardinan	L	1999-10-08	Jakarta	Menikah
P009	Rudi Harahap	L	1998-10-08	Bekasi	Menikah
P010	Rina Narni	P	1998-10-22	Jakarta	Belum Menikah
P011	Ridwan	L	1990-10-25	Jakarta	Menikah
P012	Susan	P	1992-08-03	Bekasi	Belum Menikah
P013	Intan	P	1999-08-03	Kerawang	Belum Menikah
P014	Edy	L	1995-10-03	Jakarta	Menikah
P015	Risan	L	1994-10-25	Cibitung	Menikah

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

Jika ingin melihat data pada table **pegawai2** yang telah dibuat sebelumnya dengan penomoran otomatis.

```
mysql> select * from pegawai2;
```

id	nama	jenkel	tanggal_lahir	alamat	status
1	Rufman	L	1998-12-12	Bekasi Timur	Menikah
2	Makmum	L	1998-12-02	Bekasi	Tidak Menikah
3	Mumun	P	1997-02-02	Tambun	Menikah
4	Mimin	P	1997-10-02	Tambun	Tidak Menikah
5	Mirza	L	1997-10-02	Cibatu	Tidak Menikah

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

Jika ingin melihat data tetapi field tertentu yang ingin ditampilkan. Sebagai contoh ingin menampilkan **nama**, **jenkel** dan **alamat** maka perintahnya *“select nama_field, nama_field, nama_field from nama_table ;”*

```
mysql> select nama, jenkel, alamat from pegawai;
```

nama	jenkel	alamat
Burhan	L	Bekasi
Mirna	P	Tambun
Joko Ardi	L	Cibitung
M.Sulaiman	L	Bekasi Selatan
Siska	P	Kerawang
Budiman	L	Bekasi
Riska Murni	P	Cibitung
Ardiman	L	Jakarta
Rudi Harahap	L	Bekasi
Rina Marni	P	Jakarta
Ridwan	L	Jakarta
Susan	P	Bekasi
Intan	P	Kerawang
Edy	L	Jakarta
Risam	L	Cibitung

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

Sebagai contoh ingin mengetahui data pegawai yang tinggal di **Bekasi** maka perintahnya.

```
mysql> select nama, jenkel, alamat from pegawai
-> where alamat="Bekasi";
```

nama	jenkel	alamat
Burhan	L	Bekasi
Budiman	L	Bekasi
Rudi Harahap	L	Bekasi
Susan	P	Bekasi

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

Jika ingin mengetahui data pegawai yang jenis kelamin **Lelaki** maka perintahnya.

```
mysql> select nama, jenkel, alamat from pegawai
-> where jenkel="L";
```

nama	jenkel	alamat
Burhan	L	Bekasi
Joko Ardi	L	Cibitung
M.Sulaiman	L	Bekasi Selatan
Budiman	L	Bekasi
Ardiman	L	Jakarta
Rudi Harahap	L	Bekasi
Ridwan	L	Jakarta
Edy	L	Jakarta
Risam	L	Cibitung

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

Jika melihat data ingin berdasarkan alphabet dari A ke Z, sebagai contoh ingin mengurutkan nama pegawai, maka perintahnya.


```
mysql> select * from pegawai
-> order by nama asc;
```

id	nama	jenkel	tanggal_lahir	alamat	status
P008	Ardiman	L	1999-10-08	Jakarta	Menikah
P006	Budiman	L	1999-08-08	Bekasi	Menikah
P001	Burhan	L	1998-10-10	Bekasi	Menikah
P014	Edy	L	1995-10-03	Jakarta	Menikah
P013	Intan	P	1999-08-03	Kerawang	Belum Menikah
P003	Joko Ardi	L	1997-04-25	Cibitung	Menikah
P004	M.Sulaiman	L	1997-08-20	Bekasi Selatan	Menikah
P002	Mirna	P	1999-05-09	Tambun	Menikah
P011	Ridwan	L	1998-10-25	Jakarta	Menikah
P010	Rina Marni	P	1998-10-22	Jakarta	Belum Menikah
P015	Risam	L	1994-10-25	Cibitung	Menikah
P007	Riska Murni	P	2000-08-08	Cibitung	Belum Menikah
P009	Rudi Harahap	L	1998-10-08	Bekasi	Menikah
P005	Siska	P	1997-08-08	Kerawang	Belum Menikah
P012	Susan	P	1992-08-03	Bekasi	Belum Menikah

