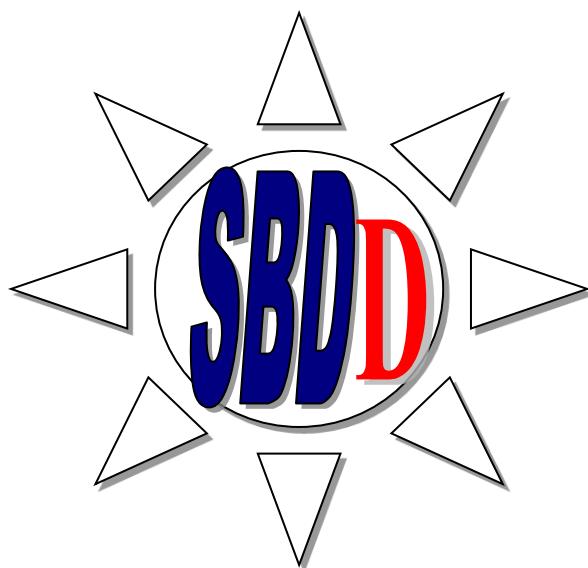


Digunakan untuk kalangan internal

## MODUL PRAKTIKUM

# PANDUAN PRAKTIS MENGUASAI BASIS DATA DENGAN MS ACCESS DAN MY SQL

DIGUNAKAN UNTUK PRAKTIKUM  
MATAKULIAH PRAKTIKUM SISTEM BASIS DATA DASAR



Disusun Oleh:  
**Z A I D I R**

PROGRAM STUDI MANAJEMEN INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS RESPATI YOGYAKARTA  
**2015**

# **KETENTUAN KEGIATAN PRAKTIKUM**

## **A. Kelengkapan Praktikum**

1. Kegiatan praktikum dilengkapi dengan modul praktikum
2. Perlu memiliki flashdisk
3. Hasil kegiatan praktikum disimpan pada folder masing-masing (bisa dibuat pada drive komputer atau flashdisk atau dibuat di server)

## **B. Tata Tertib Praktikum**

1. Peserta praktikum (praktikan) harus memenuhi atau membawa kelengkapan praktikum setiap kali mengikuti kegiatan praktikum.
2. Untuk mengikuti praktikum harus berpakaian rapi dan sopan.
3. Pada saat kegiatan praktikum, peserta praktikum harus menjaga ketenangan, ketertiban, kebersihan dan kerapian.
4. Peserta praktikum harus datang tepat waktu, dengan waktu toleransi keterlambatan 15 menit.
5. Alat komunikasi dinyalakan dalam mode silent atau dimatikan.
6. Selama praktikum, peserta praktikum tidak diperkenankan meninggalkan ruangan praktikum/lab tanpa seizin pengampu/dosen praktikum.
7. Peserta praktikum harus bertutur kata yang baik dan sopan kepada pengampu/dosen praktikum.
8. Peserta praktikum harus dapat menunjukkan sikap kejujuran. Apabila menemukan perlengkapan praktikum peserta praktikum lain yang tertinggal di komputer atau meja yang digunakan, maka dapat memberitahukan kepada petugas lab untuk selanjutnya akan dibuat pengumuman.
9. Jadwal kegiatan praktikum yang karena sesuatu sebab tidak bisa dilaksanakan, maka akan dicari jadwal pengganti dihari lain berdasarkan kesepakatan pengampu/dosen praktikum dengan peserta praktikum.
10. Peserta praktikum harus memenuhi minimal kehadiran yaitu 75% dari total pertemuan.
11. Pada saat dilaksanakan ujian responsi, peserta praktikum diwajibkan menggunakan jas almamater.
12. Hal-hal yang belum diatur pada tata tertib ini akan diatur kemudian.

## **C. Sangsi**

1. Pengampu/dosen praktikum berhak memperingatkan bahkan mengeluarkan peserta praktikum yang tidak dapat menjaga ketenangan, ketertiban, kebersihan dan kerapian.
2. Apabila peserta praktikum datang lebih dari 15 menit, maka tidak diperkenankan mengikuti praktikum.
3. Peserta praktikum yang tidak memenuhi minimal kehadiran 75% dari total pertemuan, maka tidak diperkenankan mengikuti responsi/UAS. Dengan demikian maka nilai akhir untuk mata kuliah praktikum adalah E.

#### **D. Penilaian**

Kegiatan praktikum akan dinilai dengan beberapa komponen sebagai berikut:

No	Komponen Penilaian	Bobot Penilaian
1	Praktikum Harian	30%
2	Tugas	30%
3	Responsi/UAS	40%

## DAFTAR ISI

	Hal
JUDUL	i
KETENTUAN KEGIATAN PRAKTIKUM	ii
DAFTAR ISI	iv
KATA PENGANTAR	v
BAB I      Bekerja Dengan Microsoft Access	1
BAB II     Mengelola <i>Database</i>	14
BAB III    Query Dasar	26
BAB IV    Query Lanjut	43
BAB V    Studi Kasus dan Penyelesaiannya	54
BAB VI   Belajar MySQL Dengan <i>Command Line</i>	62
BAB VII   Belajar MySQL Dengan <i>Command Line</i> Lanjut	75
BAB VIII   Studi Kasus dan Penyelesaiannya Dengan <i>Command Line</i>	96
BAB IX    Belajar MySQL Dengan <i>Command Line Expand</i>	104
BAB X    Mengelola <i>Database</i> Dengan PHP MyAdmin	119
BAB XI    Mengelola <i>Database</i> Dengan PHP MyAdmin <i>Expand</i>	133
BAB XII   Serba Serbi	147

## KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah S.W.T, karena berkat Rahmat dan Karunia-Nya modul praktikum matakuliah “Sistem Basis Data Dasar” bisa diselesaikan. Tujuan dari penyusunan modul ini adalah agar kegiatan praktikum dapat berjalan dengan panduan yang jelas dan target pembelajaran dapat tercapai dengan maksimal.

Penyelesaian modul ini juga tidak terlepas dari peran pihak-pihak memberikan inspirasi dan dukungan. Pada kesempatan ini penulis tidak lupa mengucapkan terima kasih kepada:

1. Dekan Fakultas Sains dan Teknologi yang selalu memberikan dukungan agar setiap kegiatan praktikum memiliki modul.
2. Ketua Program Studi Manajemen Informatika yang telah memberikan arahan dan dukungan agar semua dosen memiliki modul dengan standar kualitas yang baik agar kualitas pembelajaran di Program Studi Manajemen Informatika dapat terjamin.
3. Semua rekan dosen Program Studi Manajemen Informatika yang telah memberikan inspirasi dan koreksi terhadap penyelesaian modul ini.
4. Semua pihak yang tidak bisa disebutkan satu persatu.

Penulis menyadari, bahwa modul ini masih banyak kekurangannya. Penulis berharap kritik dan saran pembaca demi peningkatan kualitas modul ini dimasa yang akan datang.

Akhirnya penulis mengucapkan, semoga modul ini bisa bermanfaat bagi siapa saja yang telah menggunakannya.

Yogyakarta, April 2015  
Penulis

Zaidir

# BAB 1

## BEKERJA DENGAN MICROSOFT ACCESS

### 1. Pengantar Microsoft Access

**Microsoft Access** (atau **Microsoft Office Access**) adalah sebuah program aplikasi basis data komputer relasional yang ditujukan untuk kalangan rumahan dan perusahaan kecil hingga menengah. Aplikasi ini merupakan anggota dari beberapa aplikasi **Microsoft Office**, selain tentunya Microsoft Word, Microsoft Excel, dan Microsoft PowerPoint. Aplikasi ini menggunakan mesin basis data **Microsoft Jet Database Engine**, dan juga menggunakan tampilan grafis yang intuitif sehingga memudahkan pengguna.

Program ini mulai diperkenalkan pada tahun 1993 dan banyak mendapatkan sambutan. **Microsoft Access** merupakan sistem basis data berbasis Windows yang mendukung kemampuan *DBMS (Database Management System)*. Memiliki banyak kelebihan dan kemudahan dari jenis pengolahan data lainnya (misal: Dbase, FoxPro, dll). **Microsoft Access** (disingkat: MS Access) dapat membuat dan memodifikasi tabel, formulir entry data, laporan, dan *query* basis data.

MS Access dapat mengelola berbagai macam informasi sederhana hingga menengah seperti data pegawai, data perhotelan, data pegadaian, data rawat jalan, data penjualan, data pembelian, data persediaan barang, dsb. Dalam tingkat selanjutnya dapat mengembangkan aplikasi pengolahan data dengan bahasa pemrograman Visual seperti **Visual Basic** atau **Borland Delphi**.

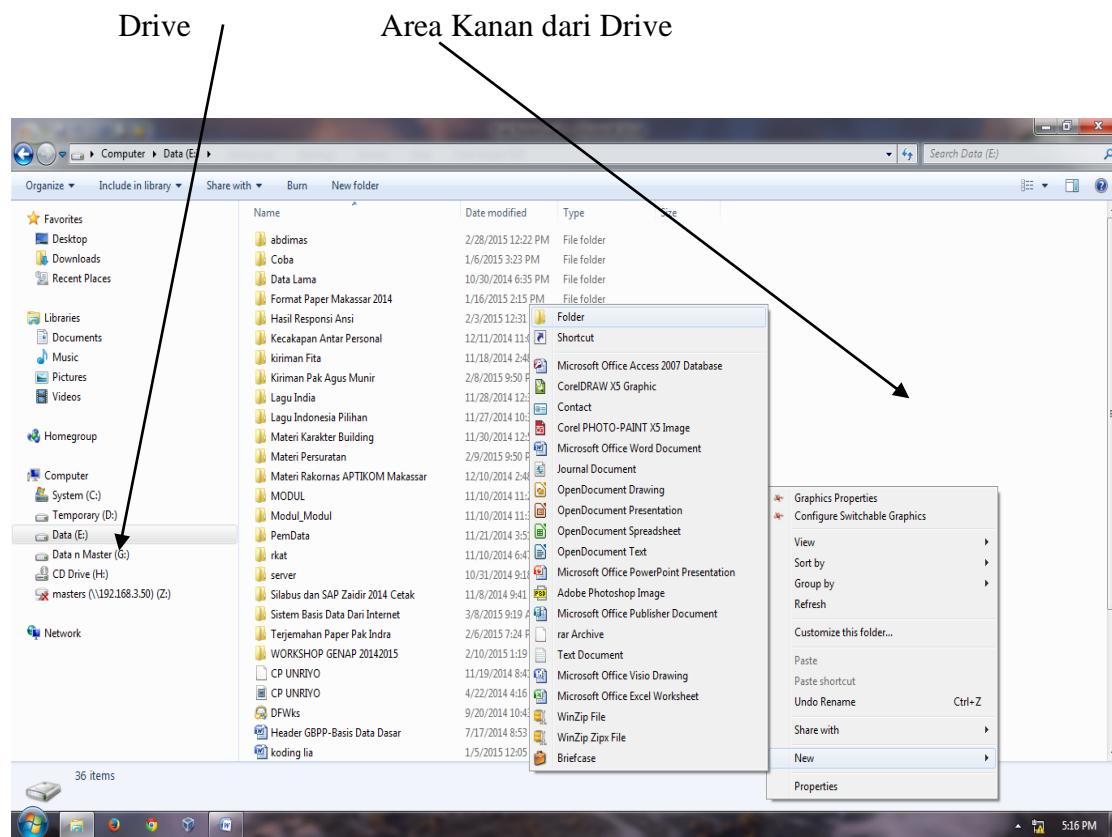
Dengan **Microsoft Accces**, dapat merancang dan mengelola *database* yang saling terkait, antara pembuatan **Tabel**, **Form**, **Query**, **Report**, **Macro**, **Modul**, dan **Pages**, sehingga dapat dikoordinasikan dalam pembuatannya: tombol Macro, Menu Drop Down dan Menu **Switchboard**.

Seiring dengan perkembangan perangkat lunak (*software*) khususnya Microsoft Office, maka Microsoft Access juga mengalami perubahan. Perubahan yang dimaksudkan tidak saja dari segi bentuk tampilannya, namun juga dari segi nama perluasanya (ekstensi). Pada versi sebelum 2007, maka file yang dibuat dengan Microsoft Access memiliki ekstensi **.MDB**. Pada versi 2007 ke atas maka ekstensinya adalah **ACCDB**. Pemahaman ekstensi ini menjadi penting, apabila *file database* yang dibuat dengan Microsoft Access digunakan untuk aplikasi lain seperti Pemrograman Visual. Apabila akan digunakan untuk pemrograman Visual Basic 6 atau Delphi 7, maka sebaiknya versi yang dibuat adalah MDB (dibuat dengan cara **Save As**).

## 2. Menyiapkan Folder Kerja

Supaya dalam kegiatan praktikum bisa lebih teratur, maka perlu dipersiapkan folder yang akan digunakan selama kegiatan praktikum. Untuk kegiatan praktikum mata kuliah ini, diminta kepada semua peserta membuat *folder* yang seragam dengan aturan sebagai berikut:

- a. Nama *folder* diawali dengan kata SBD dilanjutkan dengan NIM Anda masing-masing
- b. Sebagai contoh, jika Nomor Induk Masiswa (NIM) Anda adalah **17240001**, maka nama folder yang Anda buat adalah **SBD17240001**.
- c. Langkah untuk membuat folder adalah dengan cara:
  - Aktifkan **Windows Explorer**.
  - Klik **Drive** dimana *folder* akan diciptakan.
  - Klik kanan pada area bagian kanan yang kosong dari *drive* yang telah dipilih.
  - Pilih **New**.
  - Pilih **Folder**.
  - Ketik nama folder yang diinginkan, misalnya **SBD17240001**.



Gambar 1 : Membuat Folder Kerja

### 3. Memanggil Microsoft Access

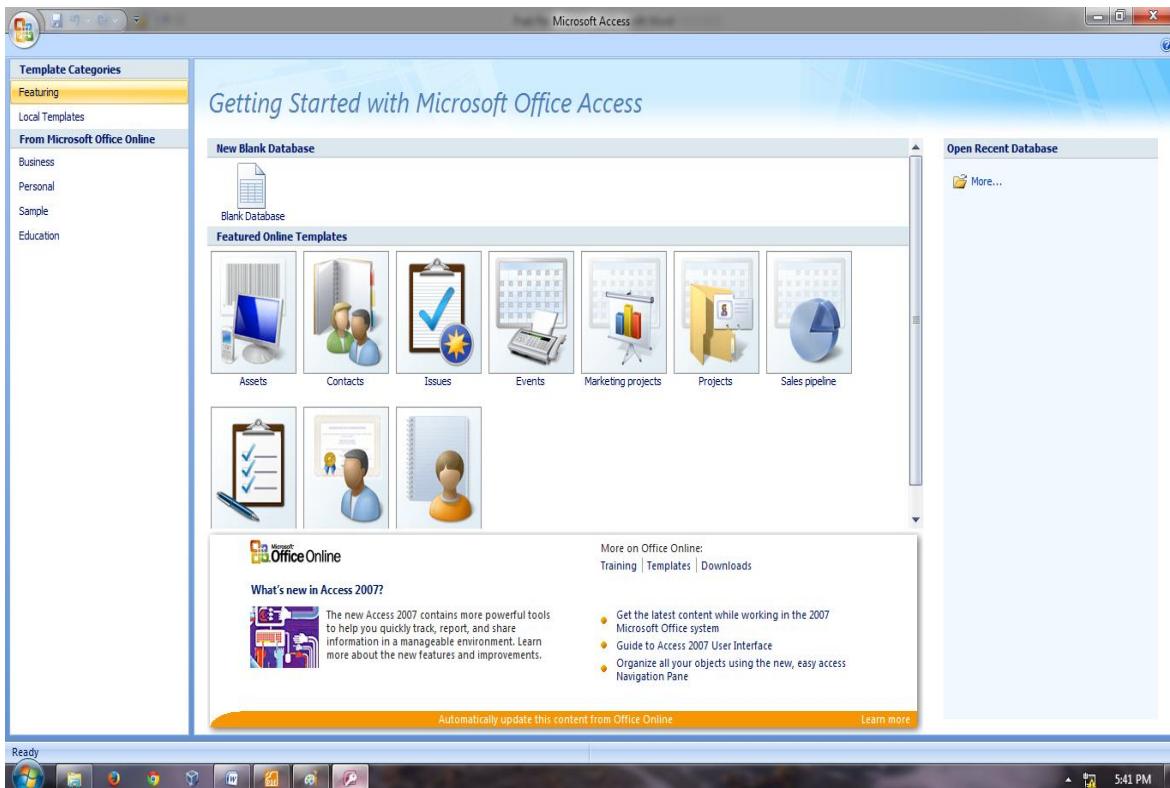
Pada kegiatan praktikum ini versi Microsoft Access yang digunakan adalah versi Microsoft Access 2007. Apabila saat ini Anda memiliki versi yang lebih baru, maka langkah-langkah penggunaannya tidak jauh berbeda. Untuk memanggil aplikasi **Microsoft Access** dapat dilakukan dengan beberapa langkah sebagai berikut:

- a. Klik tombol **Start**.
- b. Klik **All Programs**.
- c. Cari folder dimana **Microsoft Office** berada dan lanjutkan dengan mengklik folder tersebut.
- d. Klik pilihan **Microsoft Access 2007**, seperti terlihat pada Gambar 2.



Gambar 2 Memanggil Microsoft Access

- e. Setelah mengklik pilihan **Microsoft Access 2007**, maka tunggu beberapa saat sampai ditampilkan area atau jendela kerja dari Microsoft Access seperti Gambar 3.



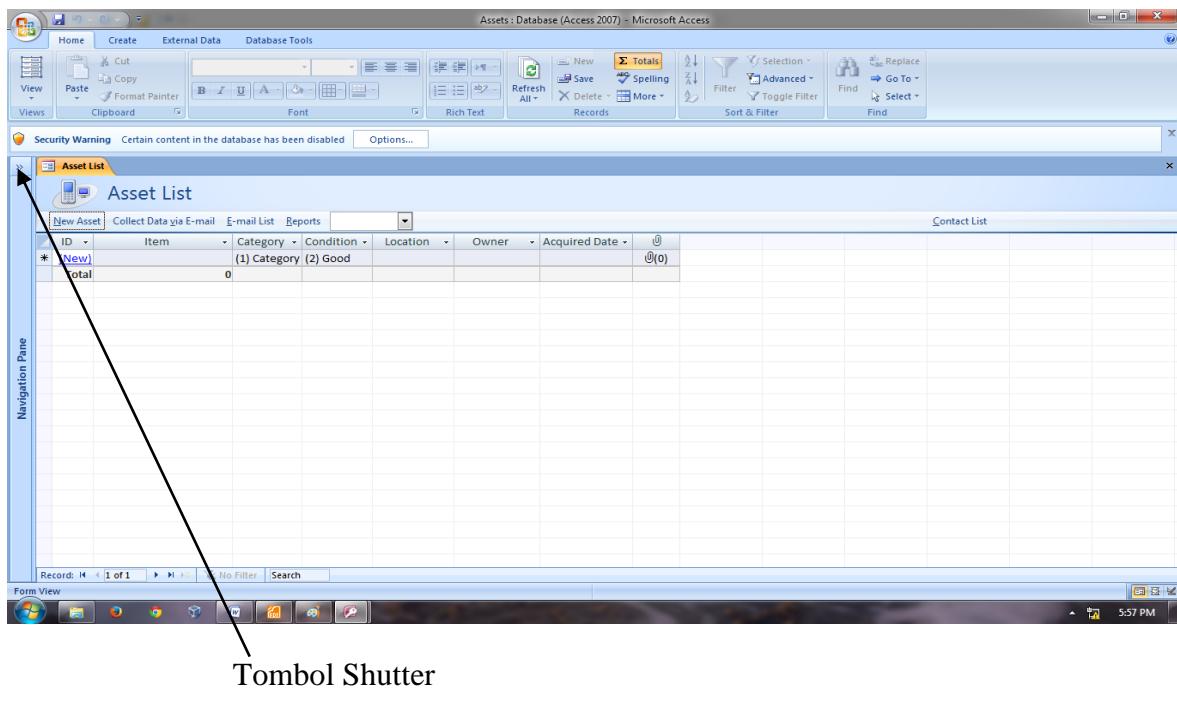
Gambar 3 Jendela Kerja Microsoft Access

#### 4. Membuat Database Dengan Microsoft Access

Untuk membuat *database* dengan **Microsoft Access** dapat dilakukan dengan dua cara yaitu secara manual (*blank database*) atau secara otomatis dengan memanfaatkan template yang ada (templates).

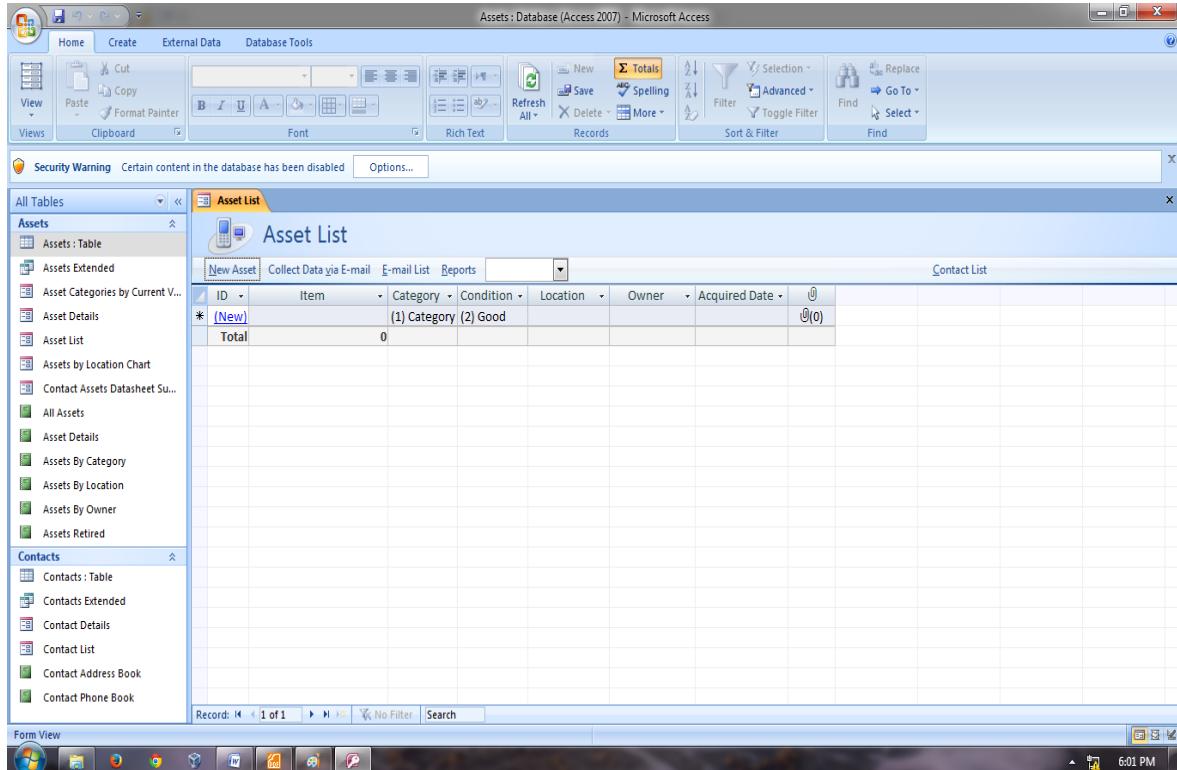
a. Membuat database secara otomatis (templates)

- 1) Klik kategori templates yang diinginkan (**Local Templates**)
- 2) Pilih dengan mengklik salah satu pilihan yang ada ada kotak Local Templates, misalnya Assets
- 3) Ketik nama *file database* yang diinginkan (secara *default* nama *file database* adalah **Asset**). Silahkan gunakan saja
- 4) Tentukan *directory* atau *folder* dimana database akan disimpan (secara *default* nama *directory* atau *foldernya* adalah **Document**). Silahkan gunakan saja
- 5) Jika sudah benar nama *file* dan *directory*, maka lanjutkan dengan mengklik tombol **Create**. Tunggu beberapa saat sampai ditampilkan hasil seperti terlihat pada Gambar 4.



Gambar 4 Tampilan Hasil Database Secara Otomatis

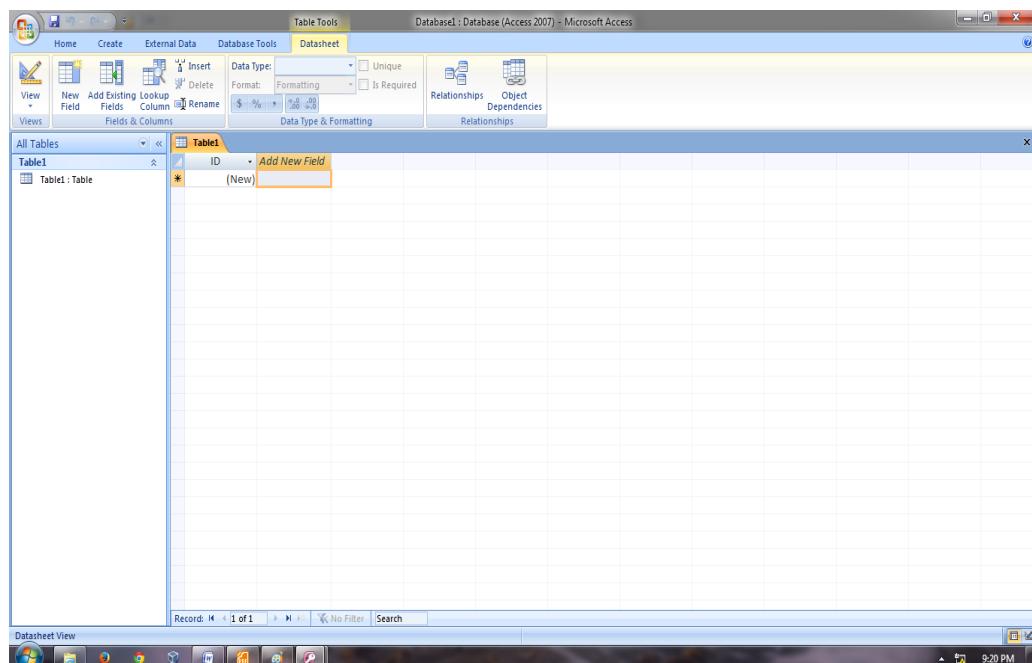
Klik tombol **Shuttle bar** sehingga akan ditampilkan *database* lengkap dari **Asset** yang diciptakan seperti terlihat pada Gambar 5.



Gambar 5 Hasil Lengkap Database Assets

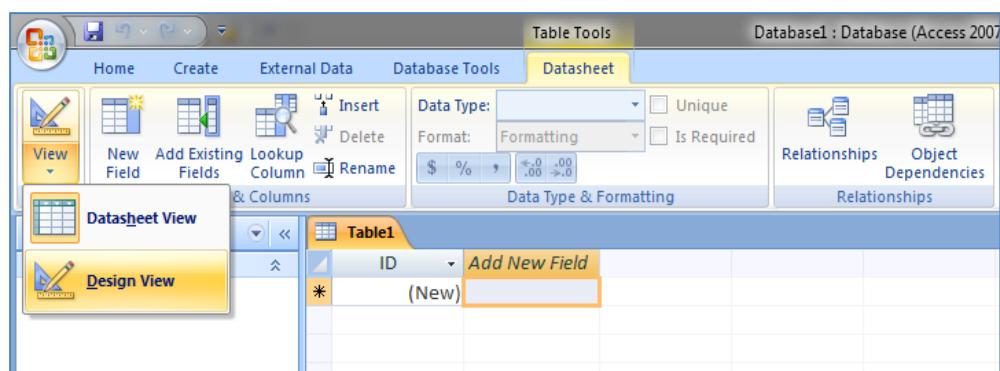
b. Membuat *database* secara manual (*blank database*)

- 1) Klik **Blank Database** pada bagian **New Blank Database**
- 2) Ketik nama *file database* yang diinginkan (secara default nama *file database* adalah **Database1**). Silahkan gunakan saja.
- 3) Tentukan *directory* atau *folder* dimana *database* akan disimpan (secara default nama *directory* atau *foldernya* adalah **Document**). Silahkan gunakan saja.
- 4) Jika sudah benar nama *file* dan *directory*, maka lanjutkan dengan mengklik tombol **Create**. Tunggu beberapa saat sampai ditampilkan hasil seperti terlihat pada Gambar 6.



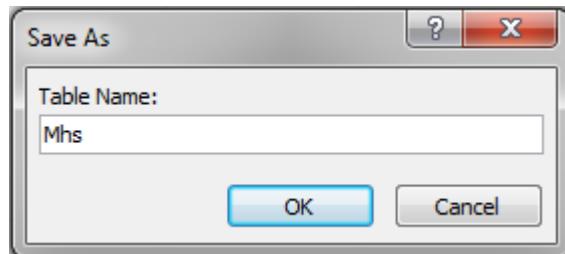
Gambar 6 Tampilan *Blank Database*

- 5) Pada saat ditampilkan seperti Gambar 6, proses pengelolaan *database* sudah bisa dilakukan terutama dalam hal pembuatan tabel. Supaya pembuatan tabel bisa lebih interaktif maka klik menu **View** lalu pilih **Design View** seperti terlihat pada Gambar 7



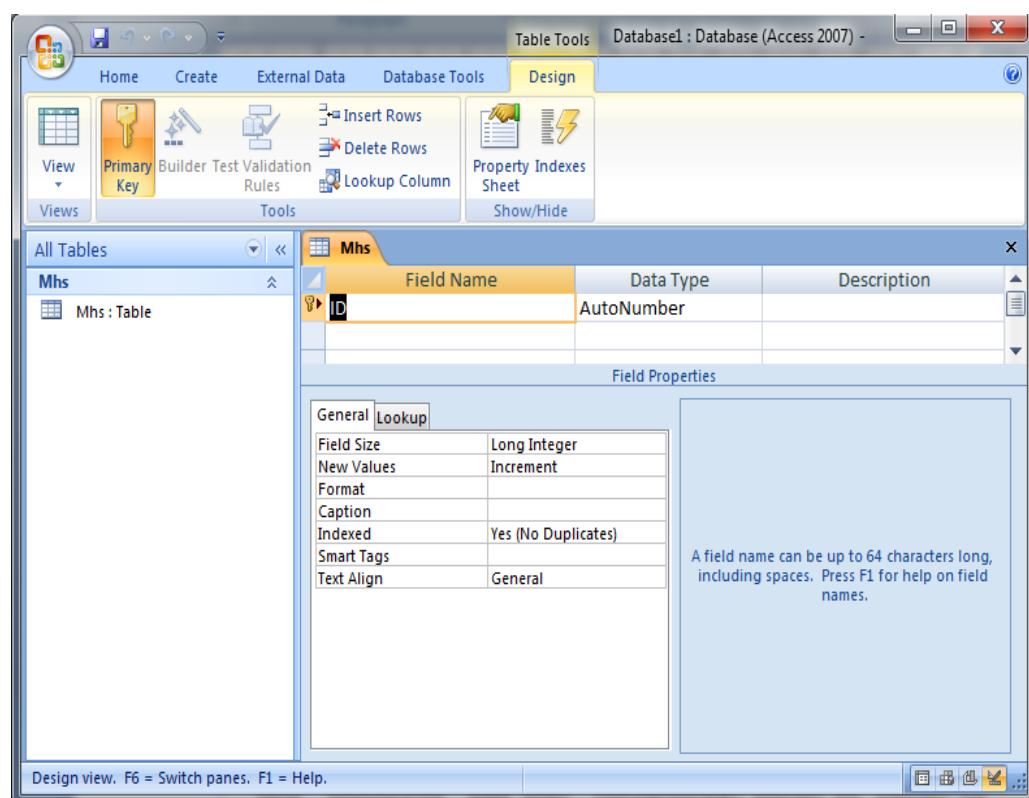
Gambar 7 Proses Merubah Tampilan Design Tabel

- 6) Setelah melakukan langkah 5 di atas, maka selanjutkan diminta untuk memberi nama tabel (secara default diberi nama **Table1**). Silahkan memberi nama *file* sesuai keinginan atau menggunakan nama yang ditawarkan, lihat Gambar 8. Untuk contoh nama *file* yang diberikan adalah **Mhs**.



Gambar 8 Proses Memberi nama Tabel

- 7) Setelah memberi nama tabel maka tampilan akan berubah seperti Gambar 9.



Gambar 9 Tampilan *Design Table*

Pada saat tampilan seperti Gambar 9, proses membuat tabel bisa dilanjutkan dengan menambahkan *field-field* yang dibutuhkan sekalian menentukan jenis datanya.

Perlu diketahui bahwa komponen *database* tidak hanya tabel masih ada komponen lain.

## 5. Objek Database Pada Microsoft Access

Objek *database* merupakan tempat objek-objek (bisa berupa **Table**, **Form**, **Report**, **Query** atau yang lainnya). Kesatuan objek atau komponen ini akan membentuk *database*. Dalam penggunaannya, pada Microsoft Access versi sebelumnya juga telah mencantumkan elemen ini. Elemen ini mempermudah dalam mengelola atau mengedit objek -objek dari *database*.

Nama Objek	Keterangan/Kegunaan
<b>Table</b>	Merupakan struktur data untuk menentukan nama field-field beserta perancangan suatu tabel.
<b>Form</b>	Untuk membuat rancangan dalam tampilan form sebagai bahan untuk memasukan data.
<b>Queries</b>	Untuk membuat relasi atau pengabungan dari beberapa tabel
<b>Report</b>	Untuk menampilkan bentuk laporan dari hasil <i>output</i> yang sesuai dengan data yang diproses dan di entri.
<b>Macros</b>	Untuk mengkoordinasikan dari menu Tabel, form, Report, Query menjadi tombol dan menu secara otomatis
<b>Pages</b>	Untuk merancang tampilan data access pada halaman web melalui sistem jaringan ( <i>server</i> ).
<b>Modules</b>	Untuk membuat rancangan modul pengolahan database yang dibuat dengan menggunakan kode pada Visual Basic untuk aplikasinya

## 6. Membuat Tabel

Tabel dalam MS Access adalah bagian dari *database* yang merupakan tempat menampung data dan dibuat sesuai dengan klasifikasi data tersebut.

Dalam Tabel juga dikenal istilah:

- **Field**
- **Record**
- **Data**

Contoh tabel:

- Tabel Pegawai:  
Memuat data-data seperti No.induk pegawai, nama pegawai, alamat, golongan/pangkat, jabatan, tanggal lahir, tanggal mulai Tugas, pendidikan terakhir, dsb.
- Tabel Gaji/Upah:  
Memuat data-data seperti nomor induk, gaji pokok, pajak, dsb.

### a. *Field*

*Field* adalah bagian dari tabel yang merupakan item-item (kolom) dari data. *Field* dibagi tiga komponen yaitu:

- Nama *Field*, pada sistem operasi windows boleh dibuat bebas seperti contoh: Nomor Induk, Nama Pegawai, Golongan, Tgl\_lahir, dsb
- Tipe *Field*, Ada beberapa macam tipe dari *field*.
- Properti *Field*, berfungsi untuk mengatur masukan dan tampilan data.

### b. Tipe *Field*

Beberapa macam tipe dari *field*, contoh :

- String: Tipe *field* untuk data-data string/character seperti NIP,Nama, Alamat, Golongan, dsb.
- Number: Tipe *field* untuk data-data numeric/angka. Number dapat dipecah beberapa jenis seperti pada tabel berikut :

Field Size	Jangkauan	Ketelitian Desimal	Ukuran Memori
Byte	0 s/d 255	0	1 Byte
Integer	-31.768 s/d 32.767	0	2 Byte
Long Integer	-2.147.483.648 s/d 2.147.483.647	0	4 Byte
Single	-340283e38 s/d 3402823e38	7	4 Byte
Double	-1797631348623e308 – 179769313486232e308	15	8 Byte
Replication ID	Global Unit identifier (GUI)	-	16 Byte

- Date/Time: Tipe *field* untuk data-data tanggal/jam, seperti: tgl\_lahir,TMT, jamdatang, dsb.
- Currency: Tipe *field* untuk data yang berhubungan dengan uang, seperti : Gaji, Bonus, Tunjangan, dsb.
- Memo: Tipe *field* untuk data-data memo seperti keterangan, catatan, dsb.
- Yes/No: Tipe *field* untuk data-data yang berisi hanya peryataan true/false atau Ya/ tidak, seperti contoh : Lunas, Status, dsb.
- Autonumber: Tipe *field* untuk data-data penomoran otomatis, seperti contoh : Nomor, Transaction\_Id, dsb.
- OLE DB: Tipe *field* untuk data-data gambar/foto.

### c. Property Field

Properti pada *field* memuat:

- 1) Field Size: Untuk mengatur panjang pendek data yang dimasukkan pada *field*.
- 2) Format: Untuk membentuk tampilan data yang dimasukkan seperti contoh : jika jenis data tersebut number maka dapat ditampilkan dengan pemisah ribuan, jika jenis data tersebut date/tanggal maka dapat ditampilkan model angka (28/08/99) atau model text (28-Aug-99), dsb.
- 3) Decimal Places: Untuk mengatur jumlah angka desimal untuk jenis data number/currency.
- 4) Input Mask: Untuk mengatur character tambahan yang tampil pada saat memasukkan data, seperti contoh: field Nomor telepon dgn menggunakan ((9999)-99999999), field tanggal dengan menggunakan (99/99/99). Komponen Input Mask dapat dilihat pada tabel berikut.
- 5) Caption: digunakan sebagai label dari field yang bersangkutan.
- 6) Default Value: Untuk memberikan nilai baku yang diisikan pada suatu field. Apabila field tersebut tidak diisi data, maka field tersebut sudah terisi dengan data baku tersebut.
- 7) Validation Rule: Untuk membuat validasi/ ketentuan sebuah pengisian field, seperti contoh : pengisian untuk field Jumlah Barang tidak boleh > 500, field jenis kelamin hanya boleh diisi dengan "L" atau "P" saja, dsb.
- 8) Validation text: Untuk membuat tampilan pesan apabila pengisian data tidak sesuai validation rule.
- 9) Required: Untuk menetapkan apakah field ini harus diisi atau tidak (blank).
- 10) Allow zero length: Untuk menetapkan field text atau memo harus diisi atau bisa dibiarkan kosong.
- 11) Indexed: Untuk menetapkan field ini sebagai index dan tidak boleh ada duplikasi data, atau Index dan boleh duplikasi data, atau bukan index.

Tabel Komponen Input Mask

Kode	Keterangan
0	digit angka, harus diisi.
9	digit angka, boleh tidak diisi.
#	digit angka, boleh tidak diisi dan bisa menampilkan tanda (-) apabila datanya kurang dari 0.
L	Semua jenis huruf, harus diisi.
?	Semua jenis huruf , boleh tidak diisi

Kode	Keterangan
A	Semua jenis huruf/ angka, harus diisi
a	Semua jenis huruf & angka, tidak harus diisi.
&	Semua jenis huruf termasuk spasi, harus diisi.
C	Semua jenis huruf termasuk spasi, tidak harus diisi.
. : ; - /	Tanda pemisah yang biasa dipakai untuk format angka, tanggal atau jam.
<	Indikator untuk menunjukan huruf pada sebelah kanan tanda tersebut di konversi menjadi huruf kecil.
>	Indikator untuk menunjukan huruf pada sebelah kanan tanda tersebut di konversi menjadi huruf besar.
!	Model pengisian dari kanan kekiri.
\	Apabila ada data yang menggunakan tanda \ pada awal tulisan, tanda \ tersebut ditiadakan.

d. *Record dan Data*

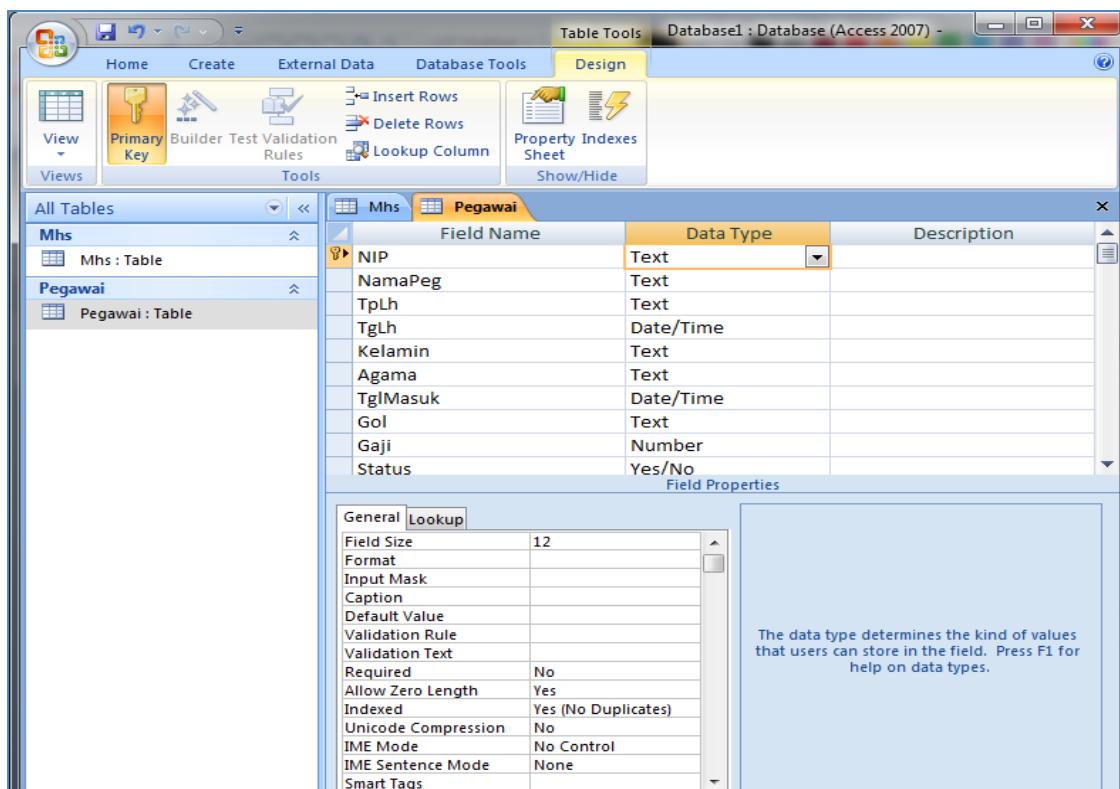
*Record* merupakan kumpulan data sesuai dengan *field-field* yang sudah ditentukan.

Data merupakan gabungan dari character-character yang mempunyai suatu arti dan pengertian.

e. Langkah Membuat Tabel

Untuk membuat tabel lakukan langkah-langkah sebagai berikut:

- 1) Klik menu **Create**
- 2) Pilih/klik icon **Table**
- 3) Selanjutnya ganti tampilan ke bentuk **Design Tabel** seperti terlihat pada Gambar 7.
- 4) Ketik nama tabel dengan nama **Pegawai**
- 5) Buatlah *field-field* yang dibutuhkan untuk table Pegawai seperti terlihat pada Gambar 10.



Gambar 10 Rancangan Tabel Pegawai

## 7. Mengisi Tabel

Setelah *field-field* yang dibutuhkan selesai dibuat, maka data-data siap untuk diisikan. Cara untuk mengisi data adalah dengan cara mengklik icon **View**. Untuk latihan isilah data seperti Gambar 11.

	NIP	NamaPeg	Tplh	TgLh	Kelamin	Agama	TglMasuk	Gol	Gaji	Status	Add New Field
	10000000001	Andi	Yogya	2/2/1970	L	Islam	1/3/2010	III/A	2500000	<input checked="" type="checkbox"/>	
	10000000002	Pamela	Sleman		P	Kristen		II/B	1550000	<input checked="" type="checkbox"/>	
	10000000003	Rita Susanti	Bantul		P	Katholik		III/C	3000000	<input checked="" type="checkbox"/>	
	10000000004	Hamka	Jakarta		L	Islam		III/A	2500000	<input checked="" type="checkbox"/>	
	10000000005	Ilham	Bandung		L	Islam		II/A	1250000	<input checked="" type="checkbox"/>	
*	10000000006	Febrina	Surabaya		P	Islam		IV/A	5000000	<input checked="" type="checkbox"/>	

Gambar 11 Tampilan Pengisian Data (*View*) Tabel Pegawai

## 8. Menutup Database

Untuk menutup *database* yang aktif kita bisa . Klik **Office Button** dan pilih **Exit Access** atau bisa juga dengan mengklik tombol perintah **Close** pada pojok kanan atas.

## 9. Membuka Database

Untuk membuka file *database* dapat dilakukan dengan langkah berikut :

- a. Klik **Office Button** pada pojok kiri atas, lalu klik **Open**.
- b. Di dalam daftar *folder*, cari folder atau alamat tempat disimpannya *file database* yang akan digunakan.
- c. Ketika sudah menemukan *file database* yang diinginkan, pilih salah satu langkah di bawah ini:
  - Double klik *file database* untuk membukanya.
  - Pilih *file database* yang diinginkan lalu klik tombol **Open** untuk membukanya.

## 10. Soal Latihan

- a. Buatlah *database* baru untuk mengelola sistem informasi perhotelan dengan nama **DBHOTEL**.
- b. Buatlah tabel untuk menyimpan data tamu dengan *field-field* dan jenis data sebagai berikut:

Nama Field	Tipe Data	Ukuran
Kdtamu	Text	8
Nmtamu	Text	40
Altamu	Text	75
Telptamu	Text	12
Kwg	Text	3
Pekerjaan	Text	15

- c. Isilah data minimal 5 record

Contoh:

Kdtamu	Nmtamu	Altamu	Tepltamu	Kwg	Pekerjaan
KT000001	Hendri	Jl. Sudirman No. 32 Yogyakarta	081324543433	WNI	Peg. Swasta
KT000002	Alex	Jl. A Yani No. 32 Bandung	085232343223	WNA	Wiraswasta

Silahkan diisi data yang lain.....

Catatan: Database yang dibuat ini akan digunakan untuk latihan pertemuan selanjutnya, maka selalu bawa disaat praktikum

## BAB 2

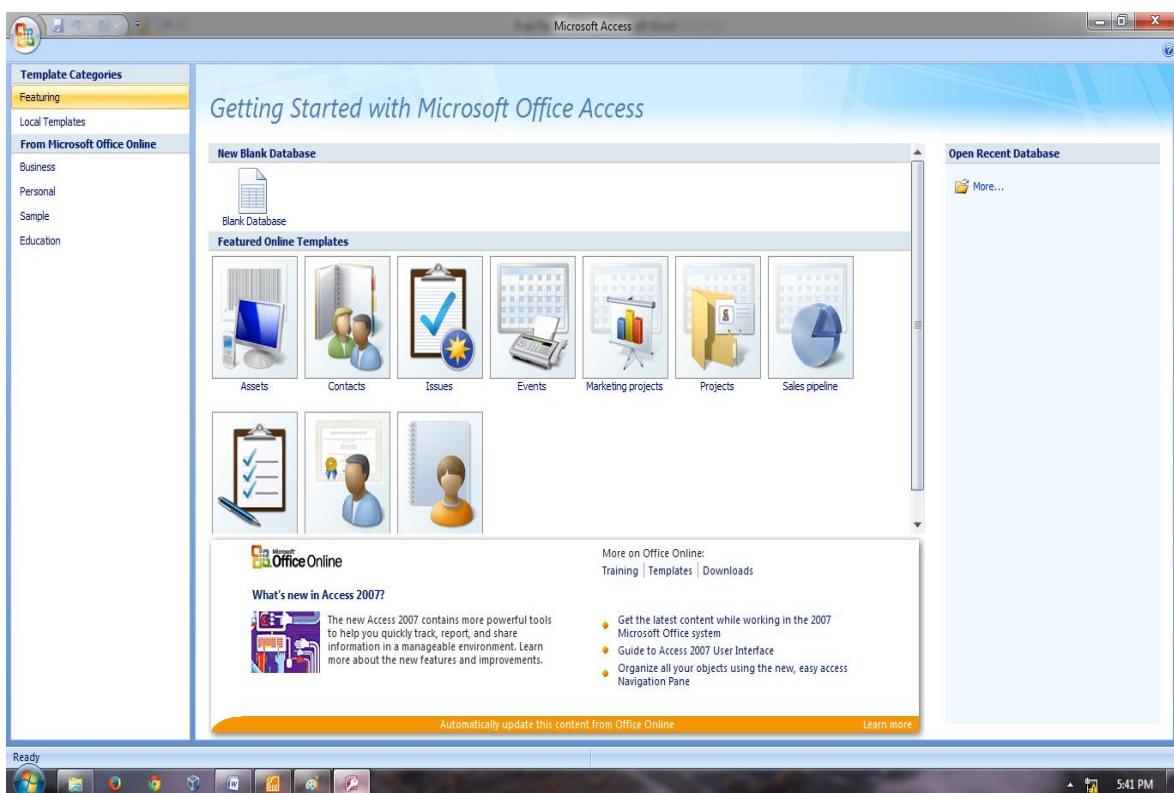
# MENGELOLA DATABASE

Supaya pemahaman terhadap basis data (*database*) lebih dalam, maka pada pertemuan praktikum kali ini akan dimulai dari awal proses mengelola *database* yang digunakan untuk pengelolaan dalam jual beli barang.

