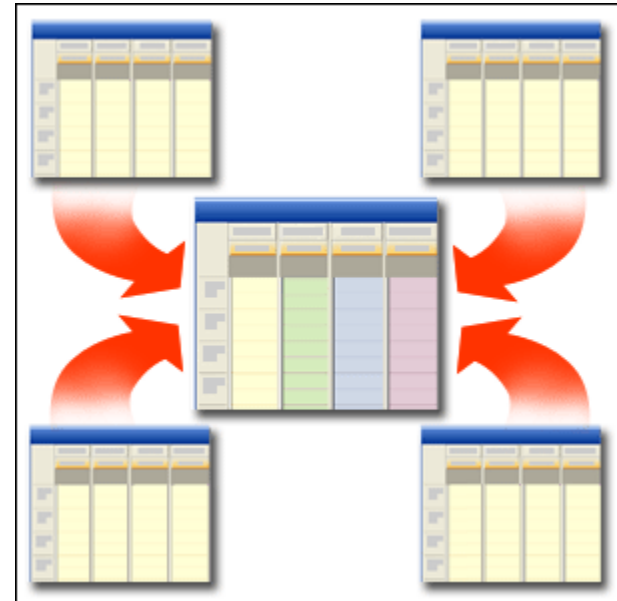


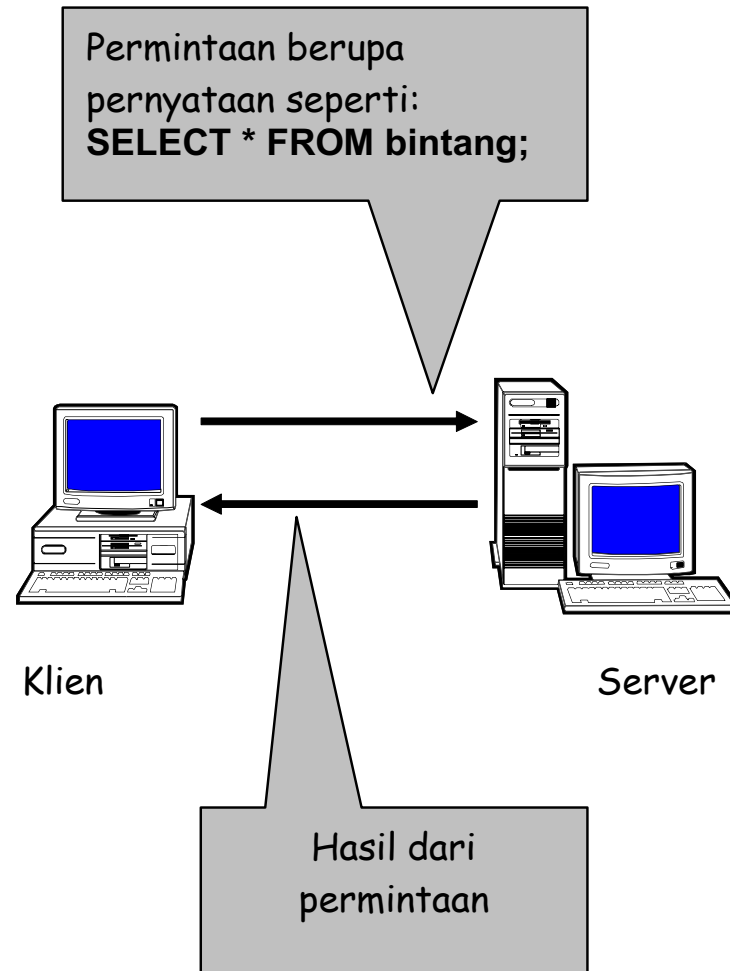
Overview MySQL

- MySQL merupakan Database Server yang bersifat :
 - *Open Source*
 - Multiplatform
 - Berbasis *database* relasional
- Bisa dipakai untuk database pribadi atau pada level korporat berskala kecil hingga besar
- Selain bersifat free, ada juga yang bersifat komersial



Overview MySQL (Lanjutan...)