15 rows in set (0.00 sec)



```
mysql> select * from pegawai
-> order by nama desc;
```

id	nama	jenkel	tanggal_lahir	alamat	status
P012	Susan	P	1992-08-03	Bekasi	Belum Menikah
P005	Siska	P	1997-08-08	Kerawang	Belum Menikah
P009	Rudi Harahap	L	1998-10-08	Bekasi	Menikah
P007	Riska Murni	P	2000-08-08	Cibitung	Belum Menikah
P015	Risam	L	1994-10-25	Cibitung	Menikah
P010	Rina Marni	P	1998-10-22	Jakarta	Belum Menikah
P011	Ridwan	L	1998-10-25	Jakarta	Menikah
P002	Mirna	P	1999-05-09	Tambun	Menikah
P004	M.Sulaiman	L	1997-08-20	Bekasi Selatan	Menikah
P003	Joko Ardi	L	1997-04-25	Cibitung	Menikah
P013	Intan	P	1999-08-03	Kerawang	Belum Menikah
P014	Edy	L	1995-10-03	Jakarta	Menikah
P001	Burhan	L	1998-10-10	Bekasi	Menikah
P006	Budiman	L	1999-08-08	Bekasi	Menikah
P008	Ardiman	L	1999-10-08	Jakarta	Menikah

15 rows in set (0.00 sec)

Melihat Data Per Group

Dalam suatu data pastilah ada yang berdasarkan group atau kesamaan data. Sebagai contoh ingin melihat group field alamat, maka perintahnya “*select nama_field from nama_table group by nama_field ;*”

```
mysql> select alamat from pegawai group by alamat;
```

alamat
Bekasi
Bekasi Selatan
Cibitung
Jakarta
Kerawang
Tambun

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

```
mysql> select status from pegawai group by status;
```

status
Belum Menikah
Menikah

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

Mencari Data

Di dalam table kumpulan beberapa data yang beraneka ragam dan bingung dalam pencarian apalagi data mempunyai kesamaan. Itulah gunanya primary key sebagai pembeda antara data yang satu dengan data yang lain. Nama aji pasti ada nama yang lain di lain tempat tetapi jika menggunakan

primary key pasti berbeda, sebagai contoh melihat data yang berkode id P007 yang sebagai primary key, maka perintahnya

```
mysql> select * from pegawai where id="P007";
+-----+-----+-----+-----+-----+-----+
| id   | nama      | jenkel | tanggal_lahir | alamat  | status      |
+-----+-----+-----+-----+-----+-----+
| P007 | Riska Murni | P      | 2000-08-08     | Cibitung | Belum Menikah |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Jika ingin mencari data tetapi hanya ditampilkan field tertentu. Sebagai contoh ingin mencari yang kode id P013 tetapi melihat datanya hanya nama, alamat dan status, maka perintahnya

```
mysql> select nama, alamat, status from pegawai where id="P013";
+-----+-----+-----+
| nama  | alamat  | status      |
+-----+-----+-----+
| Intan | Kerawang | Belum Menikah |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Yang menjadi persoalan bagaimana jika tidak tahu kode yang sebagai *primary key* ataupun lupa kode tersebut malahan ingat hanya namanya saja ataupun mencari berdasarkan kriteria yang sebagai contoh yang alamat di Bekasi. Tidak bakalan mungkin melihat data satu persatu kalau datanya hanya puluhan mungkin saja tapi kalau sampai ratusan sangat membuang waktu. Sebagai contoh mencari data yang bernama **Rina**, maka perintahnya “*select * from nama_table where nama_field like “%nama_field%” ;*”

```
mysql> select * from pegawai where nama like "%Rina%";
+-----+-----+-----+-----+-----+-----+
| id   | nama      | jenkel | tanggal_lahir | alamat  | status      |
+-----+-----+-----+-----+-----+-----+
| P010 | Rina Marni | P      | 1998-10-22     | Jakarta | Belum Menikah |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.02 sec)
```

Apabila hanya mencari nama yang **berawalan “r”** maka perintahnya

```
mysql> select * from pegawai where nama like "r%";
```

id	nama	jenkel	tanggal_lahir	alamat	status
P007	Riska Murni	P	2000-08-08	Cibitung	Belum Menikah
P009	Rudi Harahap	L	1998-10-08	Bekasi	Menikah
P010	Rina Marni	P	1998-10-22	Jakarta	Belum Menikah
P011	Ridwan	L	1998-10-25	Jakarta	Menikah
P015	Risan	L	1994-10-25	Cibitung	Menikah

5 rows in set (0.00 sec)

Apabila hanya mencari nama yang **berakhiran “n”** maka perintahnya

```
mysql> select * from pegawai where nama like "%n";
```

id	nama	jenkel	tanggal_lahir	alamat	status
P001	Burhan	L	1998-10-10	Bekasi	Menikah
P004	M.Sulaiman	L	1997-08-20	Bekasi Selatan	Menikah
P006	Budinan	L	1999-08-08	Bekasi	Menikah
P008	Ardinan	L	1999-10-08	Jakarta	Menikah
P011	Ridwan	L	1998-10-25	Jakarta	Menikah
P012	Susan	P	1992-08-03	Bekasi	Belum Menikah
P013	Intan	P	1999-08-03	Kerawang	Belum Menikah