### 11. Membuat Database Baru

Untuk membuat *database* baru diawali dengan memanggil aplikasi **Microsoft Access** sebagai berikut:

- f. Klik tombol **Start**
- g. Klik **All Programs**
- h. Cari *folder* dimana **Microsoft Office** berada dan lanjutkan dengan mengklik *folder* tersebut.
- i. Klik pilihan **Microsoft Access 2007** atau versi yang sesuai
- j. Setelah mengklik pilihan **Microsoft Access 2007** atau versi yang sesuai, maka tunggu beberapa saat sampai ditampilkan area atau jendela kerja dari Microsoft Access seperti Gambar 1.

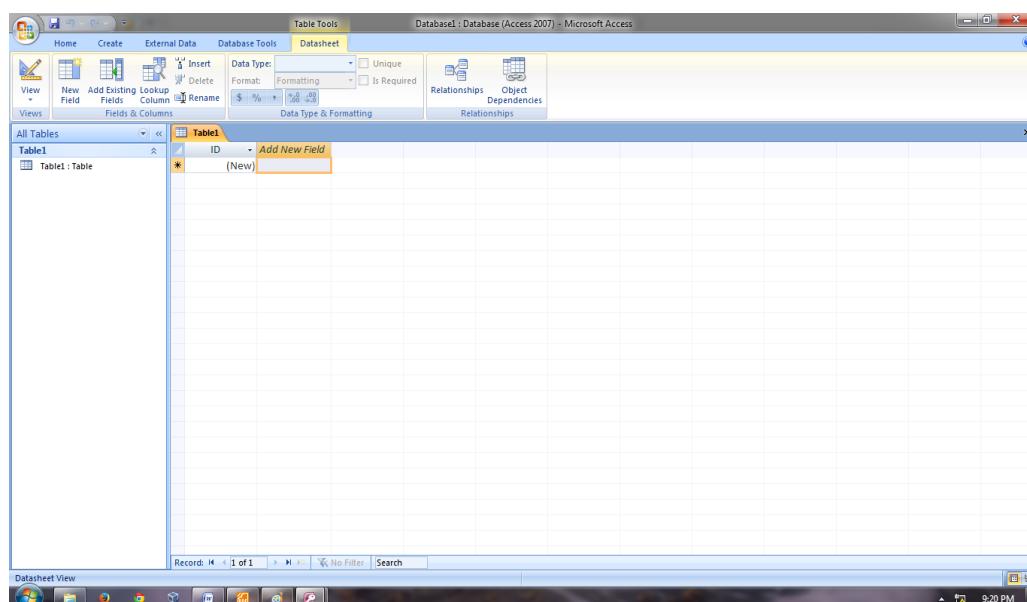


Gambar 1 Jendela Kerja Microsoft Access

## 12. Membuat Database

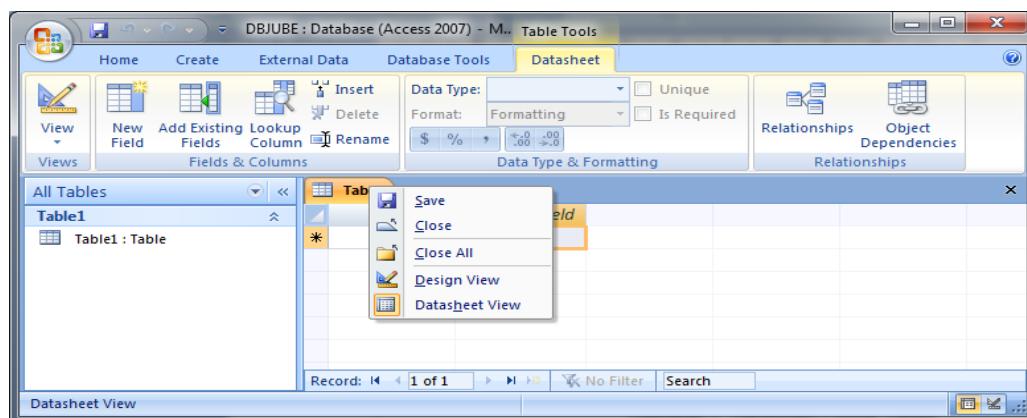
Untuk membuat *database* dengan Microsoft Access dapat dilakukan dengan cara sebagai berikut:

- c. Membuat database secara manual (**blank database**)
  - 8) Klik **Blank Database** pada bagian **New Blank Database**
  - 9) Ketik nama *file database* yang diinginkan. Untuk keperluan praktikum ini beri nama *database* dengan nama **DBJUBE**.
  - 10) Tentukan *directory* atau *folder* dimana *database* akan disimpan. Silahkan arahkan ke *folder* tempat praktikum Anda.
  - 11) Jika nama *file* dan *directory* sudah benar, lanjutkan mengklik tombol **Create**. Tunggu beberapa saat sampai ditampilkan hasil seperti terlihat pada Gambar 2.

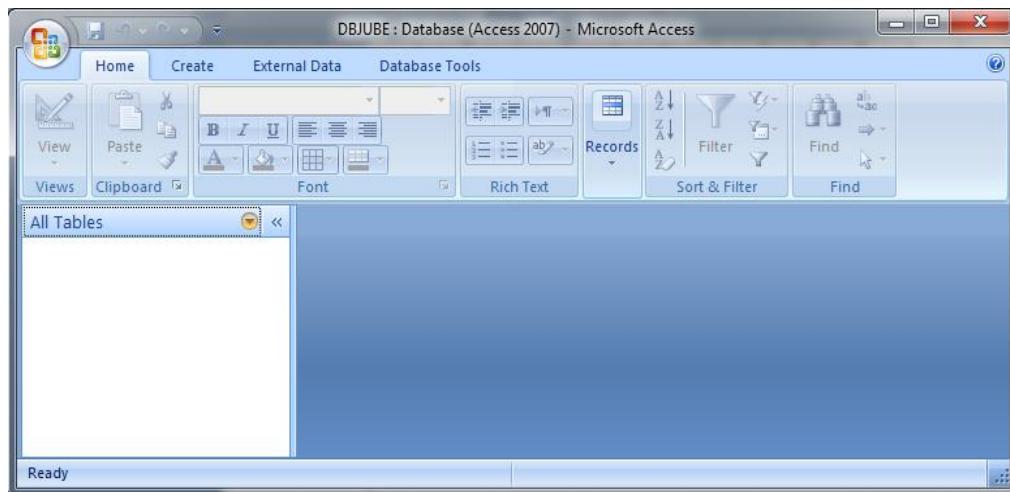


Gambar 2 Tampilan Blank Database

- 12) Untuk menutup tabel yang ditawarkan, klik kanan pada posisi Table1, lalu pilih **Close** (lihat Gambar 3). Dengan demikian tabel akan ditutup dan tampilan akan berubah seperti Gambar 4.



Gambar 3 Tampilan Close Tabel



Gambar 4 Saat Setelah Close Tabel

### 13. Membuat Tabel

Tabel-tabel yang dibutuhkan untuk pengelolaan data dalam sistem jual beli cukup banyak. Tabel-tabel tersebut dikelompokkan dalam dua bagian yaitu tabel induk dan tabel transaksi.

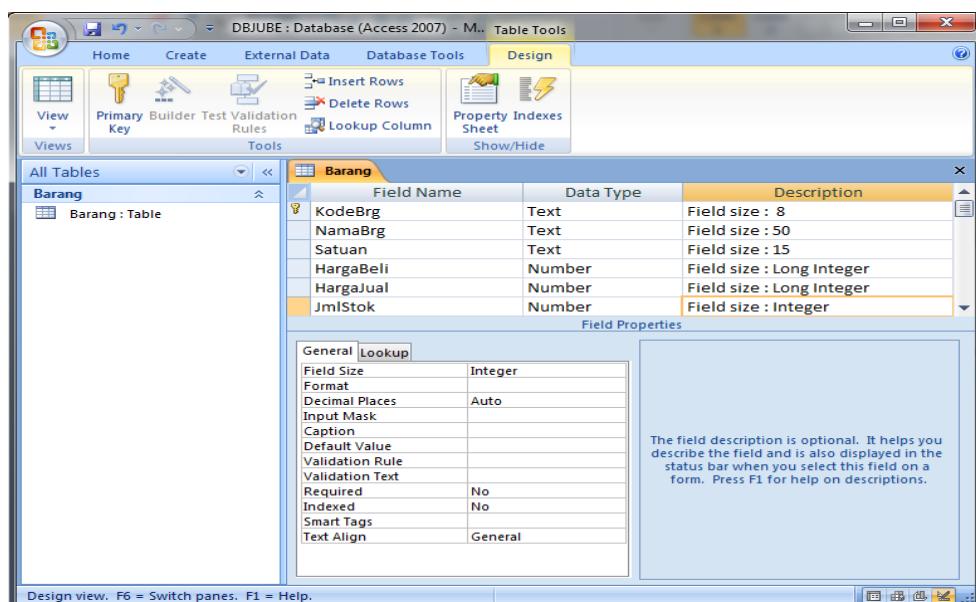
#### a. Tabel induk

Tabel induk yang dibutuhkan antara lain:

- 1) Tabel untuk menyimpan data barang (Barang)

Untuk membuat tabel baru, lakukan dengan langkah sebagai berikut:

- a) Klik menu **Create**
- b) Klik pilihan (icon) **Table**
- c) Ganti tampilan ke bentuk tampilan **Design View**
- d) Ketikkan nama tabel yang diinginkan yaitu **Barang**, kemudian klik tombol **OK**.
- e) Buat atau ketik *field-field* yang dibutuhkan (lihat gambar 5)



Gambar 5 Membuat table Barang

f) Isi data minimal 10 data (record), lihat gambar 6

KodeBrg	NamaBrg	Satuan	HargaBel	HargaJual	JmlStok
B-000001	Monitor	Unit	600000	675000	30
B-000002	Printer	Unit	650000	700000	40
B-000003	Processor Intel	Unit	450000	475000	10
B-000004	Processor AMD	Unit	350000	380000	20
B-000005	RAM 128 MB	Keping	50000	60000	50
B-000006	RAM 256 MB	Keping	75000	90000	50
B-000007	RAM 512 MB	Keping	100000	115000	50
B-000008	RAM 1 GB	Keping	150000	165000	50
B-000009	RAM 2 GB	Keping	200000	250000	50
B-000010	Motherboard	Unit	450000	500000	20
B-000011	Power Supply	Unit	75000	125000	50
B-000012	Casing	Unit	125000	175000	20
B-000013	VGA Card	Unit	250000	300000	20
B-000014	Speaker Aktif	Unit	175000	200000	30
B-000015	Tinta Printer	Dus	50000	65000	40

Gambar 6 Mengisi table Barang

2) Tabel untuk menyimpan data pembeli (Pembeli)

Untuk membuat tabel baru, lakukan dengan langkah sebagai berikut:

- Klik menu **Create**
- Klik pilihan (**icon**) **Table**
- Ganti tampilan ke bentuk tampilan **Design View**
- Ketikkan nama tabel yang diinginkan yaitu **Pembeli**, kemudian klik tombol **OK**.
- Buat atau ketik *field-field* yang dibutuhkan (lihat gambar 7)

Field Name	Data Type	Description
KdPem	Text	Fields : 5
NmPem	Text	Fields : 50
AlPem	Text	Fields : 75
KotaPem	Text	Fields : 15
Telpem	Text	Fields : 15

Gambar 7 Membuat table Pembeli

- f) Isi data minimal 10 data (record), lihat gambar 8

KdPem	NmPem	AlPem	KotaPem	Telpem
P0001	Mr Burhanudin	Jl. Adisucipto No. 42	Yogya	0274543212
P0002	PT. Adi Jaya	Jl. Slamet Riyadi No. 65	Solo	081234543675
P0003	Ny Suci Darwanti	Jl. Pancasila No. 87	Semarang	087232432543
P0004	CV Sneher Jaya	Jl. Janti No. 67	Yogya	0274665478
P0005	Mr Hermantio	Jl. Beo No. 78	Yogya	085320435443
P0006	CV Indo Pratama	Jl. Cibodas No. 98	Bandung	081234323212
P0007	Ny Titik Wijaya	Jl. Gejayan No. 87	Yogya	081234343245
P0008	Mr Ahmad W	Jl. Kartini No. 342	Solo	081345675872
P0009	Ny Liliana Sutarjo	Jl. Sudirman No. 24	Yogya	087324323456
P0010	Nn Windarwati M	Jl. Gejayan No. 98	Yogya	0274986543

Gambar 8 Mengisi tabel Pembeli

- 3) Tabel untuk menyimpan data supplier (Suplier)

Untuk membuat tabel baru, lakukan dengan langkah sebagai berikut:

- Klik menu **Create**
- Klik pilihan (**icon**) **Table**
- Ganti tampilan ke bentuk tampilan **Design View**
- Ketikkan nama tabel yang diinginkan yaitu **Suplier**, kemudian klik tombol **OK**.
- Buat atau ketik *field-field* yang dibutuhkan (lihat Gambar 9)

Field Name	Data Type	Description
KdSup	Text	Field Size : 5
NmSup	Text	Field Size : 50
AlSup	Text	Field Size : 15
KotaSup	Text	Field Size : 15
TelpSup	Text	Field Size : 15

Gambar 9 Membuat tabel Suplier

f) Isi data minimal 10 data (record), lihat gambar 10.

The screenshot shows the Microsoft Access 2007 interface with the 'Datasheet' tab selected. On the left, there's a list of tables: 'Barang', 'Pembeli', 'Suplier'. The 'Suplier' table is open and displayed in a grid format. The columns are labeled: KdSup, NmSup, AlSup, KotaSup, and TelpSup. The data entries are as follows:

KdSup	NmSup	AlSup	KotaSup	TelpSup
S-001	PT. Tecno Indotama	Jl. Sudirman No. 87	Jakarta	081567898765
S-002	PT. Megah Jaya	Jl. Suprapto No. 23	Surabaya	023865754
S-003	PT. Sumber Makmur	Jl. Tendean No. 54	Surabaya	085876777654
S-004	CV. Mitra Tekno	Jl. Kartini No. 87	Jakarta	021456781
S-005	CV. Perkasa Jaya	Jl. C Simanjuntak No. 76	Bandung	085899765434
S-006	PT. MicroCom	Jl. Senopati No. 64	Jakarta	088765432125
S-007	CV. Inti Megasindo	Jl. Sultan Agung No. 80	Jakarta	021456453
S-008	PT. Makmur Sentosa	Jl. A Yani No. 77	Surabaya	085666765431
S-009	CV. Lintas Dunia	Jl. MT Haryono No. 43	Surabaya	081234556576
S-010	PT. Buana Jaya	Jl. Sudirman No. 92	Bandung	022876546
*				

Gambar 10 Mengisi tabel Suplier

b. Tabel Transaksi

Tabel transaksi yang dibutuhkan antara lain:

- 1) Tabel untuk menyimpan data nota penjualan (Jual)

Untuk membuat tabel baru, lakukan dengan langkah sebagai berikut:

- a) Klik menu **Create**
- b) Klik pilihan (**icon**) **Table**
- c) Ganti tampilan ke bentuk tampilan **Design View**
- d) Ketikkan nama tabel yang diinginkan yaitu **Jual**, kemudian klik tombol **OK**.
- e) Buat atau ketik *field-field* yang dibutuhkan (lihat gambar 11)

The screenshot shows the Microsoft Access 2007 interface with the 'Design' tab selected. On the left, there's a list of tables: 'Barang', 'Pembeli', 'Suplier', 'Jual'. The 'Jual' table is open and displayed in a grid format. The columns are labeled: Field Name, Data Type, and Description. The data entries are as follows:

Field Name	Data Type	Description
NoNota	Text	Field Size : 10
TglJual	Date/Time	
KdPem	Text	Field Size : 5

The 'Field Properties' pane is open for the 'KdPem' field, showing various properties such as Field Size (5), Format, Input Mask, Caption, Default Value, Validation Rule, Validation Text, Required (No), Allow Zero Length (Yes), Indexed (No), Unicode Compression (Yes), IME Mode (No Control), IME Sentence Mode (None), and Smart Tags. A note in the pane states: 'The field description is optional. It helps you describe the field and is also displayed in the status bar when you select this field on a form. Press F1 for help on descriptions.'

Gambar 11 Membuat tabel Jual

f) Isi data minimal 10 data (*record*), lihat Gambar 12.

The screenshot shows the Microsoft Access 2007 interface. The ribbon at the top has tabs for Home, Create, External Data, Database Tools, and Datasheet. The Datasheet tab is selected. On the left, there's a navigation pane titled 'All Tables' listing 'Barang', 'Pembeli', 'Suplier', and 'Jual'. The 'Jual' table is currently selected and displayed in the main area. The data consists of 10 rows with columns: NoNotaJ, TglJual, and KdPem. The last row, record 10, is highlighted with an orange border. The status bar at the bottom shows 'Record: 10 of 10'.

NoNotaJ	TglJual	KdPem
NJ00000001	1/6/2014	P0002
NJ00000002	2/8/2014	P0005
NJ00000003	3/12/2014	P0003
NJ00000004	12/12/2014	P0007
NJ00000005	1/2/2015	P0001
NJ00000006	1/5/2015	P0002
NJ00000007	2/3/2015	P0006
NJ00000008	2/5/2015	P0002
NJ00000009	3/6/2015	P0004
NJ00000010	4/5/2015	P0008

Gambar 12 Mengisi tabel Jual

- 2) Tabel untuk menyimpan data rincian/detail penjualan (DJual)  
Untuk membuat tabel baru, lakukan dengan langkah sebagai berikut:
- Klik menu **Create**
  - Klik pilihan (**icon**) **Table**
  - Ganti tampilan ke bentuk tampilan **Design View**
  - Ketikkan nama tabel yang diinginkan yaitu **DJual**, kemudian klik tombol **OK**.
  - Buat atau ketik *field-field* yang dibutuhkan (lihat Gambar 13)

The screenshot shows the Microsoft Access 2007 interface in Design view. The ribbon at the top has tabs for Home, Create, External Data, Database Tools, and Design. The Design tab is selected. On the left, there's a navigation pane titled 'All Tables' listing 'Barang', 'Pembeli', 'Suplier', 'Jual', and 'DJual'. The 'DJual' table is currently selected and displayed in the main area. The table structure shows four fields: 'Field Name' (NoNotaJ, KdBrg, JmlJual, HrgJual) and 'Data Type' (Text, Text, Number, Number). The 'JmlJual' field is highlighted with an orange border. Below the table, the 'Field Properties' pane is open, showing specific settings for the 'JmlJual' field, such as 'Format: Integer' and 'Data Type: Number'. A tooltip on the right provides information about number sizes.

Gambar 13 Membuat tabel DJual

f) Isi data minimal 10 data (record), lihat gambar 14.

NoNotaJ	KdBrg	JmlJual	HrgJual
NJ00000001	B-000003	2	475000
NJ00000001	B-000005	4	60000
NJ00000001	B-000006	2	90000
NJ00000002	B-000001	5	675000
NJ00000003	B-000002	4	700000
NJ00000003	B-000003	3	475000
NJ00000004	B-000006	3	90000
NJ00000004	B-000007	4	115000
NJ00000005	B-000002	6	700000
NJ00000005	B-000004	5	380000

Gambar 14 Mengisi tabel DJual

3) Tabel untuk menyimpan data nota pembelian (Beli)

Untuk membuat tabel baru, lakukan dengan langkah sebagai berikut:

- Klik menu **Create**
- Klik pilihan (**icon**) **Table**
- Ganti tampilan ke bentuk tampilan **Design View**
- Ketikkan nama tabel yang diinginkan yaitu **Beli**, kemudian klik tombol **OK**.
- Buat atau ketik *field-field* yang dibutuhkan (lihat gambar 15)

Field Name	Data Type	Description
NoNotaB	Text	Field Size : 10
TglBeli	Date/Time	
KdSup	Text	Field Size : 5

General Field Properties

Field Size: 5  
Format:  
Input Mask:  
Caption:  
Default Value:  
Validation Rule:  
Validation Text:  
Required: No  
Allow Zero Length: Yes  
Indexed: No  
Unicode Compression: Yes  
IME Mode: No Control  
IME Sentence Mode: None  
Smart Tags:

The field description is optional. It helps you describe the field and is also displayed in the status bar when you select this field on a form. Press F1 for help on descriptions.

Gambar 15 Membuat table Beli

- f) Isi data minimal 10 data (record), lihat gambar 16.

The screenshot shows the Microsoft Access 2007 interface with the 'DBJUBE : Database (Access 2007)' window open. The ribbon tabs include Home, Create, External Data, Database Tools, and Datasheet. The 'Datasheet' tab is active. On the left, the 'All Tables' list shows tables like Barang, Pembeli, Suplier, Jual, DJual, Beli, and DBeli. The 'Beli' table is currently selected and displayed in the main grid. The grid contains 10 records with columns: NoNotaB, TglBeli, and KdSup. The last record (NoNotaB NB00000010) has its 'KdSup' value 'S-001' highlighted with a yellow selection bar. Navigation buttons at the bottom indicate '10 of 10' records.

Gambar 16 Mengisi tabel Beli

- 4) Tabel untuk menyimpan data rincian/detail pembelian (DBeli)  
Untuk membuat tabel baru, lakukan dengan langkah sebagai berikut:
- Klik menu **Create**
  - Klik pilihan (icon) **Table**
  - Ganti tampilan ke bentuk tampilan **Design View**
  - Ketikkan nama tabel yang diinginkan yaitu **DBeli**, kemudian klik tombol **OK**.
  - Buat atau ketik *field-field* yang dibutuhkan (lihat gambar 17)

The screenshot shows the Microsoft Access 2007 interface with the 'DBJUBE : Database (Access 2007)' window open. The ribbon tabs include Home, Create, External Data, Database Tools, and Design. The 'Design' tab is active. On the left, the 'All Tables' list shows tables like Barang, Pembeli, Suplier, Jual, DJual, Beli, and DBeli. The 'DBeli' table is currently selected and displayed in the main grid. The grid shows four fields: NoNotaB, KdBrrg, JmlBeli, and HrgBeli. The 'Field Properties' pane on the right shows detailed settings for the 'NoNotaB' field, such as a field size of 255 and various validation rules. A note in the pane states: 'A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.'

Gambar 17 Membuat tabel DBeli

f) Isi data minimal 10 data (*record*), lihat Gambar 18.

NoNotaB	KdBrrg	JmlBeli	HrgBeli
NB00000001	B-000002	2	650000
NB00000001	B-000004	2	350000
NB00000002	B-000001	3	600000
NB00000002	B-000002	4	650000
NB00000002	B-000005	5	50000
NB00000003	B-000001	2	600000
NB00000004	B-000003	1	450000
NB00000004	B-000006	3	75000
NB00000005	B-000007	2	100000
NB00000005	B-000002	3	650000

Gambar 18 Mengisi tabel DBeli

#### 14. Soal Latihan

- Bukalah *database* **DBHOTEL** yang telah dibuat pada pertemuan sebelumnya.
- Tambahkan atau buat tabel baru untuk menyimpan data jenis kamar dan data kamar sebagai berikut:
  - Tabel untuk menyimpan data jenis kamar (simpan dengan nama **JKamar**)

Nama Field	Tipe Data	Ukuran
KdJKm	Text	4
NmJKm	Text	40
Fasilitas	Text	75
TarifWD	Number	LongInteger
TarifWE	Number	LongInteger

Keterangan:

Field KdJKm (kode jenis kamar) dijadikan sebagai *Primary Key*

TarifWD (Tarif Week Day)

TarifWE (Tarif Week End)

- 2) Tabel untuk menyimpan data kamar (simpan dengan nama **Kamar**)

<b>Nama Field</b>	<b>Tipe Data</b>	<b>Ukuran</b>
NoKm	Text	3
KdJKm	Text	4
StatusKm	Text	1

Keterangan:

Field NoKm dijadikan sebagai *Primary Key*. Field KdJKm dihubungkan dengan field KdJKm pada tabel JKamar dengan cara membuat sebagai *Lookup Wizard*. Field StatusKm dibuat sebagai Combo dengan pilihan (B, I, K)

- 3) Tabel untuk menyimpan data tamu masuk (simpan dengan nama **CekIn**)

<b>Nama Field</b>	<b>Tipe Data</b>	<b>Ukuran</b>
NoReg	Text	10
TglCekin	Date	
KdTamu	Text	8
NoKm	Text	3
Status	Text	1

Keterangan:

Field NoReg dijadikan sebagai *Primary Key*. Field KdTamu dihubungkan dengan field KdTamu pada tabel Tamu dengan cara membuat sebagai *Lookup Wizard*. Field NoKm dihubungkan dengan field NoKm pada tabel Kamar dengan cara membuat sebagai *Lookup Wizard*.

- 4) Tabel untuk menyimpan data tamu keluar (simpan dengan nama **CekOut**)

<b>Nama Field</b>	<b>Tipe Data</b>	<b>Ukuran</b>
NoReg	Text	10
TglCekout	Date	
BeaKamar	Number	Long Integer
BeaLain	Number	Long Integer
Pajak	Number	Long Integer

Keterangan:

Field NoReg dijadikan sebagai *Primary Key*. Field NoReg dihubungkan dengan field NoReg pada tabel **CekIn** dengan cara membuat sebagai *Lookup Wizard*.

- c. Isilah tabel **JKamar** dengan data minimal 5 record. Isi juga tabel **Kamar** dengan data minimal 15 record. Isilah tabel **CekIn** dengan data minimal 10 record, dan tabel **Cekout** juga 10 record.

Contoh data Jenis kamar

<b>KdJKm</b>	<b>NmJKm</b>	<b>Fasilitas</b>	<b>TarifWD</b>	<b>TarifWE</b>
KJ01	Standard	Kipas angin	125000	175000
KJ02	Superior/Premium	Kipas angin, tv	200000	250000
KJ03	Deluxe	AC, TV	300000	350000

Silahkan lanjutkan data yang lain.....

Contoh data Kamar

<b>NoKm</b>	<b>KdJKm</b>	<b>StatusKm</b>
001	KJ01	K
002	KJ01	K
003	KJ01	K
004	KJ01	K
005	KJ01	K
006	KJ02	K
007	KJ02	K
008	KJ02	K
009	KJ02	K
010	KJ02	K

Silahkan lanjutkan data yang lain.....

Contoh data Tamu Masuk

<b>NoReg</b>	<b>TglCekin</b>	<b>KdTamu</b>	<b>NoKm</b>	<b>Status</b>
RT00000001	01/03/2015	KT000001	001	I
RT00000002	01/03/2015	KT000002	005	I
RT00000003	01/03/2016	KT000003	006	I
RT00000004	01/03/2016	KT000004	009	I
RT00000005	01/04/2016	KT000005	010	I

Silahkan lanjutkan data yang lain.....

Contoh data Tamu Keluar

<b>NoReg</b>	<b>TglCekout</b>	<b>Beakamar</b>	<b>Bealain</b>	<b>Pajak</b>
RT00000001	03/03/2015	250000	50000	30000
RT00000002	04/03/2015	375000	125000	50000
RT00000003	02/03/2016	200000	20000	22000
RT00000004	03/03/2016	400000	50000	45000
RT00000005	04/04/2016	600000	100000	70000

Silahkan lanjutkan data yang lain.....

## BAB 3

### QUERY DASAR

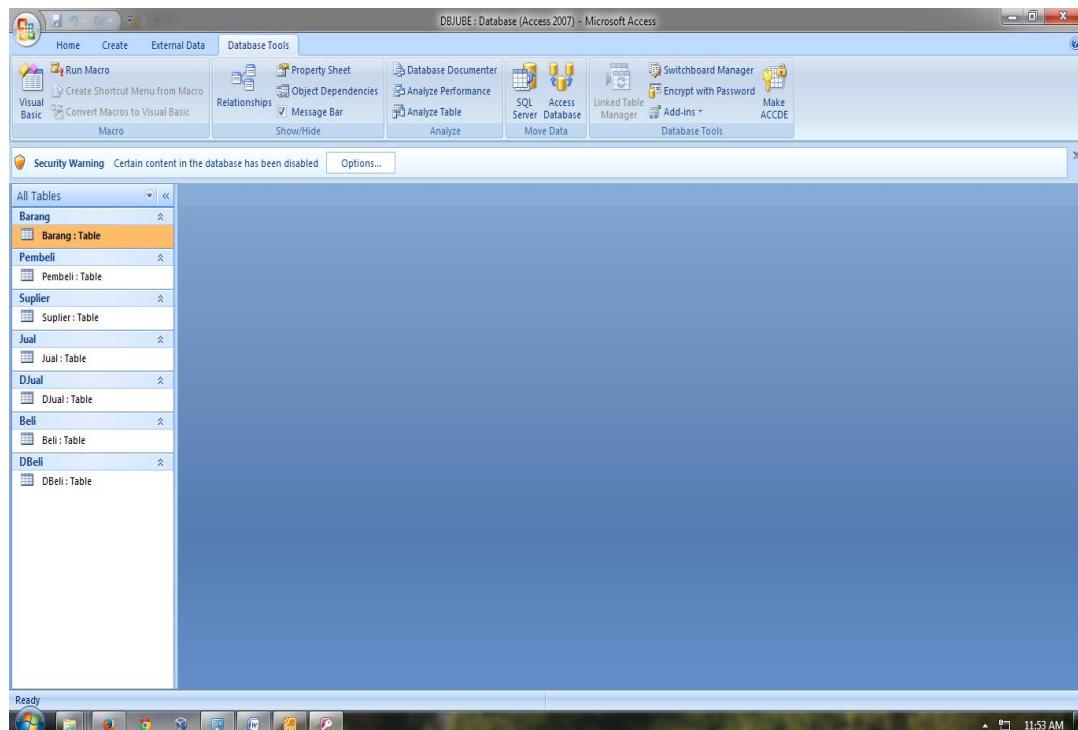
Pada pertemuan ini akan dibahas mengenai dua hal penting yaitu tentang pembuatan **relasi (relationship)** dan tentang **query**. Pembahasan *relationship* akan memberikan kepastian dan kemantapan mengenai pola relasi dari satu tabel ke tabel yang lain. Pembahasan *query* akan memberikan gambaran yang jelas mengenai cara melakukan manipulasi data yang baik dan efesien.

#### 1. Membuat Relasi

**Relationship** merupakan hubungan antara tabel yang satu dengan tabel lainnya, dalam hal ini tabel master dengan tabel transaksi. Kegunaan dari relasi ini adalah jika terjadi perubahan di tabel master maka hasil perubahan tersebut akan mempengaruhi tabel yang berelasi (tabel transaksi), misalnya jika suatu kdebrg pada tabel master dihapus, maka semua data pada tabel transaksi akan ikut terhapus.

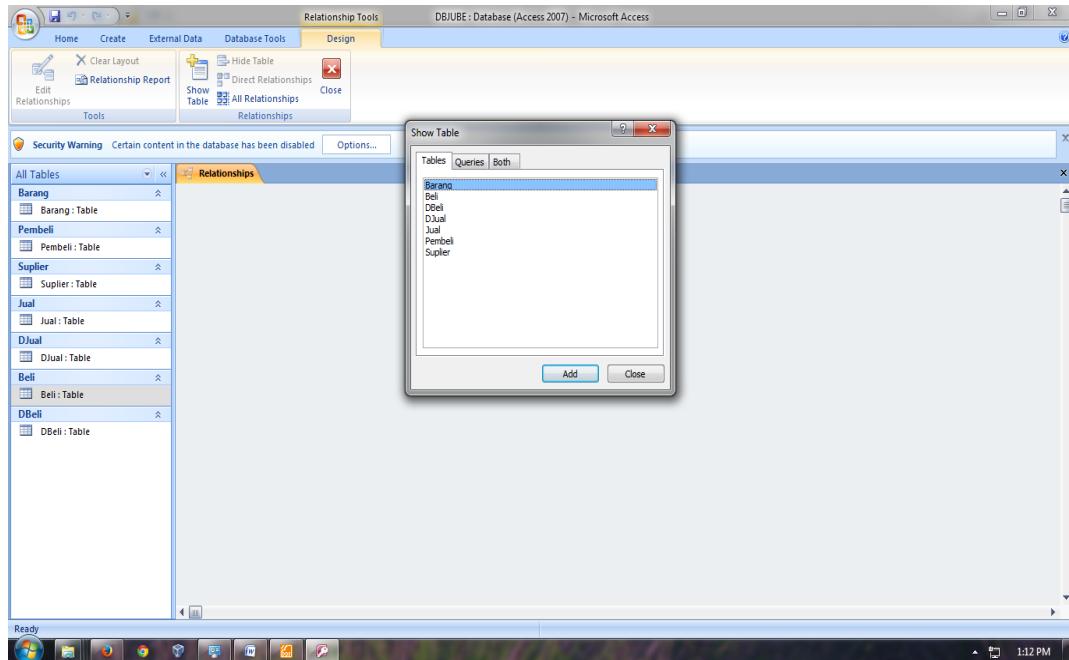
Untuk membuat relasi dari *database* yang sudah ada, lakukan langkah-langkah sebagai berikut:

- a. Buka *database* yang telah dibuat pada pertemuan sebelumnya (buka file **database DBJUBE**)
- b. Pada saat *database* sudah dibuka, klik menu **database tools** (lihat tampilan Gambar 1).



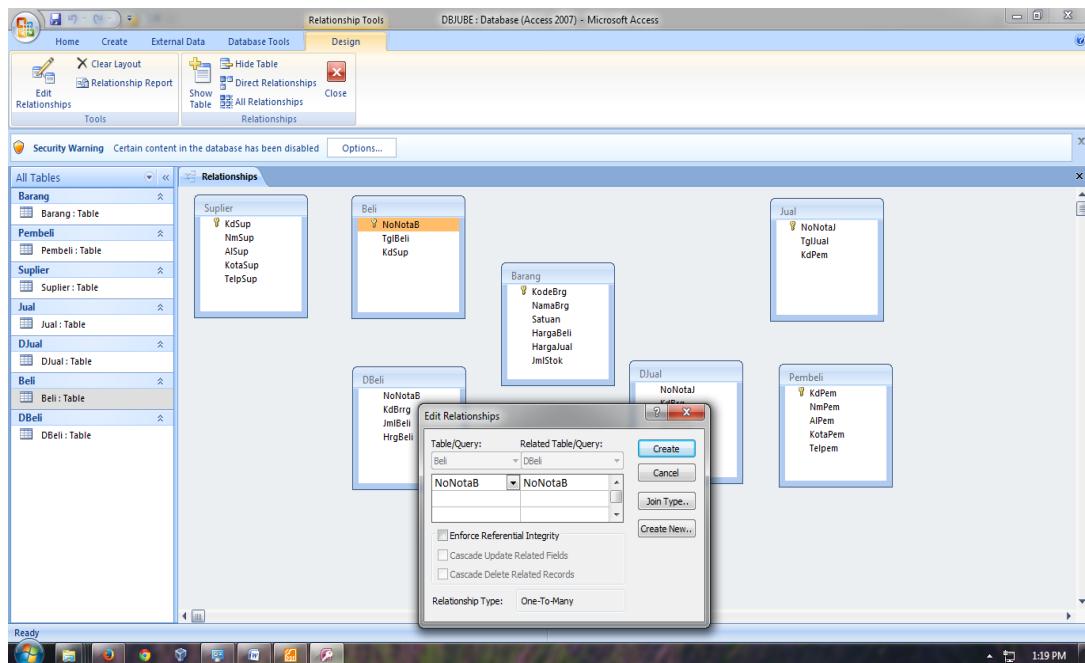
Gambar 1 Tampilan Awal Pembuatan Relationship

- c. Selanjutnya klik tab **relationships**. Jika **relationships** sudah ada sebelumnya, maka akan langsung ditampilkan. Jika belum ada, maka akan disajikan kotak dialog untuk memasukkan tabel seperti terlihat pada Gambar 2.



Gambar 2 Design Relationships

- d. Tambahkan tabel-tabel yang akan dibuatkan relationships nya dengan cara mengklik nama tabel lalu klik tombol **Add**. Lakukan proses tersebut sampai semua tabel ditambahkan.
- e. Setelah semua tabel selesai diambil/ditambahkan, klik tombol **Close**.
- f. Kemudian hubungkan *field* dengan status *primary key* pada tabel master ke *field* yang bersesuaian pada tabel transaksi (**NonotaB** pada tabel Beli dengan **NonotaB** pada tabel DBeli). Pada option relasi pilih semua **check** yang menunjukkan perubahan pada tabel master akan berpengaruh terhadap tabel transaksi (check integritas data, ubah data dan hapus data). Kemudian tekan tombol **Create**. Lihat Gambar 3. Lakukan langkah ke-6 ini untuk menghubungkan tabel-tabel yang lain.

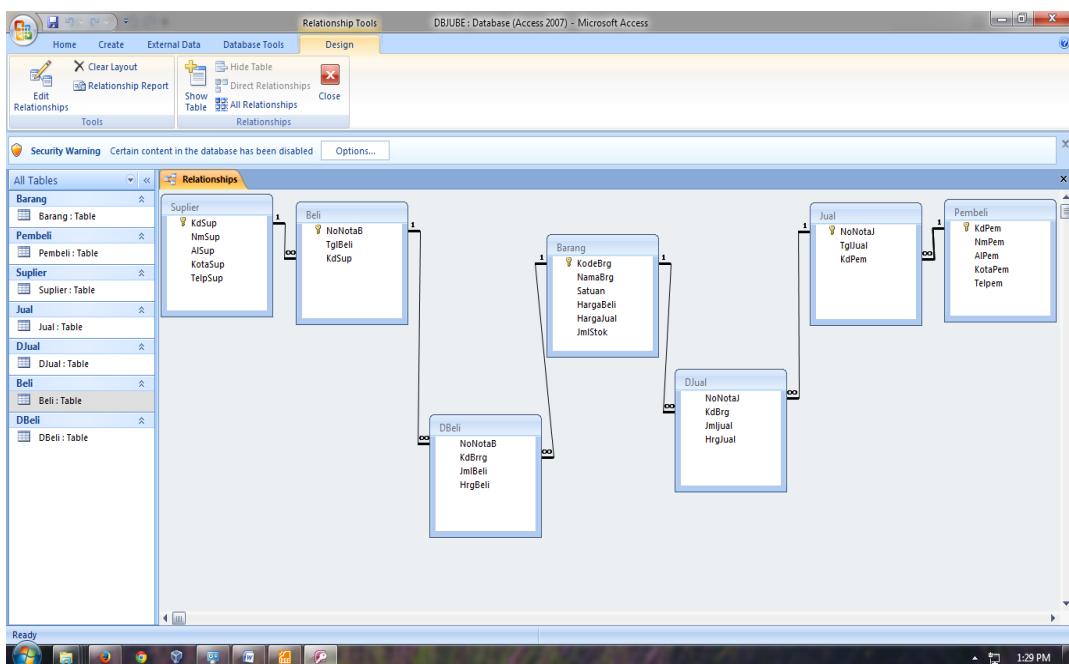


Gambar 3 Proses Pembuatan Relasi

Pada gambar di atas terlihat bahwa pola hubungannya adalah **one to many**, karena terjadi hubungan antara *primary key* (NonotaB pada tabel Beli) dengan *foreign key* (NonotaB pada table DBeli).

Apabila hubungan yang terjadi antara *primary key* dengan *primary key* maka pola hubungannya adalah **one to one**. Apabila hubungan yang terjadi antara *foreign key* dengan *foreign key* maka pola hubungannya adalah **many to many**

- Setelah selesai, hasil **Relationships** dapat dilihat pada Gambar 4.



Gambar 4 Hasil *Relationships*

## 2. Pengantar Query

Query atau juga disebut SQL (*structure query language*) merupakan ‘bahasa’ yang digunakan dalam sistem *database* untuk membuat menyeleksi atau memilih sejumlah data dari semua data yang ada dalam *database*. **Query** merupakan suatu tabel virtual artinya data yang ada pada *query* bisa diberlakukan sebagaimana data yang ada pada sebuah tabel, akan tetapi secara fisik data tersebut tidak tersimpan dalam satu tabel tertentu. *Query* bisa berasal dari satu tabel, bisa juga berasal dari dua tabel bahkan lebih.

*Query* adalah objek *database* yang berfungsi untuk melakukan manipulasi data. Pada *query* kita dapat memasukkan rumus-rumus aritmatika, logika, melakukan penyortiran data berdasarkan kriteria tertentu, mengatur penyortiran yang dilakukan dari form dan lain sebagainya.

## 3. Jenis-Jenis *Query*

Dalam Microsoft Access, ada beberapa jenis *query* yang bisa dibuat sesuai keperluan, yaitu:

### a. **Select Query**

*Select Query* adalah *query* yang paling umum digunakan. *Query* ini mengambil data dari satu tabel atau lebih yang kemudian menampilkan data tersebut dalam bentuk *datasheet*, dimana kita bisa mengolah data tersebut untuk keperluan selanjutnya seperti grouping, sorting, kalkulasi dan lain-lain.

### b. **Crosstab Query**

Anda menggunakan *crosstab query* untuk menghitung dan restrukturisasi data untuk analisis data Anda agar lebih mudah. *Crosstab query* menghitung jumlah, rata-rata, atau yang lain untuk data yang dikelompokkan oleh dua jenis informasi, satu di sisi kiri dari *datasheet* dan satu lagi di bagian atas.

### c. **Make-Tabel Query**

*Query* ini berfungsi untuk membuat tabel baru dengan sumber dari satu atau 2 tabel lebih. *Make-tabel query* sangat membantu untuk membuat tabel yang bisa disimpan di *database* itu sendiri atau *database* lain.

### d. **Update Query**

*Update query* membuat perubahan global kepada sekelompok *record* dalam satu atau lebih tabel. Sebagai contoh, Anda dapat menaikkan harga sebesar 5 persen.

### e. **Append Query**

*Append query* bisa menambahkan sebuah *record* dengan kriteria tertentu secara otomatis pada sebuah tabel, sehingga tidak perlu mengetikkan informasi baru tersebut tapi cukup dengan menjalankan *query* ini.

### f. **Delete Query**

*Query* ini berfungsi untuk menghapus data dengan kriteria tertentu dari sebuah tabel.

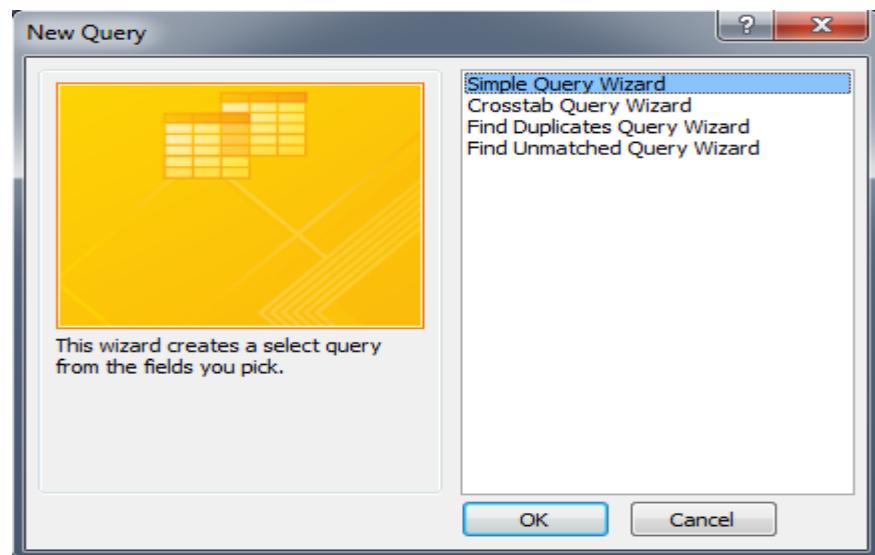
#### 4. Membuat Query

Query dapat dibuat apabila database sudah dibuka. Jika database belum dibuka, silahkan dibuka terlebih dahulu (untuk contoh gunakan database **DBJUBE**).

Untuk membuat query dengan Microsoft Access dapat dilakukan dengan cara sebagai berikut:

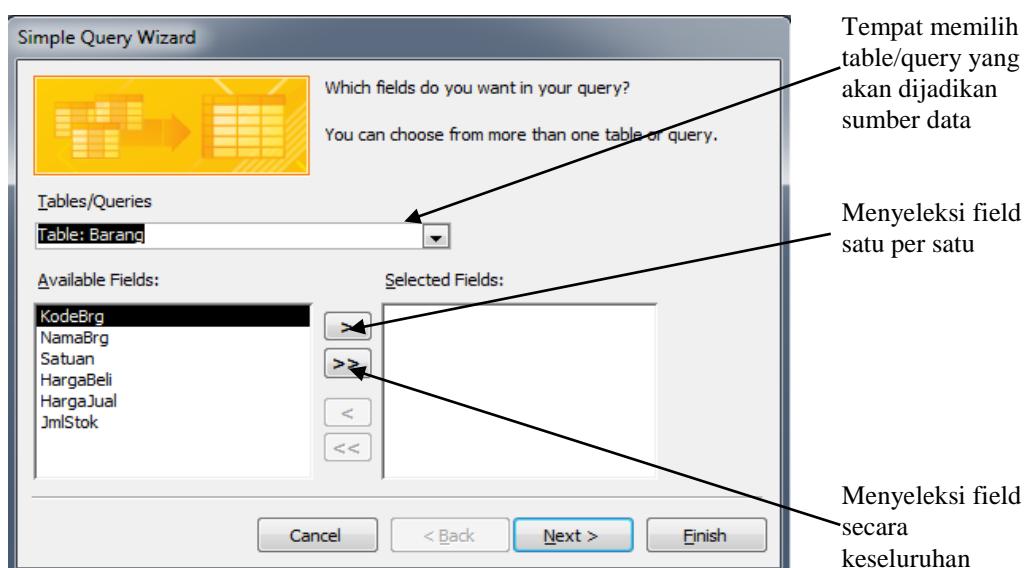
- Membuat query dari satu tabel secara otomatis (*Query Wizard*)

- 1) Klik menu **Create**
- 2) Pilih **Query Wizard**, sehingga akan ditampilkan kotak dialog seperti Gambar 5.



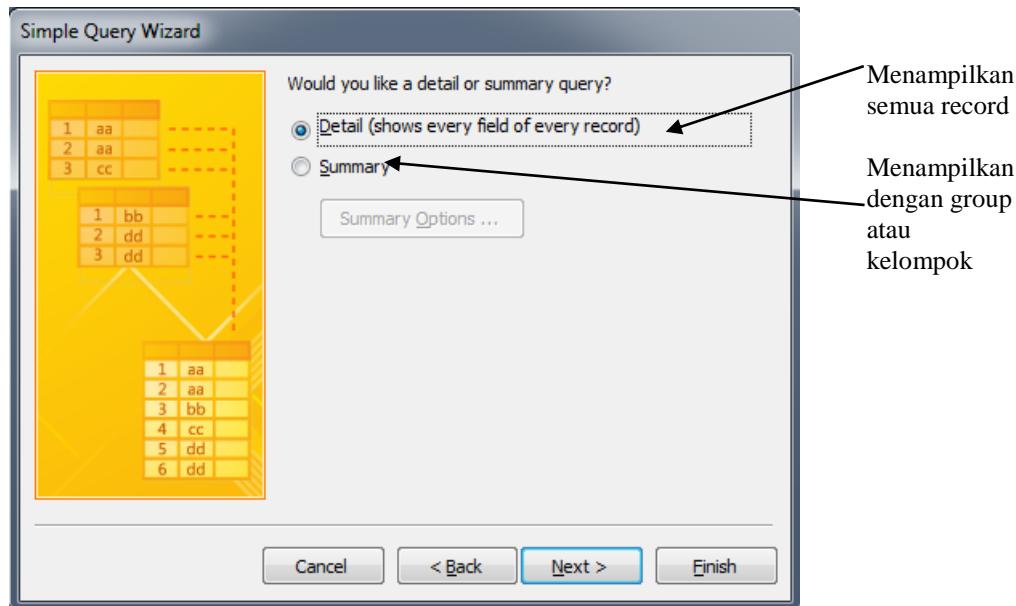
Gambar 5 Kotak Dialog Pembuatan Query Secara Wizard

- 3) Pilih jenis query yang akan dibuat, dalam contoh ini pilihlah **Simple Query Wizard**, dan lanjutkan dengan mengklik tombol **OK**. Selanjutnya akan ditampilkan kotak dialog seperti terlihat pada Gambar 6.



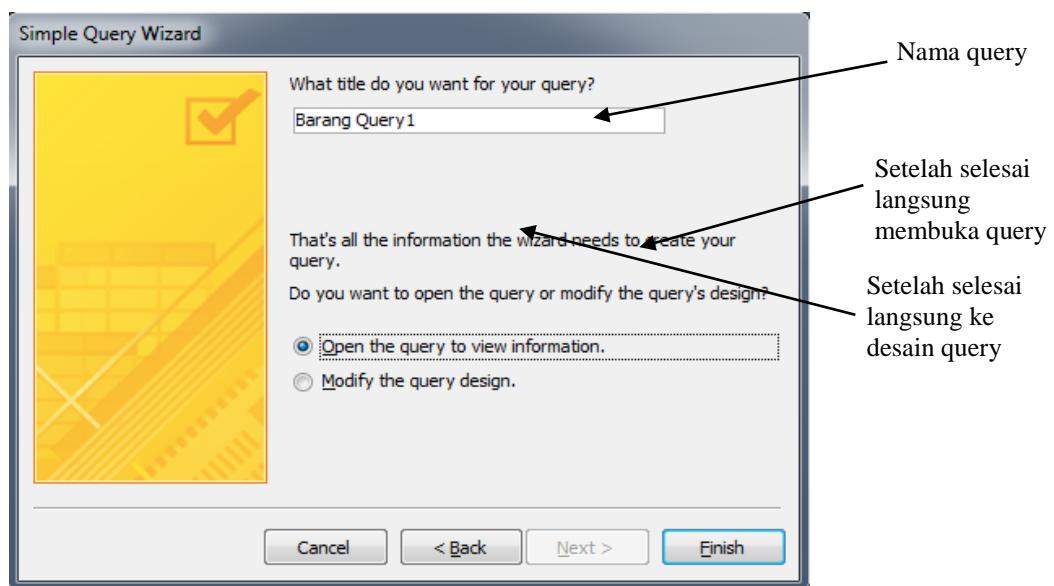
Gambar 6 Kotak Dialog Memilih Tabel dan Field

- 4) Tentukan dan pilih **table** atau **query** yang akan dijadikan sumber data. Selanjutnya pilih *field-field* yang akan ditampilkan (bisa satu per satu atau langsung keseluruhan). Setelah selesai klik tombol **Next** sehingga akan ditampilkan kotak dialog seperti terlihat pada Gambar 7.