- Menggunakan SQL untuk mendukung pengaksesan data (*query*)



Dasar SQL

- SQL = Structured Query Language
- Digunakan untuk mengakses basis data relasional
- Bersifat standar; bisa dipakai untuk basis data relasional lainnya
- Perintah SQL dapat dibagi menjadi DDL dan DML

Perintah DDL

- DDL = *Definition Data Language*
- Digunakan untuk kepentingan penciptaan database, tabel, hingga penghapusan database atau tabel
- Contoh:
 - CREATE DATABASE
 - CREATE TABLE
 - DROP TABLE
 - ALTER TABLE

Perintah DML

- DML = *Data Manipulation Language*
- Digunakan untuk memanipulasi data
- Contoh:
 - SELECT – mengambil data
 - DELETE – menghapus data
 - INSERT – menyisipkan data
 - UPDATE – mengubah data

Persiapan Pemakaian MySQL

- Lakukan instalasi MySQL terlebih dulu
- MySQL memiliki sejumlah *tool*; salah satu di antaranya adalah:
 - program **mysql**, yang dipakai untuk mengakses *database* dari sisi klien
 - Program **mysqladmin**, untuk mengelola MySQL dari sisi administrator sistem

Masuk ke Direktori Program

- Masuk ke prompt DOS
- Ketik perintah:
 `cd c:\\"Program Files"\MySQL\"MySQL Server 5.0"`
- Berikan perintah `cd bin`

Mengatur Password root

- **root** adalah pemakai dengan wewenang paling tinggi dan digunakan untuk admin sistem
- Perintah untuk mengubah password root:
mysqladmin –uroot password **rahasia**

Masuk ke Program Klien mysql

- Berikan perintah:
mysql -uroot -p
- Ketikkan password dan tekan Enter

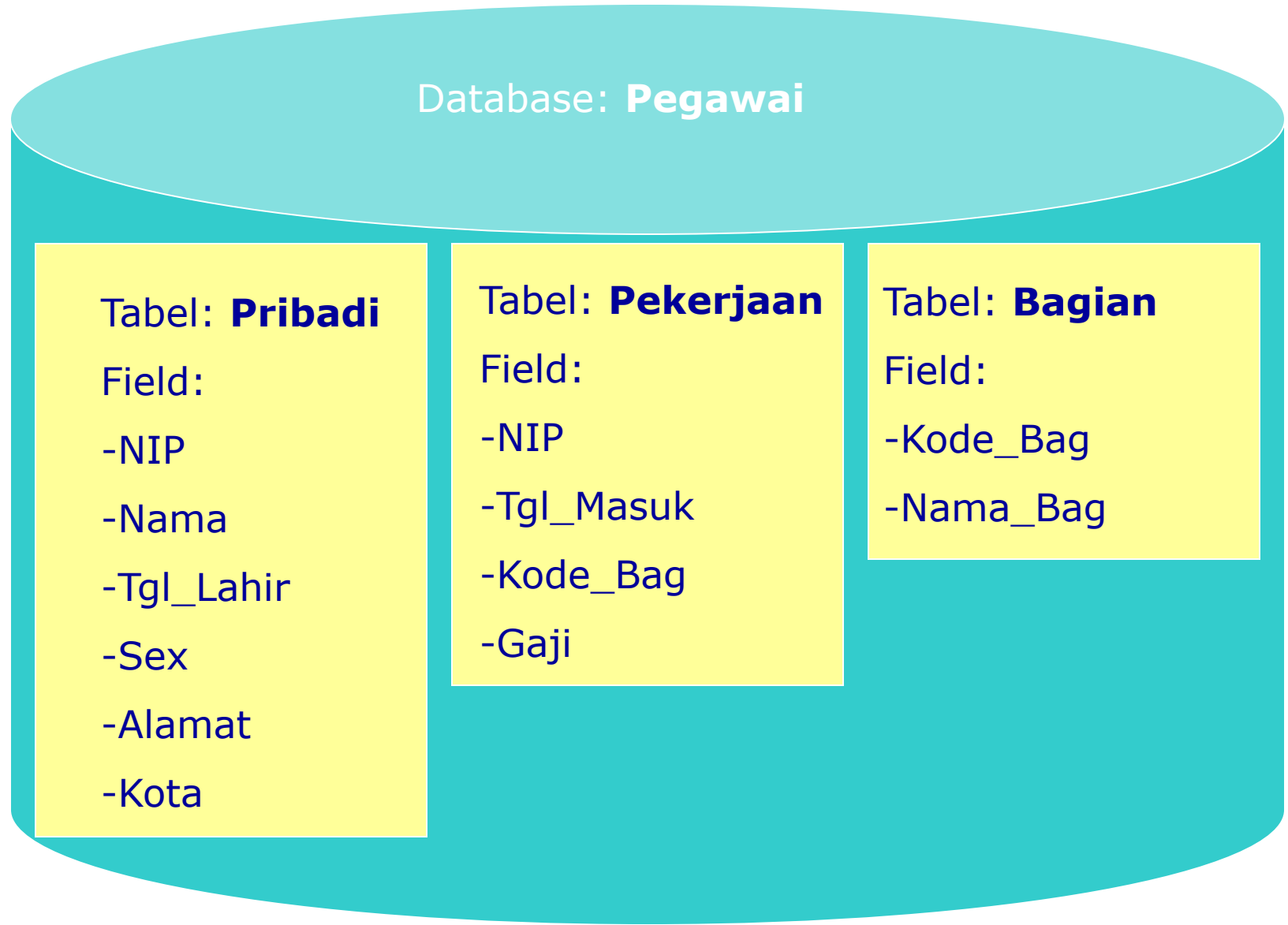
```
C:\Program Files\MySQL\MySQL Server 5.0\bin>mysql -uroot -prahasia
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 8 to server version: 5.0.15-nt
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