7 rows in set (0.00 sec)

Bisa juga menggunakan pencarian suatu data menggunakan fungsi operator pembandingan dan logika

Pembandingan		Logika	
Lebih Besar	>	Dan	AND atau &&
Lebih Kecil	<=	Atau	OR atau
Lebih Besar atau Sama Dengan	>=	Lebih Besar atau Sama Dengan	NOT atau !
Lebih Kecil atau Sama Dengan	<=		
Sama Dengan	=		

Tidak Sama Dengan

<>

Sebagai contoh mencari data yang lahir dibawah tahun 2000, maka perintahnya

```
mysql> select nama, tanggal_lahir from pegawai  
-> where tanggal_lahir <="2000"  
-> order by nama;
```

nama	tanggal_lahir
Ardinan	1999-10-08
Budiman	1999-08-08
Burhan	1998-10-10
Edy	1995-10-03
Intan	1999-08-03
Joko Ardi	1997-04-25
M.Sulaiman	1997-08-20
Mirna	1999-05-09
Ridwan	1990-10-25
Rina Marni	1998-10-22
Risam	1994-10-25
Rudi Harahap	1998-10-08
Siska	1997-08-08
Susan	1992-08-03

14 rows in set, 1 warning (0.00 sec)

Jika ingin mencari data yang lahir di antara tahun 1995 dengan tahun 2000.

```
mysql> select nama, tanggal_lahir from pegawai  
-> where tanggal_lahir >="1995"  
-> and tanggal_lahir <="2000"  
-> order by nama;
```

nama	tanggal_lahir
Ardinan	1999-10-08
Budiman	1999-08-08
Burhan	1998-10-10
Edy	1995-10-03
Intan	1999-08-03
Joko Ardi	1997-04-25
M.Sulaiman	1997-08-20
Mirna	1999-05-09
Rina Marni	1998-10-22
Rudi Harahap	1998-10-08
Siska	1997-08-08

11 rows in set, 2 warnings (0.00 sec)

Jika ingin mencari data yang tinggal di kota Bekasi, maka perintahnya

```
mysql> select * from pegawai
-> where alamat="Bekasi";
```

id	nama	jenkel	tanggal_lahir	alamat	status
P001	Burhan	L	1998-10-10	Bekasi	Menikah
P006	Budiman	L	1999-08-08	Bekasi	Menikah
P009	Rudi Harahap	L	1998-10-08	Bekasi	Menikah
P012	Susan	P	1992-08-03	Bekasi	Belum Menikah

4 rows in set (0.00 sec)

Jika ingin mencari data yang tinggal di kota Tambun dan Cibitung, maka perintahnya

```
mysql> select * from pegawai
-> where alamat="Tambun"
-> or alamat="Cibitung";
```

id	nama	jenkel	tanggal_lahir	alamat	status
P002	Mirna	P	1999-05-09	Tambun	Menikah
P003	Joko Ardi	L	1997-04-25	Cibitung	Menikah
P007	Riska Murni	P	2000-08-08	Cibitung	Belum Menikah
P015	Risam	L	1994-10-25	Cibitung	Menikah

4 rows in set (0.00 sec)

Jika ingin melihat data yang tidak tinggal di Jakarta, maka perintahnya

```
mysql> select * from pegawai
-> where alamat != "Jakarta";
```

id	nama	jenkel	tanggal_lahir	alamat	status
P001	Burhan	L	1998-10-10	Bekasi	Menikah
P002	Mirna	P	1999-05-09	Tambun	Menikah
P003	Joko Ardi	L	1997-04-25	Cibitung	Menikah
P004	M. Sulaiman	L	1997-08-20	Bekasi Selatan	Menikah
P005	Siska	P	1997-08-08	Kerawang	Belum Menikah
P006	Budiman	L	1999-08-08	Bekasi	Menikah
P007	Riska Murni	P	2000-08-08	Cibitung	Belum Menikah
P009	Rudi Harahap	L	1998-10-08	Bekasi	Menikah
P012	Susan	P	1992-08-03	Bekasi	Belum Menikah
P013	Intan	P	1999-08-03	Kerawang	Belum Menikah
P015	Risam	L	1994-10-25	Cibitung	Menikah

11 rows in set (0.00 sec)