Gambar 7 Memilih Model Tampilan Query

- 5) Pilih model tampilan query yang diinginkan (pada contoh ini pilih **Detail** (*shows every field of every record*) dan lanjutkan mengklik tombol **Next**, sampai ditampilkan kotak dialog seperti Gambar 8.



Gambar 8 Akhir Pembuatan Query

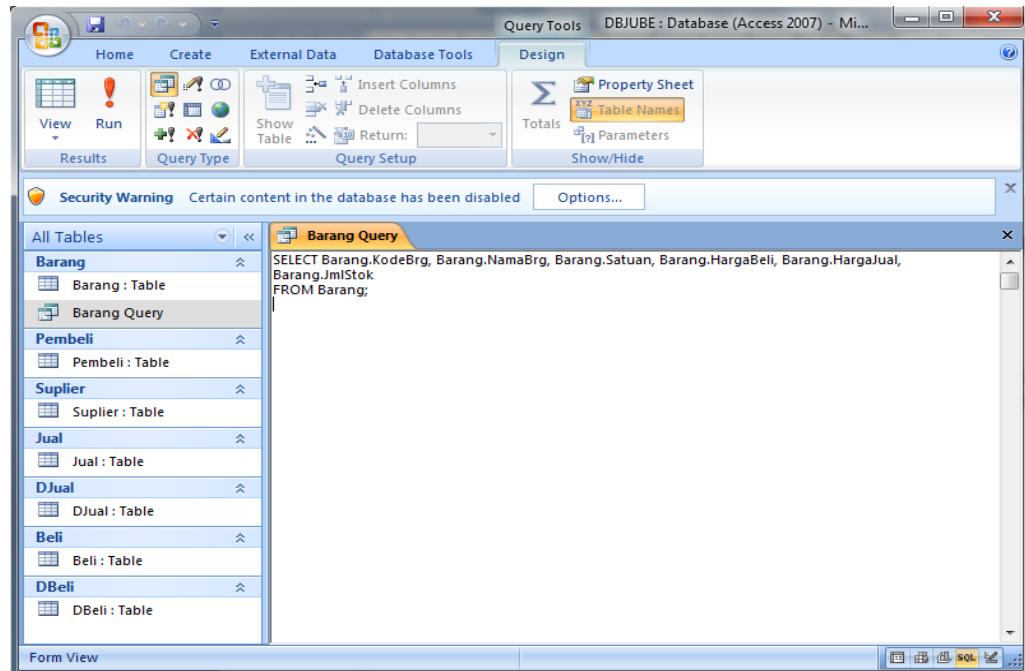
- 6) Ketik nama untuk menyimpan *query*, pilih mode apa yang akan digunakan setelah *query* selesai dibuat. Selanjutnya akhiri dengan mengklik tombol **Finish**, sehingga terlihat hasil seperti Gambar 9.

Gambar 9 Tampilan Hasil Query (*Datasheet View*)

- 7) Untuk mengganti ke bentuk *desain query* dapat dilakukan dengan cara memilih menu **View** lalu pilih **Design View**, tampilan berubah seperti terlihat pada Gambar 10.

Gambar 10 Tampilan Desain Query (*Desain View*)

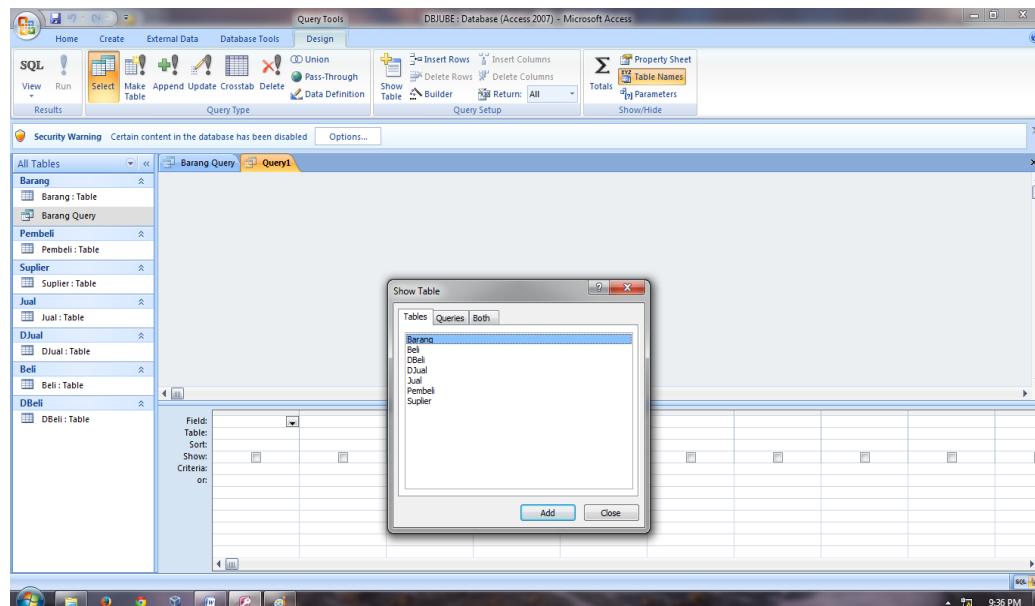
8) Untuk mengganti ke bentuk tampilan SQL dapat dilakukan dengan cara memilih **SQL View**, tampilan berubah seperti terlihat pada Gambar 11.



Gambar 11 Tampilan SQL (SQL View)

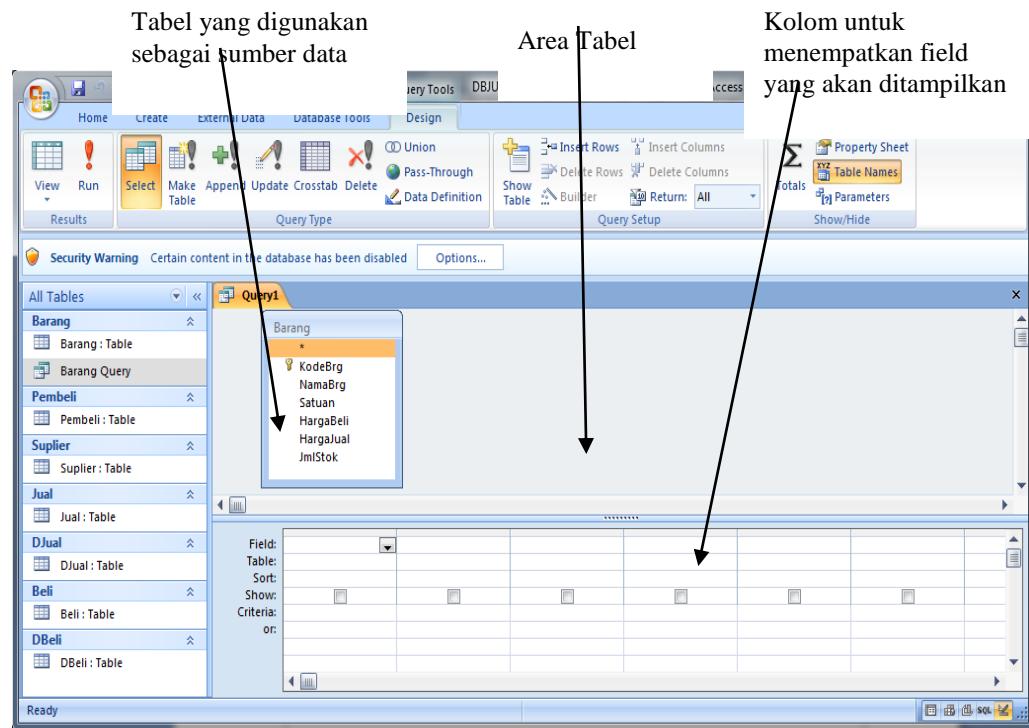
b. Membuat quey dari satu table secara manual (*Query Design*)

- 1) Klik menu **Create**
- 2) Pilih **Query Design**, sehingga akan ditampilkan kotak dialog seperti Gambar 12
- 3) Pilih/klik tabel atau query yang akan digunakan sebagai sumber datanya, selanjutnya klik tombol **Add**.



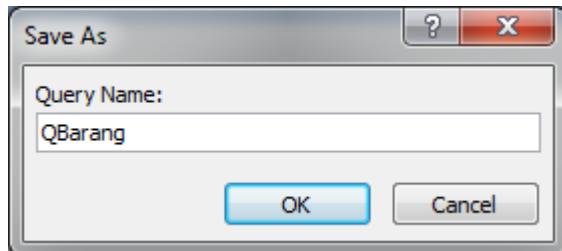
Gambar 12 Tampilan Membuat Query Secara Manual

- 4) Pemilihan sumber data (bisa tabel atau *query*) dari satu tabel atau *query* atau lebih dari satu tabel atau *query*. Untuk contoh ini gunakan satu tabel yaitu tabel **Barang**. Klik tabel Barang kemudian klik tombol Add. Setelah itu klik tombol **Close** sehingga akan ditampilkan proses desain *query* seperti terlihat pada Gambar 13.



Gambar 13 Proses Desain Query Secara Manual

- 5) Silahkan pilih *field-field* yang akan ditampilkan. Cara memilih *field* bisa satu per satu atau langsung semuanya. **Jika cara pertama** yang dipilih maka klik *field* yang diinginkan, lalu drag ke arah kolom dimana *field* tersebut akan diletakkan kemudian lepas. **Jika cara kedua** yang dipilih maka klik *field* yang pertama lalu tekan tombol **Shift** dan klik *field* terakhir, selanjutnya drag ke arah area kolom untuk menampilkan *field-field* tersebut kemudian lepas.  
 6) Jika sudah selesai memilih *field* yang dibutuhkan, simpanlah *query* yang dihasilkan dengan mengklik icon **Save** atau klik tombol **Office Button** lalu pilih **Save** atau bisa juga dengan cara langsung dengan menekan tombol **Ctrl+S**. Setelah melakukan langkah tersebut, maka akan ditampilkan kotak dialog penyimpanan *query* seperti terlihat pada Gambar 14.



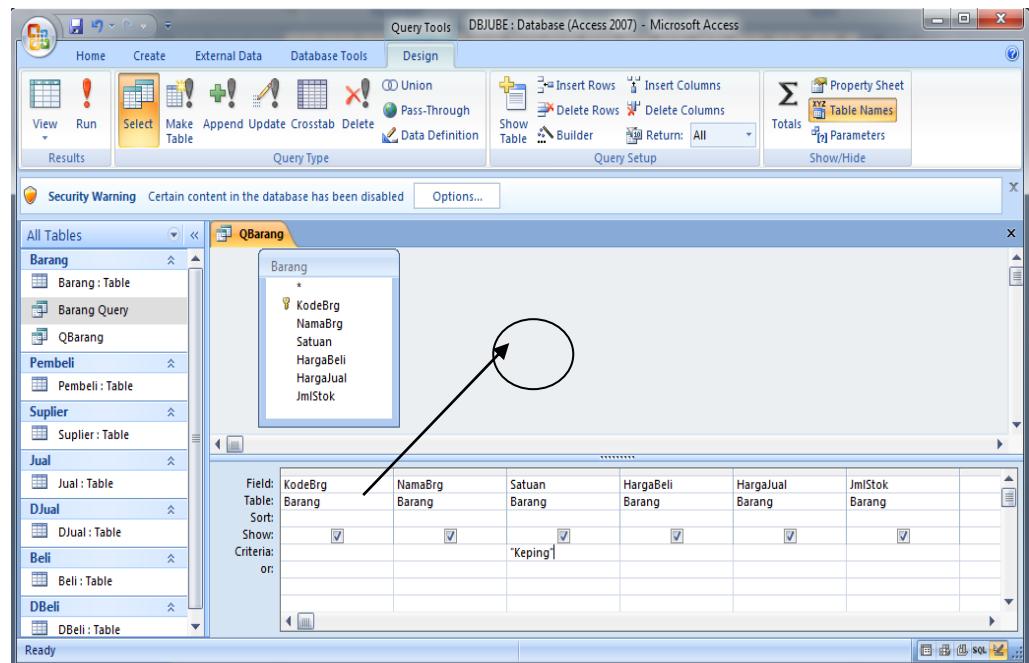
Gambar 14 Kotak Dialog Penyimpanan Query

- 7) Ketik nama *file* yang akan diberikan untuk menyimpan *query*. Untuk contoh ini diberi nama **QBarang**. Hasil akhir query yang dihasilkan dapat dilihat pada Gambar 15.

Gambar 15 Hasil Akhir Query Secara Manual

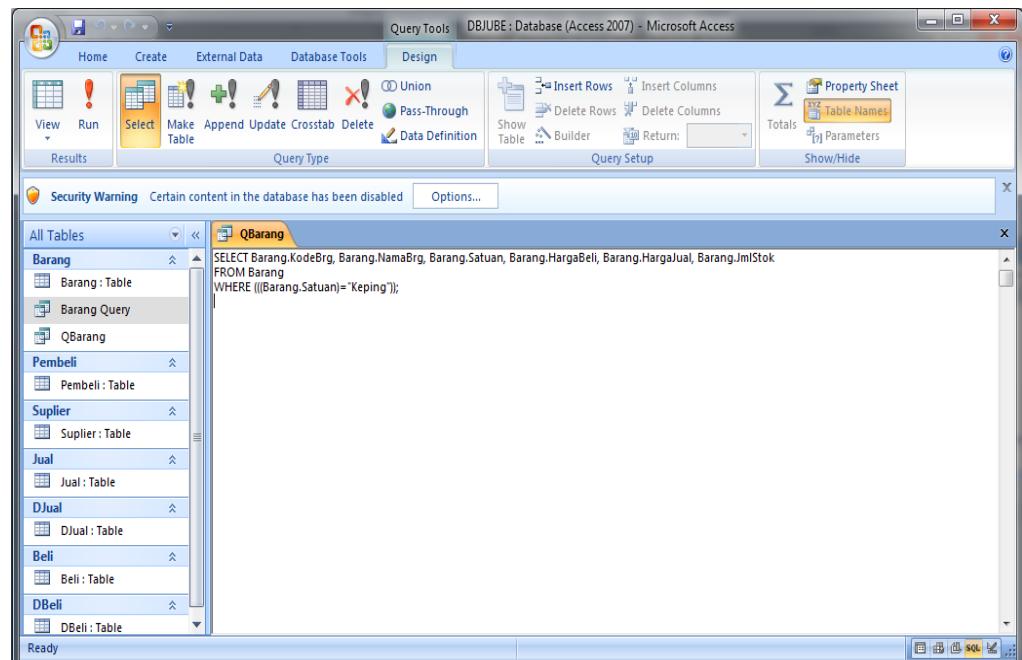
- 8) Jika diinginkan untuk melihat *output* atau hasil *query* gantilah modus ke **Datasheet View** (hasil seperti Gambar 9), sedangkan jika ingin melihat SQL gantilah ke modus **SQL View** (hasil seperti Gambar 11).
- c. Membuat *query* dengan kriteria
- Query* dengan kriteria ini dimaksudkan untuk mendapatkan informasi berdasarkan kondisi-kondisi tertentu. Untuk membuat kriteria dapat dilakukan dengan cara memasukkan kriteria yang diinginkan pada kolom yang akan dibuat kriterianya. Kriteria yang dimasukkan bisa berupa konstanta atau ekspresi dengan menggunakan operator relasional (>, <, =, >=, <=, <>). Kriteria yang dibuat tidak hanya satu, tapi juga bisa lebih (diletakkan pada kolom-kolom yang sesuai. Pada saat membuat kriteria, dapat juga menggunakan *wildcard* (tanda \* atau ?).
- Untuk uji coba, bukalah *query* yang telah dihasilkan sebelumnya (QBarang).

### 1) Membuat kriteria pada kolom satuan



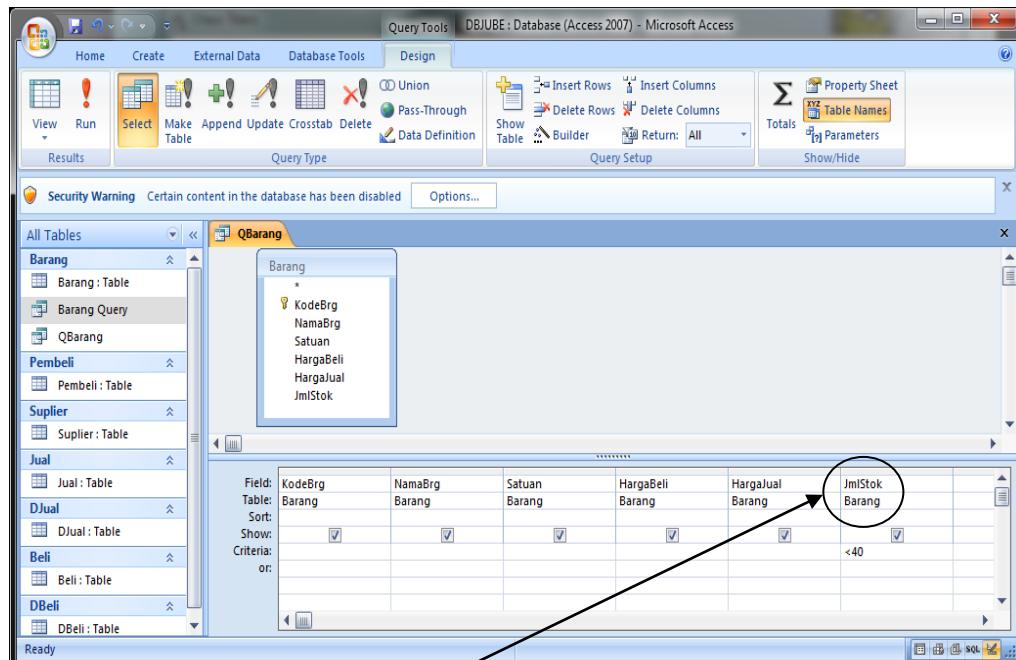
Gambar 16 Query Dengan Kriteria Pada Kolom Satuan

- Ketikkan kriteria Keping pada kolom satuan
- Tampilkan hasil (ganti ke mode *Datasheet View*). Data yang ditampilkan adalah semua barang yang memiliki satuan Keping
- Lihat SQL (ganti ke mode *SQL View*). Bentuk penulisan SQL ada penambahan ***klausa Where***



Gambar 17 SQL Dengan Kriteria Satuan

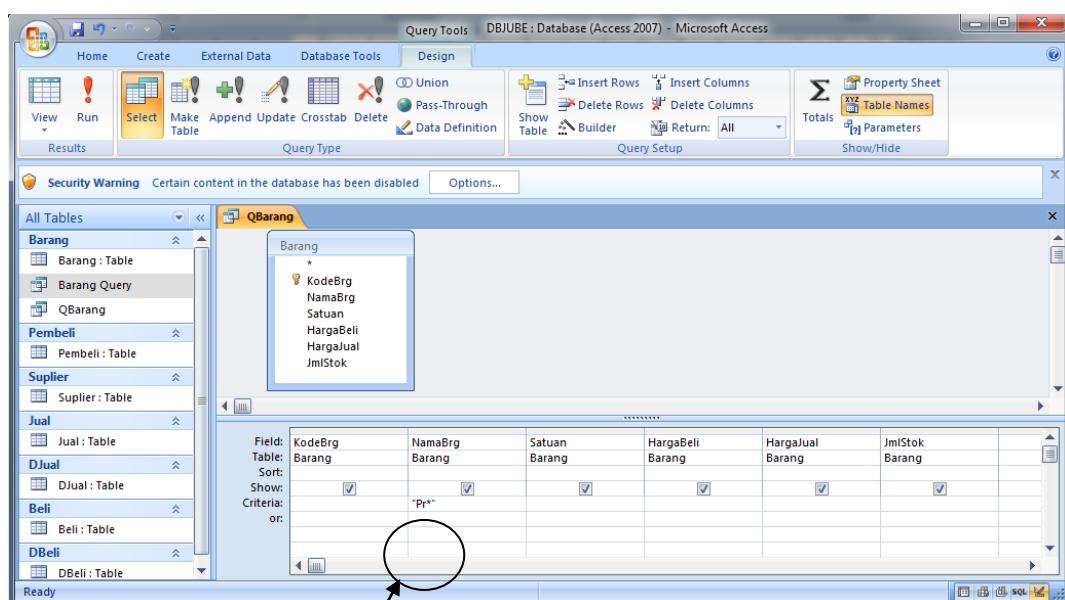
## 2) Membuat kriteria pada kolom JmlStok



Gambar 18 Query Dengan Kriteria Pada Kolom JmlStok

- Ketikkan kriteria <40 pada kolom JmlStok
- Tampilkan hasil (ganti ke mode *Datasheet View*). Data yang ditampilkan adalah semua barang yang memiliki jumlah stok di bawah 40.
- Lihat SQL (ganti ke mode *SQL View*). Bentuk penulisan SQL ada penambahan **klausa Where**.

## 3) Membuat kriteria pada kolom NamaBrg menggunakan wildcard



Gambar 19 Query Dengan Kriteria Pada Kolom NamaBrg

- Ketikkan kriteria Pr\* pada kolom NamaBrg
- Tampilkan hasil (ganti ke mode *Datasheet View*). Data yang ditampilkan adalah semua barang dengan huruf awal adalah Pr
- Lihat SQL (ganti ke mode *SQL View*). Bentuk penulisan SQL ada penambahan *klausa Where* dengan ***Like***

The screenshot shows the Microsoft Access Query Designer window. The ribbon at the top has 'Query Tools' selected. The 'Design' tab is active. In the center pane, there is a query named 'QBarang' with the following SQL code:

```
SELECT Barang.KodeBrg, Barang.NamaBrg, Barang.Satuan, Barang.HargaBel, Barang.HargaJual, Barang.JmlStok
FROM Barang
WHERE (((Barang.NamaBrg) Like "Pr*"));
```

The left pane shows a list of tables: Barang, Pembeli, Suplier, Jual, DJual, Beli, DBeli. The 'Barang' table is selected.

Gambar 20 SQL Dengan Kriteria Pada Kolom NamaBrgn

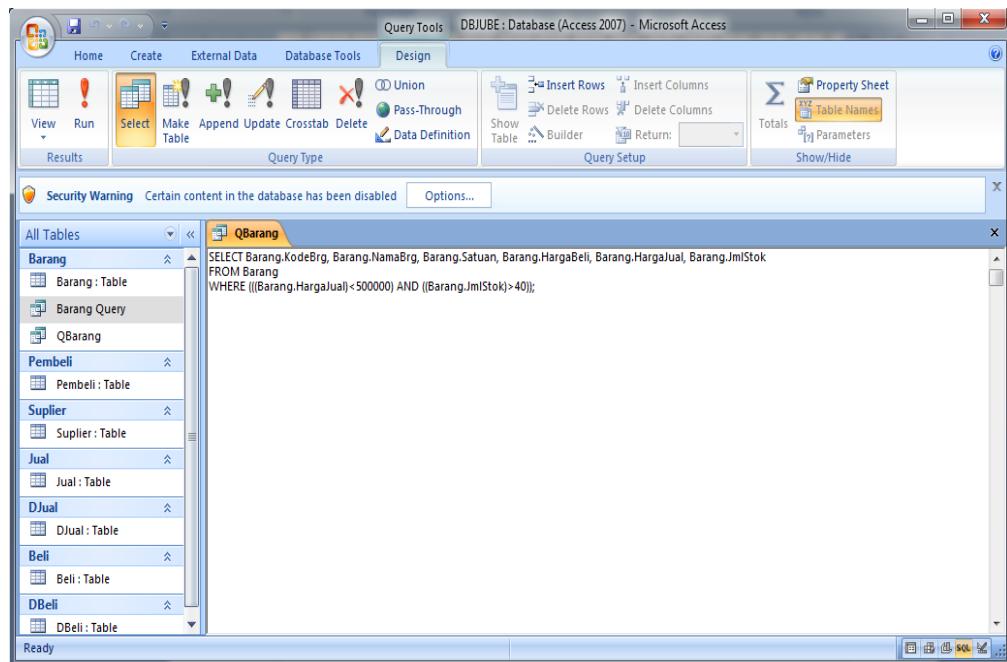
#### 4) Membuat kriteria pada kolom HargaJual dan JmlStok

The screenshot shows the Microsoft Access Query Designer window. The 'Design' tab is active. The query 'QBarang' is displayed with the following table structure and criteria:

Field:	KodeBrg	NamaBrg	Satuan	HargaBel	HargaJual	JmlStok
Table:	Barang	Barang	Barang	Barang	Barang	Barang
Show:	<input checked="" type="checkbox"/>					
Criteria:				<500000		>40
or:						

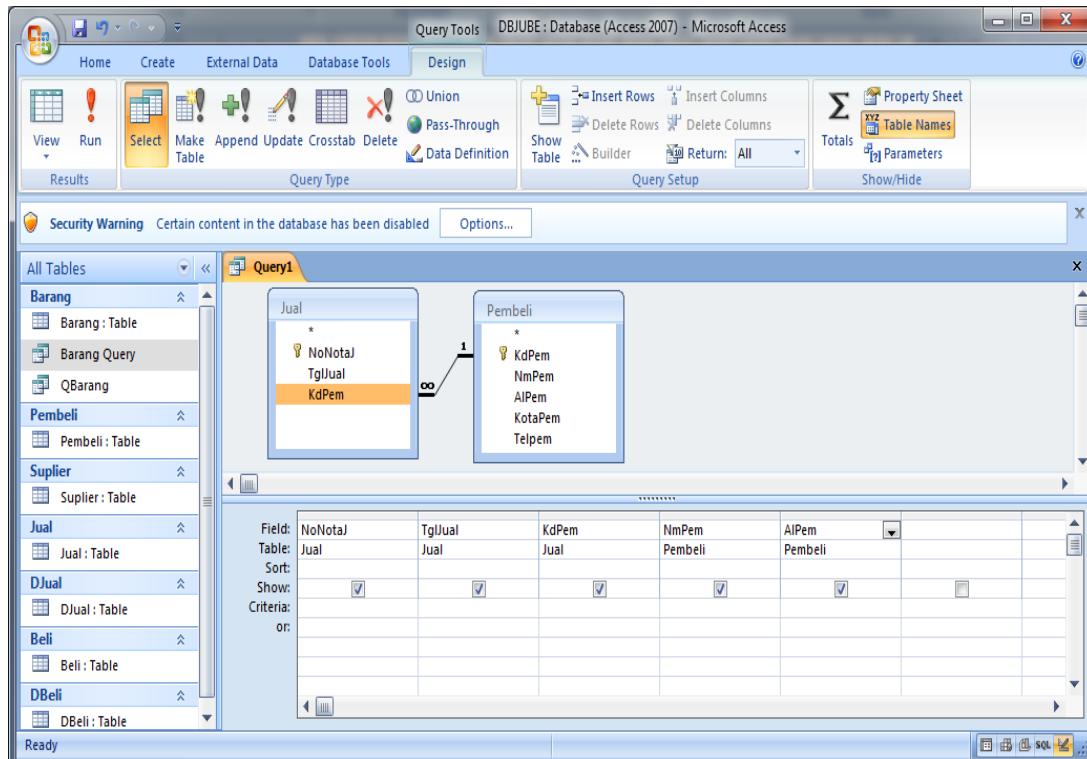
Gambar 21 Query Dengan Kriteria di Kolom HargaJual dan JmlStok

- a) Ketikkan kriteria <500000 pada kolom HargaJual dan ketikkan criteria >40 pada kolom JmlStok.
- b) Tampilkan hasil (ganti ke mode Datasheet View). Data yang ditampilkan adalah semua barang dengan harga jual di bawah 500000 dan jumlah stok di atas 40.
- c) Lihat SQL (ganti ke mode SQL View). Bentuk penulisan SQL ada penambahan *klausa Where* beserta dengan operator **AND**



Gambar 22 SQL Kriteria di Kolom HargaJual dan JmlStok

- d. Membuat *query* dari beberapa tabel secara manual (*query design*)  
Untuk contoh berikut akan digunakan dua tabel yaitu tabel Jual dan tabel Pembeli.
- 1) Klik menu **Create**
  - 2) Pilih **Query Design**, sehingga akan ditampilkan kotak dialog untuk memilih tabel yang digunakan sebagai sumber data.
  - 3) Pilih/klik tabel Jual dilanjutkan dengan mengklik tombol **Add**.
  - 4) Pilih/klik tabel Pembeli dilanjutkan dengan mengklik tombol **Add**.
  - 5) Klik tombol **Close**.
  - 6) Ambil semua *field* dari tabel Jual dan letakkan pada bagian kolom query (blok dan drag ke area kolom query).
  - 7) Ambil field NmPem dan field AlPem dari tabel Pembeli (drag satu persatu ke area kolom query)
  - 8) Simpan *query* yang dihasilkan dengan nama QJual.
  - 9) Setelah selesai mengambil *field* yang dibutuhkan, hasil desain *query* dapat dilihat seperti Gambar 23.

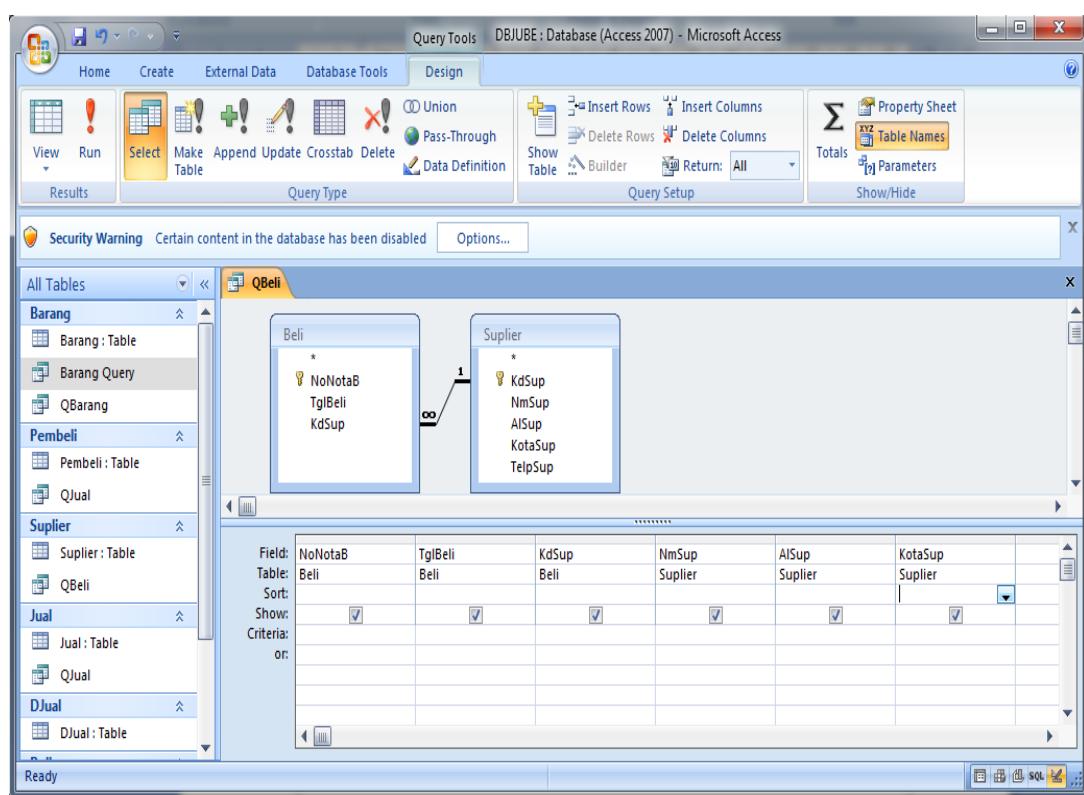


Gambar 23 Desain Query Gabungan Tabel Jual dan Tabel Pembeli

- e. Membuat *query* dari beberapa tabel secara manual (*query design*)

Untuk contoh berikut akan digunakan dua tabel yaitu tabel Beli dan tabel Suplier.

- 1) Klik menu **Create**
- 2) Pilih **Query Design**, sehingga akan ditampilkan kotak dialog untuk memilih tabel yang digunakan sebagai sumber data.
- 3) Pilih/klik tabel Beli dilanjutkan dengan mengklik tombol **Add**.
- 4) Pilih/klik tabel Suplier dilanjutkan dengan mengklik tombol **Add**.
- 5) Klik tombol **Close**.
- 6) Ambil semua *field* dari tabel Beli dan letakkan pada bagian kolom query (blok dan drag ke area kolom query).
- 7) Ambil field NmSup, field AlPem, dan field KotaSup dari tabel Suplier (drag satu persatu ke area kolom query)
- 8) Simpan *query* yang dihasilkan dengan nama QBeli.
- 9) Setelah selesai mengambil *field* yang dibutuhkan, hasil *desain query* dapat dilihat seperti Gambar 24.



Gambar 24 Desain Query Gabungan Tabel Beli dan Tabel Suplier

## 5. Soal Latihan

- Bukalah database **DBHOTEL** yang telah dibuat pada pertemuan sebelumnya.
- Buatlah Query dari tabel Tamu, dimana field yang digunakan adalah semuanya. Simpan query dengan nama file **QTamu**
  - Lihat hasilnya (*Datasheet View*)
  - Lihat SQL (*SQL View*), dipahami dan dicatat.

Tambahan:

Coba modifikasi query yang telah dihasilkan dengan membuat kriteria untuk menampilkan datanya.

- Buat kriteria pada kolom Kwg dengan isian WNI
- Buat kriteria pada kolom Pekerjaan dengan isian Wiraswasta
- Buat kriteria pada kolom Kwg dan kolom Pekerjaan
- Buat kriteria pada kolom NmTamu dengan isian A\*

- c. Buatlah Query dengan menggabung tabel **Kamar** dengan tabel **JKamar**.  
**Query** disimpan dengan nama file **QKamar**.
  - 1) Lihat hasilnya (*Datasheet View*)
  - 2) Lihat SQL (*SQL View*), dipahami dan dicatat.
- d. Buatlah Query dengan menggabungkan tabel **CekIn** dengan tabel **Tamu** dan tabel **Kamar** serta **JKamar**. Query disimpan dengan nama file **QCekIn**.
  - 1) Lihat hasilnya (*Datasheet View*)
  - 2) Lihat SQL (*SQL View*), dipahami dan dicatat.
- e. Buatlah Query dengan menggabungkan tabel **CekOut** dengan tabel **CekIn**, **Tamu**, **Kamar** serta **JKamar**. Query disimpan dengan nama file **QCekOut**.
  - 1) Lihat hasilnya (*Datasheet View*)
  - 2) Lihat SQL (*SQL View*), dipahami dan dicatat.

## BAB 4

### QUERY LANJUT

Pada pertemuan ini merupakan lanjutan pertemuan sebelumnya yaitu lebih mendalam mengenai query. Bagian awal akan dicobakan untuk membuat query dengan banyak tabel, dilanjutkan dengan query dengan menggunakan ekspresi atau rumus atau fungsi, dan terakhir adalah *query crosstab*.

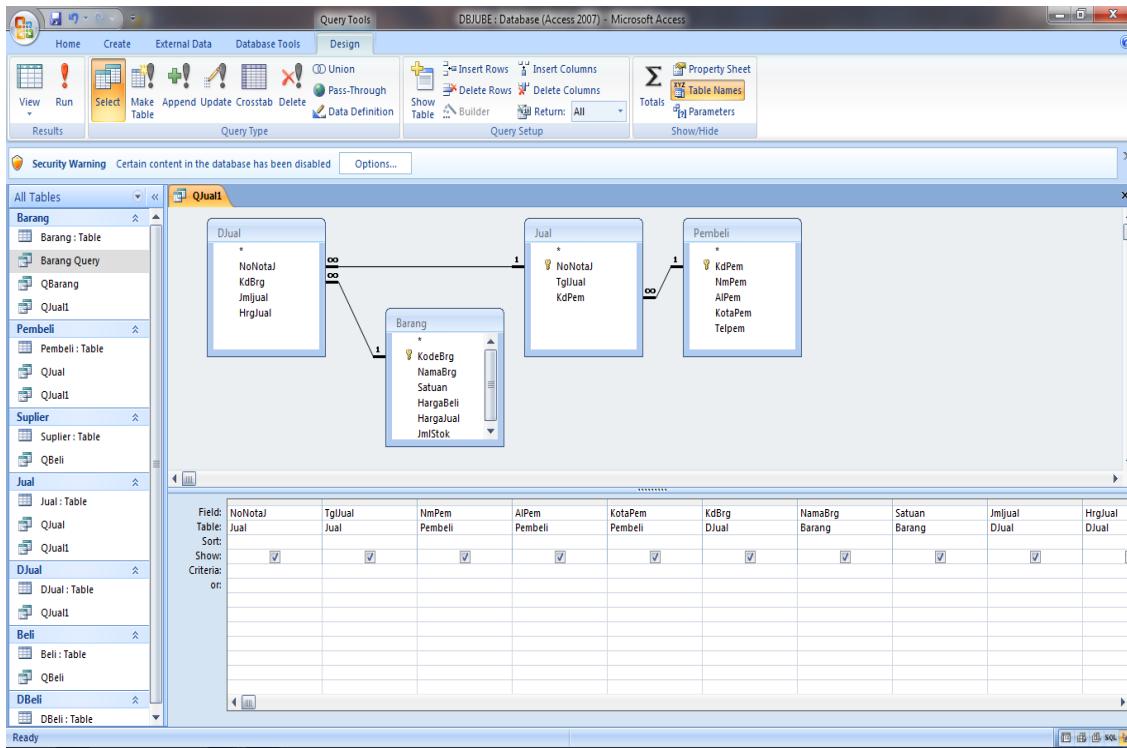
#### 1. Membuat Query Dengan Banyak Tabel

Pada pertemuan sebelumnya telah dibuat query dari dua buah tabel. Pada pertemuan ini akan dicobakan query dari lebih dari dua tabel.

Untuk membuat query maka buka *database* yang telah dibuat pada pertemuan sebelumnya (buka file database **DBJUBE**)

a. Query dari transaksi penjualan

- 1) Klik menu **Create**
- 2) Pilih **Query Design**, sehingga akan ditampilkan kotak dialog untuk memilih tabel yang digunakan sebagai sumber data.
- 3) Pilih/klik tabel DJual dilanjutkan dengan mengklik tombol **Add**.
- 4) Pilih/klik tabel Barang dilanjutkan dengan mengklik tombol **Add**.
- 5) Pilih/klik tabel Jual dilanjutkan dengan mengklik tombol **Add**
- 6) Pilih/klik tabel Pembeli dilanjutkan dengan mengklik tombol **Add**
- 7) Klik tombol **Close**. Coba perhatikan apakah semua table sudah saling berelasi/terhubung? Jika ada yang belum terhubung berarti ada yang keliru.
- 8) Ambil *field NonotaJ* dan **TglJual** dari table Jual dan letakkan pada bagian kolom *query* (blok dan drag ke area kolom query).
- 9) Ambil *field NmPem*, **AIPem** dan **KotaPem** dari tabel Pembeli (drag dan letakkan dikolom setelah *field TglJual*)
- 10) Ambil *field KdBrg* dari tabel DJual (drag dan letakkan dikolom setelah *field KotaPem*)
- 11) Ambil *field NamaBrg* dan *field Satuan* dari tabel Barang (drag dan letakkan dikolom setelah *field KdBrg*)
- 12) Ambil *field JmlJual* dari tabel DJual (drag dan letakkan dikolom setelah *field Satuan*)
- 13) Ambil *field HrgJual* dari tabel DJual (drag dan letakkan dikolom setelah *field JmlJual*)
- 14) Simpan query yang dihasilkan dengan nama **QJual1**.
- 15) Setelah selesai mengambil *field* yang dibutuhkan, hasil *desain query* dapat dilihat seperti Gambar 1.



Gambar 1 Desain Query dari table traksaksi Penjualan

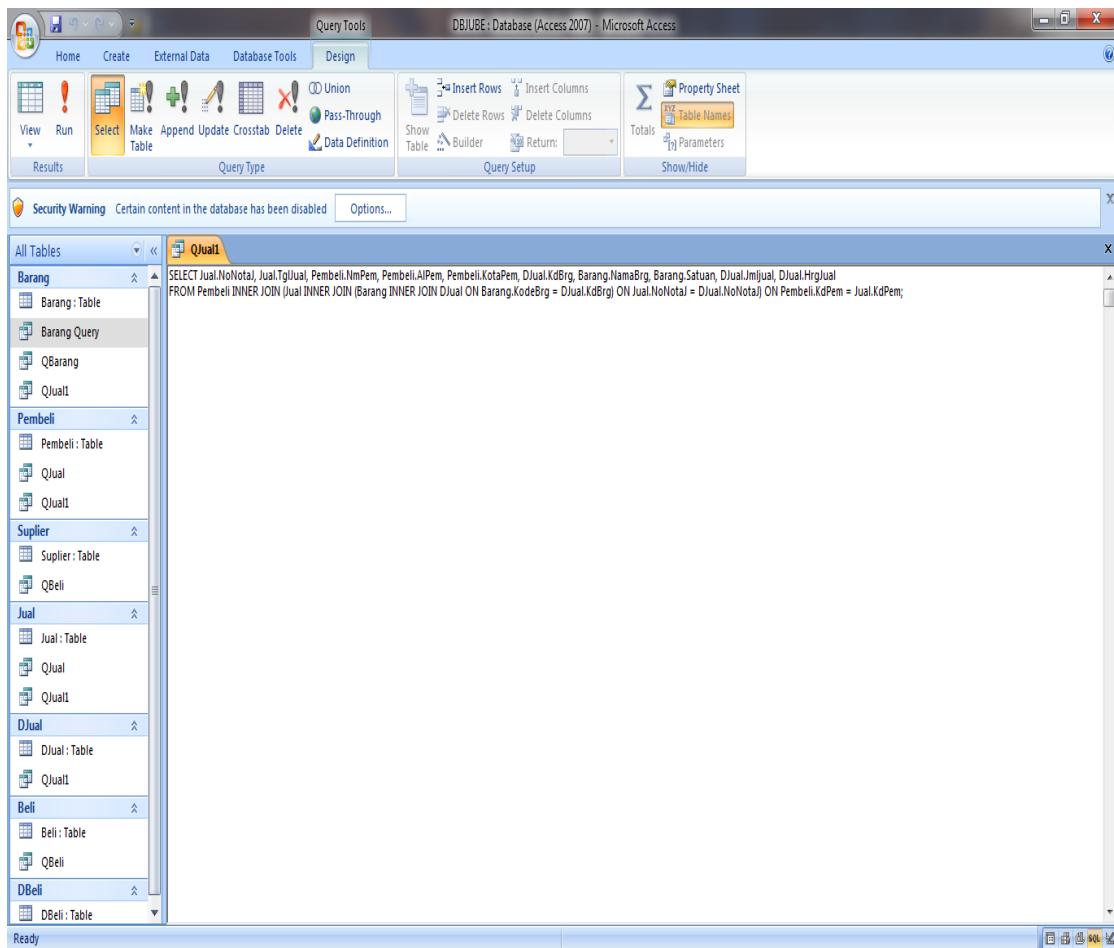
Untuk melihat hasil dari *query* yang telah dibuat gantilah modus ke **Datasheet View** dan hasilnya seperti Gambar 2.

The screenshot shows the Microsoft Access 2007 interface in 'Datasheet' view. The 'Tables' pane on the left lists tables: Barang, Jual, and Pembeli. The main area displays the query results with the following data:

NoNotaJ	TglJual	NmPem	AlPem	KotaPem	KdBrg	NamaBrg	Satuan	JmlJual	HrgJual
NJ0000001	1/6/2014 PT. Adi Jaya	Jl. Slamet Riyadi No. 65	Solo	B-00003	Processor Intel	Unit	2	475000	
NJ0000001	1/6/2014 PT. Adi Jaya	Jl. Slamet Riyadi No. 65	Solo	B-00005	RAM 128 MB	Keping	4	60000	
NJ0000001	1/6/2014 PT. Adi Jaya	Jl. Slamet Riyadi No. 65	Solo	B-00006	RAM 256 MB	Keping	2	90000	
NJ0000002	2/6/2014 Mr Hermantio	Jl. Beo No. 78	Yogyakarta	B-00001	Monitor	Unit	5	675000	
NJ0000003	3/12/2014 Ny Suci Darwanti	Jl. Pancasila No. 87	Semarang	B-00002	Printer	Unit	4	700000	
NJ0000003	3/12/2014 Ny Suci Darwanti	Jl. Pancasila No. 87	Semarang	B-00003	Processor Intel	Unit	3	475000	
NJ0000004	12/12/2014 Ny Titik Wijaya	Jl. Gejayan No. 87	Yogyakarta	B-00006	RAM 256 MB	Keping	3	90000	
NJ0000004	12/12/2014 Ny Titik Wijaya	Jl. Gejayan No. 87	Yogyakarta	B-00007	RAM 512 MB	Keping	4	115000	
NJ0000005	1/2/2015 Mr Burhanudin	Jl. Adisucipto No. 42	Yogyakarta	B-00002	Printer	Unit	6	700000	
NJ0000005	1/2/2015 Mr Burhanudin	Jl. Adisucipto No. 42	Yogyakarta	B-00004	Processor AMD	Unit	5	380000	

Gambar 2 View Query dari tabel traksaksi Penjualan

Untuk melihat *Structure Query Language* (SQL) gantilah ke modus SQL View dan hasilnya seperti Gambar 3.

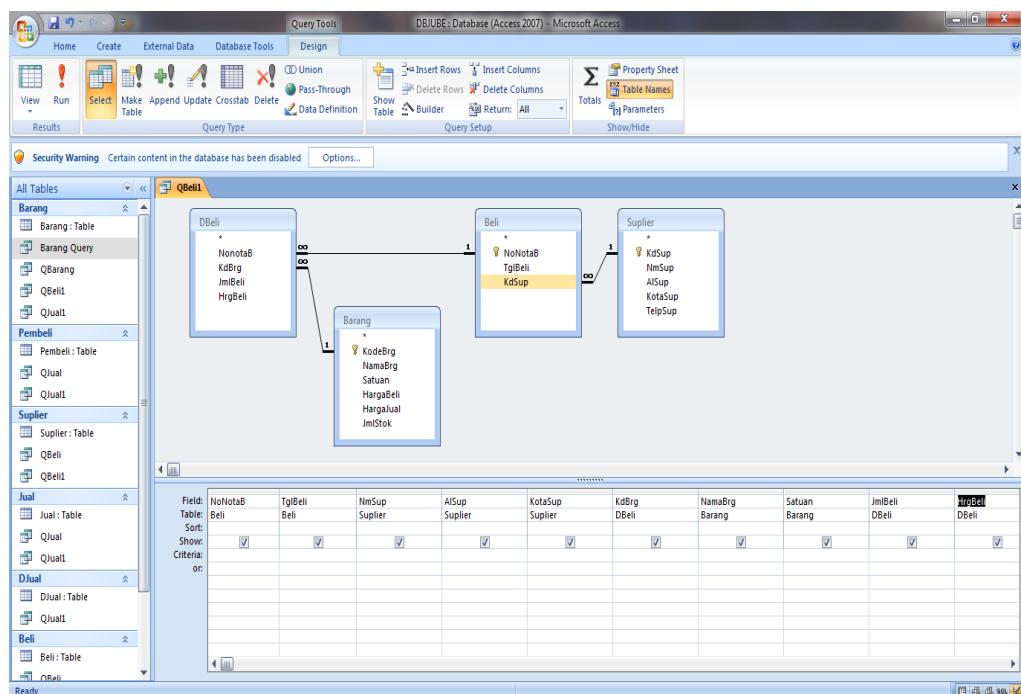


Gambar 3 SQL View Query dari table traksaksi Penjualan

Untuk lebih memahami berbagai cara untuk menampilkan data, maka lakukanlah percobaan dengan menggunakan criteria. Misalnya dibuat criteria pada kolom KotaPem dengan mengetik “Yogya”  
Gantilah ke modus Desain View, kemudian tambahkan criteria dari query yang akan ditampilkan. Silahkan lihat hasilnya seperti apa dan silahkan catat SQL yang dihasilkan seperti apa.  
Kriteria yang diberikan bisa saja lebih dari satu, misalnya dibuat kriteria pada kolom KotaPem dan Satuan  
Cobalah anda mempraktekkan dengan berbagai criteria yang lain sehingga dapat tambahan pengetahuan

b. Query dari transaksi pembelian

- 1) Klik menu **Create**
- 2) Pilih **Query Design**, sehingga akan ditampilkan kotak dialog untuk memilih tabel yang digunakan sebagai sumber data.
- 3) Pilih/klik tabel DBeli dilanjutkan dengan mengklik tombol **Add**.
- 4) Pilih/klik tabel Barang dilanjutkan dengan mengklik tombol **Add**.
- 5) Pilih/klik tabel Beli dilanjutkan dengan mengklik tombol **Add**
- 6) Pilih/klik tabel Suplier dilanjutkan dengan mengklik tombol **Add**
- 7) Klik tombol **Close**. Coba perhatikan apakah semua tabel sudah saling berelasi/terhubung? Jika ada yang belum terhubung berarti ada yang keliru.
- 8) Ambil *field* **NoNotaB** dan **TglBeli** dari tabel Beli dan letakkan pada bagian kolom query (blok dan drag ke area kolom query).
- 9) Ambil *field* **NmSup**, **AlSup** dan **KotaSup** dari tabel Suplier (drag dan letakkan dikolom setelah *field* TglJual)
- 10) Ambil *field* **KdBrg** dari tabel DJual (drag dan letakkan dikolom setelah *field* KotaSup)
- 11) Ambil *field* **NamaBrg** dan *field* **Satuan** dari tabel Barang (drag dan letakkan dikolom setelah *field* KdBrg)
- 12) Ambil *field* **JmlBeli** dari tabel DBeli (drag dan letakkan dikolom setelah *field* Satuan)
- 13) Ambil *field* **HrgBeli** dari tabel DBeli (drag dan letakkan dikolom setelah *field* JmlBeli)
- 14) Simpan query yang dihasilkan dengan nama **QBeli1**.
- 15) Setelah selesai mengambil *field* yang dibutuhkan, hasil desain query dapat dilihat seperti Gambar 4.