```
mysql>
```

Pembuatan Database dan Tabel



Membuat Database Pegawai

- Berikan perintah pada prompt mysql:
CREATE DATABASE Pegawai;

```
mysql> CREATE DATABASE Pegawai;  
Query OK, 1 row affected (0.09 sec)
```

```
mysql>
```

Memilih Database

- Berikan perintah:
USE Pegawai;



Nama
Database

Membuat Tabel Pribadi

- Berikan perintah:

Penjelasan Tipe Data

KARAKTER

- **CHAR:** Teks dengan maksimal 255 karakter
- **VARCHAR:** Teks maksimal 255 karakter dan bersifat variabel
- **TEXT:** Teks dengan panjang maksimal 65535

Penjelasan Tipe Data

BILANGAN

- **TINYINT**: Bilangan 1 byte
- **SMALLINT**: Bilangan 2 byte
- **INT** atau **INTEGER** Bilangan 4 byte
- **BIGINT**: Bilangan 8 byte
- **FLOAT**: Bilangan pecahan (4 byte)
- **DOUBLE** atau **REAL**: Bilangan pecahan (8 byte)
- **DECIMAL(M, D)** atau **NUMERIC(M, D)**: Bilangan pecahan

Penjelasan Tipe Data

LAIN-LAIN

- **DATE**: Tanggal
- **DATETIME**: Waktu (tanggal dan jam)
- **TIME** : Jam
- **ENUM**('nilai1', 'nilai2', ...): Nilai enumerasi
- **BOOLEAN**: tipe benar atau salah

Kata Tambahan

- NOT NULL : Tidak boleh kosong
- PRIMARY KEY: Kunci primer
- AUTO_INCREMENT: Nilai naik secara otomatis tanpa diisi

Penambahan Data

- Penambahan data dilakukan dengan menggunakan pernyataan INSERT
- Bentuk dasar :

```
INSERT INTO nama_tabel (nama_field, nama_field, ...)  
VALUES (nilai, nilai, ...);
```
- Contoh:

```
INSERT INTO Pribadi(Nip, Nama, Tgl_lahir, Sex, Alamat, Kota)  
VALUES ('12345', 'A. Hamzah Sianturi', '1965/12/23',  
'P', 'Jl. Kudus 1', 'Yogya');
```

Melihat Isi Tabel

- Perintah SELECT

- Bentuk Umum:

`SELECT * FROM Nama_Tabel`

- Contoh:

`SELECT * FROM Pribadi`

Penambahan Data (Lanjutan...)

- Tambahkan data berikut:

NIP	Tanggal lahir	Nama	Alamat	Kota	Jenis Kelamin

- Untuk data yang kosong, tuliskan NULL

Penambahan Data dengan Field Tertentu

- Contoh tanpa mengisi Tanggal lahir, alamat dan kota

```
INSERT INTO Pribadi(Nip, Nama, Sex)  
VALUES ('12355', 'Andini', 'W');
```

Pengaruh Kunci Primer

- Cobalah berikan pernyataan berikut:

```
INSERT INTO Pribadi(Nip, Nama, Tgl_lahir, Sex, Alamat, Kota)  
VALUES ('12345', 'Dona Harun', '1978/01/01',  
'W', 'Jl. Kreasi 77', 'Yogya');
```

- Apa yang terjadi? Kenapa?

Pengaruh NOT NULL

- Cobalah berikan pernyataan berikut:

```
INSERT INTO Pribadi(Nip, Sex)  
VALUES ('12360', 'W');
```

- Apa yang terjadi? Kenapa?

Pengaruh ENUM

- Cobalah perintah berikut:

```
INSERT INTO Pribadi(Nip, Nama, Kelamin)  
VALUES ('12361', 'Edi harahap', 'L');
```

- Apa yang terjadi? Kenapa?
- Cek dengan SELECT; apakah data tersimpan?

Melihat Struktur Tabel

- Gunakan perintah:
DESC nama_tabel
- Contoh:
- DESC Pribadi;

Field	Type	Null	Key	Default	Extra
Nip	char(5)	NO	PRI		
Nama	varchar(35)	NO			
Tgl_lahir	date	YES		NULL	
Sex	enum('P','W')	YES		NULL	
Alamat	varchar(35)	YES		NULL	
kota	varchar(20)	YES		NULL	

6 rows in set (0.16 sec)

Mengganti Nama Field

- Perintah yang digunakan adalah ALTER TABLE
- Contoh:
**ALTER TABLE Pribadi
CHANGE sex kelamin ENUM('P','W');**
- Cek struktur tabel setelah Anda melakukan perintah di atas

Mengganti Ukuran/Tipe Field

- Perintah yang digunakan adalah ALTER TABLE
- Contoh:
ALTER TABLE Pribadi
CHANGE kota kota VARCHAR(20);
- Cek struktur tabel setelah Anda melakukan perintah di atas

Menambahkan DEFAULT

- DEFAULT pada struktur tabel digunakan untuk memberikan nilai bawaan pada suatu field kalau nilai bersangkutan tidak dimasukkan

- Contoh:

```
ALTER TABLE Pribadi CHANGE  
kelamin kelamin ENUM('P','W')  
DEFAULT 'P';
```

Menambahkan DEFAULT (Lanjutan...)

- Sekarang cobalah berikan perintah:
INSERT INTO Pribadi(Nip, Nama)
VALUES ('12370', 'Fahmi Idris');
- PERHATIKAN, apa isi field Kelamin untuk NIP 12370'?

Pengubahan Data

- Perintah yang digunakan adalah UPDATE
- Bentuk dasar:

```
UPDATE nama_tabel SET nama_field = nilai, nama_field = nilai, ...  
WHERE nama_field = nilai
```

- Contoh mengubah **Udin** menjadi **Udinsah**:

```
UPDATE Pribadi SET Nama= 'Udinsah'  
WHERE NIP = '12346';
```

- Ujilah dengan SELECT untuk melihat hasil perubahan

Penghapusan Data

- Perintah DELETE
- Bentuk dasar:
DELETE FROM nama_tabel WHERE
nama_field = nilai
- Contoh:
DELETE FROM Pribadi
WHERE Nip = '12355';
- Ujilah dengan SELECT untuk melihat efek perintah di atas

Penghapusan Tabel

- Gunakan DROP TABLE
- Untuk mempraktekkan, buatlah sebuah table bernama RIWAYAT, dengan isi berupa sebuah field (misalnya Nip bertipe CHAR(5))
- Kemudian, lihatlah daftar tabel dengan memberikan perintah:
SHOW TABLES;

Penghapusan Tabel (Lanjutan...)

- Berikan perintah:

DROP TABLE Riwayat;

- Kemudian, lihatlah daftar tabel dengan memberikan perintah:

SHOW TABLES;

Apa sudah terhapus?

Latihan Membuat Tabel Bagian

- Buatlah tabel **Bagian** dengan struktur seperti berikut:
 - **Kode_Bag**, CHAR, 1 karakter, primary key
 - **Nama_Bag**, VARCHAR, 20 karakter, harus diisi
- Isikan data sebagai berikut:
 - 1 untuk EDP
 - 2 untuk Pemasaran
 - 3 untuk Produksi
 - 4 untuk SDM
 - 5 untuk Akunting

Latihan Membuat Tabel Pekerjaan



Latihan Penggunaan Tipe Memo

- Buatlah tabel bernama **Riwayat**
- Isi Field:
 - NIP
 - Keterangan (bertipe Memo)
- Isikan data untuk NIP yang tersedia pada tabel Pribadi (12345 sampai dengan 12354)
- Isikan Keterangan dengan daftar riwayat pendidikan, misalnya sbb:

SD Muhammadiyah I, Yogya

SMP Negeri I, Yogya

SMA Negeri III, Semarang

Latihan Gabungan

- Masukkan data berikut ke Tabel Pribadi:
 - NIP: 12390
 - Nama: Asti Damayanti
 - Tanggal Lahir: 2 Februari 1983
 - Jenis kelamin: Wanita
- Ubahlah Tanggal lahir Asti Damayanti menjadi 23 Pebruari 1973
- Cek hasilnya

Latihan Gabungan (Lanjutan...)

- Hapuslah record yang berisi data Asti Damayanti dengan menyebutkan namanya (bukan NIP)
- Ubahlah nama field **Keterangan** pada tabel Riwayat menjadi **Ket**

Latihan Gabungan (Lanjutan...)

- Menambahkan Field:
ALTER TABLE Pribadi ADD
Bisa_Bhs_Asing BOOLEAN DEFAULT
FALSE;
- Buatlah agar field baru tersebut bernilai
TRUE untuk NIP=12345, 12347, dan
12350