Jika ingin melihat data yang tinggal di luar kota Depok dan Jakarta, maka perintahnya

```
mysql> select * from pegawai
-> where alamat != "Jakarta"
-> and alamat != "Bekasi";
```

id	nama	jenkel	tanggal_lahir	alamat	status
P002	Mirna	P	1999-05-09	Tambun	Menikah
P003	Joko Ardi	L	1997-04-25	Cibitung	Menikah
P004	M. Sulaiman	L	1997-08-20	Bekasi Selatan	Menikah
P005	Siska	P	1997-08-08	Kerawang	Belum Menikah
P007	Riska Nurni	P	2000-08-08	Cibitung	Belum Menikah
P013	Intan	P	1999-08-03	Kerawang	Belum Menikah
P015	Risam	L	1994-10-25	Cibitung	Menikah

7 rows in set (0.00 sec)

```
mysql> select * from pegawai
-> where alamat <> "Jakarta"
-> and alamat <> "Bekasi";
```

id	nama	jenkel	tanggal_lahir	alamat	status
P002	Mirna	P	1999-05-09	Tambun	Menikah
P003	Joko Ardi	L	1997-04-25	Cibitung	Menikah
P004	M. Sulaiman	L	1997-08-20	Bekasi Selatan	Menikah
P005	Siska	P	1997-08-08	Kerawang	Belum Menikah
P007	Riska Nurni	P	2000-08-08	Cibitung	Belum Menikah
P013	Intan	P	1999-08-03	Kerawang	Belum Menikah
P015	Risam	L	1994-10-25	Cibitung	Menikah

7 rows in set (0.00 sec)

Mengupdate Record/Data

Pastilah dalam mengisi data ada kesalahan dan itu lumrah, jika terjadi kesalahan dalam input data, sebagai contoh mengganti status Mirna seharusnya Belum Menikah. Perintahnya *"update nama_table set nama_field where nama_primary key = 'nama_data' ; "*

```
mysql> update pegawai
-> set status="Belum Menikah"
-> where id="P002";
```

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0

Untuk melihat hasilnya

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

id	nama	jenis_kelamin	tanggal_lahir	alamat	status
P001	Burhan	L	1998-10-10	Bekasi	Menikah
P002	Purna	P	1999-05-09	Tambun	Belum Menikah
P003	Joko Andi	L	1997-04-25	Cibitung	Menikah
P004	M. Sulaiman	L	1997-06-20	Bekasi Selatan	Menikah
P005	Siska	P	1997-08-08	Kerawang	Belum Menikah
P006	Budiman	L	1999-08-08	Bekasi	Menikah
P007	Riska Yarni	P	2000-08-08	Cibitung	Belum Menikah
P008	Ardiman	L	1999-10-08	Jakarta	Menikah
P009	Rudi Harahap	L	1998-10-08	Bekasi	Menikah
P010	Rina Yarni	P	1998-10-22	Jakarta	Belum Menikah
P011	Ridwan	L	1998-10-25	Jakarta	Menikah
P012	Susan	P	1992-08-03	Bekasi	Belum Menikah
P013	Intan	P	1999-08-03	Kerawang	Belum Menikah
P014	Edy	L	1995-10-03	Jakarta	Menikah
P015	Risam	L	1994-10-25	Cibitung	Menikah

15 rows in set (0.00 sec)

Menghapus Record/Data

Jika ada data yang ingin dihapus, maka perintah “*delete from nama_table where nama_field(PK) = “nama_data”;* “. Sebagai contoh ingin menghapus data Risam.

```
mysql> delete from pegawai where id="P015";  
Query OK, 1 row affected (0.00 sec)
```

Untuk melihat hasilnya

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

id	nama	jenis_kelamin	tanggal_lahir	alamat	status
P001	Burhan	L	1998-10-10	Bekasi	Menikah
P002	Purna	P	1999-05-09	Tambun	Belum Menikah
P003	Joko Andi	L	1997-04-25	Cibitung	Menikah
P004	M. Sulaiman	L	1997-06-20	Bekasi Selatan	Menikah
P005	Siska	P	1997-08-08	Kerawang	Belum Menikah
P006	Budiman	L	1999-08-08	Bekasi	Menikah
P007	Riska Yarni	P	2000-08-08	Cibitung	Belum Menikah
P008	Ardiman	L	1999-10-08	Jakarta	Menikah
P009	Rudi Harahap	L	1998-10-08	Bekasi	Menikah
P010	Rina Yarni	P	1998-10-22	Jakarta	Belum Menikah
P011	Ridwan	L	1998-10-25	Jakarta	Menikah
P012	Susan	P	1992-08-03	Bekasi	Belum Menikah
P013	Intan	P	1999-08-03	Kerawang	Belum Menikah
P014	Edy	L	1995-10-03	Jakarta	Menikah

14 rows in set (0.00 sec)