Gambar 4 Desain Query dari table traksaksi Pembelian

Untuk melihat hasil dari *query* yang telah dibuat gantilah modus ke **Datasheet View** dan hasilnya seperti Gambar 5.

Gambar 5 View Query dari tabel traksaksi Pembelian

Untuk melihat *Structure Query Language* (SQL) gantilah ke modus SQL View dan hasilnya seperti Gambar 6

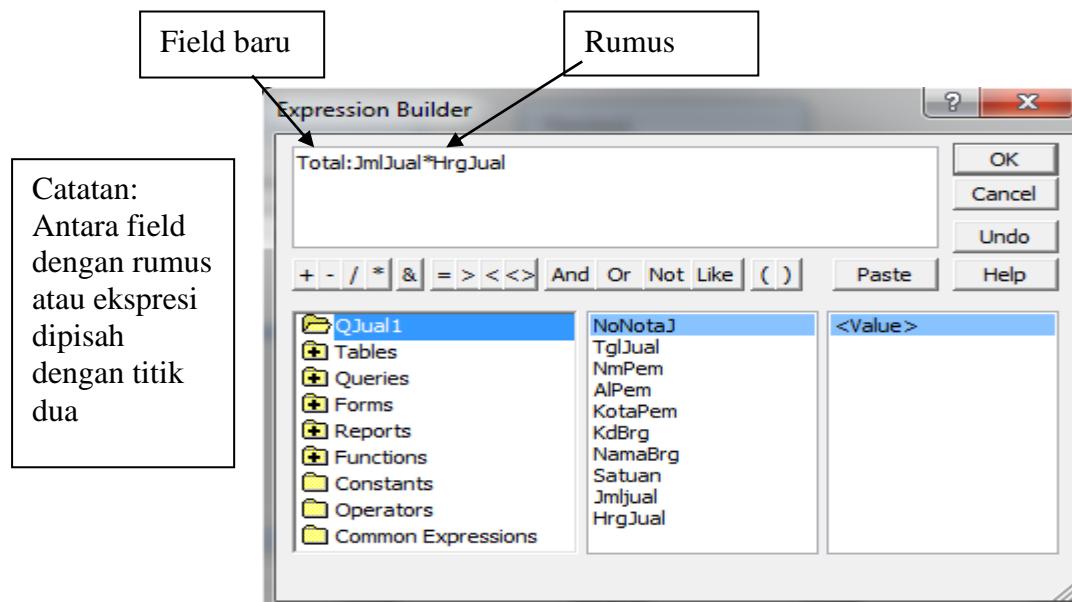
Gambar 6 SQL View Query dari table traksaksi Pembelian

Untuk lebih memahami berbagai cara untuk menampilkan data, maka lakukanlah percobaan dengan menggunakan criteria. Misalnya dibuat criteria pada kolom KotaSup dengan mengetik "Jakarta" Gantilah ke modus Desain View, kemudian tambahkan criteria dari query yang akan ditampilkan. Silahkan lihat hasilnya seperti apa dan silahkan catat SQL yang dihasilkan seperti apa. Kriteria yang diberikan bisa saja lebih dari satu, misalnya dibuat kriteria pada kolom KotaSup dan Satuan Cobalah anda mempraktekkan dengan berbagai criteria yang lain sehingga dapat tambahan pengetahuan

## 2. Membuat Query Dengan Ekspresi/Rumus/Fungsi

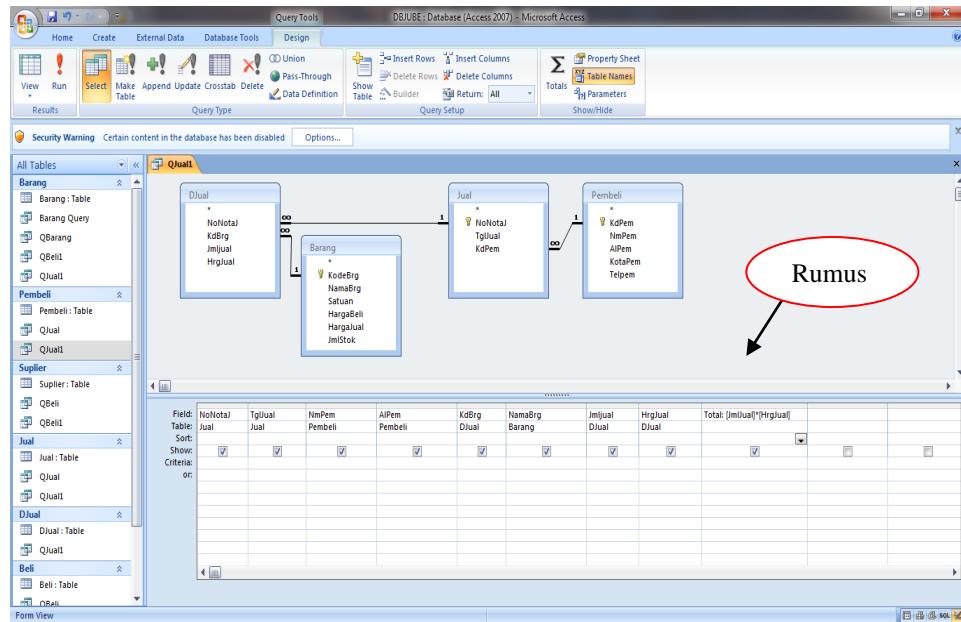
Query ini dapat mengatasi permasalahan apabila dalam tabel-tabel yang dimiliki membutuhkan operasi perhitungan atau suatu ekspresi. Sebagai contoh untuk query QJual1 memerlukan kolom berupa Total yang merupakan perkalian antara JmlJual dengan HrgJual. Langkahnya sebagai berikut:

- Buka query QJual1
- Hapus beberapa kolom supaya kolom tidak terlalu banyak. Misal yang dihapus kolom KotaPem dan kolom Satuan. Klik kolom yang akan dihapus lalu klik icon **Delete**. Lakukan penyimpanan ke nama yang baru dengan klik tombol **Office Button**, lalu pilih **Save Object As**. Ketikkan nama query yang baru yaitu **QJual2**.
- Letakkan kursor dikolom dimana ekspresi atau rumus akan dibutuhkan, lalu klik kanan dan pilih **Bulid** sampai ditampilkan kotak dialog untuk menulis ekspresi atau rumus. Silahkan mengetikkan rumus atau ekspresi yang diinginkan seperti terlihat pada Gambar 7.



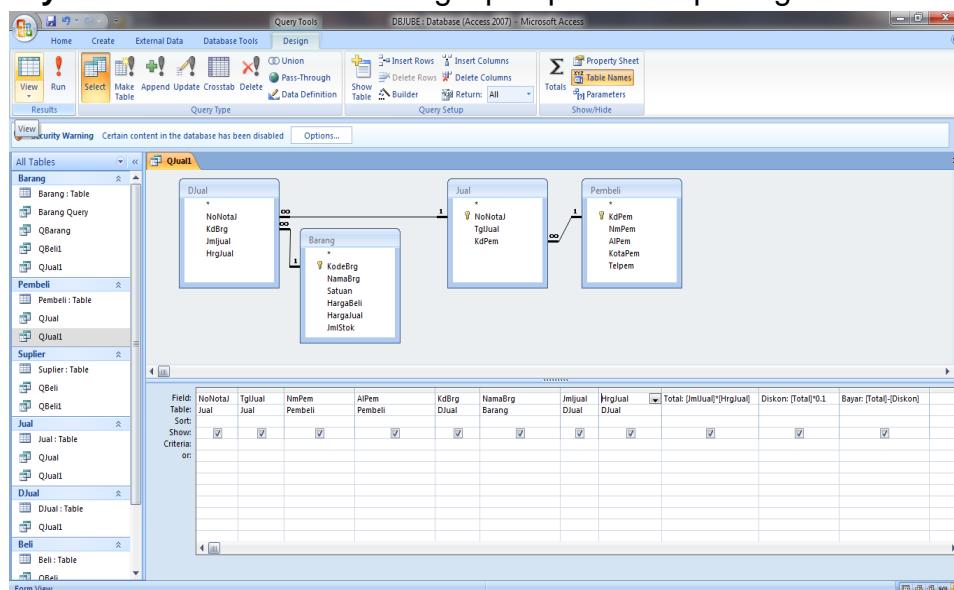
Gambar 7 Kotak Dialog Build

- d. Jika sudah selesai, klik tombol **OK**, hasilnya seperti terlihat pada Gambar 8.



Gambar 8 Desain Query Dengan Rumus/Ekspresi

- e. Silahkan lihat hasilnya (ubah ke modus *View Datasheet*)  
f. Silahkan lihat SQLnya (ubah ke modus *View SQL*)  
g. Rumus-rumus yang lain bisa ditambahkan, misalnya diskon dan Bayar. Rumus untuk Diskon adalah sebesar 10 persen dari Total. Rumus untuk **Bayar=Total-Diskon**. Arahkan kursor ke kolom kosong di kanan kolom Total, klik kanan lalu pilih **Build**. Ketik rumus **Diskon:Total\*0.1**. Selanjutnya arahkan kursor ke kolom kosong di kanan kolom Diskon, klik kanan lalu pilih **Build**. Ketik rumus **Bayar:Total-Diskon**. Hasil lengkap dapat dilihat pada gambar 9.



Gambar 9 Desain Query Dengan Rumus/Ekspresi Lengkap

**Query** yang dihasilkan seperti Gambar 9, akan menghasilkan tiga kolom (*field*) baru. Silahkan lihat hasilnya dengan merubah ke modus **View Datasheet**.

Perlu juga dilihat SQL yang dihasilkan dengan merubah ke modus SQL View. Silahkan diamati dan dipahami serta dicatat

Untuk memperlancar, silahkan membuat kolom Total untuk QBeli1 dan disimspan ke nama baru dengan nama QBeli2

### 3. Membuat **Query Crosstab**

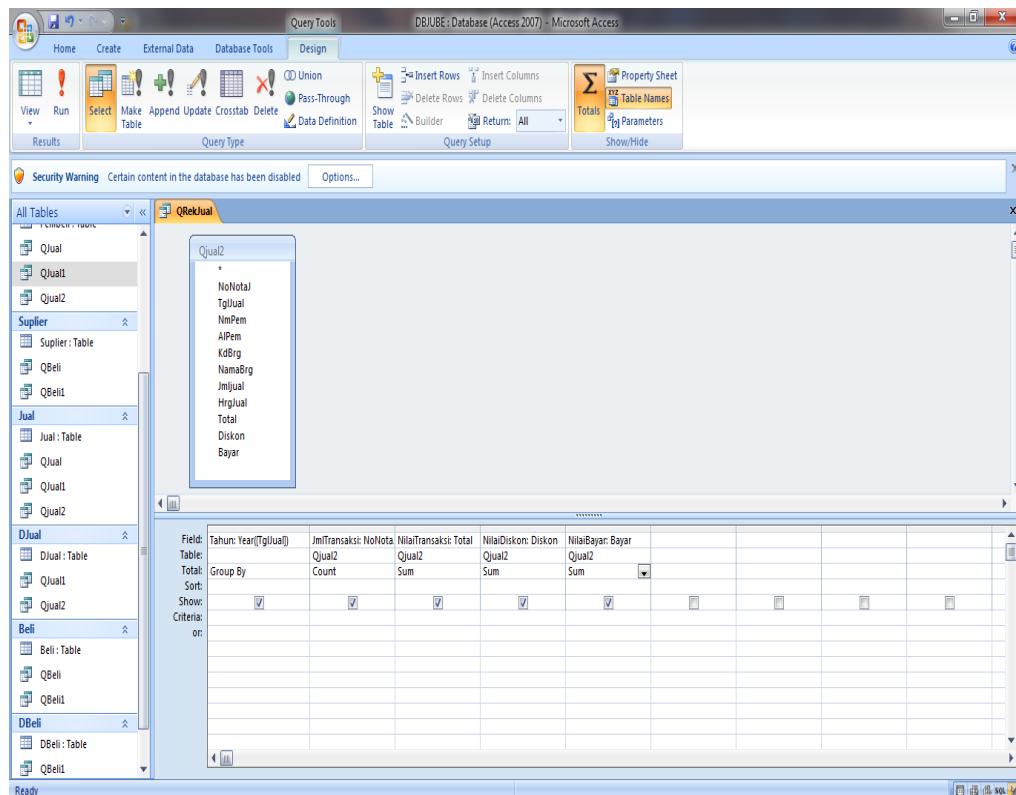
Query crosstab merupakan query yang digunakan untuk melakukan pengelompokan data berdasarkan *field* tertentu. Syarat untuk membuat jenis query ini adalah dengan menggunakan/mengaktifkan icon Total



a. Membuat query crosstab dari query query QJual2

- 1) Klik menu **Create**
- 2) Pilih **Query Design**, sehingga akan ditampilkan kotak dialog untuk memilih *table/query* yang digunakan sebagai sumber data.
- 3) Klik tab page **Queries** lalu pilih query **QJual2** sebagai sumber data dilanjutkan dengan mengklik tombol **Add**.
- 4) Klik tombol **Close**.
- 5) Ambil field **TglJual, NonotaJ, Total, Diskon, Bayar** dari query QJual2 dan letakkan secara berurutan pada bagian kolom *query*.
- 6) Untuk dasar pengelompokan data digunakan tahun. Namun field tahun tidak ada, maka dibuatlah *field* baru dengan cara menggunakan **Builder**. Klik kanan pada kolom TglJual lalu pilih Builder dan ketik ekspresi atau fungsi sebagai berikut:  
**Tahun:Year(TglJual)**
- 7) Klik kanan pada kolom NonotaJ lalu pilih **Builder** dan ketik ekspresi sebagai berikut: **JmlTransaksi:NonotaJ**
- 8) Klik kanan pada kolom Total lalu pilih **Builder** dan ketik ekspresi sebagai berikut: **NilaiTransaksi:Total**
- 9) Klik kanan pada kolom Diskon lalu pilih **Builder** dan ketik ekspresi sebagai berikut: **NilaiDiskon:Diskon**
- 10) Klik kanan pada kolom Bayar lalu pilih **Builder** dan ketik ekspresi sebagai berikut: **NilaiBayar:Bayar**

- 11) Coba lihat hasilnya dengan mengganti ke modus **View Datasheet**. Kolom tahun memang sudah tercipta, tapi data belum berkelompok. Untuk itu kembali ke modus **Design View**.
- 12) Klik atau aktifkan icon Total, sehingga akan ditambahkan baris Total pada kolom query.
- 13) Rubah tulisan *Group by* pada kolom NonotaJ di baris Total dengan **Count**.
- 14) Rubah tulisan *Group by* pada kolom Total di baris Total dengan **Sum**.
- 15) Rubah tulisan *Group by* pada kolom Diskon di baris Total dengan **Sum**.
- 16) Rubah tulisan *Group by* pada kolom Bayar di baris Total dengan **Sum**.
- 17) Simpan query dengan nama **QRekJual**, jika sudah selesai pastikan rancangan query seperti Gambar 10



Gambar 10 Desain Query Crosstab

Lihatlah hasil dari query yang dibuat dengan merubah ke modus **View Datasheet**.  
Lihatlah SQL yang dihasilkan dengan merubah ke modus **SQL View**

- b. Membuat *query crosstab* dari *query QBeli2*
  - a. Klik menu **Create**
  - b. Pilih **Query Design**, sehingga akan ditampilkan kotak dialog untuk memilih *table/query* yang digunakan sebagai sumber data.
  - c. Klik tab page **Queries** lalu pilih *query QBeli2* sebagai sumber data dilanjutkan dengan mengklik tombol **Add**.
  - d. Klik tombol **Close**.
  - e. Ambil *field TglBeli, NonotaB, Total* dari *query QBeli2* dan letakkan secara berurutan pada bagian kolom *query*.
  - f. Untuk dasar pengelompokan data digunakan tahun. Namun field tahun tidak ada, maka dibuatlah field baru dengan cara menggunakan **Builder**. Klik kanan pada kolom TglBeli lalu pilih **Builder** dan ketik ekspresi atau fungsi sebagai berikut: Tahun:Year(TglBeli)
  - g. Klik kanan pada kolom NonotaB lalu pilih **Builder** dan ketik ekspresi sebagai berikut: JmlTransaksi:NonotaB
  - h. Klik kanan pada kolom Total lalu pilih **Builder** dan ketik ekspresi sebagai berikut: NilaiTransaksi:Total
  - i. Coba lihat hasilnya dengan mengganti ke modus View Datasheet. Kolom tahun memang sudah tercipta, tapi data belum berkelompok. Untuk itu kembali ke modus **Design View**.
  - j. Klik atau aktifkan icon Total, sehingga akan ditambahkan baris Total pada kolom *query*.
  - k. Rubah tulisan *Group by* pada kolom NonotaB di baris Total dengan **Count**.
  - l. Rubah tulisan *Group by* pada kolom Total di baris Total dengan **Sum**.
  - m. Simpan *query* dengan nama **QRekBeli**

Lihatlah hasil dari *query* yang dibuat dengan merubah ke modus **View Datasheet**.

Lihatlah SQL yang dihasilkan dengan merubah ke modus **SQL View**

#### 4. Soal Latihan

- f. Bukalah database **DBHOTEL** yang telah dibuat pada pertemuan sebelumnya.
- g. Bukalah *query* QCekout.
- h. Buatlah kolom baru (*field* baru) dengan nama **LamaInap** (pengurangan TglCekout dengan TglCekIn).
- i. Buat juga kolom baru lainnya dengan nama **JmlBayar** (penjumlahan BeaKamar + Biaya Lain + Pajak)
  - 3) Lihat hasilnya (*Datasheet View*)
  - 4) Lihat SQL (*SQL View*), dipahami dan dicatat.
- j. Simpanlah *query* yang dihasilkan ke nama baru dan beri nama dengan **QCekout1**
- k. Buatlah *query crosstab* dari *query* QCekout dan bentuk tampilan mirip seperti *query crosstab* pada *query* QJual2, dan simpan *query* dengan nama **QRekCekout**

## **BAB 5**

### **STUDI KASUS DAN PENYELESAIANNYA**

Pada pertemuan ini akan dibahas suatu kasus dan solusi penyelesaian agar mendapat gambaran yang jelas mengenai basis data yang telah dikerja pada pertemuan-pertemuan sebelumnya.

#### **Kasus:**

Sebuah perusahaan pegadaian akan membuat sistem terkomputerisasi mengenai layanan administrasi gadai.

- Proses diawali dengan melakukan ketetapan mengenai golongan barang
- Orang yang akan melayani administrasi pegadaian disebut dengan petugas yang akan diberi hak akses terhadap sistem.
- Orang-orang yang melakukan transaksi ke pegadaian disebut dengan istilah Nasabah. Setiap kali melakukan transaksi, maka identitas nasabah perlu disertakan dalam bentuk foto copy kartu identitas (KTP/SIM/KTM/DII).
- Orang yang mendapat kesempatan untuk mendapatkan barang lelangan di pegadaian disebut dengan peserta lelang
- Proses memasukkan barang pada pihak perusahaan pegadaian disebut dengan istilah proses gadai
- Proses untuk melakukan pengambilan barang setelah masa jatuh tempo disebut dengan tebus.
- Proses untuk mendapatkan barang yang dilelang disebut dengan lelang.

#### **1. Membuat Database**

Proses diawali dengan memanggil aplikasi **Microsoft Access** sebagai berikut:

- a. Klik tombol **Start**
- b. Klik **All Programs**
- c. Cari *folder* dimana **Microsoft Office** berada dan lanjutkan dengan mengklik *folder* tersebut.
- d. Klik pilihan **Microsoft Access 2007** atau versi yang sesuai

Setelah mengklik pilihan **Microsoft Access 2007** atau versi yang sesuai, maka tunggu beberapa saat sampai ditampilkan area atau jendela kerja dari **Microsoft Access**.

Untuk membuat *database* dengan **Microsoft Access** dapat dilakukan dengan cara sebagai berikut:

- a. Klik **Blank Database** pada bagian **New Blank Database**
- b. Ketik nama *file database* yang diinginkan. Untuk keperluan praktikum ini beri nama *database* dengan nama **DBGADAI**.

- c. Tentukan *directory* atau *folder* dimana *database* akan disimpan. Silahkan arahkan ke *folder* tempat praktikum Anda. Jika nama *file* dan *directory* sudah benar, lanjutkan mengklik tombol **Create**

## 2. Membuat Tabel

Tabel-tabel yang dibutuhkan untuk pengelolaan data dalam sistem pegadaian cukup banyak. Tabel-tabel tersebut dikelompokkan dalam dua bagian yaitu tabel induk dan tabel transaksi.

### a. Tabel induk

Tabel induk yang dibutuhkan antara lain:

- 1) Tabel untuk menyimpan data golongan barang (GolBar)

Untuk membuat tabel baru, lakukan dengan langkah sebagai berikut:

- a) Klik menu **Create**
- b) Klik pilihan (**icon**) **Table**
- c) Ganti tampilan ke bentuk tampilan **Design View**
- d) Ketikkan nama tabel yang diinginkan yaitu **Golbar**, kemudian klik tombol **OK**.
- e) Buat atau ketik *field-field* yang dibutuhkan sbb:

Nama Field	Tipe Data	Ukuran	Keterangan/Deskripsi
Gol	Text	10	Golongan barang yang bisa digadaikan (sebagai primary key)
Kategori	Text	20	Kategori golongan barang
MaksPin	Number	Long Integer	Maksimal Pinjaman

- f) Isi data seperti contoh di bawah

Gol	Kategori	MaksPin
A	Kain	500000
B	Elektronik	2000000
C	Perhiasan	10000000
D	Kendaraan	50000000

- 2) Tabel untuk menyimpan data petugas pegadaian (Pengguna)

Untuk membuat tabel baru, lakukan dengan langkah sebagai berikut:

- a) Klik menu **Create**
- b) Klik pilihan (**icon**) **Table**
- c) Ganti tampilan ke bentuk tampilan **Design View**
- d) Ketikkan nama tabel yang diinginkan yaitu **Pengguna**, kemudian klik tombol **OK**.

- e) Buat atau ketik *field-field* yang dibutuhkan sbb:

Nama Field	Tipe Data	Ukuran	Keterangan/Deskripsi
Kodepeng	Text	3	Kode pengguna (sebagai primary key)
NamaPeng	Text	40	Nama pengguna
JabPeng	Text	20	Jabatan pengguna
Nmuser	Text	20	Nama user
Pasw	Text	20	Password pengguna
LevelP	Text	10	Level pengguna: - Admin - Kasir - User1 - User2

- f) Isi data minimal 5 data (*record*).

- 3) Tabel untuk menyimpan data nasabah (Nasabah)

Untuk membuat tabel baru, lakukan dengan langkah sebagai berikut:

- Klik menu **Create**
- Klik pilihan (**icon**) **Table**
- Ganti tampilan ke bentuk tampilan **Design View**
- Ketikkan nama tabel yang diinginkan yaitu **Nasabah**, kemudian klik tombol **OK**.
- Buat atau ketik *field-field* yang dibutuhkan sbb:

Nama Field	Tipe Data	Ukuran	Keterangan/Deskripsi
IdNas	Text	7	Kode nasabah (sebagai primary key)
NamaNas	Text	40	Nama nasabah
Jenkel	Text	1	Jenis Kelamin (L/P)
AlamatNas	Text	50	Alamat nasabah
TelpNas	Text	12	Nomor telepon nasabah
Noldentitas	Text	20	Nomor identitas nasabah
Pekerjaan	Text	15	Pekerjaan nasabah

- f) Isi data minimal 10 data (*record*).

- 4) Tabel untuk menyimpan data peserta lelang (**DPSL**)

Untuk membuat tabel baru, lakukan dengan langkah sebagai berikut:

- Klik menu **Create**
- Klik pilihan (**icon**) **Table**
- Ganti tampilan ke bentuk tampilan **Design View**

- d) Ketikkan nama tabel yang diinginkan yaitu **DPSL**, kemudian klik tombol **OK**.
- e) Buat atau ketik *field-field* yang dibutuhkan sbb:

<b>Nama Field</b>	<b>Tipe Data</b>	<b>Ukuran</b>	<b>Keterangan/Deskripsi</b>
KodePL	Text	7	Kode peserta lelang (sebagai primary key)
NamaPL	Text	40	Nama peserta lelang
AlamatPL	Text	50	Alamat peserta lelang
TelpPL	Text	12	Nomor telepon peserta lelang
Kategori	Text	20	Kategori peserta lelang: - Personal - Perusahaan

- f) Isi data minimal 10 data (*record*).
- b. Tabel Transaksi
- Tabel transaksi yang dibutuhkan antara lain:
- 1) Tabel untuk menyimpan data gadai (**Gadai**)
- Untuk membuat tabel baru, lakukan dengan langkah sebagai berikut:
- a) Klik menu **Create**
  - b) Klik pilihan (**icon**) **Table**
  - c) Ganti tampilan ke bentuk tampilan **Design View**
  - d) Ketikkan nama tabel yang diinginkan yaitu **Gadai**, kemudian klik tombol **OK**.
  - e) Buat atau ketik *field-field* yang dibutuhkan sbb:

<b>Nama Field</b>	<b>Tipe Data</b>	<b>Ukuran</b>	<b>Keterangan/Deskripsi</b>
NoSBK	Text	15	Nomor Surat Bukti Kredit (primary key)
TglGadai	Date		Tanggal gadai
Uraian	Text	75	Uraian barang gadai
Gol	Text	10	Golongan barang
HTaksiran	Number	Long Integer	Harga taksiran
UP	Number	Long Integer	Uang pinjaman
Adm	Number	Long Integer	Biaya administrasi
IdNas	Text	7	Kode nasabah
TglJT	Date		Tanggal jatuh tempo
Ket	Text	10	Keterangan: - Tebus - Perpanjang - Lelang
Kodepeng	Text	3	Kode pengguna

- f) Isi data minimal 10 data (*record*).
- 2) Tabel untuk menyimpan data tebus barang gadai (**Tebus**)  
Untuk membuat tabel baru, lakukan dengan langkah sebagai berikut:
- Klik menu **Create**
  - Klik pilihan (**icon**) **Table**
  - Ganti tampilan ke bentuk tampilan **Design View**
  - Ketikkan nama tabel yang diinginkan yaitu **Tebus**, kemudian klik tombol **OK**.
  - Buat atau ketik *field-field* yang dibutuhkan sbb:

<b>Nama Field</b>	<b>Tipe Data</b>	<b>Ukuran</b>	<b>Keterangan/Deskripsi</b>
NoSBK	Text	15	Nomor Surat Bukti Kredit (primary key)
Tgltebus	Date		Tanggal tebus
BPokok	Number	Long Integer	Besar uang pokok
Bbunga	Number	Long Integer	Besar bunga pinjaman
Kodepeng	Text	3	Kode pengguna

- f) Isi data minimal 10 data (*record*).
- 3) Tabel untuk menyimpan data lelang barang gadai (**Lelang**)  
Untuk membuat tabel baru, lakukan dengan langkah sebagai berikut:
- Klik menu **Create**
  - Klik pilihan (**icon**) **Table**
  - Ganti tampilan ke bentuk tampilan **Design View**
  - Ketikkan nama tabel yang diinginkan yaitu **Lelang**, kemudian klik tombol **OK**.
  - Buat atau ketik *field-field* yang dibutuhkan sbb:

<b>Nama Field</b>	<b>Tipe Data</b>	<b>Ukuran</b>	<b>Keterangan/Deskripsi</b>
NoSBK	Text	15	Nomor Surat Bukti Kredit (primary key)
Tgllelang	Date		Tanggal lelang
NLelang	Number	Long Integer	Nilai lelang
KodePL	Text	7	Kode peserta lelang
Kodepeng	Text	3	Kode pengguna

- f) Isi data minimal 10 data (*record*).

### 3. Membuat Relasi

*Relationship* merupakan hubungan antara tabel yang satu dengan tabel lainnya, dalam hal ini tabel master dengan tabel transaksi. Kegunaan dari relasi ini adalah jika terjadi perubahan di tabel master maka hasil perubahan tersebut akan mempengaruhi tabel yang berelasi (tabel transaksi).

Untuk membuat relasi dari *database* yang sudah ada, lakukan langkah-langkah sebagai berikut:

- a. Buka *database* yang telah dibuat (buka file database **DBGADAI**)
- b. Pada saat *database* sudah dibuka, klik menu **database tools**.
- c. Selanjutnya klik tab **relationships**. Jika **relationships** sudah ada sebelumnya, maka akan langsung ditampilkan. Jika belum ada, maka akan disajikan kotak dialog untuk memasukkan tabel.
- d. Tambahkan tabel-tabel yang akan dibuatkan *relationships* nya dengan cara mengklik nama tabel lalu klik tombol **Add**. Lakukan proses tersebut sampai semua tabel ditambahkan.
- e. Setelah tabel semua selesai diambil/ditambahkan, klik tombol **Close**.
- f. Kemudian hubungkan *field* dengan status *primary key* pada tabel master ke *field* yang bersesuaian pada tabel transaksi. Pada option relasi pilih semua **check** yang menunjukkan perubahan pada tabel master akan berpengaruh terhadap tabel transaksi (*check* integritas data, ubah data dan hapus data). Kemudian tekan tombol **Create**. Lakukan langkah ke-6 ini untuk menghubungkan tabel-tabel yang lain.

### 4. Membuat Query

Untuk membuat query dapat dilakukan dengan cara sebagai berikut:

- 1) Klik menu **Create**
- 2) Pilih **Query Design**
- 3) Pilih/klik tabel atau *query* yang akan digunakan sebagai sumber datanya, selanjutnya klik tombol **Add**. Lakukan langkah 3 ini sampai semua *table/query* yang dibutuhkan selesai ditambahkan. Setelah itu klik tombol **Close**.
- 4) Silahkan pilih *field-field* yang akan ditampilkan. Cara memilih *field* bisa satu per satu atau langsung semuanya. **Jika cara pertama** yang dipilih maka klik *field* yang diinginkan, lalu drag ke arah kolom dimana *field* tersebut akan diletakkan kemudian lepas. **Jika cara kedua** yang dipilih maka klik *field* yang pertama lalu tekan tombol Shift dan klik *field* terakhir, selanjutnya drag ke arah area kolom untuk menampilkan *field-field* tersebut kemudian lepas.
- 5) Jika sudah selesai memilih *field* yang dibutuhkan, simpanlah query yang dihasilkan dengan mengklik icon **Save** atau klik tombol **Office Button** lalu pilih **Save** atau bisa juga dengan cara langsung dengan menekan tombol **Ctrl+S**.
- 6) Ketik nama *file* yang akan diberikan untuk menyimpan *query*.

- 7) Jika diinginkan untuk melihat *output* atau hasil *query* gantilah modus ke **Datasheet View**, sedangkan jika ingin melihat SQL gantilah ke modus **SQL View**.

**a. Query dari table Nasabah**

- 1) Buatlah *query* dari tabel nasabah, dimana *field* yang diambil adalah *field* **IdNas**, **NamaNas**, **AlamatNas**, **Pekerjaan**. Data yang ditampilkan adalah nasabah yang memiliki pekerjaan “Petani”
- 2) Buatlah *query crosstab* dari tabel Nasabah (disimpan dengan nama **QRekNas**) sehingga data bisa ditampilkan seperti tampilan di bawah ini:

Pekerjaan	Laki-laki	Perempuan	Jumlah Semua

**b. Query dari tabel Gadai, tabel Golbar, tabel Nasabah, dan tabel Petugas.**

- 1) Buatlah query dalam bentuk query secara umum (disimpan dengan nama **QGadai**). *Field* yang diambil adalah semua *field* dari tabel Gadai, *field* Kategori dari tabel Golbar, *field* NamaNas, AlamatNas, TelpNas dari tabel Nasabah, *field* NamaPeng dari tabel Pengguna.
- 2) Buatlah **query crosstab** dimana sumber datanya diambil dari *query* QGadai (disimpan dengan nama **QRekGadai1**). *Field* yang diambil adalah Tlgadai, NoSBK, UP. Saat query ditampilkan, bentuk informasi yang disajikan bisa sebagai berikut:

Tahun	Jumlah Transaksi	Nilai Uang Pinjaman

- 3) Buatlah query crosstab dimana sumber datanya diambil dari *query* QGadai (disimpan dengan nama **QRekGadai2**). *Field* yang diambil adalah Tlgadai, Gol, NoSBK. Saat query ditampilkan, bentuk informasi yang disajikan bisa sebagai berikut:

Tahun	JmlGolA	JmlGolB	JmlGolC	JmlGolD	JmlSemua

c. Query dari tabel Tebus, tabel Gadai, tabel Golbar, tabel Nasabah, dan tabel Petugas

- 1) Buatlah query dalam bentuk *query* secara umum (disimpan dengan nama **QTebus**). *Field* yang diambil adalah *field* NoSBK dari tabel Tebus, TglGadai dari tabel Gadai, Field TglTebus dari tabel Tebus, *field* NamaNas dari tabel Nasabah, *field* NamaPeng dari tabel Pengguna, *field* BPokok, BBunga dari tabel Tebus, *field* LamaGadai dihasilkan dari pengurangan TglTebus-TglGadai, *field* JmlUangMasuk dihasilkan dari penjumlahan *field* BPokok+BBunga..
- 2) Buatlah **query crosstab** dimana sumber datanya diambil dari *query* QTebus (disimpan dengan nama **QRekTebus**). *Field* yang diambil adalah Tgltebus, BPokok, BBunga, JmlUangMasuk. Saat query ditampilkan, bentuk informasi yang disajikan bisa sebagai berikut:

Tahun	JmlPokok	JmlBunga	JmlTotal

d. Query dari tabel Lelang, tabel Gadai, tabel Nasabah, tabel DPSL, dan tabel Petugas

Buatlah *query* dalam bentuk *query* secara umum (disimpan dengan nama **QLelang**). *Field* yang diambil adalah *field* NoSBK dari tabel Lelang, NamaNas dari tabel Nasabah, Tlgadai, TglJT, UP dari tabel Gadai, TglLelang, NLelang dari tabel Lelang, NamaPL dari tabel DPSL, *field* UangKembali dihasilkan dari pengurangan *field* NLelang dengan UP.

## 5. Tugas

- a. Catat dan pahami semua SQL yang dihasilkan dari proses pembuatan query.
- b. Setelah studi kasus ini selesai, maka kirimkan *database* yang dihasilkan ke email: [zaidirtan@gmail.com](mailto:zaidirtan@gmail.com) paling lambat hari ini jam 23.59 WIB. Apabila melebihi waktu yang ditentukan tidak mendapat nilai Tugas.
- c. Tuliskan kesulitan apa yang saudara alami dalam penyelesaian permasalahan ini. Kesulitan yang dialami tersebut ditulis secara rinci dan jelas pada email yang dikirimkan.

## BAB 6

### BELAJAR MYSQL DENGAN COMMAND LINE

#### A. Pengantar MySQL

##### 1. Sejarah

Siapa yang tidak mengenal dengan MySQL? Apa itu MySQL? **MySQL** (dibaca : mai-se-kuel) adalah perangkat lunak *database relasi* (**Relational Database Management System** atau **RDBMS**), seperti halnya **ORACLE**, **Postgresql**, **MS SQL**, dan sebagainya. Kalau lebih jelasnya MySQL merupakan aplikasi pengolah database yang bersifat *open source*, dikembangkan oleh **Oracle** (sebelumnya Sun dan MySQL AB). Merupakan pengolah *database* yang paling banyak digunakan di dunia dan lazim diterapkan untuk aplikasi web. MySQL ini banyak digunakan oleh para programmer apalagi oleh *web developer* karena sifatnya yang *free alias gratis*. MySQL pertama kali dibuat dan dikembangkan di Swedia, yaitu oleh David Axmark, Allan Larsson dan Michael "Monty" Widenius. MySQL merupakan *software database open source* yang paling populer di dunia, dimana saat ini digunakan lebih dari 100 juta pengguna di seluruh dunia.



MySQL dikembangkan sejak tahun 1980-an. Saat ini versi **MySQL** yang sudah stabil mencapai versi 5x, dan sedang dikembangkan versi 6x dan mungkin juga selebihnya. Untuk lebih lengkapnya dapat dilihat di situs resmi MySQL. Untuk mendownload Mysql bisa langsung ke situs Resminya yaitu [www.mysql.com](http://www.mysql.com).

##### 2. Kemampuan MySql

Mysql ini kemampuannya sudah bisa diandalkan, mempunyai kapasitas yang cukup baik sekitar 60.000 tabel dengan jumlah *record* mencapai kurang lebih 5.000.000.000 bahkan untuk yang terbaru sudah lebih. Keamanan datanya pun cukup aman walaupun tidak sehebat Postgre apalagi Oracle. *Engine* ini *multiplatform* sehingga mampu diaplikasikan di berbagai sistem operasi. MySql cocok diaplikasikan diaplikasi kelas kecil dan menengah. Kelebihan paling utama *engine* ini adalah kecepatannya.

**SQL (Structured Query Language)** didefinisikan sebagai suatu sintaks perintah-perintah tertentu yang digunakan untuk mengelola suatu *database*.

Artinya MySQL adalah dua hal yang berbeda SQL. Mudahnya, MySQL adalah *softwarenya*, dan SQL adalah bahasa perintahnya.

### 3. Jenis Data pada MySql

Macam-macam Tipe Data pada **MySQL (Mai Se Kuel)** sebenarnya mempunyai beberapa jenis, secara umum tipe-tipe data MySQL ini ada empat (4), diantaranya yaitu:

- a. Tipe Data **Numeric**.
- b. Tipe Data **String**.
- c. Tipe Data **Date**.
- c. Tipe Data **Kelompok Himpunan**.

Jenis-jenis tipe data pada MySQL di atas merupakan tipe data yang harus dipelajari saat sedang belajar *Database*, karena sebagai modal awal dalam pembelajaran *Database* menggunakan MySQL, maka lebih baik mengetahui dan mengerti masing-masing kegunaan tipe-tipe data pada MySQL tersebut.

#### a. Tipe data Numeric

Tipe data numerik yaitu tipe data yang digunakan untuk menyimpan data numeric (angka).

##### 1) TINYINT

Penggunaan: digunakan untuk menyimpan data bilangan bulat positif dan negatif.

Jangkauan: -128 s/d 127

Ukuran: 1 byte (8 bit).

##### 2) SMALLINT

Penggunaan: digunakan untuk menyimpan data bilangan bulat positif dan negatif.

Jangkauan: -32.768 s/d 32.767

Ukuran: 2 byte (16 bit).

##### 3) MEDIUMINT

Penggunaan: digunakan untuk menyimpan data bilangan bulat positif dan negatif.

Jangkauan: -8.388.608 s/d 8.388.607

Ukuran: 3 byte (24 bit).

- 4) INT  
Penggunaan: digunakan untuk menyimpan data bilangan bulat positif dan negatif.  
Jangkauan: -2.147.483.648 s/d 2.147.483.647  
Ukuran: 4 byte (32 bit).
- 5) BIGINT  
Penggunaan: digunakan untuk menyimpan data bilangan bulat positif dan negatif.  
Jangkauan:  $\pm 9,22 \times 10^{18}$   
Ukuran: 8 byte (64 bit).
- 6) FLOAT  
Penggunaan: digunakan untuk menyimpan data bilangan pecahan positif dan negatif presisi tunggal.  
Jangkauan: -3.402823466E+38 s/d -1.175494351E-38, 0, dan 1.175494351E-38 s/d 3.402823466E+38.  
Ukuran: 4 byte (32 bit).
- 7) DOUBLE  
Penggunaan : digunakan untuk menyimpan data bilangan pecahan positif dan negatif presisi ganda.  
Jangkauan: -1.79...E+308 s/d -2.22...E-308, 0, dan 2.22...E-308 s/d 1.79...E+308.  
Ukuran: 8 byte (64 bit).
- 8) REAL  
Penggunaan : digunakan untuk menyimpan data bilangan pecahan positif dan negatif presisi ganda.  
Jangkauan: -1.79...E+308 s/d -2.22...E-308, 0, dan 2.22...E-308 s/d 1.79...E+308.  
Ukuran: 8 byte (64 bit).
- 9) DECIMAL  
Penggunaan : digunakan untuk menyimpan data bilangan pecahan positif dan negatif.  
Jangkauan: -1.79...E+308 s/d -2.22...E-308, 0, dan 2.22...E-308 s/d 1.79...E+308.  
Ukuran: 8 byte (64 bit).
- 10) NUMERIC  
Penggunaan : digunakan untuk menyimpan data bilangan pecahan positif dan negatif.  
Jangkauan: -1.79...E+308 s/d -2.22...E-308, 0, dan 2.22...E-308 s/d 1.79...E+308.  
Ukuran: 8 byte (64 bit).

b. Tipe data String

Tipe data string yaitu tipe data yang digunakan untuk menyimpan data string (text).

1) CHAR

Penggunaan: digunakan untuk menyimpan data string ukuran tetap.  
Jangkauan: 0 s/d 255 karakter

2) VARCHAR

Penggunaan: digunakan untuk menyimpan data string ukuran dinamis.  
Jangkauan: 0 s/d 255 karakter (versi 4.1), 0 s/d 65.535 (versi 5.0.3)

3) TINYTEXT

Penggunaan: digunakan untuk menyimpan data text.  
Jangkauan: 0 s/d 255 karakter (versi 4.1), 0 s/d 65.535 (versi 5.0.3)

4) TEXT

Penggunaan: digunakan untuk menyimpan data text.  
Jangkauan: 0 s/d 65.535 (216 - 1) karakter

5) MEDIUMTEXT

Penggunaan: digunakan untuk menyimpan data text.  
Jangkauan: 0 s/d 224 - 1 karakter

6) LONGTEXT

Penggunaan: digunakan untuk menyimpan data text.  
Jangkauan: 0 s/d 232 - 1 karakter

c. Tipe data Date dan Time

Tipe data date dan time yaitu tipe data yang digunakan untuk menyimpan data tanggal dan waktu.

1) DATE

Penggunaan: digunakan untuk menyimpan data tanggal.  
Jangkauan: 1000-01-01 s/d 9999-12-31 (YYYY-MM-DD)  
Ukuran: 3 byte.

2) TIME

Penggunaan: digunakan untuk menyimpan data waktu.  
Jangkauan: -838:59:59 s/d +838:59:59 (HH:MM:SS)  
Ukuran: 3 byte.

3) DATETIME

Penggunaan: digunakan untuk menyimpan data tanggal dan waktu.  
Jangkauan: '1000-01-01 00:00:00' s/d '9999-12-31 23:59:59'  
Ukuran: 8 byte.

4) YEAR

Penggunaan: digunakan untuk menyimpan data tahun dari tanggal.

Jangkauan: 1900 s/d 2155

Ukuran: 1 byte.

d. Tipe data Blob (biner)

Tipe data blob digunakan untuk menyimpan data biner.

1) BIT (sejak versi 5.0.3)

Penggunaan: digunakan untuk menyimpan data biner.

Jangkauan: 64 digit biner

2) TINYBLOB

Penggunaan: digunakan untuk menyimpan data biner.

Jangkauan: 255 byte

3) BLOB

Penggunaan: digunakan untuk menyimpan data biner.

Jangkauan: 216 - 1 byte

4) MEDIUMBLOB

Penggunaan: digunakan untuk menyimpan data biner.

Jangkauan: 224 - 1 byte

5) LONGBLOB

Penggunaan: digunakan untuk menyimpan data biner.

Jangkauan: 232 - 1 byte

e. Tipe data yang lain

Selain tipe data di atas, MySQL juga menyediakan tipe data yang lain, diantaranya adalah :

1) ENUM

Penggunaan: enumerasi (kumpulan data).

Jangkauan: sampai dengan 65535 string.

2) SET

Penggunaan: combination (himpunan data).

Jangkauan: sampai dengan 255 string anggota.

Jadi kurang lebih ada sekitar 27 Tipe Data pada MySQL, dimana masing-masing tipe data tersebut bisa dimanfaatkan sesuai kebutuhan yang diperlukan.

#### 4. Perhatian Khusus

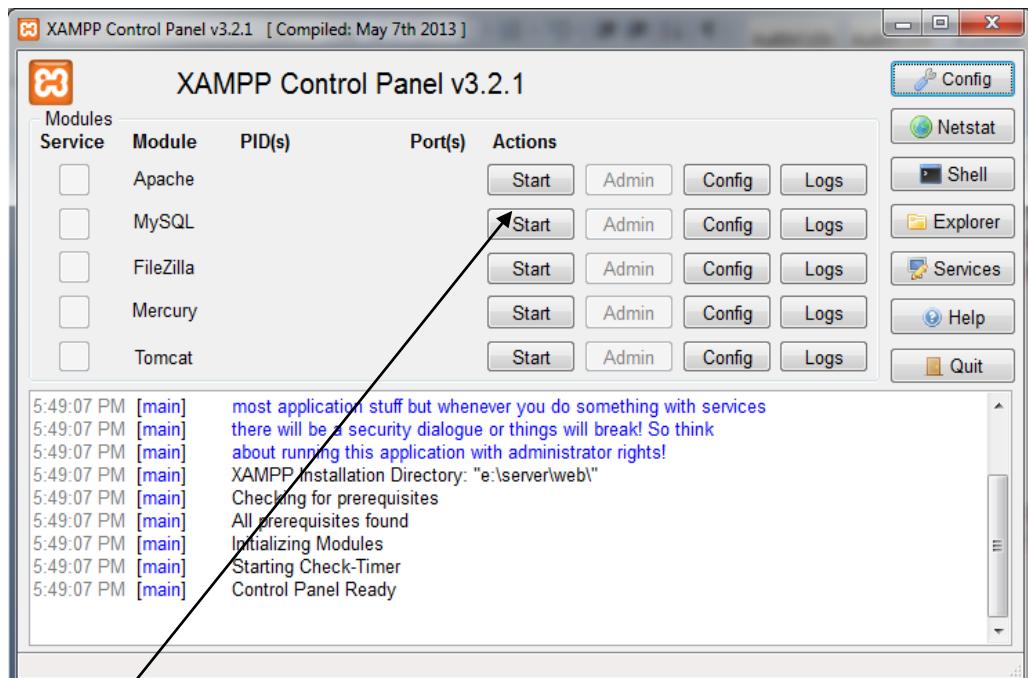
**Membuat database MySQL di Xampp dengan CMD (Command Prompt)** - memang sedikit sulit dibandingkan membuat *database* langsung menggunakan browser karena disitu sudah menggunakan mode GUI (*Graphical User Interface*) tanpa harus mengetikkan perintah maupun SQL. Agak sedikit perbedaan pembuatan *database* lewat CMD pada Xampp dengan AppServ, karena pada xampp tidak perlu menggunakan password melainkan hanya username. Langsung saja kita lihat cara membuat *database* MySQL di Xampp dengan CMD.

### B. Praktek MySQL Menggunakan Command Line

#### 1. Persiapan

##### a. Mengaktifkan Panel Xampp

Untuk masuk ke **Command Line** lakukan dengan cara klik **Start - All Programs - XAMPP - AMPP Control Panel** sampai ditampilkan kotak dialog seperti Gambar 1.

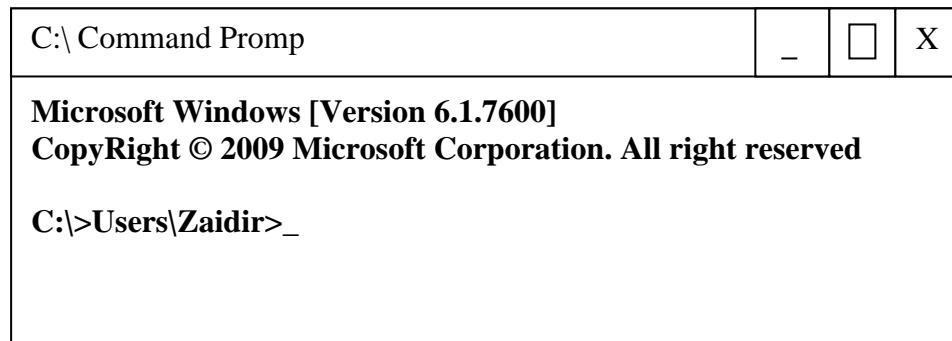


Klik Start pada modul MySQL

Gambar 1 Kotak dialog panel Xampp

##### b. Masuk ke Command Line

Untuk masuk ke Command Line lakukan dengan cara klik **Start - All Programs - Accessories - Command Prompt** atau dengan shortcut tombol **WINDOWS+R** kemudian ketikan “**CMD**” tanpa tanda kutip kemudian **ENTER**, setelah itu akan tampil seperti terlihat pada Gambar 2.



Gambar 2 Tampilan *Command Line*

Untuk keluar dari direktori yang ada saat ini maka ketik CD\ lalu tekan enter sehingga promp akan berubah menjadi C:>

Karena akan menggunakan Xampp, jadi buka directory lokasi Xampp yang terinstal biasanya terdapat di drive C dengan perintah “cd c:\xampp\mysql\bin” kemudian Enter. Dimanapun Xampp berada, cara masuk ke *directory* adalah seperti tersebut, sehingga kondisi siap untuk bekerja dengan MySQL, misalnya *prompt* menjadi:

C:\xampp\mysql\bin>\_

atau

E:\Datalama\xampp\mysql\bin>\_

## 2. Login ke MySQL

Pada saat kondisi sudah siap seperti di atas, ketik perintah **mysql -u root** Lanjutkan menekan **Enter**. Setelah mengetikkan perintah tersebut, akan ditampilkan kondisi seperti Gambar 3.

A screenshot of a Microsoft Windows Command Prompt window titled "Command Prompt - mysql -u root". The window content shows the MySQL monitor starting with "Welcome to the MySQL monitor. Commands end with ; or \g." It displays the connection id (2), server version (5.6.11 MySQL Community Server <GPL>), copyright information (Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.), and a note about Oracle trademarks. The prompt ends with "mysql> \_".

Gambar 3 Tampilan setelah login ke MySQL

### 3. Membuat database baru

Untuk membuat *database* baru, perintah yang digunakan adalah sebagai berikut:

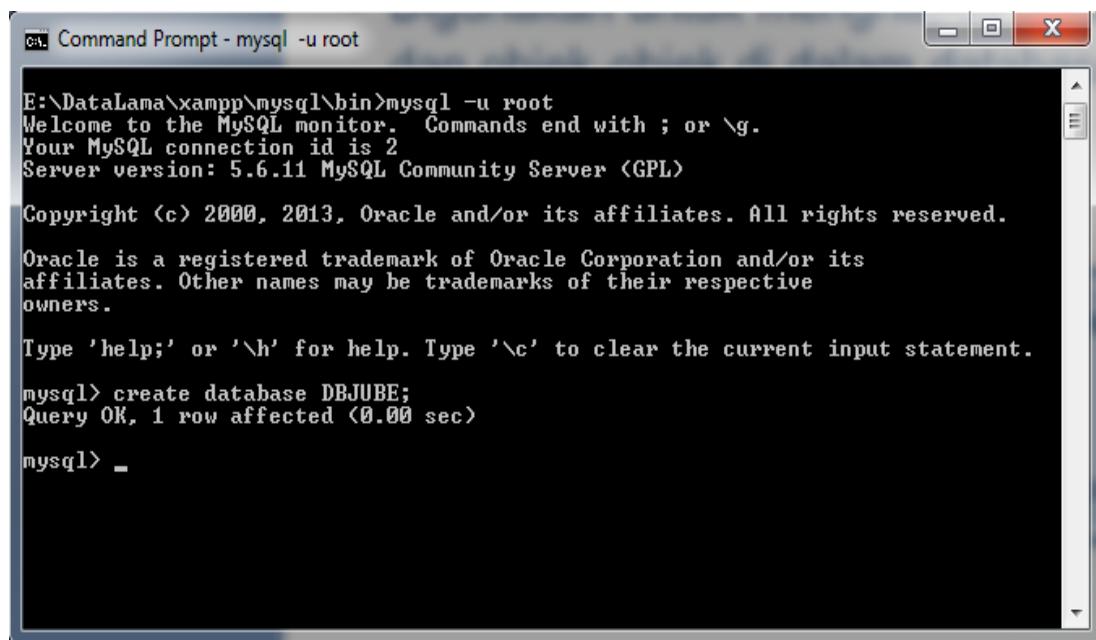
**CREATE DATABASE nama\_database;**

Contoh kalau akan membuat *database* dengan nama DBJUBE, perintah yang digunakan:

**CREATE DATABASE DBJUBE;**

jangan lupa untuk mengakhiri perintah mysql dengan tanda titik koma ( ; ) supaya pada waktu menekan **Enter** perintah tersebut akan dieksekusi. Jika lupa menuliskan tanda titik koma (;), maka kursor akan turun ke bawah dalam artian ganti baris saja, akan tetapi perintah tidak dieksekusi. Untuk mengatasi hal tersebut, ketik saja tanda titik koma (;) kemudian tekan tombol Enter.

Jika proses pembuatan *database* berhasil atau tidak ada kesalahan, maka akan terlihat tampilan seperti Gambar 4.



```
E:\DataLama\xampp\mysql\bin>mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.6.11 MySQL Community Server (GPL)

Copyright (c) 2000, 2013, 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> create database DBJUBE;
Query OK, 1 row affected (0.00 sec)

mysql> _
```

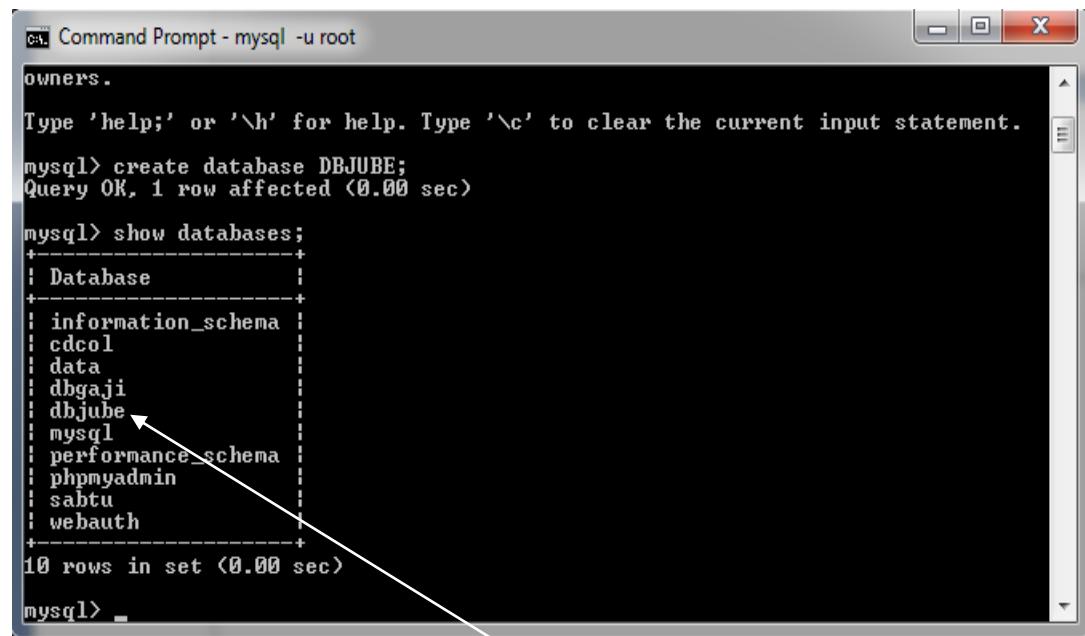
Gambar 4 Tampilan proses membuat *Database*

### 4. Melihat *Database*

Untuk melihat *database* yang sudah ada pada *directory* yang aktif adalah dengan perintah:

**SHOW Databases;**

Setelah memberikan perintah di atas, akan ditampilkan daftar *database* yang sudah ada seperti Gambar 5.



```
ca. Command Prompt - mysql -u root
owners.

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

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

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| cdcoll |
| data |
| dbgaji |
| dbjube |
| mysql |
| performance_schema |
| phpmyadmin |
| sabtu |
| webauth |
+-----+
10 rows in set (0.00 sec)

mysql> _
```

Database yang sudah dibuat

Gambar 5 Tampilan Daftar *Database*

## 5. Masuk ke *database* untuk pengelolaan

Untuk masuk ke *database* tertentu, perintah yang digunakan adalah USE, dengan struktur perintahnya yaitu:

**USE nama\_database**

Contoh penerapannya adalah:

**USE DBJUBE;**

Jika perintah yang diketikkan benar, maka akan dicetak pesan **Database Changed**, kemudian sudah bisa mengolah *database* tersebut.

## 6. Membuat table

Struktur perintah untuk membuat tabel adalah sebagai berikut:

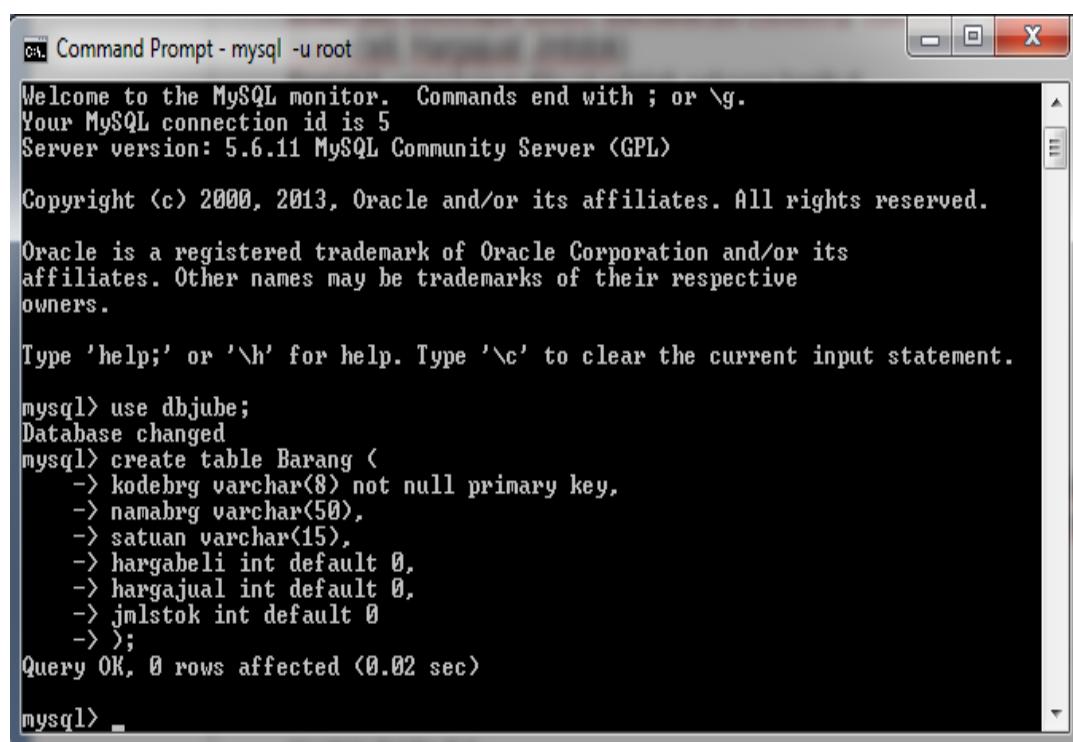
```
CREATE TABLE nama_tabel(
    kolom1 type_data(ukuran) ,
    kolom2 type_data(ukuran),
    kolom3 type_data(ukuran),
    kolom4 type_data(ukuran) ,
    ...
    kolom_n type_data(ukuran)
);
```

Pada contoh berikut akan dibuat tabel Barang dengan struktur tabel terdiri dari beberapa kolom, diantaranya (Kodebrg, Namabrg, Satuan, Hargabeli, Hargajual, Jmlstok).

Perintah yang harus dibuat adalah sebagai berikut:

```
CREATE TABLE Barang (
Kodebrg varchar(8) not null primary key,
Namabrg varchar(50),
Satuan varchar(15),
HargaBeli int default 0,
HargaJual int default 0,
JmlStok int default 0
);
```

Jika perintah di atas berhasil, maka akan dicetak pesan **Query OK** (lihat Gambar 6), sedangkan jika ditemukan kesalahan dalam perintah di atas, maka akan dicetak pesan kesalahan berserta dengan petunjuk lokasi kesalahannya.



```
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 5
Server version: 5.6.11 MySQL Community Server (GPL)

Copyright (c) 2000, 2013, 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> use dbjube;
Database changed
mysql> create table Barang (
    -> kodebrg varchar(8) not null primary key,
    -> namabrg varchar(50),
    -> satuan varchar(15),
    -> hargabeli int default 0,
    -> hargajual int default 0,
    -> jmlstok int default 0
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql> _
```

Gambar 6 Tampilan perintah berhasil

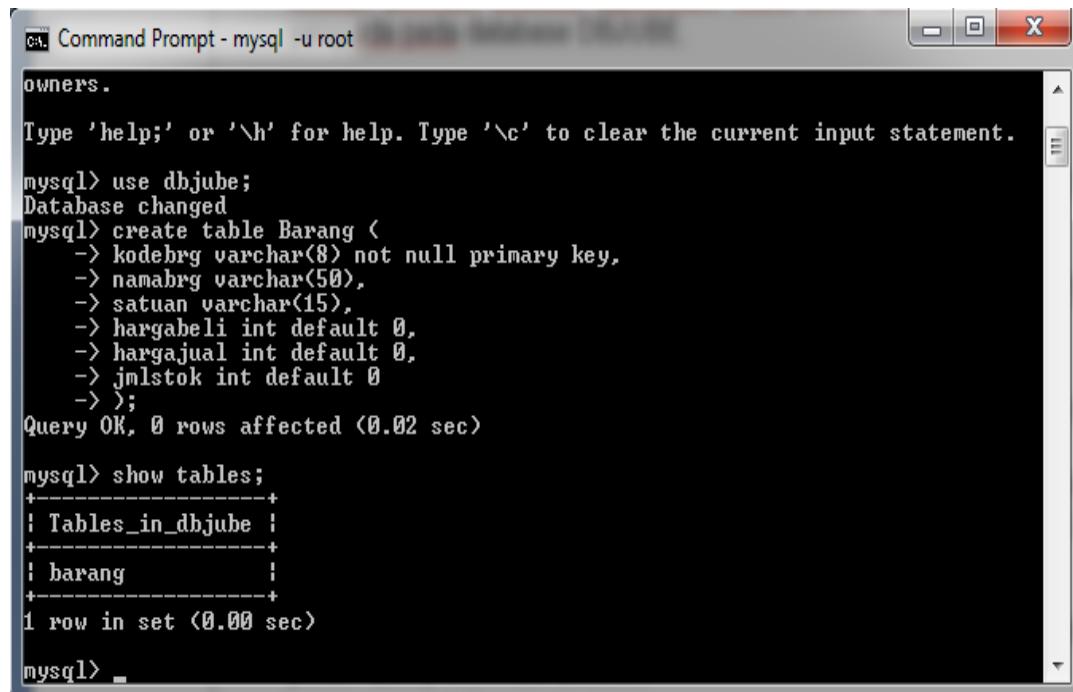
Untuk selanjutnya silahkan membuat tabel yang lain yaitu tabel **Pembeli**, dan tabel **Suplier** dengan rincian *field* menyesuaikan dengan apa yang telah dibuat pada modul pertemuan 2 atau buka *database DBJUBE* yang pernah dibuat pada praktikum sebelumnya. Jenis data **Text** diganti dengan tipe data **Char** atau **Varchar**.

## 7. Melihat daftar tabel dalam database

Untuk mengetahui ada berapa tabel dalam *database* **DBJUBE**, maka gunakan perintah berikut:

**SHOW TABLES;**

setelah perintah tersebut dijalankan, maka akan ditampilkan daftar tabel yang ada pada *database* **DBJUBE**.



```
Command Prompt - mysql -u root
owners.

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

mysql> use dbjube;
Database changed
mysql> create table Barang (
    -> kodebrg varchar(8) not null primary key,
    -> namabrg varchar(50),
    -> satuan varchar(15),
    -> hargabeli int default 0,
    -> hargajual int default 0,
    -> jmlstok int default 0
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql> show tables;
+-----+
| Tables_in_dbjube |
+-----+
| barang           |
+-----+
1 row in set (0.00 sec)

mysql> _
```

Gambar 7 Tampilan daftar tabel

## 8. Memasukkan data ke dalam table

Untuk memasukkan data/*record* pada tabel barang, maka perintah yang digunakan adalah perintah **insert**, struktur perintahnya adalah:

**INSERT INTO NamaTabel (kolom1, kolom2, kolom3, kolom\_n) values ('isi 1', 'isi 2', 'isi 3', 'isi n')**

Pada contoh berikut akan memasukkan *record* pada tabel Barang yang telah kita buat sebelumnya.

**INSERT INTO Barang (Kodebrg, Namabrg, Satuan, Hargabeli, Hargajual, Jmlstok) values ('B-000001', 'Monitor', 'Unit', 600000, 675000, 30)**

jika perintah di atas dijalankan, maka akan dimasukkan data ke dalam tabel Barang. Jika perintah berhasil, akan dicetak pesan Query Ok, 1 row affected seperti terlihat pada Gambar 8.

```
Command Prompt - mysql -u root
mysql> create table Barang (
    -> kodebrg varchar(8) not null primary key,
    -> namabrg varchar(50),
    -> satuan varchar(15),
    -> hargabeli int default 0,
    -> hargajual int default 0,
    -> jmlstok int default 0
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql> show tables;
+-----+
| Tables_in_dbjuhe |
+-----+
| barang |
+-----+
1 row in set (0.00 sec)

mysql> insert into barang
    -> (kodebrg, namabrg, satuan, hargabeli, hargajual, jmlstok) values
    -> ('B-000001', 'Monitor', 'Unit', 600000, 675000, 30)
    -> ;
Query OK, 1 row affected (0.00 sec)

mysql> _
```

Gambar 8 Tampilan setelah menyisipkan *record*

Untuk selanjutnya silahkan masukkan data barang sebanyak data barang yang pernah dibuat pada modul pertemuan 2.

Begitu juga dengan tabel Pembeli dan tabel Suplier, silahkan diisi datanya berdasarkan data yang mirip dengan isi tabel yang telah dibuat pada modul pertemuan 2

## 9. Menampilkan data atau *record table*

Untuk melihat data dari sebuah tabel, maka digunakan perintah select , yang struktur perintahnya adalah sebagai berikut:

**SELECT kolom1,kolom2,kolom\_n FROM nama\_tabel**

pada contoh berikut akan menampilkan isi dari tabel barang, maka digunakan perintah:

**SELECT \* FROM Barang;**

Perintah di atas akan menampilkan isi semua kolom dalam tabel barang, sedangkan jika perintahnya diubah menjadi:

**SELECT kodebrg,Namabrg,jmlstok FROM Barang;**

Maka hanya akan ditampilkan isi dari Kolom kodebrg,Namabrg dan kolom jmlstok

Penjelasan lebih detail dari perintah di atas akan dibahas dalam Mysql query

Untuk selanjutnya silahkan melakukan latihan untuk menampilkan data barang dengan berbagai variasi dengan mengacu pada *query* barang yang pernah dicobakan pada praktikum-praktikum sebelumnya.  
Begini juga dengan proses menampilkan data untuk tabel Pembeli dan tabel Suplier.

## 10. Mengubah data table

Untuk melakukan perubahan data pada tabel, maka perintah yang digunakan adalah perintah **UPDATE**, struktur perintahnya adalah sebagai berikut:

**UPDATE nama\_tabel SET kolom1='isi baru' , kolom2='isi baru' WHERE kolom\_unik='kode\_unik' ;**

Pada contoh berikut kita akan mengubah Nama barang yang berada pada tabel Barang. Perlu diketahui bahwa kolom unik atau primary key dari tabel barang adalah kodebrg, maka kita akan mengubah data barang dengan kodebrg tertentu.

**UPDATE barang SET Namabrg='Layar', satuan='Dus', jmlstok=50 WHERE kodebrg='B-000001 '**

Jika perintah tersebut dijalankan, maka data barang yang memiliki kodebrg = B-000001 akan diubah, yaitu Namabrg = Monitor akan diubah menjadi Layar, Satuan diubah menjadi Dus dan Jmlstok diubah menjadi 50. Untuk melihat hasilnya, silahkan tampilkan dengan menggunakan perintah **SELECT**

## 11. Menghapus data

Untuk menghapus data pada tabel, perintah yang digunakan adalah perintah **DELETE**. Struktur perintahnya adalah sebagai berikut:

**DELETE FROM nama\_tabel WHERE kolom\_unik = 'angka\_unik';**

Contoh penerapannya pada tabel barang adalah ketika akan menghapus data barang yang memiliki kodebrg B-000001, maka perintah yang digunakan adalah sebagai berikut:

**DELETE FROM barang WHERE kodebrg='B-000001';**

## BAB 7

# BELAJAR MYSQL DENGAN COMMAND LINE LANJUT

### A. Perintah-Perintah Penting

#### 1. Perintah Alter Table

Pada saat tertentu dalam mengelola *database*, khususnya pada saat bekerja dengan tabel, dibutuhkan proses untuk merubah struktur tabel yang pernah dibuat. MySql menyediakan perintah untuk mengatasi masalah-masalah tersebut. Perintahnya adalah **ALTER TABLE**.

Bentuk Umum perintahnya adalah:

**ALTER TABLE nama\_tabel alter\_options;**

dimana:

- **ALTER TABLE** merupakan perintah dasar untuk mengubah tabel.
- **nama\_tabel** merupakan nama tabel yang akan diubah strurnya.
- **alter\_options** merupakan pilihan perubahan tabel. Contoh **Option** yang bisa digunakan adalah sebagai berikut:

Nama Option	Fungsi Option
<b>ADD</b> nama_field/kolom_baru	untuk menambahkan <i>field/kolom baru</i> .
<b>ADD PRIMARY KEY</b> (nama_field/kolom)	untuk menambahkan <i>primary key</i> pada tabel
<b>CHANGE</b> nama_field_lama nama_field_baru_definisi_field_baru	untuk mengubah nama_field_lama menjadi nama_field_baru_definisi_field_baru <b>definisi_field_baru</b> artinya adalah type data.
<b>MODIFY</b> nama_field type_data	untuk mengubah type_data suatu <i>field</i>
<b>DROP</b> nama_field	untuk menghapus <i>field</i> nama_field
<b>RENAME TO</b> nama_tabel_baru	untuk mengganti nama tabel

Sekarang saatnya dibahas satu persatu supaya bisa dipahami. Sebelumnya, buatlah sebuah tabel terlebih dahulu untuk percobaan. Misal di sini dibuat tabel sederhana dengan nama **peserta** dan strurnya sebagai berikut :

Nama Kolom	Type data	Panjang Data	Keterangan
NIP	integer	5	<ul style="list-style-type: none"><li>○ bukan primary key</li><li>○ bukan auto increment</li></ul>
nama	varchar	30	



Perintahnya adalah sebagai berikut:

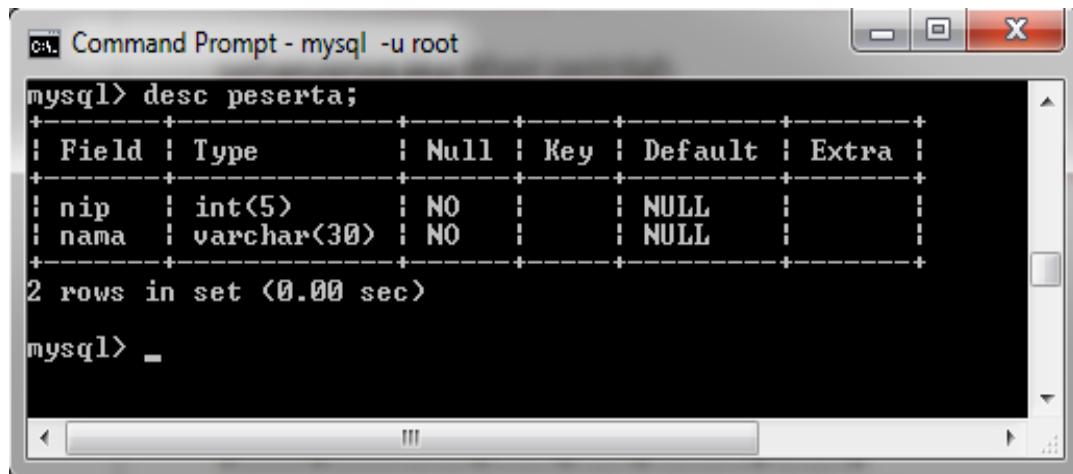
```
CREATE TABLE peserta (
    NIP int(5) NOT NULL,
    nama varchar(30) NOT NULL);
```

\*sengaja tidak di beri “primary key” dan “auto increment”

Untuk melihat struktur tabel yang telah dibuat dapat diberikan perintah sebagai berikut:

```
DESC peserta;
```

hasil dari perintah di atas dapat dilihat pada Gambar 1



The screenshot shows a Windows Command Prompt window titled "Command Prompt - mysql -u root". Inside, the MySQL command "desc peserta;" is run, resulting in the following table output:

Field	Type	Null	Key	Default	Extra
nip	int(5)	NO		NULL	
nama	varchar(30)	NO		NULL	

Below the table, the message "2 rows in set <0.00 sec>" is displayed. The MySQL prompt "mysql> \_" is at the bottom.

Gambar 1 Tampilan dengan perintah desc

Karena perintah **alter table** terdapat banyak sekali variasinya, maka akan dibahas beberapa saja yang sering digunakan, yaitu:

- Tambah Field/Kolom Baru
- Tambah/hapus Primary Key
- Hapus Field/Kolom
- Ubah Nama Field/Kolom
- Ubah Type Data Suatu Field/Kolom
- Ganti Nama Tabel

a. **Tambah Field/Kolom Baru.**

Misal ingin menambah *Field/Kolom alamat* pada tabel di atas, maka perintahnya adalah:

```
ALTER TABLE peserta ADD alamat varchar(50);
```

Setelah itu, ketik lagi perintah “**DESC peserta;**”, seharusnya hasilnya adalah seperti terlihat pada Gambar 2.

```

mysql> desc peserta
-> ;
+-----+-----+-----+-----+
| Field | Type  | Null | Key  | Default | Extra |
+-----+-----+-----+-----+
| nip   | int(5) | NO   |      | NULL    |        |
| nama  | varchar(30)| NO  |      | NULL    |        |
| alamat | varchar(50) | YES  |      | NULL    |        |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>

```

Gambar 2 Tampilan setelah menambah field alamat

Field/Kolom alamat sudah terbentuk. Timbul pertanyaan bagaimana menambah field **telepon** di tengah-tengah antara kolom **nama** dan **alamat**. Maka perintahnya adalah :

**ALTER TABLE peserta ADD telepon varchar(15) AFTER nama;**

Kata **AFTER** digunakan untuk membantu menambahkan kolom sesuai dengan yang diinginkan. Kata **BEFORE** tidak bisa digunakan. Hasilnya adalah sebagai berikut setelah ketik perintah ” **DESC peserta;** ”

```

mysql> desc peserta;
+-----+-----+-----+-----+
| Field | Type  | Null | Key  | Default | Extra |
+-----+-----+-----+-----+
| nip   | int(5) | NO   |      | NULL    |        |
| nama  | varchar(30)| NO  |      | NULL    |        |
| telepon | varchar(15) | YES  |      | NULL    |        |
| alamat | varchar(50) | YES  |      | NULL    |        |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>

```

Gambar 3 Tampilan setelah menambah field telepon

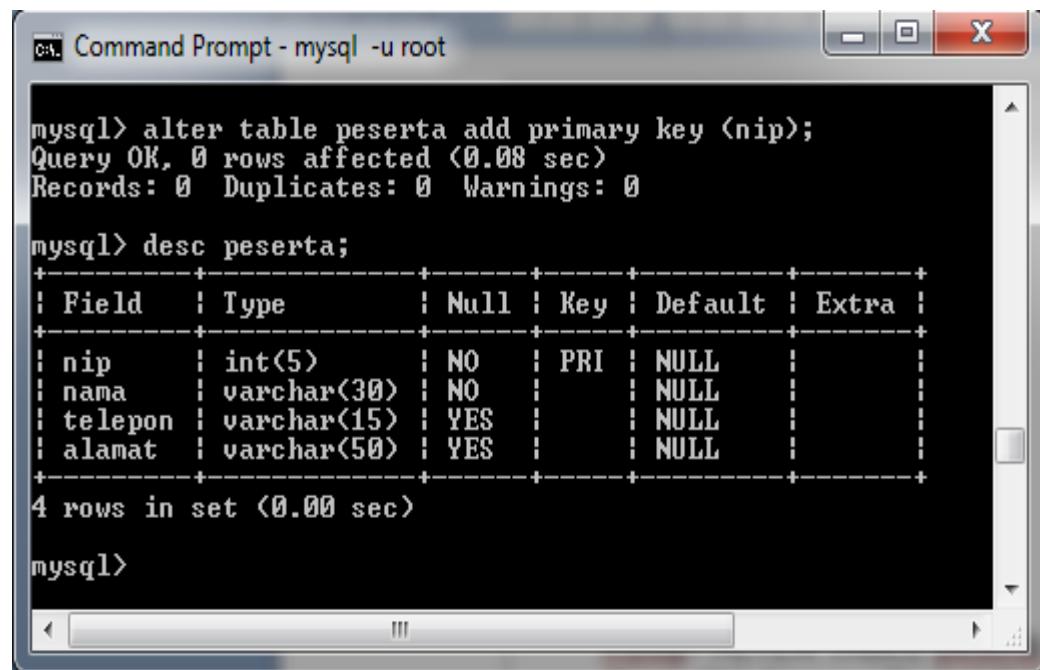
Terlihat bahwa kolom **telepon** sudah berhasil ditambahkan.

#### b. Tambah/hapus Primary Key

Untuk menambah Primary Key pada tabel, bentuk umum perintahnya adalah:

**ALTER TABLE nama\_table ADD PRIMARY KEY (nama\_kolom);**

Contoh dan hasilnya dapat dilihat pada Gambar 4.



Command Prompt - mysql -u root

```
mysql> alter table peserta add primary key (nip);
Query OK, 0 rows affected (0.08 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc peserta;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+
| nip   | int(5) | NO   | PRI  | NULL    |       |
| nama  | varchar(30)| NO  |       | NULL    |       |
| telepon | varchar(15)| YES |       | NULL    |       |
| alamat | varchar(50)| YES |       | NULL    |       |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

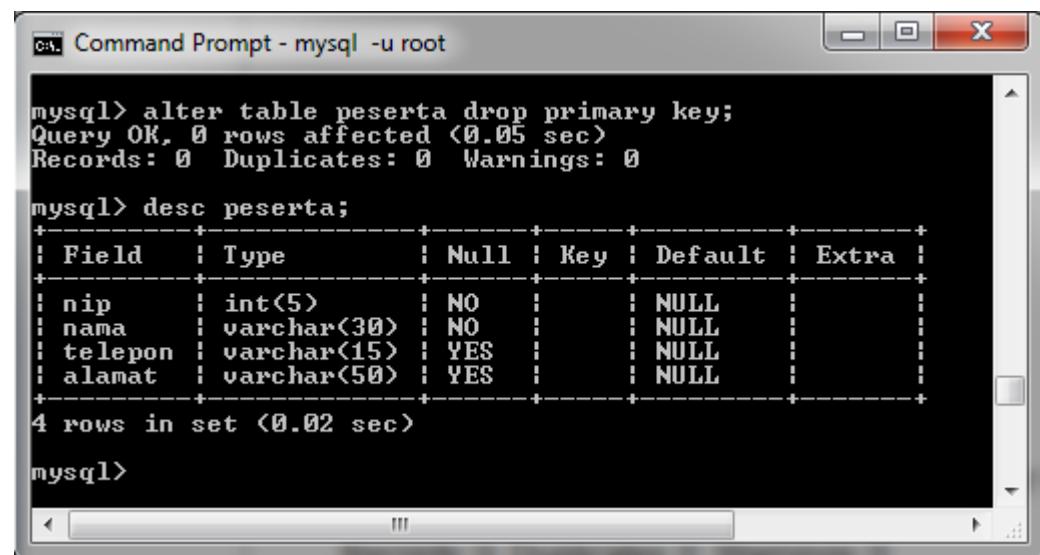
mysql>
```

Gambar 4 Tampilan setelah menambah Primary key

Untuk hapus Primary Key pada tabel, bentuk umum perintahnya adalah:

**ALTER TABLE nama\_table DROP PRIMARY KEY;**

Contoh dan hasilnya dapat dilihat pada Gambar 5



Command Prompt - mysql -u root

```
mysql> alter table peserta drop primary key;
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc peserta;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+
| nip   | int(5) | NO   |       | NULL    |       |
| nama  | varchar(30)| NO  |       | NULL    |       |
| telepon | varchar(15)| YES |       | NULL    |       |
| alamat | varchar(50)| YES |       | NULL    |       |
+-----+-----+-----+-----+-----+
4 rows in set (0.02 sec)

mysql>
```

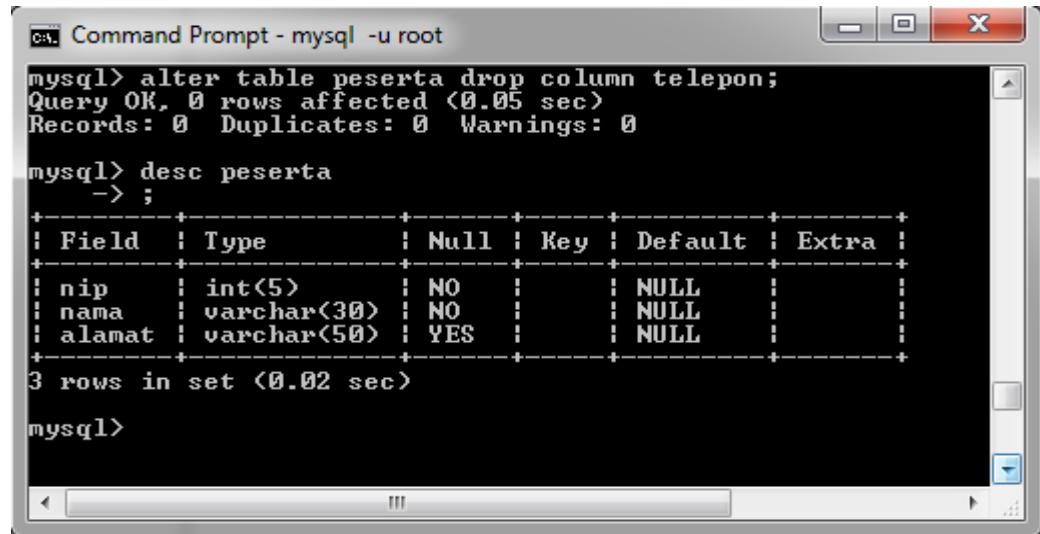
Gambar 5 Tampilan setelah menghapus *primary key*

### c. Hapus Field/Kolom

Bentuk umum perintahnya adalah sebagai berikut:

**ALTER TABLE nama\_tabel DROP COLUMN nama\_kolom;**

Misalnya ingin menghapus kolom **telepon**, maka contohnya dan hasilnya bisa dilihat pada gambar 6.



```
Command Prompt - mysql -u root
mysql> alter table peserta drop column telepon;
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc peserta
-> ;
+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| nip   | int(5) | NO   |   | NULL    |       |
| nama  | varchar(30) | NO  |   | NULL    |       |
| alamat | varchar(50) | YES  |   | NULL    |       |
+-----+-----+-----+-----+-----+
3 rows in set (0.02 sec)

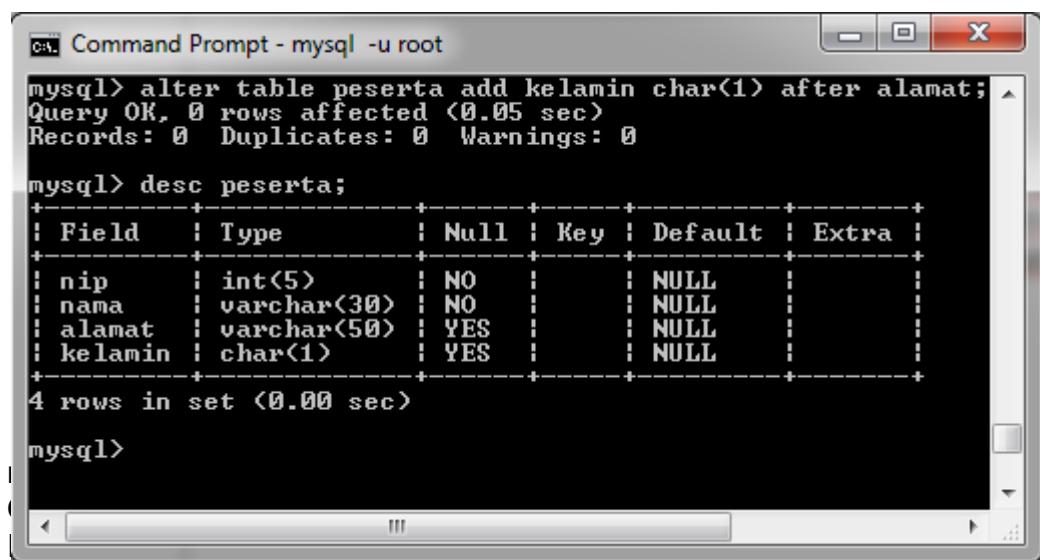
mysql>
```

Gambar 6 Tampilan setelah menghapus *field* telepon

#### d. Ubah Nama Field/Kolom

Mengubah nama kolom/field dapat menggunakan option **change**. Sebelumnya, dibuat dulu kolom baru yang bernama **kelamin** dengan jenis data “char” yang berjumlah 1 dan diletakkan setelah kolom **alamat**.

Contoh dan hasilnya dapat dilihat pada Gambar 7



```
Command Prompt - mysql -u root
mysql> alter table peserta add kelamin char(1) after alamat;
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc peserta;
+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| nip   | int(5) | NO   |   | NULL    |       |
| nama  | varchar(30) | NO  |   | NULL    |       |
| alamat | varchar(50) | YES  |   | NULL    |       |
| kelamin | char(1) | YES  |   | NULL    |       |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

Gambar 7 Tampilan setelah menambah field kelamin

Setelah berhasil, kita rubah kolom **kelamin** tersebut menjadi **sex**.

Contoh perintahnya dan hasilnya dapat dilihat pada Gambar 8.

```

Command Prompt - mysql -u root

mysql> alter table peserta change kelamin sex char(1);
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc peserta;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| nip   | int<5> | NO   |   | NULL    |       |
| nama  | varchar<30> | NO  |   | NULL    |       |
| alamat | varchar<50> | YES |   | NULL    |       |
| sex   | char<1>  | YES |   | NULL    |       |
+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)

mysql>

```

Gambar 8 Tampilan setelah merubah *field* kelamin menjadi sex

#### e. Ubah Type Data Field/Kolom

Pada materi sebelumnya sudah dicobakan membuat kolom/field baru bernama sex. Berikutnya ingin membuat kolom/field tersebut hanya terisi oleh karakter “L” untuk laki-laki dan “P” untuk perempuan. Untuk itu perlu merubah dari type data **char(1)** menjadi **enum** ('L','P'). Perintah yang digunakan adalah **MODIFY**.

Contoh dan hasilnya dapat dilihat pada Gambar 9

```

Command Prompt - mysql -u root

mysql> alter table peserta modify sex enum('L','P') not null;
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc peserta;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| nip   | int<5> | NO   |   | NULL    |       |
| nama  | varchar<30> | NO  |   | NULL    |       |
| alamat | varchar<50> | YES |   | NULL    |       |
| sex   | enum('L','P') | NO  |   | NULL    |       |
+-----+-----+-----+-----+-----+
4 rows in set (0.02 sec)

mysql>

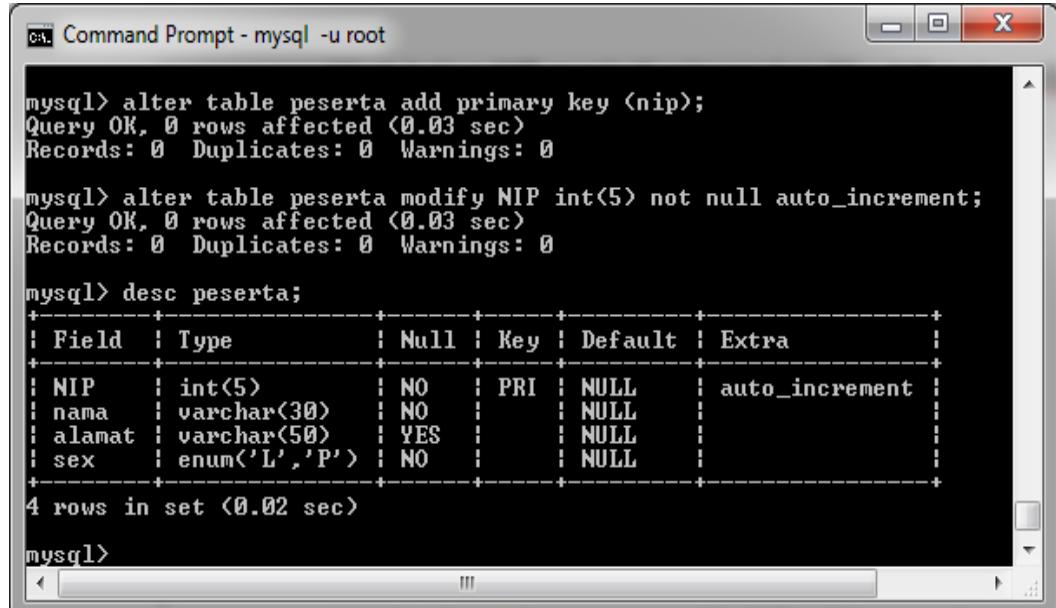
```

Gambar 9 Tampilan setelah mengubah field sex

Skenario selanjutnya adalah: merubah NIP menjadi **auto\_increment**. Hal tersebut bisa dilakukan dengan option **modify** dengan catatan yang di-**auto\_increment**-kan harus sebagai **primary key**. Langkahnya ada 2 tahap, yaitu:

- Ubah kolom NIP menjadi primary key (jika NIP belum dijadikan Primary Key)
- Seting kolom NIP menjadi auto\_increment

Contoh dan hasilnya dapat dilihat pada Gambar 10



```
mysql> alter table peserta add primary key (nip);
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> alter table peserta modify NIP int<5> not null auto_increment;
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0

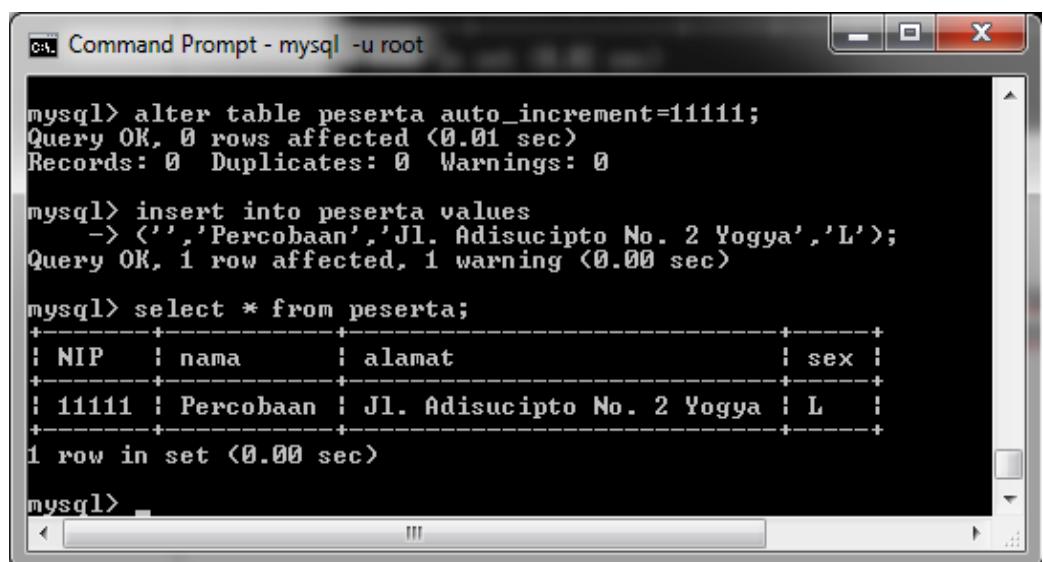
mysql> desc peserta;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key  | Default | Extra       |
+-----+-----+-----+-----+-----+-----+
| NIP   | int<5> | NO   | PRI   | NULL    | auto_increment |
| nama  | varchar(30) | NO   |        | NULL    |              |
| alamat | varchar(50) | YES  |        | NULL    |              |
| sex   | enum('L','P') | NO   |        | NULL    |              |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.02 sec)

mysql>
```

Gambar 10 Tampilan setelah mengubah *field* NIP

Misalnya ingin men-SET auto\_increment menjadi 11111 sebagai angka awal, karena sudah melakukannya set auto\_increment sebelumnya, langkahnya adalah sebagai berikut :

Contoh dan hasilnya dapat dilihat pada Gambar 11



```
mysql> alter table peserta auto_increment=11111;
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> insert into peserta values
-> ('','Percobaan','Jl. Adisucipto No. 2 Yogyga','L');
Query OK, 1 row affected, 1 warning (0.00 sec)

mysql> select * from peserta;
+-----+-----+-----+-----+
| NIP   | nama  | alamat          | sex  |
+-----+-----+-----+-----+
| 11111 | Percobaan | Jl. Adisucipto No. 2 Yogyga | L   |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

Gambar 11 Tampilan setelah menset *field* nip dengan 11111

Karena sudah auto\_increment, maka kolom NIP bisa dikosongkan dan akan terisi otomatis sesuai angka yang ditetapkan/SET sebelumnya.

## f. Ganti Nama Tabel

Mengganti nama tabel bisa memakai perintah **alter table**. Bentuk umum perintahnya adalah:

**ALTER TABLE nama\_table\_lama RENAME TO nama\_table\_baru;**

**Alter Table peserta rename to tbcoba;**

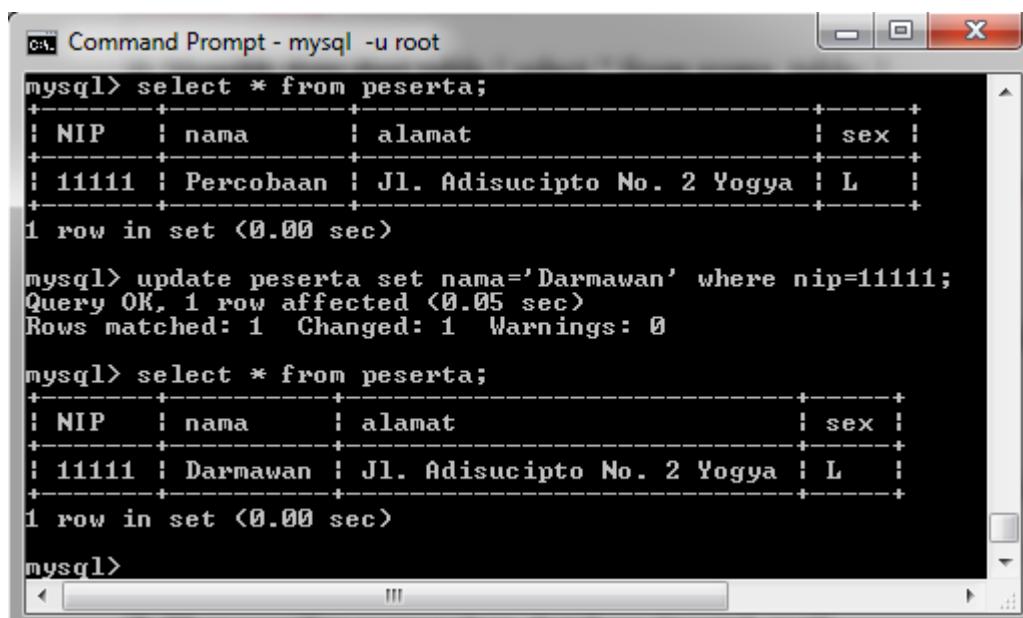
## 2. Perintah Update

Perintah ini digunakan untuk merubah isi data yang pernah dimasukkan pada record.

Bentuk umum perintahnya adalah:

**update nama\_table set field = 'value' where field = 'value';**

Contoh dan hasil sebelum dan sesudahnya dapat dilihat pada Gambar 12



The screenshot shows a Windows Command Prompt window titled "Command Prompt - mysql -u root". It displays the following MySQL session:

```
mysql> select * from peserta;
+-----+-----+-----+
| NIP  | nama   | alamat           | sex |
+-----+-----+-----+
| 11111 | Percobaan | Jl. Adisucipto No. 2 Yogyka | L   |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> update peserta set nama='Darmawan' where nip=11111;
Query OK, 1 row affected (0.05 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from peserta;
+-----+-----+-----+
| NIP  | nama   | alamat           | sex |
+-----+-----+-----+
| 11111 | Darmawan | Jl. Adisucipto No. 2 Yogyka | L   |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

Gambar 12 Tampilan setelah mengupdate field nama

## 3. Perintah Delete

Perintah ini digunakan untuk menghapus record pada table yang ditunjuk. Bentuk umum perintahnya adalah sebagai berikut:

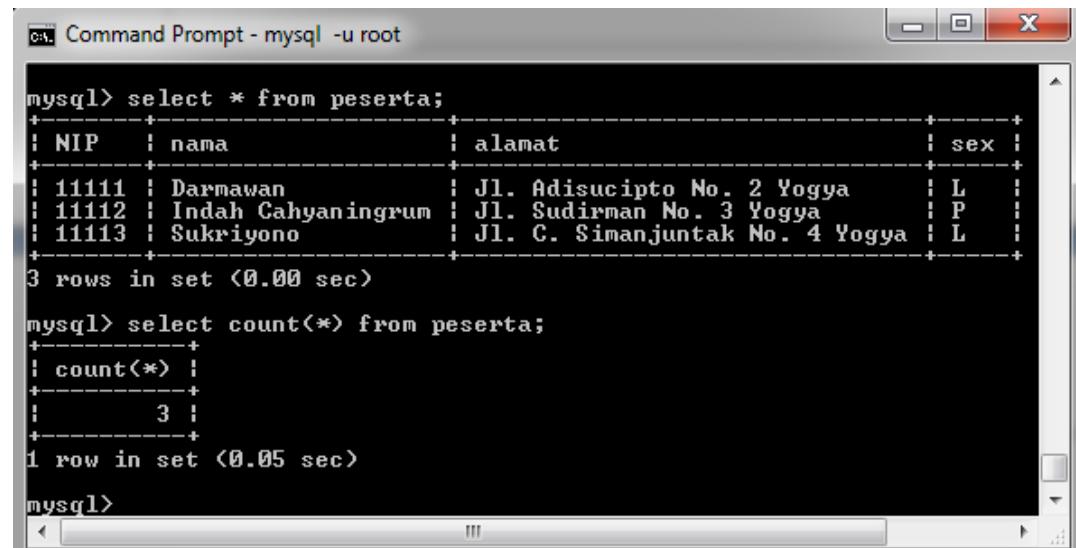
**DELETE FROM table-name WHERE condition(s)**

*Contoh :*

**mysql> DELETE FROM peserta WHERE nip=11111;**

#### 4. Perintah Count

Perintah ini digunakan untuk menghitung jumlah record.  
Contoh dan hasilnya dapat dilihat pada Gambar 13



```
mysql> select * from peserta;
+-----+-----+-----+
| NIP  | nama | alamat          | sex |
+-----+-----+-----+
| 11111 | Darmawan | Jl. Adisucipto No. 2 Yogyga | L |
| 11112 | Indah Cahyaningrum | Jl. Sudirman No. 3 Yogyga | P |
| 11113 | Sukriyono | Jl. C. Simanjuntak No. 4 Yogyga | L |
+-----+-----+-----+
3 rows in set (0.00 sec)

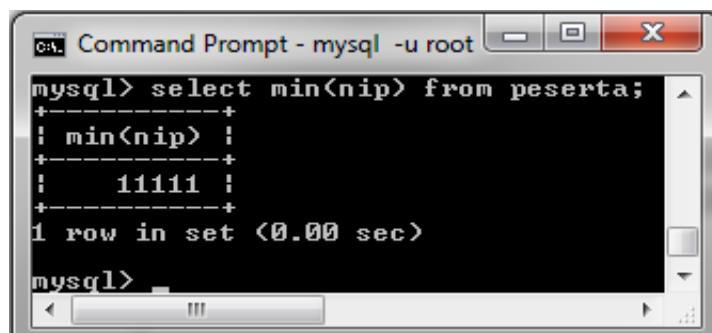
mysql> select count(*) from peserta;
+-----+
| count(*) |
+-----+
|      3   |
+-----+
1 row in set (0.05 sec)

mysql>
```

Gambar 13 Tampilan setelah perintah count

#### 5. Perintah Min

Perintah ini digunakan untuk mencari nilai terkecil.  
Contoh dan hasilnya dapat dilihat pada Gambar 14



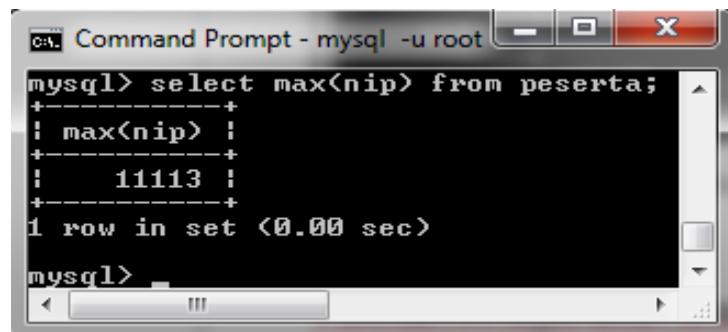
```
mysql> select min(nip) from peserta;
+-----+
| min(nip) |
+-----+
|    11111 |
+-----+
1 row in set (0.00 sec)

mysql>
```

Gambar 14 Tampilan setelah perintah min

#### 6. Perintah Max

Perintah ini digunakan untuk mencari nilai terbesar.  
Berikut contoh dan hasilnya:



```
mysql> select max(nip) from peserta;
+-----+
| max(nip) |
+-----+
|    11113 |
+-----+
1 row in set (0.00 sec)

mysql>
```

Gambar 15 Tampilan setelah perintah max

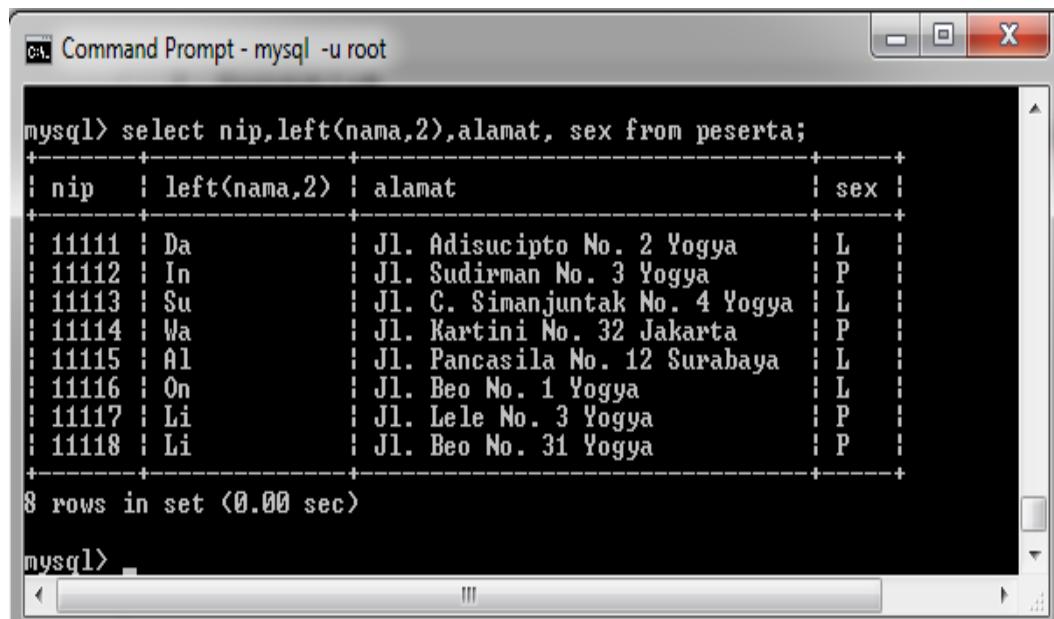
7. Perintah Left

Perintah ini dapat digunakan untuk mengambil bagian kiri dari data tertentu sejumlah yang telah ditentukan.

Bentuk umum perintah ini adalah:

**Left(string,n)**

Contoh dan hasilnya dapat dilihat pada Gambar 16 dan gambar 17.:

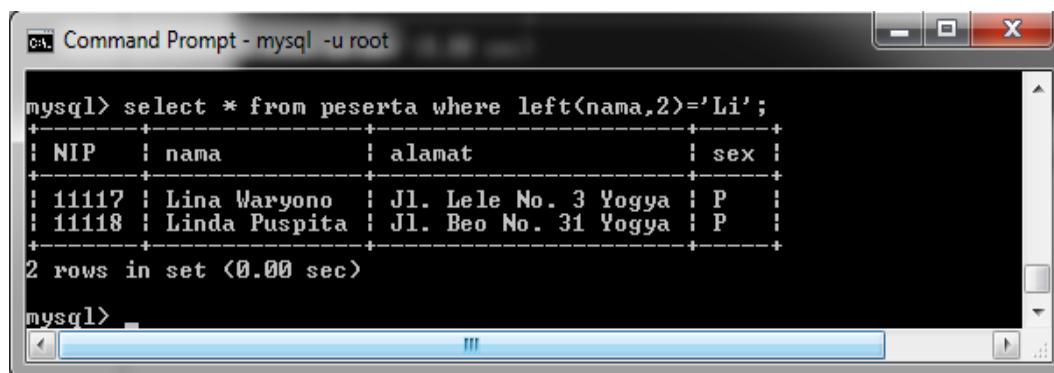


Command Prompt - mysql -u root

```
mysql> select nip, left(nama,2), alamat, sex from peserta;
+-----+-----+-----+-----+
| nip  | left(nama,2) | alamat          | sex |
+-----+-----+-----+-----+
| 11111 | Da          | Jl. Adisucipto No. 2 Yogyga | L   |
| 11112 | In          | Jl. Sudirman No. 3 Yogyga  | P   |
| 11113 | Su          | Jl. C. Simanjuntak No. 4 Yogyga | L   |
| 11114 | Wa          | Jl. Kartini No. 32 Jakarta | P   |
| 11115 | Al          | Jl. Pancasila No. 12 Surabaya | L   |
| 11116 | On          | Jl. Beo No. 1 Yogyga       | L   |
| 11117 | Li          | Jl. Lele No. 3 Yogyga     | P   |
| 11118 | Li          | Jl. Beo No. 31 Yogyga     | P   |
+-----+-----+-----+-----+
8 rows in set (0.00 sec)

mysql>
```

Gambar 16 Tampilan setelah perintah `left(nama,2)`



Command Prompt - mysql -u root

```
mysql> select * from peserta where left(nama,2)='Li';
+-----+-----+-----+-----+
| NIP  | nama        | alamat          | sex |
+-----+-----+-----+-----+
| 11117 | Lina Waryono | Jl. Lele No. 3 Yogyga | P   |
| 11118 | Linda Puspita | Jl. Beo No. 31 Yogyga | P   |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

Gambar 17 Tampilan setelah perintah `left left(nama,2)='Li'`

8. Perintah Right

Perintah ini dapat digunakan untuk mengambil bagian kanan dari data tertentu sejumlah yang telah ditentukan.

Bentuk umum perintah ini adalah:

**Right(string,n)**

Contoh dan hasilnya dapat dilihat pada Gambar 18 dan gambar 19.

```
mysql> select nip,right(nama,3),alamat, sex from peserta;
+-----+-----+-----+-----+
| nip  | right(nama,3) | alamat          | sex |
+-----+-----+-----+-----+
| 11111 | wan        | Jl. Adisucipto No. 2 Yogya | L   |
| 11112 | rum        | Jl. Sudirman No. 3 Yogya  | P   |
| 11113 | ono        | Jl. C. Simanjuntak No. 4 Yogya | L   |
| 11114 | ria        | Jl. Kartini No. 32 Jakarta | P   |
| 11115 | kma        | Jl. Pancasila No. 12 Surabaya | L   |
| 11116 | ana        | Jl. Beo No. 1 Yogya       | L   |
| 11117 | ono        | Jl. Lele No. 3 Yogya      | P   |
| 11118 | ita        | Jl. Beo No. 31 Yogya     | P   |
+-----+-----+-----+-----+
8 rows in set (0.00 sec)

mysql>
```

Gambar 18 Tampilan setelah perintah right(nama,3)

```
mysql> select * from peserta where right(nama,3)='ono';
+-----+-----+-----+-----+
| NIP  | nama       | alamat          | sex |
+-----+-----+-----+-----+
| 11113 | Sukriyono  | Jl. C. Simanjuntak No. 4 Yogya | L   |
| 11117 | Lina Waryono | Jl. Lele No. 3 Yogya      | P   |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

Gambar 19 Tampilan setelah perintah right(nama,3)='ono'

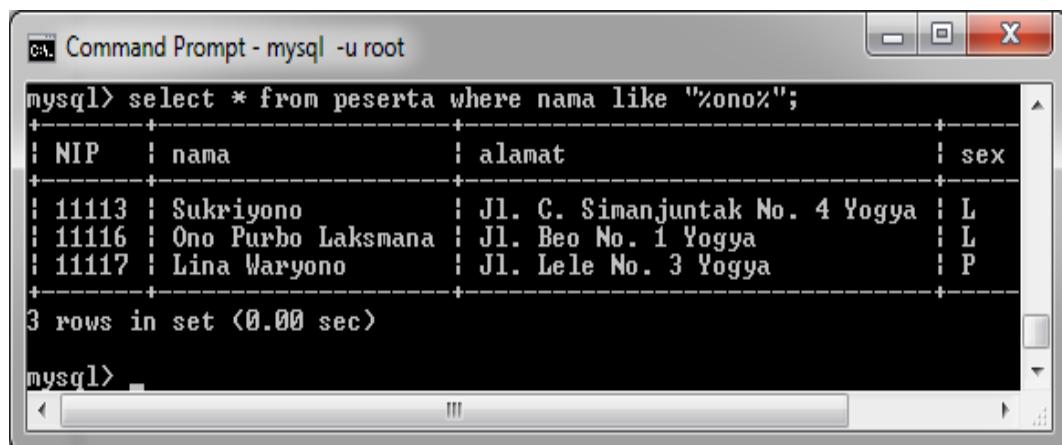
## 9. Perintah Like

Perintah ini dapat digunakan untuk mengambil bagian sub data dari data tertentu.

Bentuk umum perintah ini adalah:

**Like(%string%)**

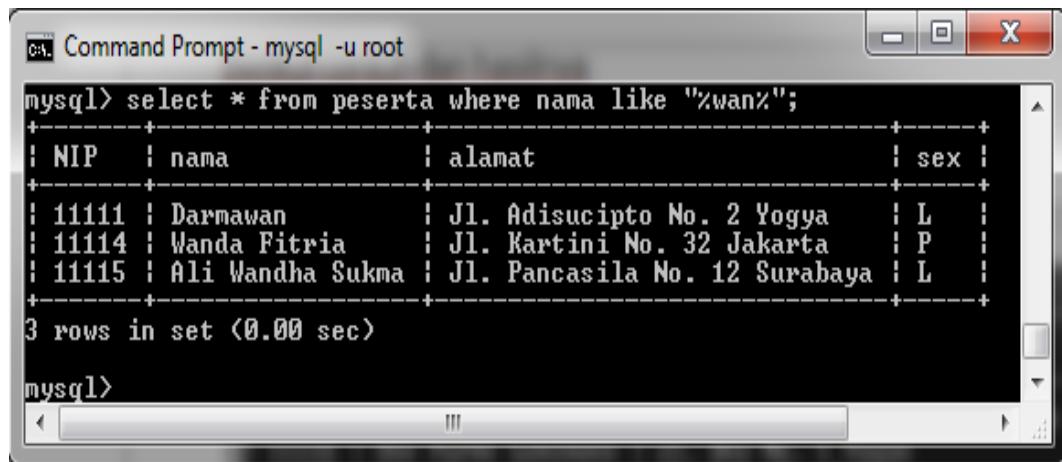
Contoh dan hasilnya dapat dilihat pada Gambar 20 dan Gambar 21.



```
c:\ Command Prompt - mysql -u root
mysql> select * from peserta where nama like "%ono%";
+-----+-----+-----+
| NIP | nama | alamat | sex |
+-----+-----+-----+
| 11113 | Sukriyono | Jl. C. Simanjuntak No. 4 Yogya | L |
| 11116 | Ono Purbo Laksmana | Jl. Beo No. 1 Yogya | L |
| 11117 | Lina Waryono | Jl. Lele No. 3 Yogya | P |
+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

Gambar 20 Tampilan setelah perintah like%ono%



```
c:\ Command Prompt - mysql -u root
mysql> select * from peserta where nama like "%wan%";
+-----+-----+-----+
| NIP | nama | alamat | sex |
+-----+-----+-----+
| 11111 | Darmawan | Jl. Adisucipto No. 2 Yogya | L |
| 11114 | Wanda Fitria | Jl. Kartini No. 32 Jakarta | P |
| 11115 | Ali Wandha Sukma | Jl. Pancasila No. 12 Surabaya | L |
+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

Gambar 21 Tampilan setelah perintah like%wan%

#### 10. Perintah Show Table

Perintah ini digunakan untuk melihat table yang ada atau yang pernah dibuat pada database yang sedang aktif saat ini;

Contoh dan hasilnya dapat dilihat pada Gambar 22



```
c:\ Command Prompt - m...
mysql> show tables;
+-----+
| Tables_in_dbjube |
+-----+
| barang           |
| peserta          |
+-----+
2 rows in set (0.00 sec)

mysql>
```

Gambar 22 Tampilan setelah perintah Show Table

## B. Praktek MySQL Menggunakan Command Line Lanjutan

Pada pertemuan ini akan melanjutkan proses penggunaan command line untuk menyelesaikan basis data jual beli (DBJUBE).

### 1. Membuat table Lanjutan

Struktur perintah untuk membuat tabel adalah sebagai berikut :

```
CREATE TABLE nama_tabel(  
    kolom1 type_data(ukuran),  
    kolom2 type_data(ukuran),  
    kolom3 type_data(ukuran),  
    kolom4 type_data(ukuran),  
    ...  
    kolom_n type_data(ukuran)  
);
```

#### a. Membuat Tabel Jual

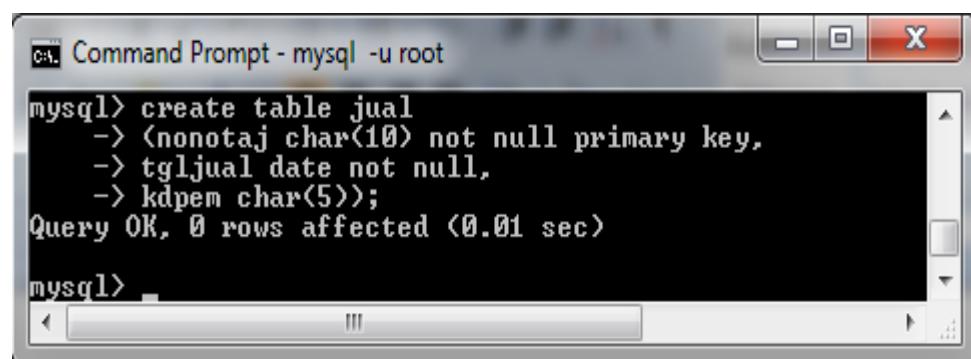
Pada contoh berikut akan dibuat tabel Jual dengan struktur tabel tersebut terdiri dari beberapa kolom, diantaranya (Nonotaj, Tgljual, Kdpem).

Perintah yang harus dibuat adalah sebagai berikut :

```
CREATE TABLE Jual (  
    nonotaj char(10) not null primary key,  
    tgljual date not null,  
    kdpmem char(5));
```

Jika perintah di atas berhasil, maka akan dicetak pesan Query Ok , sedangkan jika ditemukan kesalahan dalam perintah di atas, maka akan dicetak pesan kesalahan berserta dengan petunjuk lokasi kesalahannya.

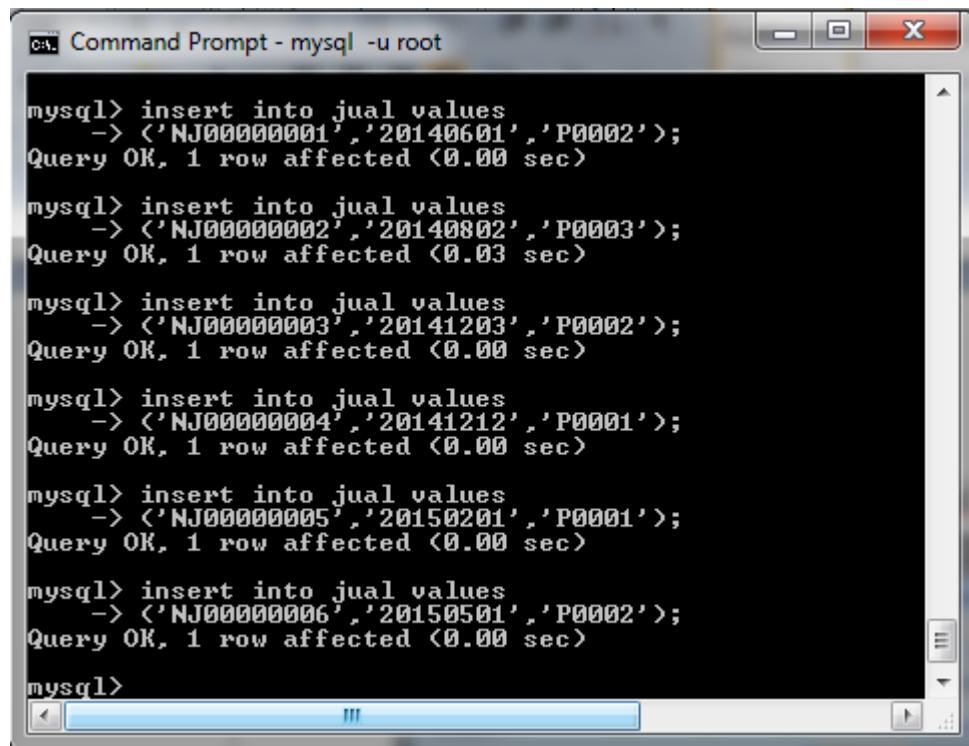
Contoh dan hasilnya dapat dilihat pada Gambar 23.



```
mysql> create table jual  
    -> (nonotaj char(10) not null primary key,  
    -> tgljual date not null,  
    -> kdpmem char(5));  
Query OK, 0 rows affected (0.01 sec)
```

Gambar 23 Tampilan setelah perintah like%ono%

Untuk mengisi data pada table jual dapat dilakukan seperti terlihat pada Gambar 24



```
mysql> insert into jual values
-> ('NJ00000001','20140601','P0002');
Query OK, 1 row affected (0.00 sec)

mysql> insert into jual values
-> ('NJ00000002','20140802','P0003');
Query OK, 1 row affected (0.03 sec)

mysql> insert into jual values
-> ('NJ00000003','20141203','P0002');
Query OK, 1 row affected (0.00 sec)

mysql> insert into jual values
-> ('NJ00000004','20141212','P0001');
Query OK, 1 row affected (0.00 sec)

mysql> insert into jual values
-> ('NJ00000005','20150201','P0001');
Query OK, 1 row affected (0.00 sec)

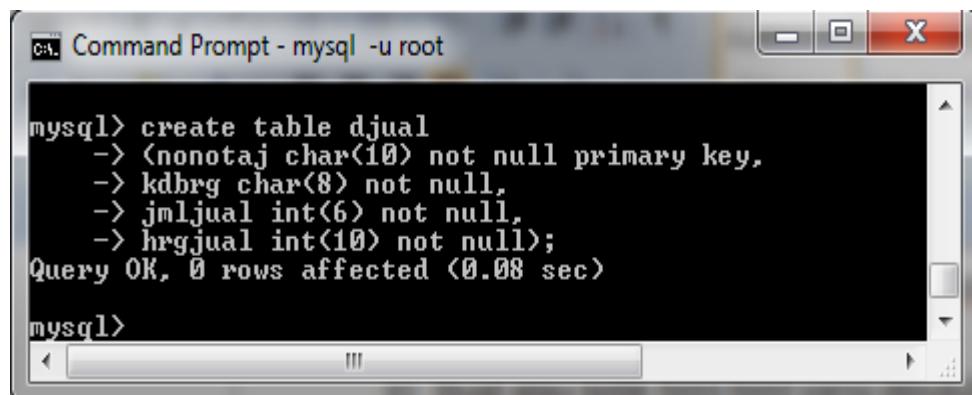
mysql> insert into jual values
-> ('NJ00000006','20150501','P0002');
Query OK, 1 row affected (0.00 sec)

mysql>
```

Gambar 24 Tampilan proses mengisi tabel jual

b. Membuat Tabel Detail Jual

Contoh dan hasilnya dapat dilihat pada gambar 25

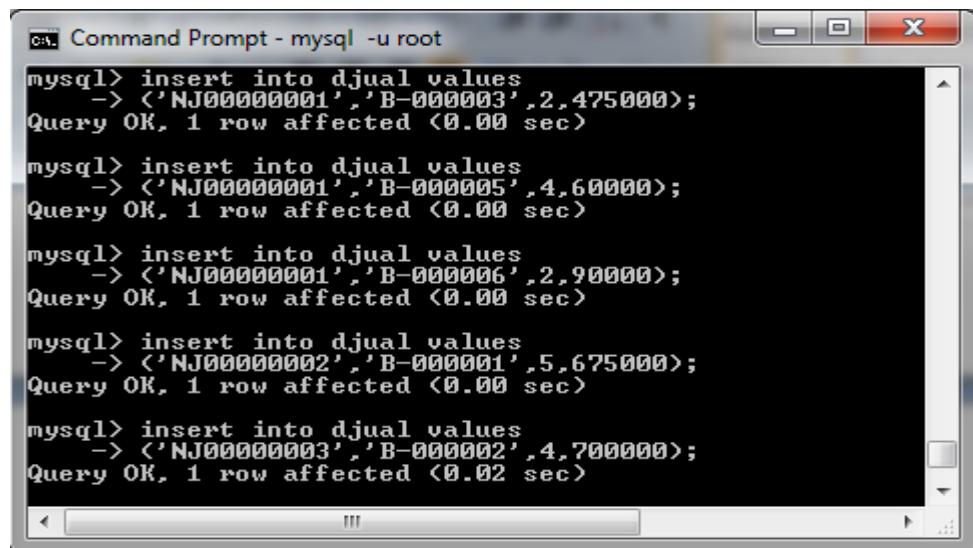


```
mysql> create table djual
-> (nonotaj char(10) not null primary key,
-> kdbrg char(8) not null,
-> jmljual int(6) not null,
-> hrgjual int(10) not null);
Query OK, 0 rows affected (0.08 sec)

mysql>
```

Gambar 25 Tampilan perintah membuat tabel detail jual

Untuk mengisi data pada table detail jual dapat dilakukan seperti terlihat pada Gambar 26.



```
ca. Command Prompt - mysql -u root
mysql> insert into djual values
-> ('NJ00000001','B-000003',2,475000);
Query OK, 1 row affected (0.00 sec)

mysql> insert into djual values
-> ('NJ00000001','B-000005',4,60000);
Query OK, 1 row affected (0.00 sec)

mysql> insert into djual values
-> ('NJ00000001','B-000006',2,90000);
Query OK, 1 row affected (0.00 sec)

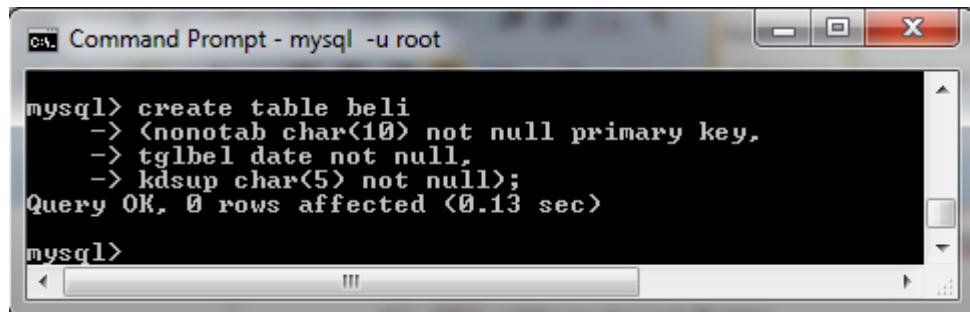
mysql> insert into djual values
-> ('NJ00000002','B-000001',5,675000);
Query OK, 1 row affected (0.00 sec)

mysql> insert into djual values
-> ('NJ00000003','B-000002',4,700000);
Query OK, 1 row affected (0.02 sec)
```

Gambar 26 Tampilan menisci tabel detail jual

c. Membuat Tabel Beli

Contoh dan hasilnya dapat dilihat pada Gambar 27

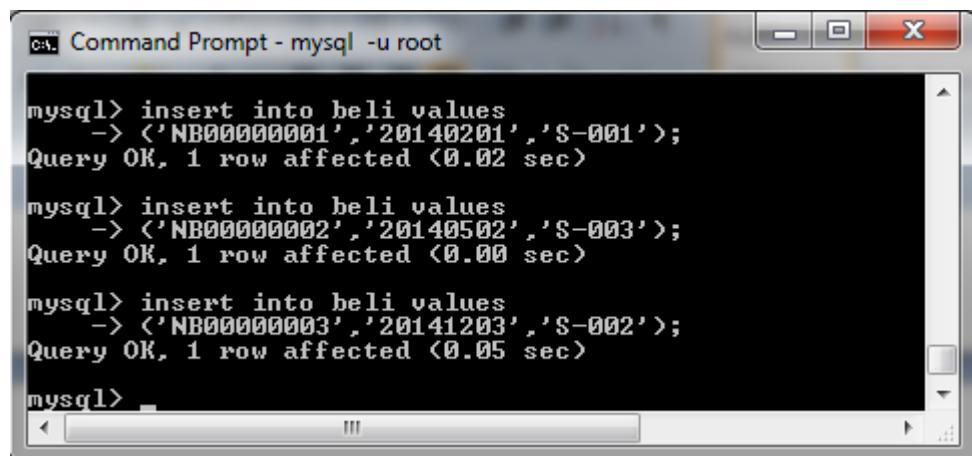


```
ca. Command Prompt - mysql -u root
mysql> create table beli
-> (nonatab char(10) not null primary key,
-> tglbel date not null,
-> kdsup char(5) not null);
Query OK, 0 rows affected (0.13 sec)

mysql>
```

Gambar 27 Tampilan perintah membuat tabel beli

Untuk mengisi data pada table beli dapat dilakukan seperti terlihat pada Gambar 28.



```
ca. Command Prompt - mysql -u root
mysql> insert into beli values
-> ('NB00000001','20140201','S-001');
Query OK, 1 row affected (0.02 sec)

mysql> insert into beli values
-> ('NB00000002','20140502','S-003');
Query OK, 1 row affected (0.00 sec)

mysql> insert into beli values
-> ('NB00000003','20141203','S-002');
Query OK, 1 row affected (0.05 sec)

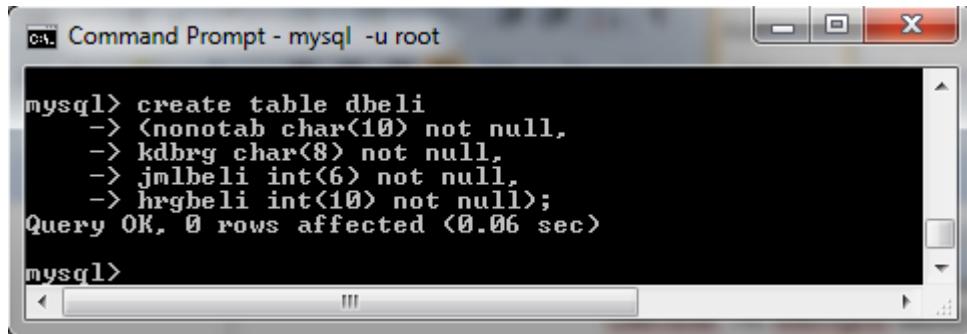
mysql>
```

Gambar 28 Tampilan proses mengisi tabel beli

Silahkan dilanjutkan untuk data yang lain

d. Membuat Tabel Detail Beli

Contoh dan hasilnya adalah sebagai berikut:

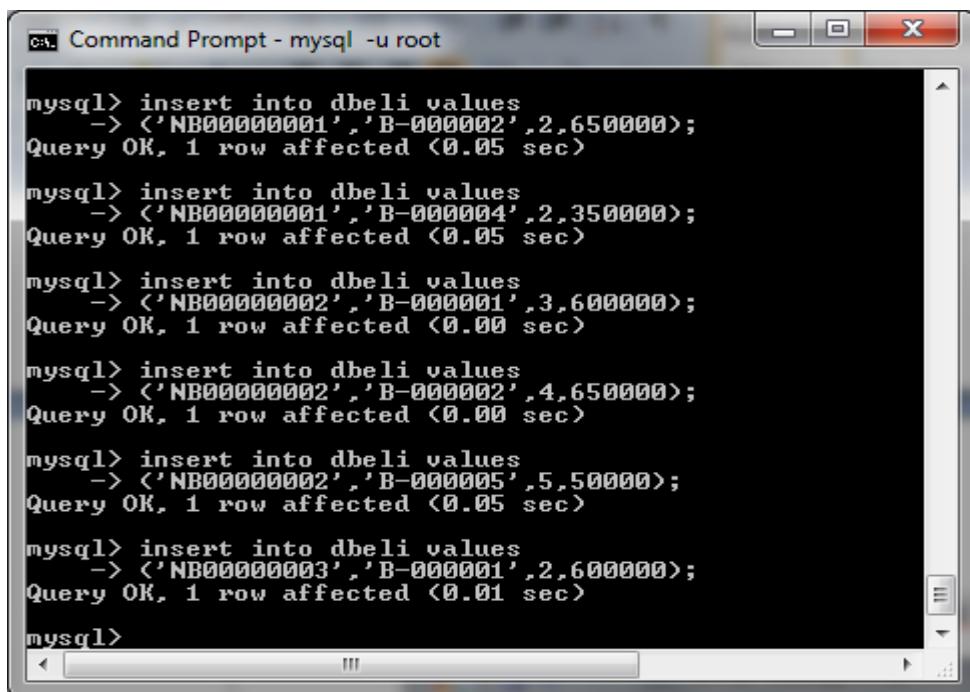


```
mysql> create table dbeli
->      nonotab char(10) not null,
->      kdbrg char(8) not null,
->      jmlbeli int(6) not null,
->      hrgbeli int(10) not null;
Query OK, 0 rows affected <0.06 sec>

mysql>
```

Gambar 29 Tampilan untuk membuat tabel detail beli

Untuk mengisi data pada table detail beli dapat dilakukan seperti terlihat pada Gambar 30.



```
mysql> insert into dbeli values
-> ('NB00000001','B-000002',2,650000);
Query OK, 1 row affected <0.05 sec>

mysql> insert into dbeli values
-> ('NB00000001','B-000004',2,350000);
Query OK, 1 row affected <0.05 sec>

mysql> insert into dbeli values
-> ('NB00000002','B-000001',3,600000);
Query OK, 1 row affected <0.00 sec>

mysql> insert into dbeli values
-> ('NB00000002','B-000002',4,650000);
Query OK, 1 row affected <0.00 sec>

mysql> insert into dbeli values
-> ('NB00000002','B-000005',5,500000);
Query OK, 1 row affected <0.05 sec>

mysql> insert into dbeli values
-> ('NB00000003','B-000001',2,600000);
Query OK, 1 row affected <0.01 sec>

mysql>
```

Gambar 30 Tampilan untuk mengisi tabel detail beli

Silahkan dilanjutkan untuk data yang lain

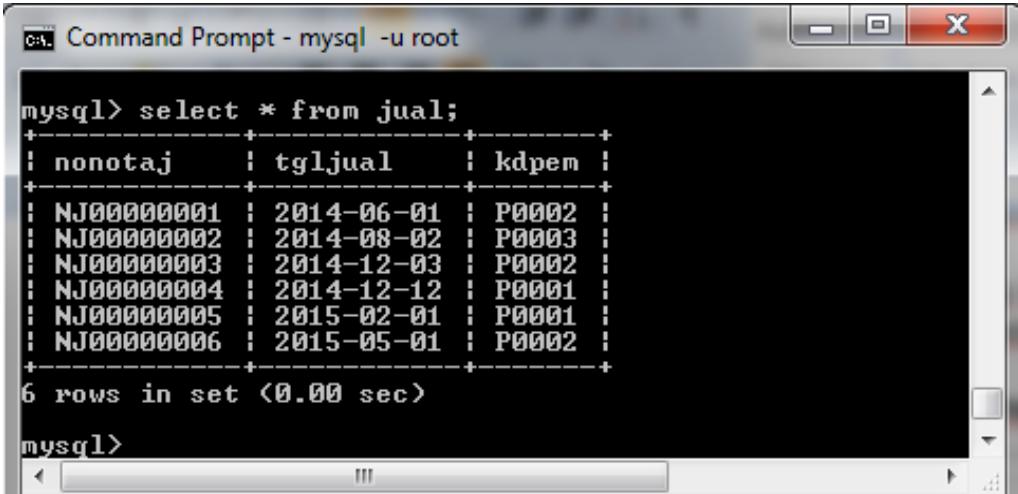
## 2. Menampilkan data atau record table Lanjutan

Untuk melihat data dari sebuah tabel, maka kita gunakan perintah select , yang struktur perintahnya adalah sebagai berikut :

**SELECT kolom1,kolom2,kolom\_n FROM nama\_tabel**

- a. Menampilkan data dari satu table

Tabel yang digunakan adalah table jual, bentuk contoh dan hasilnya dapat dilihat pada Gambar 31.



```
mysql> select * from jual;
+-----+-----+-----+
| nonotaj | tgljual | kd pem |
+-----+-----+-----+
| NJ00000001 | 2014-06-01 | P0002 |
| NJ00000002 | 2014-08-02 | P0003 |
| NJ00000003 | 2014-12-03 | P0002 |
| NJ00000004 | 2014-12-12 | P0001 |
| NJ00000005 | 2015-02-01 | P0001 |
| NJ00000006 | 2015-05-01 | P0002 |
+-----+-----+-----+
6 rows in set (0.00 sec)

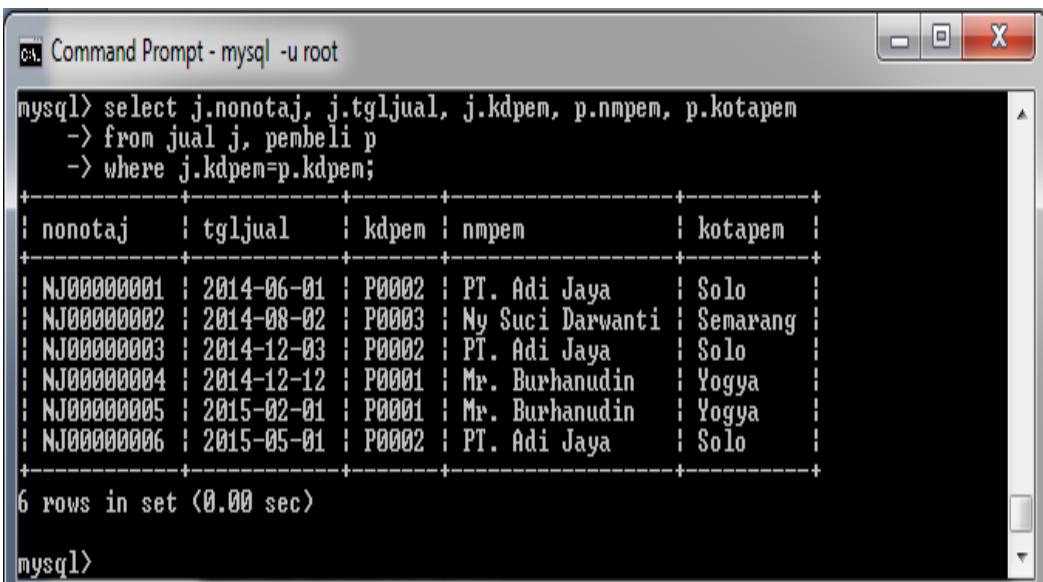
mysql>
```

Gambar 31 Hasil perintah menampilkan data dari satu tabel

- b. Menampilkan data dari dua table

Tabel yang digunakan adalah table jual dan table pembeli, bentuk contoh dan hasilnya dapat dilihat pada Gambar 32.

**Select j.nonotaj, j.tgljual, j.kd pem, p.nmpem, p.kotapem  
From jual j, pembeli p  
Where j.kd pem=p.kd pem;**

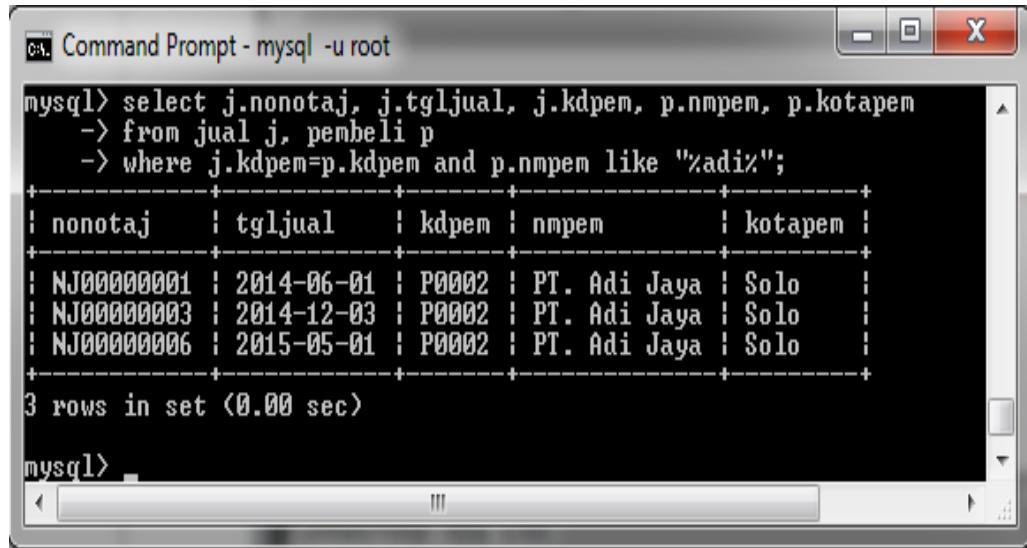


```
mysql> select j.nonotaj, j.tgljual, j.kd pem, p.nmpem, p.kotapem
    -> from jual j, pembeli p
    -> where j.kd pem=p.kd pem;
+-----+-----+-----+-----+-----+
| nonotaj | tgljual | kd pem | nmpem | kotapem |
+-----+-----+-----+-----+-----+
| NJ00000001 | 2014-06-01 | P0002 | PT. Adi Jaya | Solo |
| NJ00000002 | 2014-08-02 | P0003 | Ny Suci Darwanti | Semarang |
| NJ00000003 | 2014-12-03 | P0002 | PT. Adi Jaya | Solo |
| NJ00000004 | 2014-12-12 | P0001 | Mr. Burhanudin | Yogya |
| NJ00000005 | 2015-02-01 | P0001 | Mr. Burhanudin | Yogya |
| NJ00000006 | 2015-05-01 | P0002 | PT. Adi Jaya | Solo |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

Gambar 32 Hasil perintah menampilkan data dari dua tabel

```
Select j.nonotaj, j.tgljual, j.kdpem, p.nmpem, p.kotapem
From jual j, pembeli p
Where j.kdpem=p.kdpem and p.nmpem like "%adi%";
```



```
mysql> select j.nonotaj, j.tgljual, j.kdpem, p.nmpem, p.kotapem
-> from jual j, pembeli p
-> where j.kdpem=p.kdpem and p.nmpem like "%adi%";
+-----+-----+-----+-----+-----+
| nonotaj | tgljual | kdpem | nmpem | kotapem |
+-----+-----+-----+-----+-----+
| NJ00000001 | 2014-06-01 | P0002 | PT. Adi Jaya | Solo |
| NJ00000003 | 2014-12-03 | P0002 | PT. Adi Jaya | Solo |
| NJ00000006 | 2015-05-01 | P0002 | PT. Adi Jaya | Solo |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

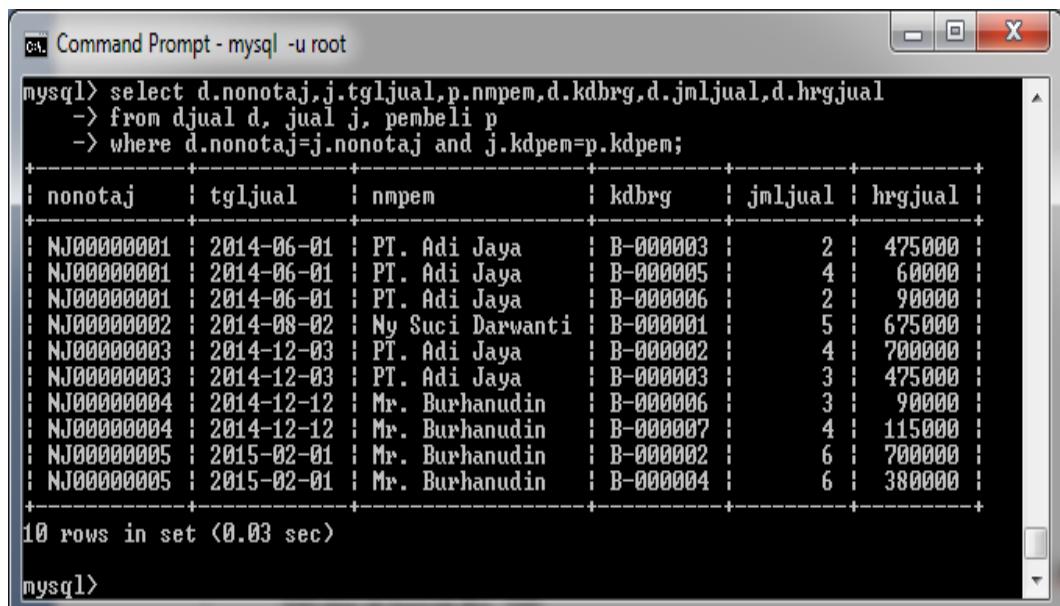
mysql>
```

Gambar 33 Hasil perintah menampilkan data dari dua tabel dengan like

c. Menampilkan data dari tiga table

Tabel yang digunakan adalah table djual, table jual dan tabel pembeli bentuk contoh dan hasilnya dapat dilihat pada Gambar 34.

```
Select d.nonotaj, j.tgljual, p.nmpem, d.kdbrg, d.jmljual, d.hrgjual
From djual d, jual j, pembeli p
Where d.nonotaj=j.nonotaj and j.kdpem=p.kdpem;
```



```
mysql> select d.nonotaj,j.tgljual,p.nmpem,d.kdbrg,d.jmljual,d.hrgjual
-> from djual d, jual j, pembeli p
-> where d.nonotaj=j.nonotaj and j.kdpem=p.kdpem;
+-----+-----+-----+-----+-----+-----+
| nonotaj | tgljual | nmpem | kdbrg | jmljual | hrgjual |
+-----+-----+-----+-----+-----+-----+
| NJ00000001 | 2014-06-01 | PT. Adi Jaya | B-00003 | 2 | 475000 |
| NJ00000001 | 2014-06-01 | PT. Adi Jaya | B-00005 | 4 | 60000 |
| NJ00000001 | 2014-06-01 | PT. Adi Jaya | B-00006 | 2 | 90000 |
| NJ00000002 | 2014-08-02 | Ny Suci Darwanti | B-00001 | 5 | 675000 |
| NJ00000003 | 2014-12-03 | PT. Adi Jaya | B-00002 | 4 | 700000 |
| NJ00000003 | 2014-12-03 | PT. Adi Jaya | B-00003 | 3 | 475000 |
| NJ00000004 | 2014-12-12 | Mr. Burhanudin | B-00006 | 3 | 90000 |
| NJ00000004 | 2014-12-12 | Mr. Burhanudin | B-00007 | 4 | 115000 |
| NJ00000005 | 2015-02-01 | Mr. Burhanudin | B-00002 | 6 | 700000 |
| NJ00000005 | 2015-02-01 | Mr. Burhanudin | B-00004 | 6 | 380000 |
+-----+-----+-----+-----+-----+-----+
10 rows in set (0.03 sec)

mysql>
```

Gambar 34 Hasil perintah menampilkan data dari tiga tabel

- d. Menampilkan data dari empat tabel

Tabel yang digunakan adalah tabel djual, tabel barang, tabel jual dan tabel pembeli bentuk contoh dan hasilnya dapat dilihat pada Gambar 35

```
Select d.nonotaj, j.tgljual, p.nmpem, b.namabrg,
d.jmljual, d.hrgjual
from djual d, barang b, jual j, pembeli p
where d.nonotaj=j.nonotaj      and      d.kdbrg=b.kdbrg      and
j.kdpem=p.kdpem;
```

The screenshot shows a Windows Command Prompt window titled "Command Prompt - mysql -u root". The user has run the following SQL query:

```
mysql> select d.nonotaj, j.tgljual, p.nmpem, b.namabrg,
-> d.jmljual,d.hrgjual
-> from djual d, barang b, jual j, pembeli p
-> where d.nonotaj=j.nonotaj and d.kdbrg=b.kdbrg and j.kdpem=p.kdpem;
```

The results are displayed in a table format:

nonotaj	tgljual	nmpem	namabrg	jmljual	hrgju
NJ00000004	2014-12-12	Mr. Burhanudin	RAM 256 MB	3	900
NJ00000004	2014-12-12	Mr. Burhanudin	RAM 512 MB	4	1150
NJ00000005	2015-02-01	Mr. Burhanudin	Printer	6	7000
NJ00000005	2015-02-01	Mr. Burhanudin	Processor AMD	6	3800
NJ00000001	2014-06-01	PT. Adi Jaya	Processor Intel	2	4750
NJ00000001	2014-06-01	PT. Adi Jaya	RAM 128 MB	4	600
NJ00000001	2014-06-01	PT. Adi Jaya	RAM 256 MB	2	900
NJ00000003	2014-12-03	PT. Adi Jaya	Printer	4	7000
NJ00000003	2014-12-03	PT. Adi Jaya	Processor Intel	3	4750
NJ00000002	2014-08-02	Ny Suci Darwanti	Monitor	5	6750

10 rows in set <0.00 sec>

Gambar 35 Hasil perintah menampilkan data dari empat tabel

- e. Menampilkan data dari empat tabel dengan membuat *field* baru

Pada kasus tertentu, *field* yang ada tidak ada untuk ditampilkan, sementara kebutuhan informasi mengharuskan untuk menampilkan kolom tertentu. Solusi dari masalah tersebut adalah memanfaatkan *field* lain dengan membuat formula atau fungsi.

Sebagai contoh adalah untuk menampilkan kolom Total. Padahal kolom Total tidak ada di tabel manapun. Namun secara logika Total dapat dihasilkan dengan melakukan perkalian antara *field* **jmljual** dengan *field* **hrgjual**.

Tabel yang digunakan adalah tabel djual, tabel barang, tabel jual dan tabel pembeli bentuk contoh dan hasilnya adalah sebagai berikut:

```
Select d.nonotaj, j.tgljual, p.nmpem, b.namabrg,
d.jmljual, d.hrgjual, d.jmljual*d.hrgjual as total
from djual d, barang b, jual j, pembeli p
where d.nonotaj=j.nonotaj      and      d.kdbrg=b.kdbrg      and
j.kdpem=p.kdpem;
```

```

mysql> select d.nonotaj, j.tgljual, p.nmpem, d.kdbrg, b.namabrg,
-> d.jmljual*d.hrgjual, d.jmljual*d.hrgjual as total
-> from djual d, barang b, jual j, pembeli p
-> where d.nonotaj=j.nonotaj and d.kdbrg=b.kdbrg and j.kdpem=p.kdpem;
+-----+-----+-----+-----+-----+
| nonotaj | tgljual | nmpem      | kdbrg     | namabrg      | jmlj
| ual | hrgjual | total      |           |             | ual |
+-----+-----+-----+-----+-----+
| NJ00000004 | 2014-12-12 | Mr. Burhanudin | B-000006 | RAM 256 MB | 1
| 3 | 90000 | 270000 |
| NJ00000004 | 2014-12-12 | Mr. Burhanudin | B-000007 | RAM 512 MB | 1
| 4 | 115000 | 460000 |
| NJ00000005 | 2015-02-01 | Mr. Burhanudin | B-000002 | Printer      | 1
| 6 | 700000 | 4200000 |
| NJ00000005 | 2015-02-01 | Mr. Burhanudin | B-000004 | Processor AMD | 1
| 6 | 380000 | 2280000 |
| NJ00000001 | 2014-06-01 | PT. Adi Jaya   | B-000003 | Processor Intel | 1
| 2 | 475000 | 950000 |
| NJ00000001 | 2014-06-01 | PT. Adi Jaya   | B-000005 | RAM 128 MB | 1
| 4 | 60000 | 240000 |
| NJ00000001 | 2014-06-01 | PT. Adi Jaya   | B-000006 | RAM 256 MB | 1
| 2 | 90000 | 180000 |
| NJ00000003 | 2014-12-03 | PT. Adi Jaya   | B-000002 | Printer      | 1
| 4 | 700000 | 2800000 |
| NJ00000003 | 2014-12-03 | PT. Adi Jaya   | B-000003 | Processor Intel | 1
| 3 | 475000 | 1425000 |
| NJ00000002 | 2014-08-02 | Ny Suci Darwanti | B-000001 | Monitor      | 1
| 5 | 675000 | 3375000 |
+-----+-----+-----+-----+-----+
10 rows in set (0.05 sec)

mysql>

```

Gambar 36 Hasil perintah menampilkan data dari empat tabel dengan membuat field baru

- f. Menampilkan data dari beberapa dengan **group by** dan membuat *field* baru

Tabel yang digunakan adalah tabel djual dan tabel jual, bentuk contoh dan hasilnya adalah sebagai berikut:

```

Select year(j.tgljual) as tahun, count(d.nonotaj) as jmltransaksi
Sum(d.jmljual*d.hrgjual) as nilaitransaksi
From djual d, jual j
Where d.nonotaj=j.nonotaj
Group by year(j.tgljual);

```

```

mysql> select year(j.tgljual) as tahun, count(d.nonotaj) as jmltransaksi,
-> sum(d.jmljual*d.hrgjual) as nilaitransaksi
-> from djual d, jual j
-> where d.nonotaj=j.nonotaj
-> group by year(j.tgljual);
+-----+-----+-----+
| tahun | jmltransaksi | nilaitransaksi |
+-----+-----+-----+
| 2014 | 8 | 9700000 |
| 2015 | 2 | 6480000 |
+-----+-----+-----+
2 rows in set (0.03 sec)

mysql>

```

Gambar 37 Menampilkan data dari beberapa dengan group by dan membuat field baru

**3. Tugas**

- a. Kerjakan perintah menampilkan data untuk tabel beli dan dbeli yang dikombinasikan juga dengan tabel yang lain seperti proses menampilkan tabel jual dan djual.
- b. Lakukan juga proses pembuatan *database* hotel sampai pengelolaan tabel dan menampilkan data (lihat langkah-langkah pada modul 6 dan modul 7)
- c. Langkah-langkah atau perintah yang dihasilkan dari pekerjaan untuk poin a dan poin b di pindahkan ke MS Word dan kirimkan ke email: **[zaidirtan@gmail.com](mailto:zaidirtan@gmail.com)**.

## BAB 8

### STUDI KASUS DAN PENYELESAIAN DENGAN MYSQL COMMAND LINE

#### A. Kasus

##### 1. Permasalahan

Sebuah perusahaan rental kendaraan yang baru berkembang ingin melakukan peningkatan kualitas layanan kepada para pelanggannya sekaligus ingin menetibkan administrasinya. Pemilik ingin melakukan pengolahan data secara terkomputerisasi.

Pada saat ini kendaraan yang disewakan terdiri dari motor dan mobil. Untuk melayani para peminjam, pemilik selama ini menyediakan data berupa data kendaraan beserta besaran harga sewanya.

Apabila ada masyarakat atau pelanggan yang ingin menyewa kendaraan maka dicatat identitasnya dan diharus meninggalkan identitas yang dimiliki. Selanjutnya diisi juga mengenai data peminjaman yang dilakukan. Pada saat peminjam mengembalikan kendaraan, maka pemilik mencatatkan mengenai data pengembalian dan mengembalikan identitas peminjam yang ditinggalkan pada saat peminjaman.

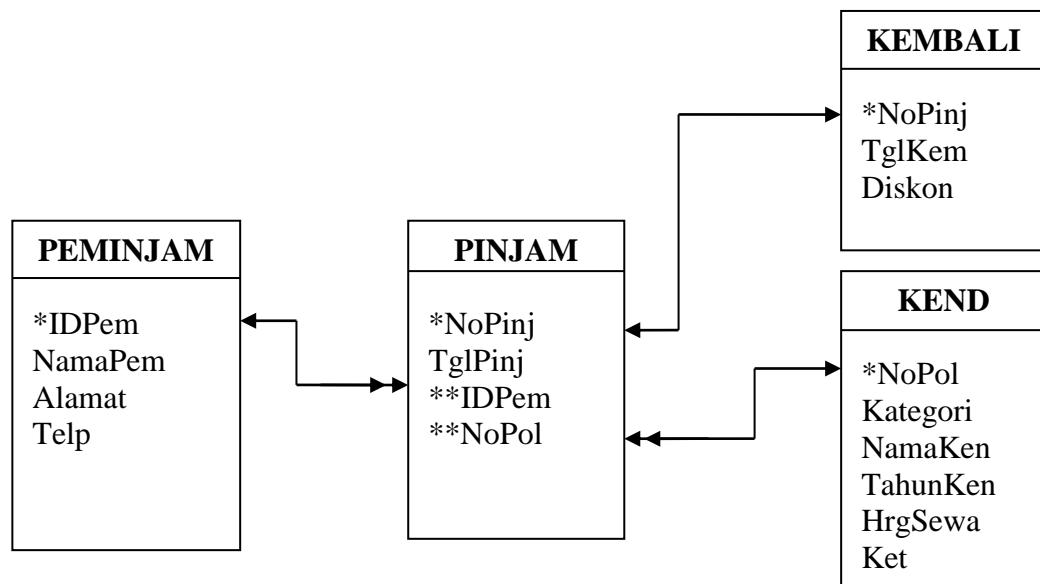
##### 2. Analisis Masalah

Berdasarkan permasalahan di atas, ada beberapa hal yang bisa dijadikan kunci untuk penyelesaiannya. Pertama untuk pengelolaan data kendaraan yang terdiri dari dua yaitu motor dan mobil, maka penyimpanan cukup dibuat satu saja dan dibedakan berdasarkan kategorinya saja, sedangkan identitas kendaraan lainnya bisa sama. Data yang terkait dengan identitas bias disimpan dalam satu tabel untuk mengatasi jangan sampai banyak terjadi perulangan data apabila peminjam yang sama melakukan penyewaan berulangkali.

Untuk pencatatan layanan kepada penyewa atau peminjam dapat dibuat satu penyimpanan atau dua penyimpanan. Apabila dibuat satu penyimpanan kelebihannya adalah lebih praktis (hanya satu tabel), namun akan ada masalah tertentu dalam hal penghapusan ataupun dalam hal pertanggungjawaban pelayanan. Apabila dibuat dua penyimpanan maka tabel lebih banyak, namun masalah tertentu dalam hal penghapusan dan pertanggungjawaban pelayanan biasa diatasi. Pemikiran ini tentu akan lebih baik dengan pertimbangan masa yang akan dating bilamana perusahaan berkembang dan membutuhkan sistem yang lebih valid. Untuk contoh penyelesaian dari kasus ini akan diterapkan yang menggunakan dua penyimpanan saja.

### 3. Rancangan (Desain) Tabel

Berdasarkan analisis di atas, maka setelah melalui proses pertimbangan yang mendalam maka rancangan tabel yang dihasilkan adalah sebagai berikut.



Gambar 1 Relasi antar tabel kasus rental kendaraan

Catatan:

Rancangan tabel di atas hanya sebagai contoh, dan masih memungkinkan untuk dikembangkan berdasarkan kebutuhan dan permintaan pengguna. Sebagai contoh untuk memberikan pertanggungjawaban pelayanan dapat ditambah table petugas/pengguna yang dihubungkan dengan table pinjam dan table kembali.

## B. Praktek Penyelesaian Studi Kasus Dengan MySQL Command Line

Berdasarkan hasil rancangan tabel di atas, maka buatlah *database* baru menggunakan **command line** dengan nama RENTAL. Setelah *database* tercipta lanutkan dengan membuat masing-masing table sekaligus mengisi datanya.

### 1. Membuat Database

Topik yang dibahas pada pertemuan praktikum ini adalah sesuatu yang baru, maka ciptakanlah *database* yang baru dengan nama RENTAL.

**Create database rental;**

### 2. Membuat table dan mengisi data

Untuk membuat tabel, mohon diperhatikan *field-field* yang akan diciptakan termasuk jenis data yang sesuai. Berdasarkan rancangan tabel, maka *field* yang diberi tanda satu bintang (\*) berarti *field* tersebut adalah *primary key*. *Field* yang diberi tanda dua bintang (\*\*) berarti *field* tersebut adalah *foreign key*.

#### a. Membuat tabel dan mengisi data kendaraan (kend)

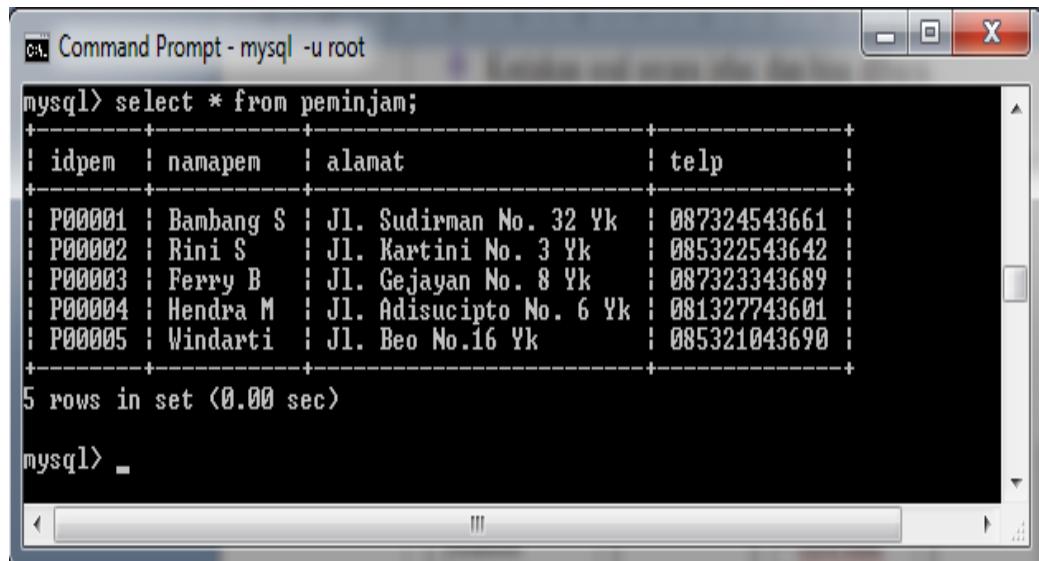
Buatlah tabel baru yang bernama **kend**, dengan jumlah *field* sebanyak 6 *field*. Untuk jenis data dan ukurannya silahkan disesuaikan. *Field* nopol dijadikan sebagai kunci utama yang tidak boleh kosong. Selanjutnya isilah datanya, sehingga bila ditampilkan bisa terlihat seperti Gambar 2.

nopol	kategori	namaken	tahunken	hrgsewa	ket
AB1219YA	Motor	Yamaha Mio	2010	50000	K
AB2145AZ	Mobil	Avanza	2012	250000	K
AB2341TR	Motor	Supra X	2014	60000	K
AB5321MK	Mobil	Camry	2014	1200000	K
AB5647HG	Motor	Beat	2013	40000	K
AB6106PL	Motor	Suzuki Shogun	2010	30000	I
AB7654RU	Mobil	Pajero	2013	750000	K
AB7809FD	Motor	Vario	2014	50000	I
AB9065PK	Mobil	Mobilio	2014	500000	K
AB9878XP	Mobil	Xenia	2013	250000	K
AB9980HK	Mobil	Avanza	2010	200000	I

Gambar 2 Tampilan data kendaraan

b. Membuat tabel dan mengisi data penyewa (peminjam)

Buatlah tabel baru yang bernama **peminjam**, dengan jumlah *field* sebanyak 4 *field*. Untuk jenis data dan ukurannya silahkan disesuaikan. *Field idpem* dijadikan sebagai kunci utama yang tidak boleh kosong Selanjutnya isilah datanya, sehingga bila ditampilkan bisa terlihat seperti Gambar 3..



MySQL command prompt window showing the output of a SELECT query on the 'peminjam' table. The table has four columns: idpem, namapem, alamat, and telp. The data consists of five rows:

idpem	namapem	alamat	telp
P00001	Bambang S	Jl. Sudirman No. 32 Yk	087324543661
P00002	Rini S	Jl. Kartini No. 3 Yk	085322543642
P00003	Ferry B	Jl. Gejayan No. 8 Yk	087323343689
P00004	Hendra M	Jl. Adisucipto No. 6 Yk	081327743601
P00005	Windarti	Jl. Beo No.16 Yk	085321043690

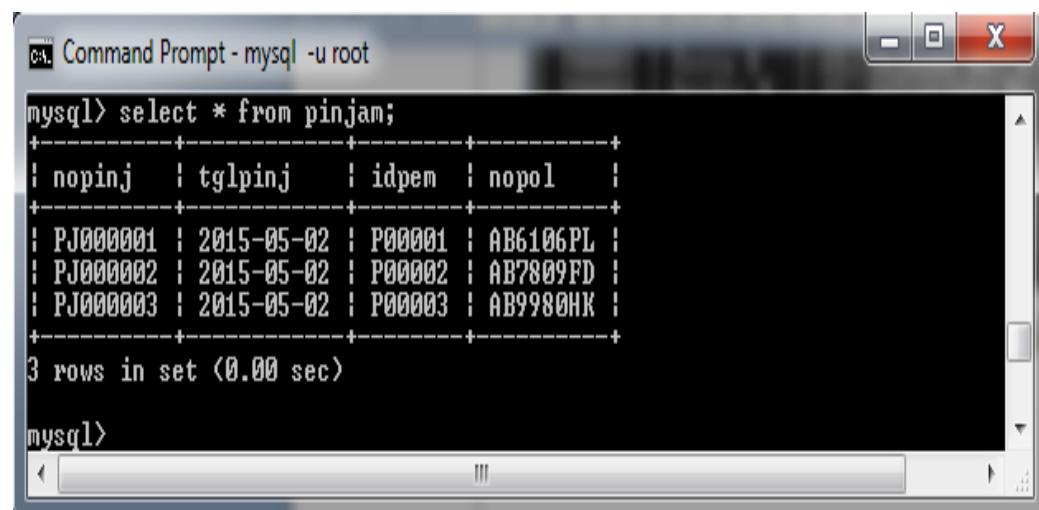
5 rows in set (0.00 sec)

mysql> \_

Gambar 3 Tampilan data peminjam

c. Membuat tabel dan mengisi data peminjaman (pinjam)

Buatlah tabel baru yang bernama **pinjam**, dengan jumlah *field* sebanyak 4 *field*. Untuk jenis data dan ukurannya silahkan disesuaikan. *Field nopinj* dijadikan sebagai kunci utama yang tidak boleh kosong Selanjutnya isilah datanya, sehingga bila ditampilkan bisa terlihat seperti Gambar 4.



MySQL command prompt window showing the output of a SELECT query on the 'pinjam' table. The table has four columns: nopinj, tglpinj, idpem, and nopol. The data consists of three rows:

nopinj	tglpinj	idpem	nopol
PJ000001	2015-05-02	P00001	AB6106PL
PJ000002	2015-05-02	P00002	AB7809FD
PJ000003	2015-05-02	P00003	AB9980HK

3 rows in set (0.00 sec)

mysql>

Gambar 4 Tampilan data pinjam

- d. Membuat tabel dan mengisi data pengembalian (kembali)

Buatlah tabel baru yang bernama **kembali**, dengan jumlah *field* sebanyak 3 *field*. Untuk jenis data dan ukurannya silahkan disesuaikan. *Field* **nopinj** dijadikan sebagai kunci utama yang tidak boleh kosong Selanjutnya isilah datanya, sehingga bila ditampilkan bisa terlihat seperti Gambar 5.

```
mysql> select * from kembali;
+-----+-----+-----+
| nopinj | tglkem   | diskon |
+-----+-----+-----+
| PJ000001 | 2015-05-05 |      5 |
| PJ000002 | 2015-05-04 |     10 |
| PJ000003 | 2015-05-06 |      0 |
+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

Gambar 5 Tampilan data kembali

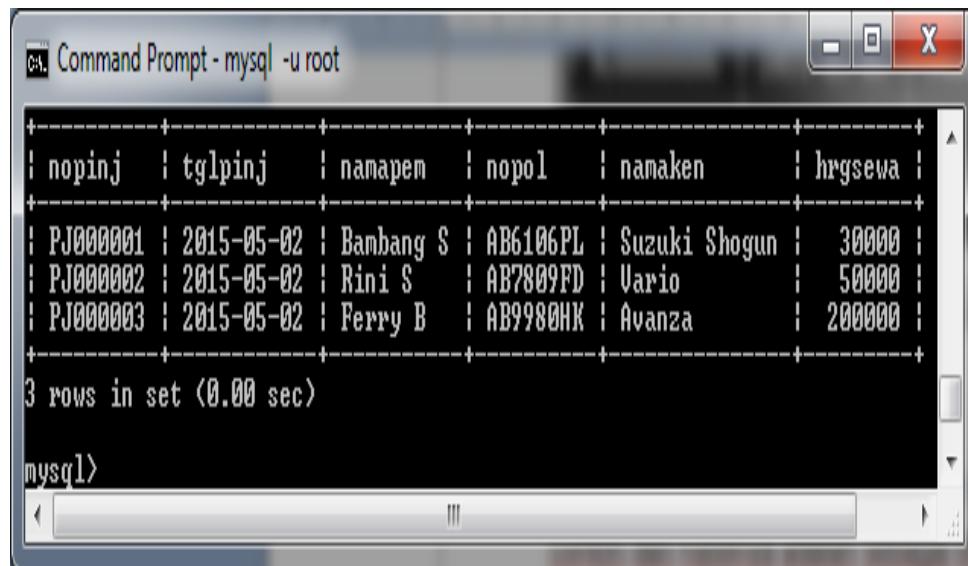
### 3. Menampilkan data

- d. Menampilkan data kendaraan
  - 1) Tampilkanlah data kendaraan khusus untuk mobil
  - 2) Tampilkanlah data kendaraan khusus untuk motor
  - 3) Tampilkanlah data kendaraan yang harga sewanya diantara 50000 dan 500000
  - 4) Tampilkan data kendaraan khusus mobil yang harga sewanya di atas 500000
  - 5) Tampilkan data kendaraan yang sedang dipinjam saja
  - 6) Tampilkan data kendaraan khusus motor yang tidak dipinjam

e. Menampilkan data peminjaman

1) Menampilkan semua data peminjaman

Tulislah perintah SQL untuk menampilkan semua data peminjaman dengan kolom dan data yang ditampilkan seperti terlihat pada Gambar 6.



nopinj	tglpinj	namapem	nopol	namaken	hrgsewa
PJ000001	2015-05-02	Bambang S	AB6106PL	Suzuki Shogun	30000
PJ000002	2015-05-02	Rini S	AB7809FD	Vario	50000
PJ000003	2015-05-02	Ferry B	AB9980HK	Avanza	200000

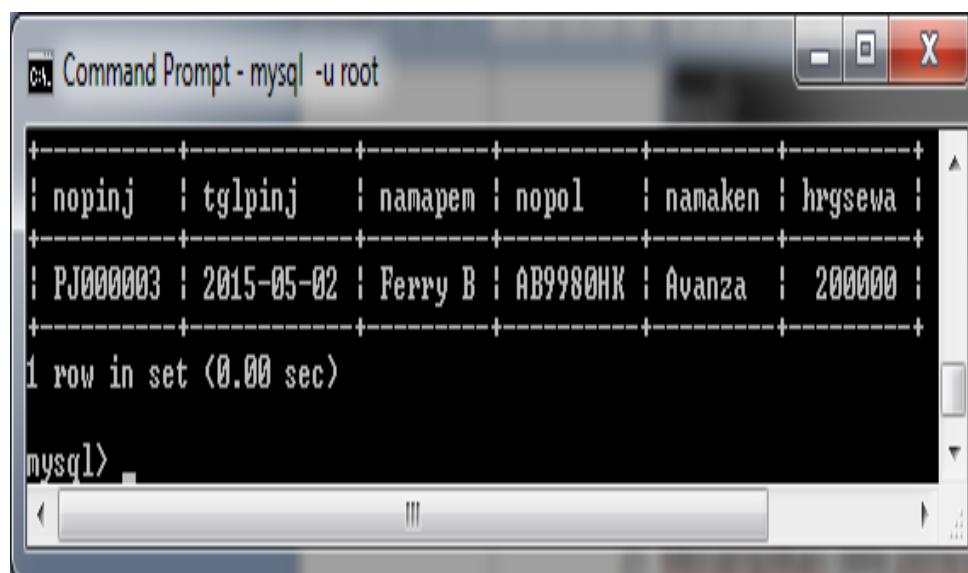
3 rows in set (0.00 sec)

mysql>

Gambar 6 Tampilan setelah perintah menampilkan semua data peminjaman

2) Menampilkan data peminjaman khusus mobil

Tulislah perintah SQL untuk menampilkan data peminjaman dengan kolom dan data yang ditampilkan seperti terlihat pada Gambar 7.



nopinj	tglpinj	namapem	nopol	namaken	hrgsewa
PJ000003	2015-05-02	Ferry B	AB9980HK	Avanza	200000

1 row in set (0.00 sec)

mysql>

Gambar 7 Tampilan setelah perintah menampilkan data peminjaman khusus mobil

- f. Menampilkan data pengembalian

The screenshot shows a Windows Command Prompt window titled "Command Prompt - mysql -u root". The output displays a table of data from a database. The columns are labeled: namapem, namaken, lama, hrgsewa, Jmlsewa, Diskon, and Bayar. The data consists of three rows:

namapem	namaken	lama	hrgsewa	Jmlsewa	Diskon	Bayar
Bambang S	Suzuki Shogun	3	30000	90000	4500	85500
Rini S	Vario	2	50000	100000	10000	90000
Ferry B	Avanza	4	200000	800000	0	800000

Below the table, the message "3 rows in set (0.00 sec)" is displayed. The MySQL prompt "mysql>" is at the bottom.

Gambar 8 Tampilan setelah perintah menampilkan data pengembalian

Catatan:

Tabel yang digunakan yaitu table pinjam, table kend, table peminjam, dan table kembali. Manfaatkan operasi-operasi matematis untuk menghasilkan perhitungan dan simpan ke field baru. Gunakan fungsi round untuk pembulatan sampai 0 titik decimal.

- g. Menampilkan data dengan pengelompokan (grouping)
- 1) Menampilkan grouping dari peminjaman

The screenshot shows a Windows Command Prompt window titled "Command Prompt - mysql -u root". The output displays a table of data from a database, grouped by the year. The columns are labeled: Tahun, JmlMobil, JmlMotor, and JmlTotal. The data consists of one row:

Tahun	JmlMobil	JmlMotor	JmlTotal
2015	1	2	3

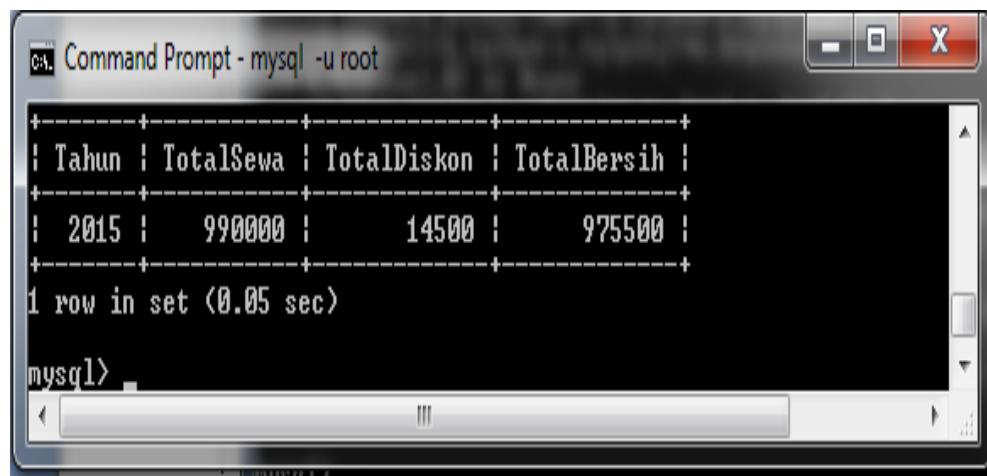
Below the table, the message "1 row in set (0.00 sec)" is displayed. The MySQL prompt "mysql>" is at the bottom.

Gambar 9 Tampilan setelah perintah menampilkan data peminjaman dengan group

Catatan:

Tabel yang digunakan yaitu tabel pinjam, tabel kend, tabel kembali. Tahun dihasilkan dari fungsi tahun dari field tglpinj. Jmlmobil dan Jmlmotor didapat dengan menggunakan fungsi sum dan fungsi if. JmlTotal didapat dengan fungsi **count**.

2) Menampilkan grouping dari pengembalian



```
+-----+-----+-----+
| Tahun | TotalSewa | TotalDiskon | TotalBersih |
+-----+-----+-----+
| 2015 | 990000 | 14500 | 975500 |
+-----+-----+-----+
1 row in set (0.05 sec)

mysql>
```

Gambar 9 Tampilan setelah perintah menampilkan data pengembalian dengan group

Catatan:

Tabel yang digunakan yaitu tabel pinjam, tabel kend, tabel kembali. Tahun dihasilkan dari fungsi tahun dari field tglkem

## BAB 9

### BELAJAR MYSQL DENGAN COMMAND LINE EXPAND

#### A. Menggabungkan Tabel Dengan Inner Join

Pada pertemuan sebelumnya telah digunakan perintah untuk menampilkan data dari satu tabel atau dari beberapa tabel. Perintah yang digunakan adalah memanfaat perintah Select.. From... Where....Group By. Hasil yang diperoleh memang telah mencukupi dengan berbagai variasi.

Pada kesempatan praktikum ini akan digunakan perintah menggabungkan tabel (query) dengan menggunakan Inner Join. Query **INNER JOIN** memiliki beberapa variasi, yaitu:

##### 1. **SELECT..INNER JOIN..USING**

Digunakan dengan syarat tabel-tabel yang digabungkan harus memiliki nama kolom (*field*) yang sama

##### 2. **SELECT..INNER JOIN..ON**

Digunakan dengan syarat tabel-tabel yang digabungkan boleh memiliki nama kolom (*field*) yang tidak sama

Untuk mencoba perintah-perintah tersebut akan digunakan *database RENTAL* yang telah dihasilkan pada pertemuan sebelumnya.

```
Command Prompt - mysql -u root
mysql> use rental;
Database changed
mysql> show tables;
+-----+
| Tables_in_rental |
+-----+
| kembali
| kend
| peminjam
| pinjam
+-----+
4 rows in set <0.00 sec>

mysql>
```

##### a. Struktur dan data tabel peminjam

###### 1) Struktur tabel peminjam

```
Command Prompt - mysql -u root
mysql> desc peminjam;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idpem | char(6) | NO  | PRI  | NULL    |       |
| namapem | varchar(40) | YES |       | NULL    |       |
| alamat | varchar(75) | YES |       | NULL    |       |
| telp | varchar(15) | YES |       | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set <0.11 sec>

mysql>
```

2) Data tabel peminjam

```
mysql> select * from peminjam;
+-----+-----+-----+-----+
| idpem | namapem | alamat | telp |
+-----+-----+-----+-----+
| P00001 | Bambang S | Jl. Sudirman No. 32 Yk | 087324543661 |
| P00002 | Rini S | Jl. Kartini No. 3 Yk | 085322543642 |
| P00003 | Ferry B | Jl. Gejayan No. 8 Yk | 087323343689 |
| P00004 | Hendra M | Jl. Adisucipto No. 6 Yk | 081327743601 |
| P00005 | Windarti | Jl. Beo No.16 Yk | 085321043690 |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

b. Struktur dan data tabel kend

1) Struktur tabel kend

```
mysql> desc kend;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nopol | char(8) | NO | PRI | NULL |       |
| kategori | varchar(10) | YES |       | NULL |       |
| namaken | varchar(30) | YES |       | NULL |       |
| tahunken | char(4) | YES |       | NULL |       |
| hrgsewa | int(11) | YES |       | 0 |       |
| ket | char(1) | YES |       | NULL |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.09 sec)

mysql>
```

2) Data tabel kend

```
mysql> select * from kend;
+-----+-----+-----+-----+-----+-----+
| nopol | kategori | namaken | tahunken | hrgsewa | ket |
+-----+-----+-----+-----+-----+-----+
| AB1219VA | Motor | Yamaha Mio | 2010 | 50000 | K |
| AB2145AZ | Mobil | Avanza | 2012 | 250000 | K |
| AB2341TR | Motor | Supra X | 2014 | 60000 | K |
| AB5321MK | Mobil | Camry | 2014 | 1200000 | K |
| AB5647HG | Motor | Beat | 2013 | 40000 | K |
| AB6106PL | Motor | Suzuki Shogun | 2010 | 30000 | I |
| AB7654RU | Mobil | Pajero | 2013 | 750000 | K |
| AB7809FD | Motor | Vario | 2014 | 50000 | I |
| AB9065PK | Mobil | Mobilio | 2014 | 500000 | K |
| AB9878XP | Mobil | Xenia | 2013 | 250000 | K |
| AB9980HK | Mobil | Avanza | 2010 | 200000 | I |
+-----+-----+-----+-----+-----+-----+
11 rows in set (0.00 sec)

mysql>
```

c. Struktur dan data tabel pinjam

1) Struktur tabel pinjam

```
mysql> desc pinjam;
+-----+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nopol | char(8)| NO   | PRI   | NULL    |       |
| tglpinj| date   | YES  |       | NULL    |       |
| idpem  | char(6) | NO   |       | NULL    |       |
| nopol  | char(8) | NO   |       | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.03 sec)

mysql>
```

2) Data tabel pinjam

```
mysql> select * from pinjam;
+-----+-----+-----+-----+
| nopol | tglpinj | idpem | nopol |
+-----+-----+-----+-----+
| PJ000001 | 2015-05-02 | P00001 | AB6106PL |
| PJ000002 | 2015-05-02 | P00002 | AB7809FD |
| PJ000003 | 2015-05-02 | P00003 | AB9980HK |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

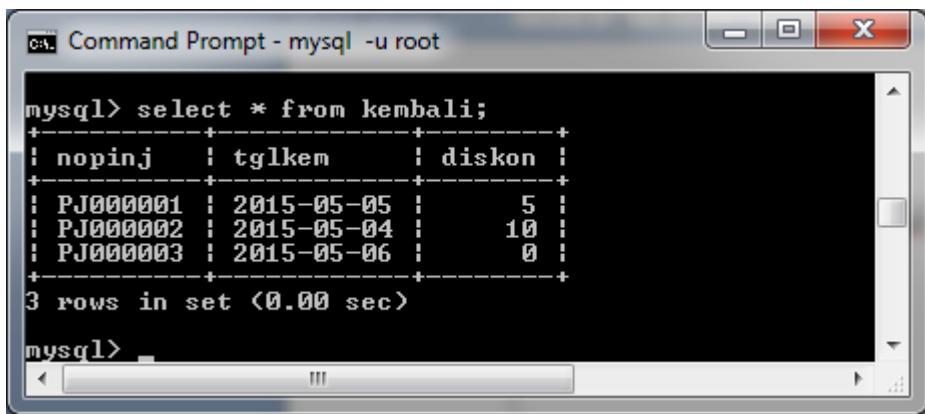
d. Struktur dan data tabel kembali

1) Struktur tabel kembali

```
mysql> desc kembali;
+-----+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nopol | char(8)| NO   | PRI   | NULL    |       |
| tglkem | date   | YES  |       | NULL    |       |
| diskon | int(11) | YES  |       | 0       |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.03 sec)

mysql>
```

## 2) Data tabel kembali



```
Command Prompt - mysql -u root
mysql> select * from kembali;
+-----+-----+-----+
| nopinj | tglkem | diskon |
+-----+-----+-----+
| PJ000001 | 2015-05-05 |      5 |
| PJ000002 | 2015-05-04 |     10 |
| PJ000003 | 2015-05-06 |      0 |
+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

## 4. Cara Menggabungkan Tabel MySQL dengan SELECT..INNER JOIN..USING

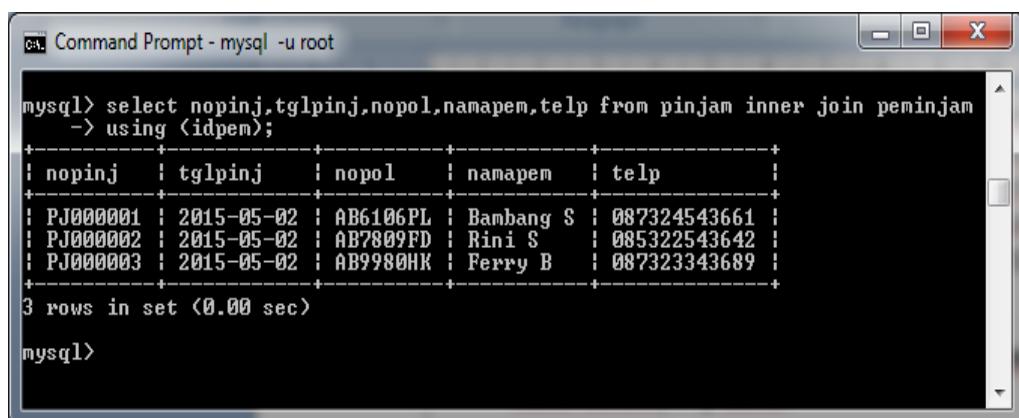
Format dasar dari penulisan query **SELECT..INNER JOIN..USING** adalah:

**SELECT nama\_kolom\_tampil FROM nama\_tabel\_pertama  
INNER JOIN  
nama\_tabel\_kedua USING (nama\_kolom\_join)**

- nama\_kolom\_tampil adalah nama dari kolom yang akan kita tampilkan, bisa semua kolom dalam tabel, atau hanya kolom tertentu saja.
- nama\_tabel\_pertama adalah nama tabel pertama yang akan digabung.
- nama\_tabel\_kedua adalah nama tabel kedua yang akan digabung.
- nama\_kolom\_join adalah nama kolom yang akan digunakan sebagai join.

Syarat untuk **INNER JOIN..USING** adalah ***kedua tabel harus memiliki nama kolom yang sama.***

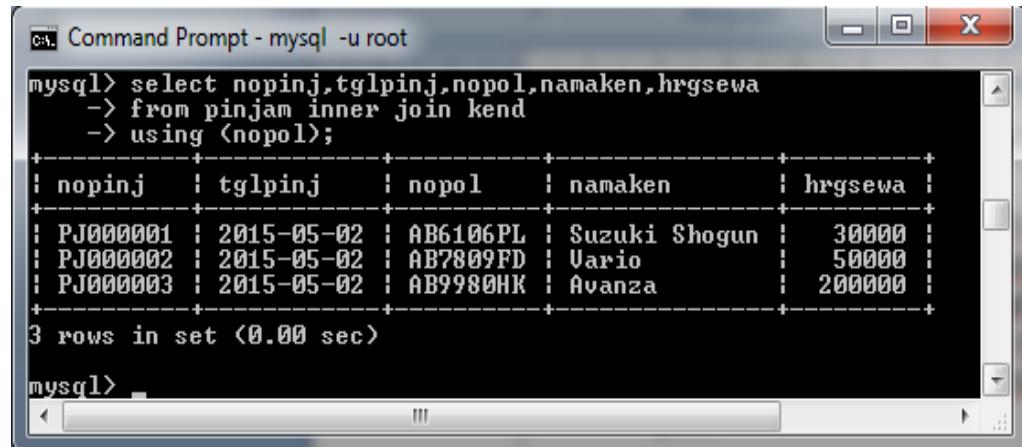
### a. Menggabung tabel Pinjam dengan tabel Peminjam



```
Command Prompt - mysql -u root
mysql> select nopinj,tglpinj,nopol,namapem,telp from pinjam inner join peminjam
-> using (idpem);
+-----+-----+-----+-----+
| nopinj | tglpinj | nopol | namapem | telp |
+-----+-----+-----+-----+
| PJ000001 | 2015-05-02 | AB6106PL | Bambang S | 087324543661 |
| PJ000002 | 2015-05-02 | AB7809FD | Rini S | 085322543642 |
| PJ000003 | 2015-05-02 | AB9980HK | Ferry B | 087323343689 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

b. Menggabung tabel Pinjam dengan tabel Kend

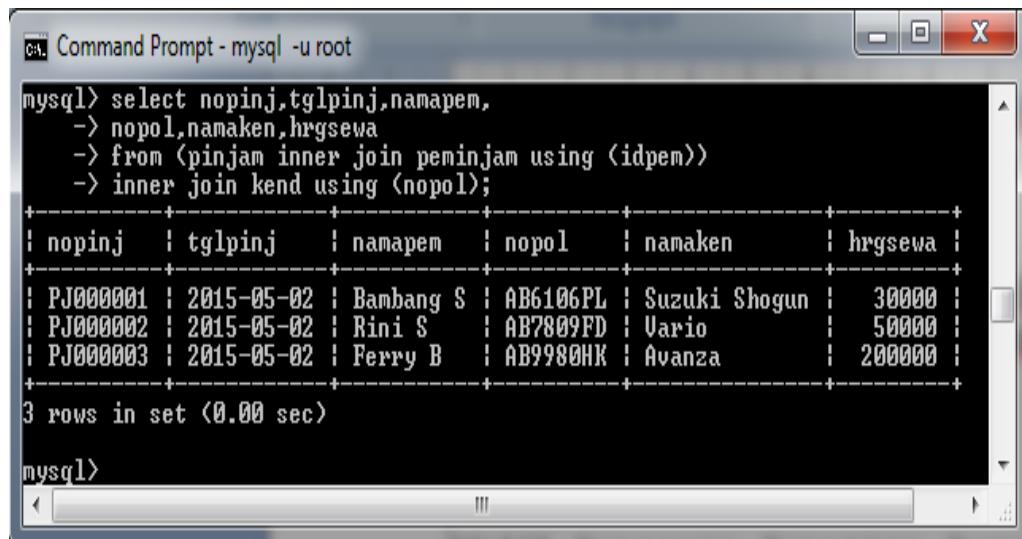


Command Prompt - mysql -u root

```
mysql> select nopinj,tglpinj,nopol,namaken,hrgsewa
-> from pinjam inner join kend
-> using <nopol>;
+-----+-----+-----+-----+-----+
| nopinj | tglpinj | nopol | namaken | hrgsewa |
+-----+-----+-----+-----+-----+
| PJ000001 | 2015-05-02 | AB6106PL | Suzuki Shogun | 30000 |
| PJ000002 | 2015-05-02 | AB7809FD | Vario | 50000 |
| PJ000003 | 2015-05-02 | AB9980HK | Avanza | 200000 |
+-----+-----+-----+-----+-----+
3 rows in set <0.00 sec>

mysql>
```

c. Menggabung tabel Pinjam dengan tabel Peminjam sekaligus tabel Kend

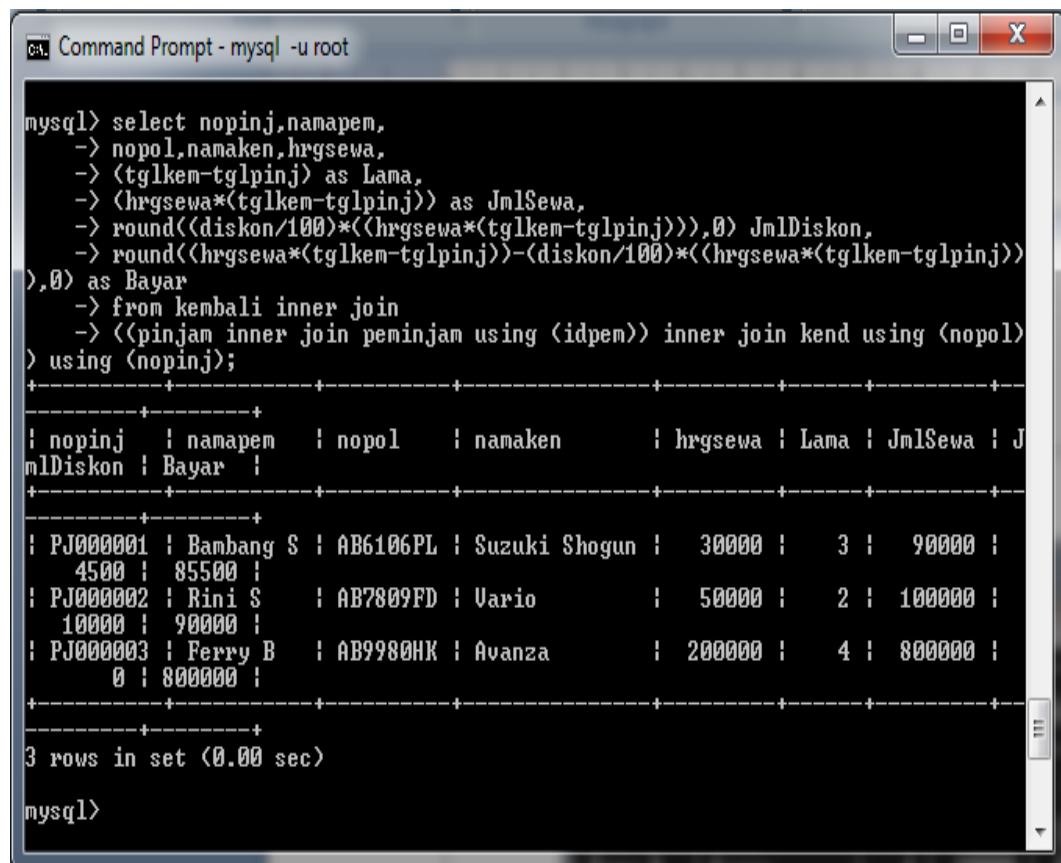


Command Prompt - mysql -u root

```
mysql> select nopinj,tglpinj,namapem,
-> nopol,namaken,hrgsewa
-> from <pinjam inner join peminjam using <idpem>
-> inner join kend using <nopol>;
+-----+-----+-----+-----+-----+-----+
| nopinj | tglpinj | namapem | nopol | namaken | hrgsewa |
+-----+-----+-----+-----+-----+-----+
| PJ000001 | 2015-05-02 | Bambang S | AB6106PL | Suzuki Shogun | 30000 |
| PJ000002 | 2015-05-02 | Rini S | AB7809FD | Vario | 50000 |
| PJ000003 | 2015-05-02 | Ferry B | AB9980HK | Avanza | 200000 |
+-----+-----+-----+-----+-----+-----+
3 rows in set <0.00 sec>

mysql>
```

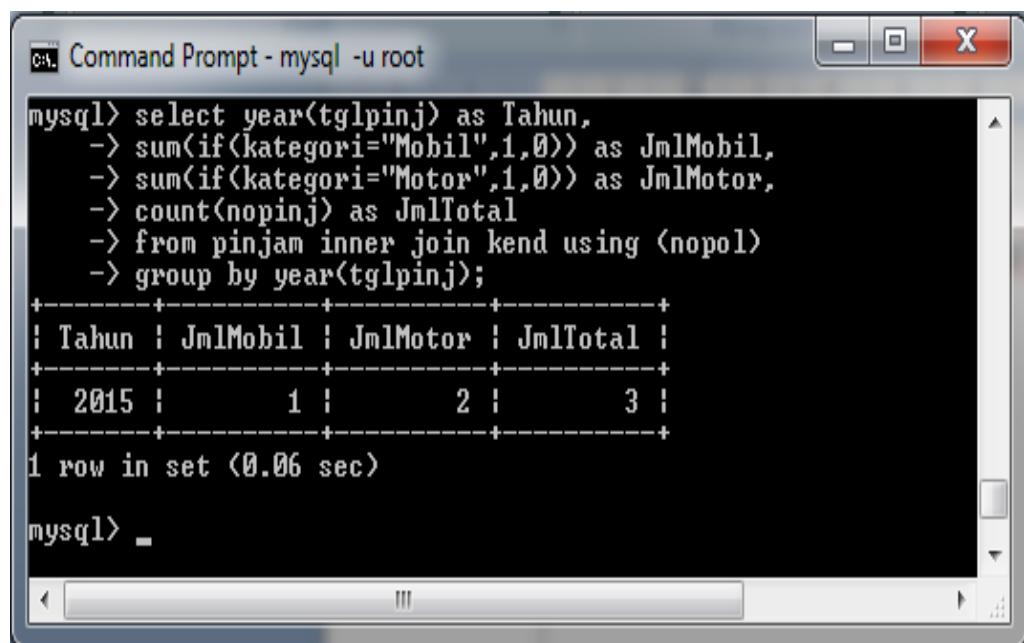
- d. Menggabung tabel Pinjam dengan tabel Peminjam, tabel Kend, tabel Kembali



```
mysql> select nopinj,namapem,
-> nopol,namaken,hrgsewa,
-> <tgkem-tglpinj> as Lama,
-> <hrgsewa*(tgkem-tglpinj)> as JmlSewa,
-> round((diskon/100)*(<hrgsewa*(tgkem-tglpinj)>),0) JmlDiskon,
-> round((hrgsewa*(tgkem-tglpinj))-<diskon/100)*(<hrgsewa*(tgkem-tglpinj)>,0) as Bayar
-> from kembali inner join
-> ((pinjam inner join peminjam using (idpem)) inner join kend using (nopol))
-> using (nopinj);
+-----+-----+-----+-----+-----+
| nopinj | namapem | nopol | namaken | hrgsewa | Lama | JmlSewa | JmlDiskon | Bayar |
+-----+-----+-----+-----+-----+
| PJ000001 | Bambang S | AB6106PL | Suzuki Shogun | 30000 | 3 | 90000 | 4500 | 85500 |
| PJ000002 | Rini S | AB7809FD | Vario | 50000 | 2 | 100000 | 10000 | 90000 |
| PJ000003 | Ferry B | AB9980HK | Avanza | 200000 | 4 | 800000 | 0 | 800000 |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

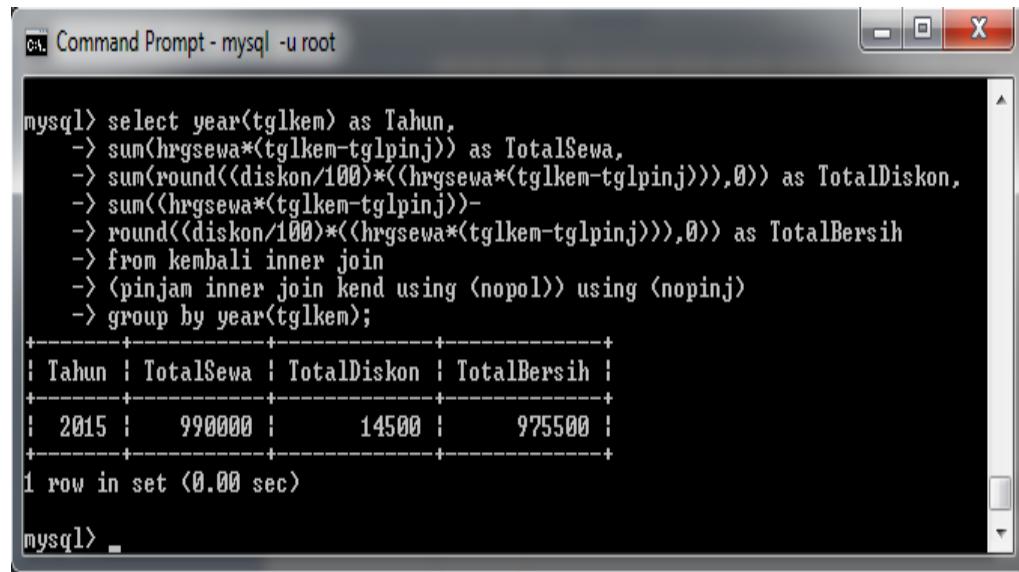
- e. Menggabung tabel Pinjam dengan tabel Kend untuk menghasilkan rekapitulasi peminjaman



```
mysql> select year(tglpinj) as Tahun,
-> sum(if(kategori="Mobil",1,0)) as JmlMobil,
-> sum(if(kategori="Motor",1,0)) as JmlMotor,
-> count(nopinj) as JmlTotal
-> from pinjam inner join kend using (nopol)
-> group by year(tglpinj);
+-----+-----+-----+
| Tahun | JmlMobil | JmlMotor | JmlTotal |
+-----+-----+-----+
| 2015 | 1 | 2 | 3 |
+-----+-----+
1 row in set (0.06 sec)

mysql> _
```

- f. Menggabung tabel Kembali dengan tabel Pinjam, tabel Kend untuk menghasilkan rekapitulasi pengembalian



```
Command Prompt - mysql -u root
mysql> select year(tglkem) as Tahun,
-> sum(hrgsewa*(tglkem-tglpinj)) as TotalSewa,
-> sum(round((diskon/100)*((hrgsewa*(tglkem-tglpinj))),0)) as TotalDiskon,
-> sum((hrgsewa*(tglkem-tglpinj))-round((diskon/100)*((hrgsewa*(tglkem-tglpinj))),0)) as TotalBersih
-> from kembali inner join
-> (pinjam inner join kend using (nopol) using (nopinj)
-> group by year(tglkem);
+-----+-----+-----+
| Tahun | TotalSewa | TotalDiskon | TotalBersih |
+-----+-----+-----+
| 2015 | 990000 | 14500 | 975500 |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> _
```

5. Cara Menggabungkan Tabel MySQL dengan query **SELECT..INNER JOIN..ON**

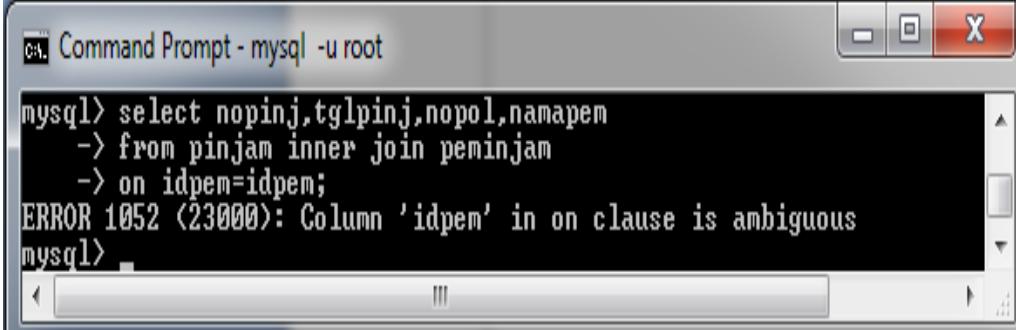
Format dasar dari penulisan query **SELECT..INNER JOIN..ON** adalah:

**SELECT nama\_kolom\_tampil FROM nama\_tabel\_pertama  
INNER JOIN nama\_tabel\_kedua  
ON nama\_kolom\_join\_tabel\_pertama =  
nama\_kolom\_join\_tabel\_kedua**

- nama kolom tampil adalah nama dari kolom yang akan kita tampilkan, bisa semua kolom dalam tabel, atau hanya kolom tertentu saja.
- nama tabel pertama adalah nama tabel pertama yang akan digabung.
- nama tabel kedua adalah nama tabel kedua yang akan digabung.
- nama kolom join tabel pertama adalah nama kolom yang akan digunakan sebagai join dari tabel pertama.
- nama kolom join tabel kedua adalah nama kolom yang akan digunakan sebagai join dari tabel kedua.

Syarat untuk **INNER JOIN** adalah kedua tabel **harus memiliki sebuah kolom dengan nilai yang sama** yang akan digunakan dalam proses **JOIN**. Apabila nama kolom (*field*) dari tabel yang digabungkan ternyata sama, maka sebelum *field* tersebut harus disebutkan nama tabelnya dan dipisah dengan tanda titik.

a. Menggabung tabel Pinjam dengan tabel Peminjam



```
Command Prompt - mysql -u root
mysql> select nopinj,tglpinj,nopol,namapem
-> from pinjam inner join peminjam
-> on idpem=idpem;
ERROR 1052 (23000): Column 'idpem' in on clause is ambiguous
mysql>
```

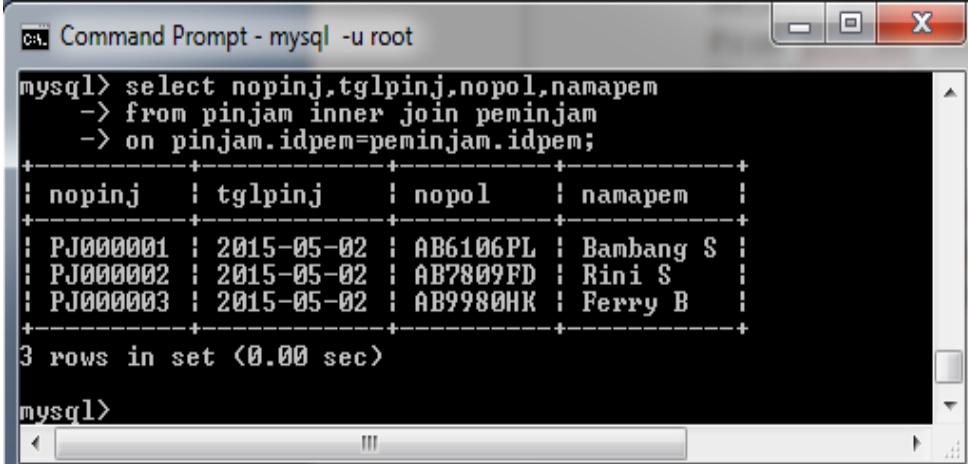
Dari pesan error yang terjadi, sebenarnya MySQL ‘bingung’ dalam memilih kolom. Query **ON idpem=idpem** menjadi ambigu karena kedua tabel kita memiliki nama kolom yang sama.

Untuk mengatasi ambiguitas ini, ada dua cara

1. Membuat *field* sama menjadi berbeda, misalnya *field* idpem pada tabel Peminjam dirubah menjadi idP, sehingga perintahnya menjadi:

```
Select nopinj,tglpinj,nopol,namapem
From pinjam inner join peminjam
On idpem=idp
```

2. MySQL mengharuskan secara *eksplisit* menyebutkan nama tabelnya. Cara penulisan ini menggunakan tanda titik sebagai pemisah, sehingga perintahnya menjadi:



```
Command Prompt - mysql -u root
mysql> select nopinj,tglpinj,nopol,namapem
-> from pinjam inner join peminjam
-> on pinjam.idpem=peminjam.idpem;
+-----+-----+-----+-----+
| nopinj | tglpinj | nopol | namapem |
+-----+-----+-----+-----+
| PJ000001 | 2015-05-02 | AB6106PL | Bambang S |
| PJ000002 | 2015-05-02 | AB7809FD | Rini S |
| PJ000003 | 2015-05-02 | AB9980HK | Ferry B |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

b. Menggabung tabel Pinjam dengan tabel Kend

```
ca. Command Prompt - mysql -u root
mysql> select nopinj,tglpinj,pinjam.nopol,namaken,hrgsewa
-> from pinjam inner join kend
-> on pinjam.nopol=kend.nopol;
+-----+-----+-----+-----+-----+
| nopinj | tglpinj | nopol | namaken | hrgsewa |
+-----+-----+-----+-----+-----+
| PJ000001 | 2015-05-02 | AB6106PL | Suzuki Shogun | 30000 |
| PJ000002 | 2015-05-02 | AB7809FD | Vario | 50000 |
| PJ000003 | 2015-05-02 | AB9980HK | Avanza | 200000 |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

c. Menggabung tabel Pinjam dengan tabel Peminjam sekaligus tabel Kend

```
ca. Command Prompt - mysql -u root
mysql> select nopinj,tglpinj,namapem,
-> pinjam.nopol,namaken,hrgsewa
-> from (pinjam inner join peminjam on pinjam.idpem=peminjam.idpem)
-> inner join kend on pinjam.nopol=kend.nopol;
+-----+-----+-----+-----+-----+
| nopinj | tglpinj | namapem | nopol | namaken | hrgsewa |
+-----+-----+-----+-----+-----+
| PJ000001 | 2015-05-02 | Bambang S | AB6106PL | Suzuki Shogun | 30000 |
| PJ000002 | 2015-05-02 | Rini S | AB7809FD | Vario | 50000 |
| PJ000003 | 2015-05-02 | Ferry B | AB9980HK | Avanza | 200000 |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

d. Menggabung tabel Pinjam dengan tabel Peminjam, tabel Kend, tabel Kembali.

```
Command Prompt - mysql -u root
mysql> select kembali.nopinj,namapem,
-> pinjam.nopol,namaken,hrgsewa,
-> <tglkem-tglpinj> as Lama,
-> <hrgsewa*(tglkem-tglpinj)> as JmlSewa,
-> round(<diskon/100>*<(hrgsewa*(tglkem-tglpinj))>,0) as JmlDiskon,
-> round(<hrgsewa*(tglkem-tglpinj)>-
-> <diskon/100>*<(hrgsewa*(tglkem-tglpinj))>,0) as Bayar
-> from kembali inner join
-> <<pinjam inner join peminjam on pinjam.idpem=peminjam.idpem>
-> inner join kend on pinjam.nopol=kend.nopol>
-> on kembali.nopinj=pinjam.nopinj;
+-----+-----+-----+-----+-----+
| nopinj | namapem | nopol   | namaken      | hrgsewa | Lama | JmlSewa | J
mDiskon | Bayar  |
+-----+-----+-----+-----+-----+
| PJ000001 | Bambang S | AB6106PL | Suzuki Shogun | 30000  | 3 | 90000 |
| 4500 | 85500 |
| PJ000002 | Rini S   | AB7809FD | Vario       | 50000  | 2 | 100000 |
| 10000 | 90000 |
| PJ000003 | Ferry B   | AB9980HK | Avanza     | 200000 | 4 | 800000 |
| 0 | 800000 |
+-----+-----+-----+-----+-----+
3 rows in set (0.35 sec)

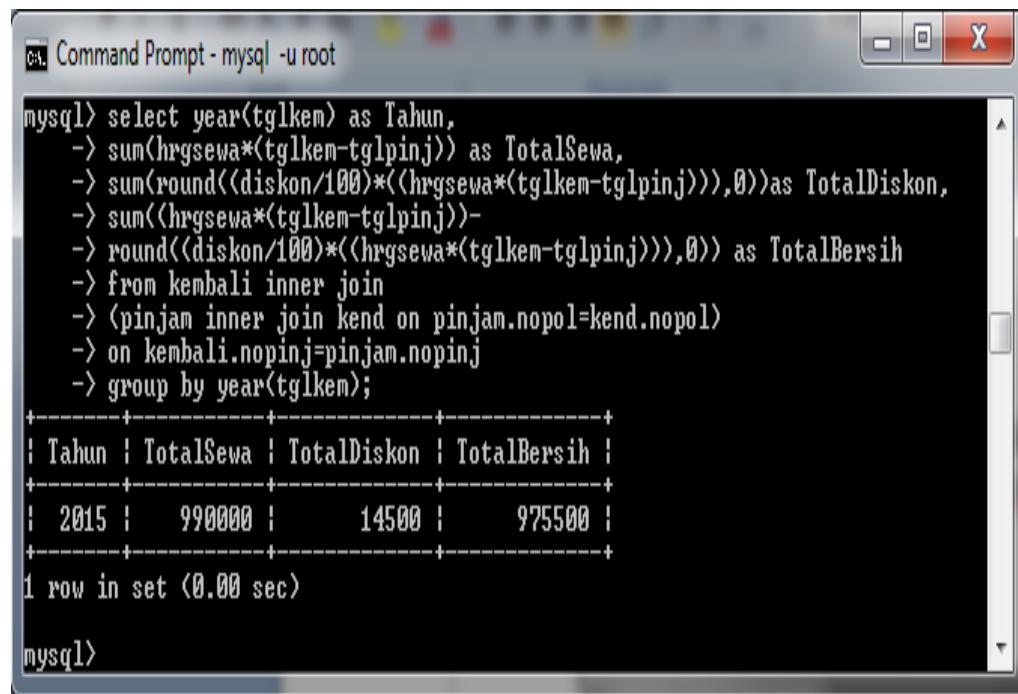
mysql>
```

e. Menggabung tabel Pinjam dengan tabel Kend untuk menghasilkan rekapitulasi Peminjaman.

```
Command Prompt - mysql -u root
mysql> select year(tglpinj) as Tahun,
-> sum(if(kategori="Mobil",1,0)) as JmlMobil,
-> sum(if(kategori="Motor",1,0)) as JmlMotor,
-> count(nopinj) as JmlTotal
-> from pinjam inner join kend on pinjam.nopol=kend.nopol
-> group by year(tglpinj);
+-----+-----+-----+
| Tahun | JmlMobil | JmlMotor | JmlTotal |
+-----+-----+-----+
| 2015 | 1 | 2 | 3 |
+-----+-----+-----+
1 row in set (0.07 sec)

mysql>
```

- f. Menggabung tabel Kembali dengan tabel Pinjam, tabel Kend untuk menghasilkan rekapitulasi pengembalian



The screenshot shows a Windows Command Prompt window titled "Command Prompt - mysql -u root". Inside, a MySQL query is run to calculate annual statistics. The query uses multiple joins between tables "kembali", "pinjam", and "kend" to calculate total rent, discount, and clean amounts for the year 2015. The results are displayed in a table with columns: Tahun, TotalSewa, TotalDiskon, and TotalBersih. The output shows one row for the year 2015.

```
mysql> select year(tglkem) as Tahun,
-> sum(hrgsewa*(tglkem-tglpinj)) as TotalSewa,
-> sum(round((diskon/100)*((hrgsewa*(tglkem-tglpinj))),0))as TotalDiskon,
-> sum((hrgsewa*(tglkem-tglpinj))-_
-> round((diskon/100)*((hrgsewa*(tglkem-tglpinj))),0)) as TotalBersih
-> from kembali inner join
-> (pinjam inner join kend on pinjam.nopol=kend.nopol)
-> on kembali.nopinj=pinjam.nopinj
-> group by year(tglkem);
+-----+-----+-----+
| Tahun | TotalSewa | TotalDiskon | TotalBersih |
+-----+-----+-----+
| 2015 | 990000 | 14500 | 975500 |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

#### Catatan Khusus:

Di dalam **MySQL**, kata **INNER JOIN** dapat diganti dengan **CROSS JOIN**, atau hanya **JOIN** saja. Ketiga kata kunci ini memiliki fungsi yang sama. MySQL menyediakannya agar dapat kompatible dengan SQL dari **RDBMS** lainnya seperti **ORACLE** dan **MS SQL**.

## B. Praktek

Kegiatan praktikum pertemuan ini adalah pemantapan penggunaan MySQL dengan **Command Line** yang lebih khusus untuk menggabung dua tabel atau lebih.

### 4. Mengaktifkan Database

Topik yang dibahas pada pertemuan praktikum ini adalah pengelolaan *database* yang sudah dibuat sebelumnya yaitu DBJUBE.

**Use DBJUBE;**

### 5. Menggabung dua tabel

- Mengabung tabel Beli dengan tabel Suplier

Gunakanlah perintah **SELECT..INNER JOIN..USING** dan **SELECT..INNER JOIN..ON** sehingga bisa menampilkan data seperti gambar di bawah ini.

```
+-----+-----+-----+-----+-----+
| nonotab | tglbeli | nmnotab | kotasup | telpsup |
+-----+-----+-----+-----+-----+
| NB00000001 | 2014-02-01 | PT. Tecno Indotama | Jakarta | 081567898765 |
| NB00000003 | 2014-12-03 | PT. Megah Jaya | Surabaya | 023865754 |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

- Menggabung tabel Jual dengan tabel Pembeli

Gunakanlah perintah **SELECT..INNER JOIN..USING** dan **SELECT..INNER JOIN..ON** sehingga bisa menampilkan data seperti gambar di bawah ini.

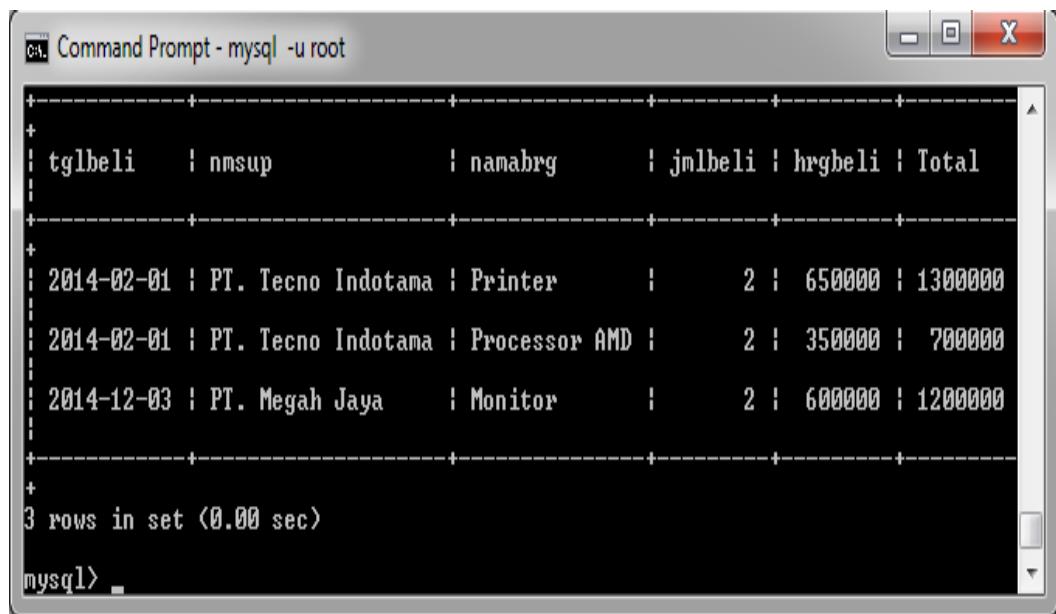
```
+-----+-----+-----+-----+-----+
| nonotaj | tgljual | nmnotaj | kotapem | telpem |
+-----+-----+-----+-----+-----+
| NJ00000001 | 2014-06-01 | PT. Adi Jaya | Solo | 081234543675 |
| NJ00000002 | 2014-08-02 | Ny Suci Darwanti | Semarang | 087232432543 |
| NJ00000003 | 2014-12-03 | PT. Adi Jaya | Solo | 081234543675 |
| NJ00000004 | 2014-12-12 | Mr. Burhanudin | Yogyakarta | 0274543212 |
| NJ00000005 | 2015-02-01 | Mr. Burhanudin | Yogyakarta | 0274543212 |
| NJ00000006 | 2015-05-01 | PT. Adi Jaya | Solo | 081234543675 |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

## 6. Menggabung lebih dari dua tabel

- a. Menggabung tabel Beli, DBeli, Suplier, Barang untuk menampilkan daftar pembelian

Gunakanlah perintah **SELECT..INNER JOIN..USING** dan **SELECT..INNER JOIN..ON** sehingga bisa menampilkan data seperti gambar di bawah ini.



c:\ Command Prompt - mysql -u root

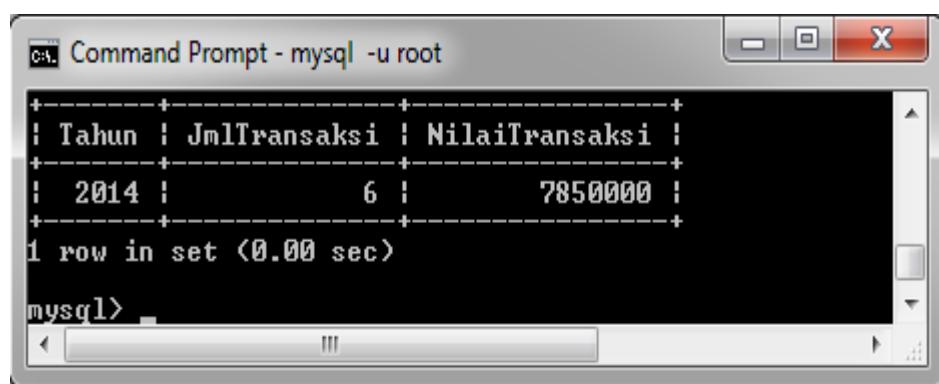
tgmbeli	nmsup	namabrg	jmlbeli	hrgbeli	Total
2014-02-01	PT. Tecno Indotama	Printer	2	650000	1300000
2014-02-01	PT. Tecno Indotama	Processor AMD	2	350000	700000
2014-12-03	PT. Megah Jaya	Monitor	2	600000	1200000

3 rows in set (0.00 sec)

mysql> \_

- b. Menggabung tabel Beli, DBeli, Suplier, Barang untuk menampilkan rekap pembelian

Gunakanlah perintah **SELECT..INNER JOIN..USING** dan **SELECT..INNER JOIN..ON** sehingga bisa menampilkan data seperti gambar di bawah ini.



c:\ Command Prompt - mysql -u root

Tahun	JmlTransaksi	MilaiTransaksi
2014	6	7850000

1 row in set (0.00 sec)

mysql> \_

- c. Menggabung tabel Jual, Djual, Pembeli, Barang untuk menampilkan daftar penjualan

Gunakanlah perintah **SELECT..INNER JOIN..USING** dan **SELECT..INNER JOIN..ON** sehingga bisa menampilkan data seperti gambar di bawah ini.

```

C:\ Command Prompt - mysql -u root
+-----+-----+-----+-----+-----+
| tgljual | nmpem      | namabrg      | jmljual | hrgjual | Total
+-----+-----+-----+-----+-----+
| 2014-12-12 | Mr. Burhanudin | RAM 256 MB | 3 | 90000 | 270000
| 2014-12-12 | Mr. Burhanudin | RAM 512 MB | 4 | 115000 | 460000
| 2015-02-01 | Mr. Burhanudin | Printer      | 6 | 700000 | 4200000
| 2015-02-01 | Mr. Burhanudin | Processor AMD | 6 | 380000 | 2280000
| 2014-06-01 | PT. Adi Jaya   | Processor Intel | 2 | 475000 | 950000
| 2014-06-01 | PT. Adi Jaya   | RAM 128 MB   | 4 | 60000 | 240000
| 2014-06-01 | PT. Adi Jaya   | RAM 256 MB   | 2 | 90000 | 180000
| 2014-12-03 | PT. Adi Jaya   | Printer      | 4 | 700000 | 2800000
| 2014-12-03 | PT. Adi Jaya   | Processor Intel | 3 | 475000 | 1425000
| 2014-08-02 | Ny Suci Darwanti | Monitor      | 5 | 675000 | 3375000
+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql>

```

- d. Menggabung tabel Jual, Djual, Pembeli, Barang untuk menampilkan rekap penjualan

Gunakanlah perintah **SELECT..INNER JOIN..USING** dan **SELECT..INNER JOIN..ON** sehingga bisa menampilkan data seperti gambar di bawah ini.

```

C:\ Command Prompt - mysql -u root
+-----+-----+-----+
| Tahun | JmlTransaksi | NilaiTransaksi |
+-----+-----+-----+
| 2014 | 8 | 9700000 |
| 2015 | 2 | 6480000 |
+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>

```

## C. Tugas

Untuk memperlancar materi pada praktikum ini maka gunakan *database* Hotel yang pernah dibuat pada modul 7 yang lalu.

Gunakanlah perintah **SELECT..INNER JOIN..USING** dan **SELECT..INNER JOIN..ON** sehingga tabel-tabel yang ada di *database* Hotel dapat dikelola sedemikian rupa dengan berbagai variasi.

## BAB 10

# MENGELOLA DATABASE DENGAN PHPMYADMIN

### A. Mengenal PHPMyAdmin

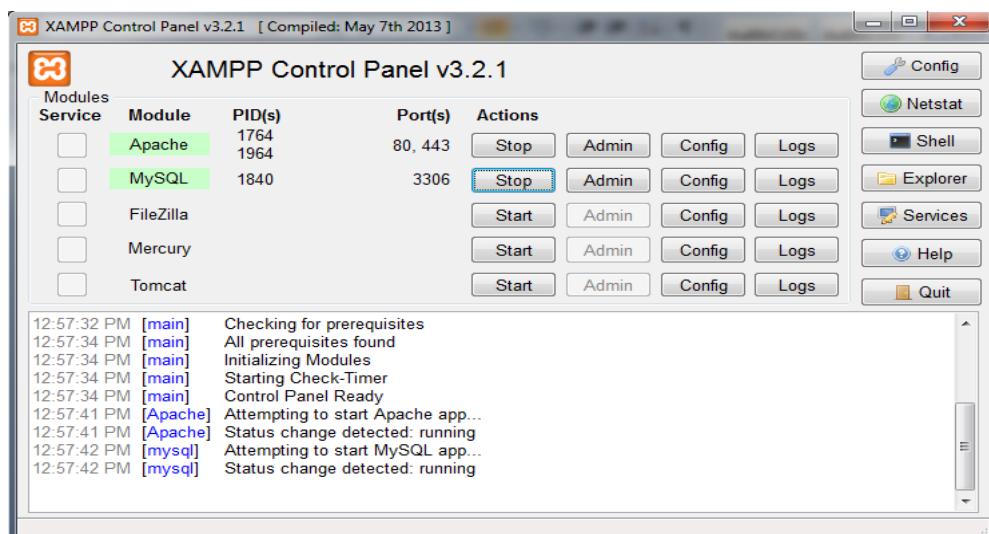
#### 1. Pengantar

Kemungkinan ada beberapa diantara pembaca yang belum mengenal apa itu phpMyAdmin, karena biasanya dalam pembuatan *database* masih menggunakan MySQL manual (*command line*). Disini akan dipaparkan cara pembuatan *database* dengan menggunakan phpMyAdmin yang sebenarnya jauh lebih mudah dibandingkan dengan MySQL manual.

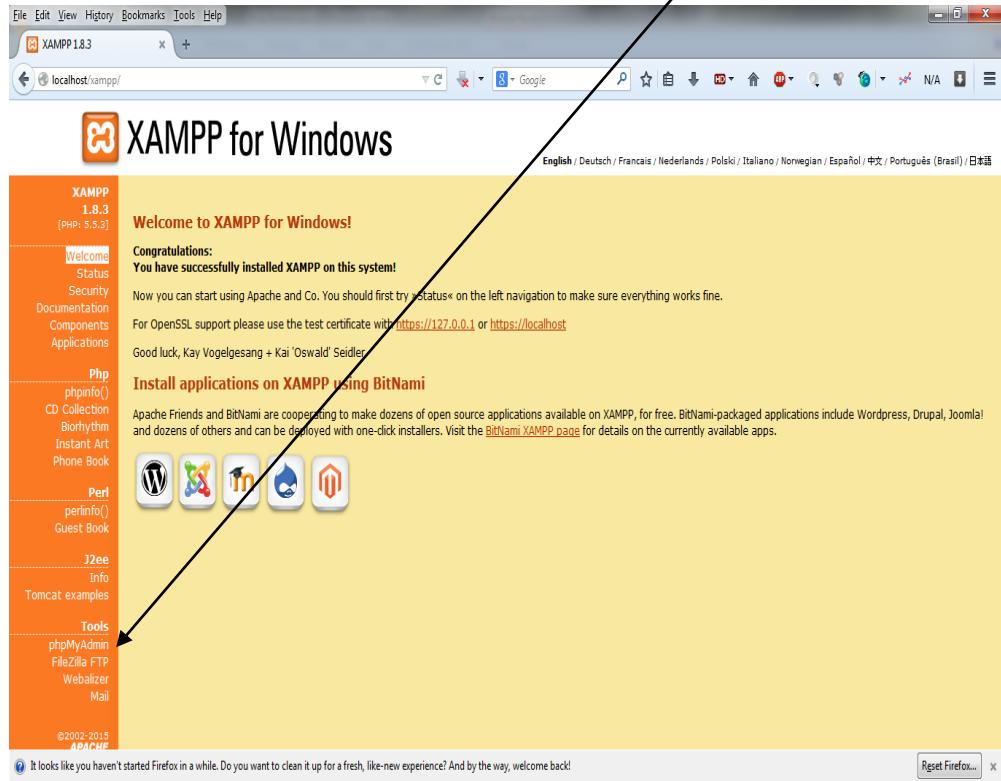
Apa itu **PhpMyAdmin**? . PhpMyAdmin adalah suatu *software* yang ditulis dalam bahasa pemrograman PHP yang digunakan untuk membuat *database/menangani administrasi MySQL*, karena *software* ini mendukung berbagai operasi MySQL. Dengan menggunakan phpMyAdmin pembuatan *database* tanpa perlu menghafal dan menuliskan query seperti pada saat membuat *database* menggunakan MySQL manual, karena bisa dibilang phpMyAdmin ini adalah MySQL dengan *Graphical User Interface (GUI)*, jadi hanya perlu memasukkan *field-field* apa saja yang ingin dibuat tanpa perlu mengetikkan barisan-barisan query MySQL.

#### 2. Persiapan Menggunakan PHPMyAdmin

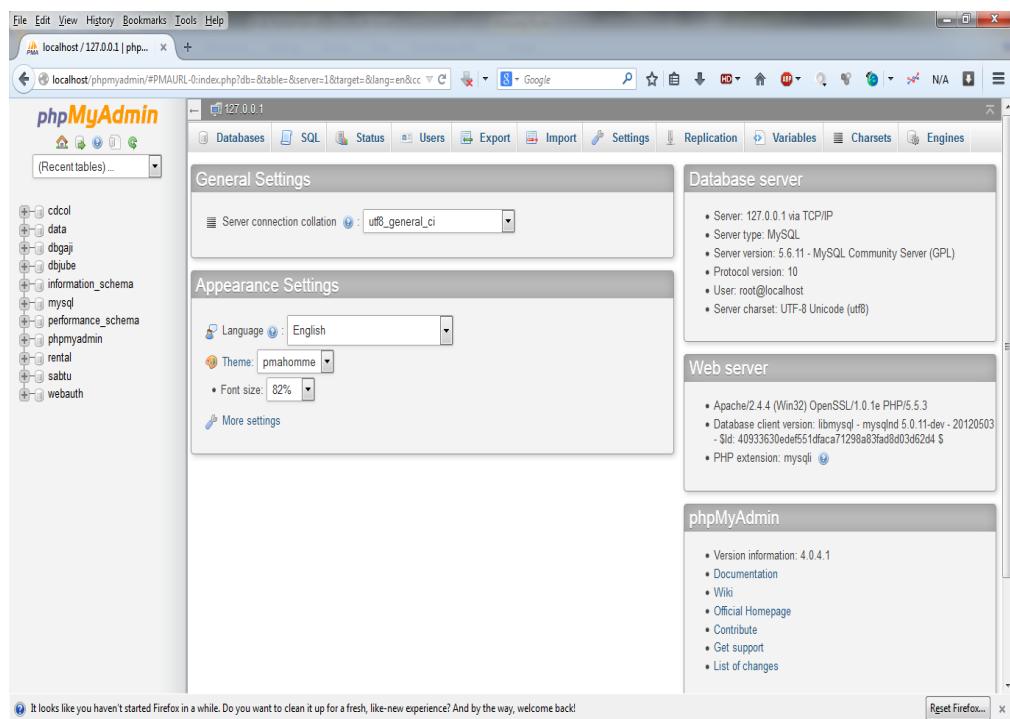
- a. Harus memiliki *software* XAMPP yaitu sebuah *web server apache* yang mensupport php programming, di dalamnya tersedia *database server mysql*, apache, PHP support, dan modul lainnya.
- b. Bila XAMPP sudah diinstal, aktifkan Apache dan MySQL pada XAMPP tersebut dengan mengklik button ‘Start’ pada kolom **Action** di tiap modulennya.



- c. Bila keduanya sudah running, buka browser kemudian masuk ke alamat <http://localhost/xampp/>. kemudian pilih phpMyAdmin pada tools yang disediakan



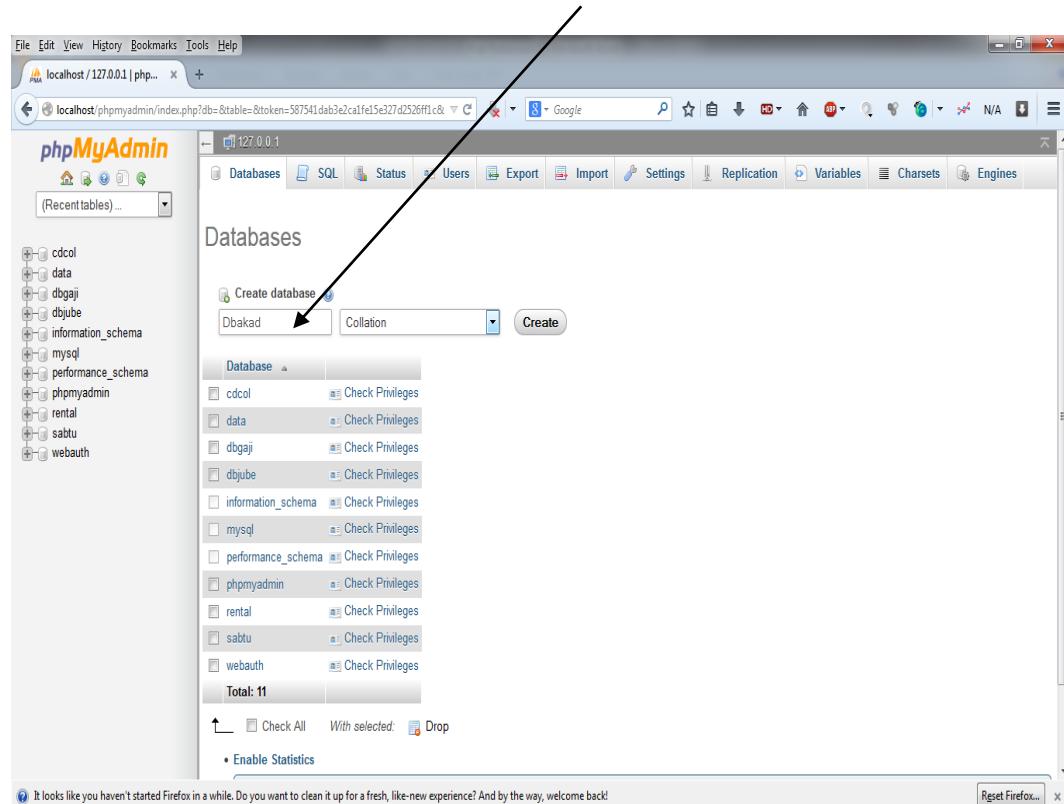
- d. Tunggu beberapa saat sampai ditampilkan layar kerja PHPMyAdmin seperti gambar berikut ini



Pada saat setelah ditampilkan layar seperti gambar di atas, maka pengelolaan basis data sudah dapat dilakukan.

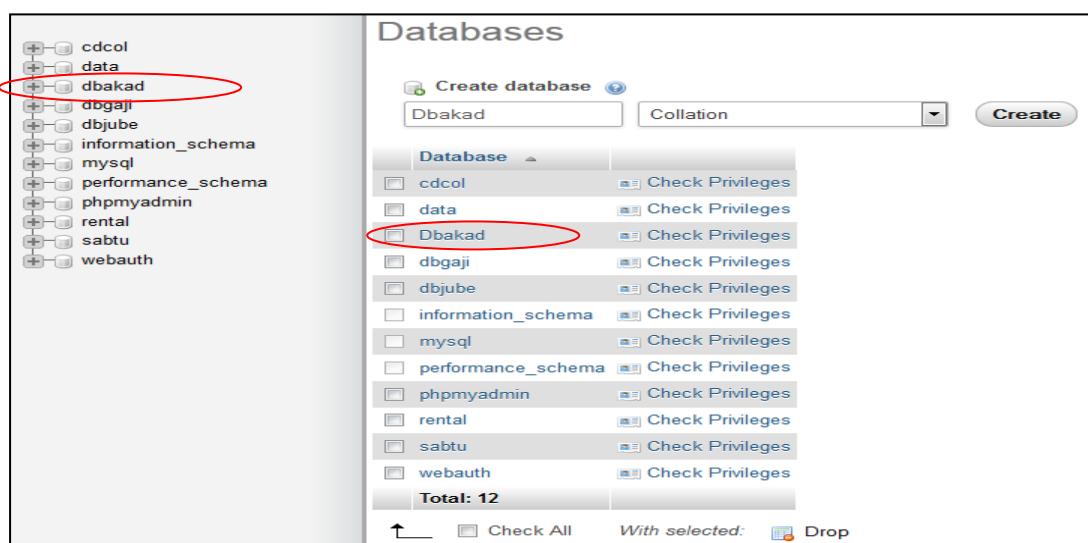
### 3. Membuat Database

Cara membuat *database* adalah dengan mengklik attribute ‘**Database**’ lalu isikan nama *database* sesuai dengan keinginan pada ‘**Create database**’ kemudian **Create**. Sebagai contoh akan dibuat *database* mengenai akademik mahasiswa, dengan nama **Dbakad**.



The screenshot shows the phpMyAdmin interface for MySQL version 5.6.37. The 'Databases' tab is selected. On the left, there's a tree view of existing databases: cdcoll, data, dbgaji, dbjube, information\_schema, mysql, performance\_schema, phpmyadmin, rental, sabtu, webauth. In the main area, there's a 'Create database' form with 'Dbakad' entered in the 'Database' field. Below it is a list of databases with their status: cdcoll (Check Privileges), data (Check Privileges), dbgaji (Check Privileges), dbjube (Check Privileges), information\_schema (Check Privileges), mysql (Check Privileges), performance\_schema (Check Privileges), phpmyadmin (Check Privileges), rental (Check Privileges), sabtu (Check Privileges), webauth (Check Privileges). At the bottom, there are buttons for 'Check All' and 'Drop'.

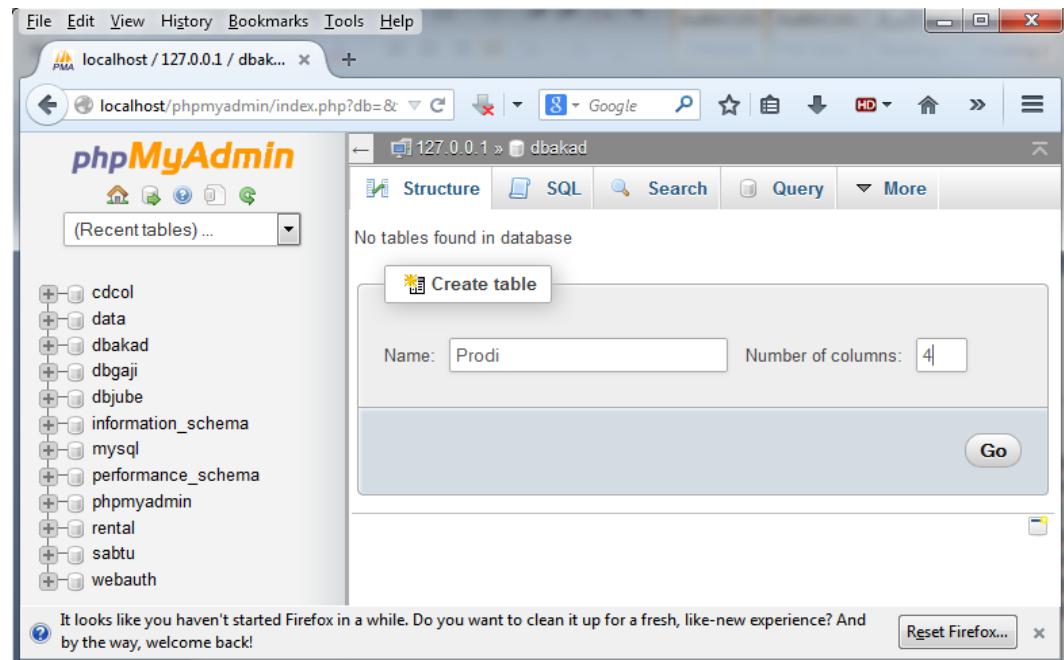
- Ketik nama *database* yang dibuat (misal Dbakad),
- Klik tombol **Create** sehingga *database* yang dibuat akan masuk ke dalam kumpulan *database* yang sudah ada seperti gambar berikut:



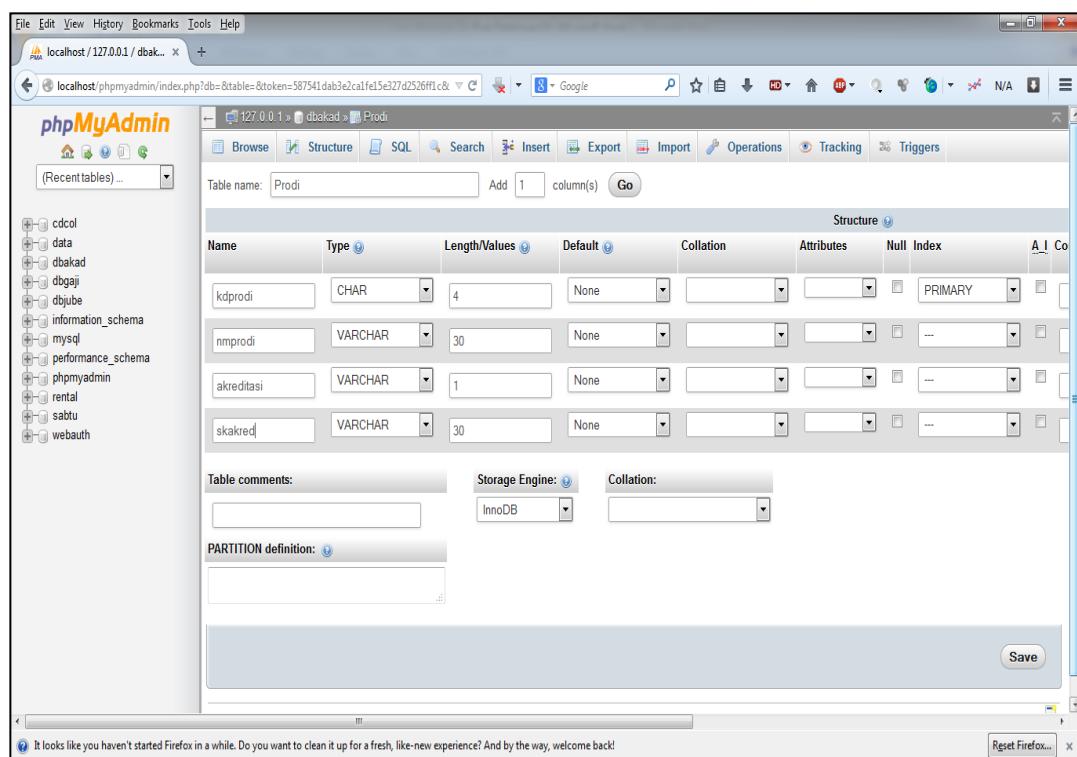
This screenshot shows the same phpMyAdmin interface after the 'Dbakad' database has been created. The tree view on the left now includes 'Dbakad' under the 'data' database. The main 'Databases' list also includes 'Dbakad'. Both instances of 'Dbakad' are highlighted with red circles. The rest of the interface is identical to the previous screenshot, showing other databases and their privileges.

#### 4. Membuat Tabel

Sekarang akan mulai membuat tabel. Seperti yang sudah diterangkan di awal, membuat tabel dengan phpMyAdmin tidak perlu membuat barisan-barisan query (pada MySQL kita menggunakan CREATE) , cukup dengan mengklik *database* ‘Dbakad’, klik Structure.



- Klik **Structure**
- Ketik nama tabel, misal Prodi
- Masukkan jumlah kolom, misal 4 kolom
- Klik ‘Go’



- e. Masukkan *field* yang akan dibuat, misalnya kdprodi (char 4), nmprodi (varchar 30), akreditasi (varchar 1), dan skakred (varchar 30)[
- f. Tentukan jenis datanya dan lebarnya (ukurannya)
- g. Tentukan kunci utama (*primary*) pada kolom Index (kdprodi dijadikan *primary key*)
- h. Klik tombol **Save** untuk menyimpan tabel

Jika menginginkan untuk membuat tabel yang baru, lakukan langkah a sampai h. Untuk melengkapi *database* di atas, dapat dibuat tabel yang lain yaitu:

- 1) Tabel Mahasiswa dengan 5 field saja yaitu field NIM Char (8), Namamhs Varchar(30), Alamat Varchar (40) , Telp Varchar (12), Kdprodi Char(4).
- 2) Tabel Matakuliah dengan 5 field yaitu field Kodemk Char (15), Namamk varchar (40), SKS Int(1), Sem Varchar (4), Kdprodi Char (4).

## 5. Mengisi Data

- a. Klik tombol 'Insert'
- b. Isikan record sesuai dengan urutan field, yaitu kdprodi, nmprodi, akreditasi, skakred.
- c. Klik tombol Go jika sudah selesai

Column	Type	Function	Null	Value
kdprodi	char(4)			KP01
nmprodi	varchar(30)			Manajemen Informatika
akreditasi	varchar(1)			B
skakred	varchar(30)			001/Dikti/SK/D-III/2015

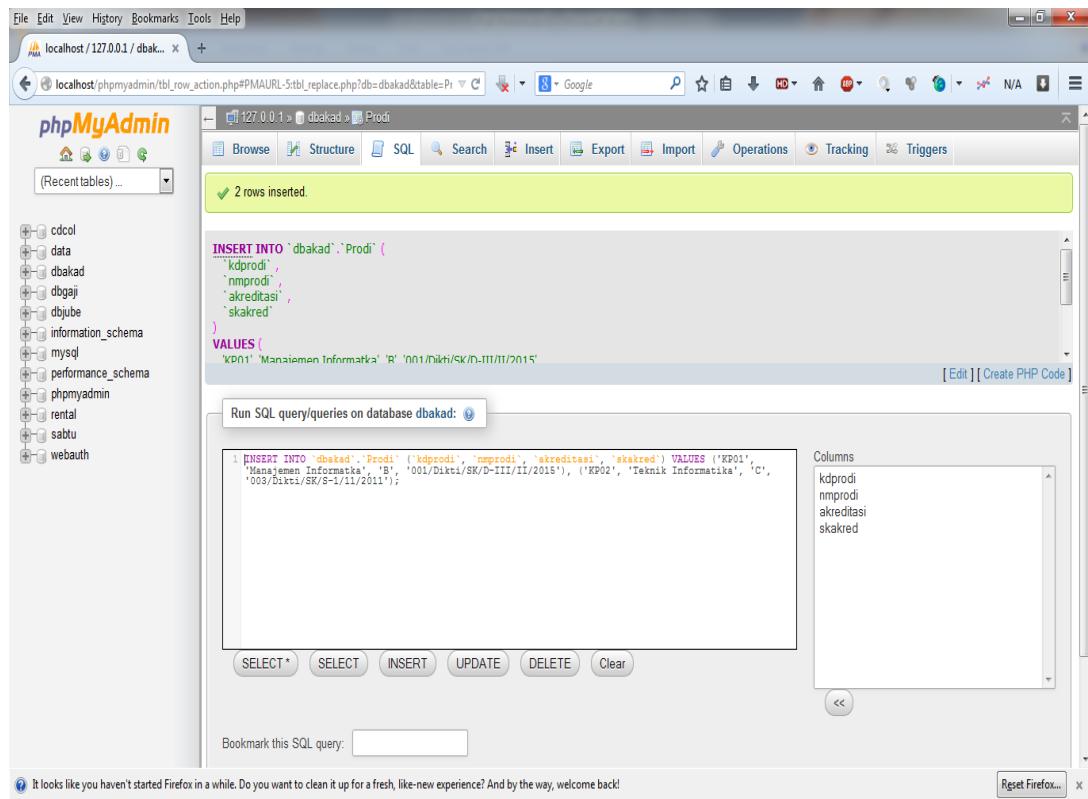
  

Column	Type	Function	Null	Value
kdprodi	char(4)			KP02
nmprodi	varchar(30)			Teknik Informatika
akreditasi	varchar(1)			C
skakred	varchar(30)			003/Dikti/SK/S-1/11/2011

### Contoh data

KP01	Manajemen Informatika	B	001/Dikti/SK/D-III/2015
KP02	Teknik Informatika	C	003/Dikti/SK/S-1/11/2011

Setelah mengklik tombol **Go** maka data yang diisikan akan dimasukkan ke dalam tabel. Selanjutnya tampilan akan berubah seperti gambar di bawah ini.



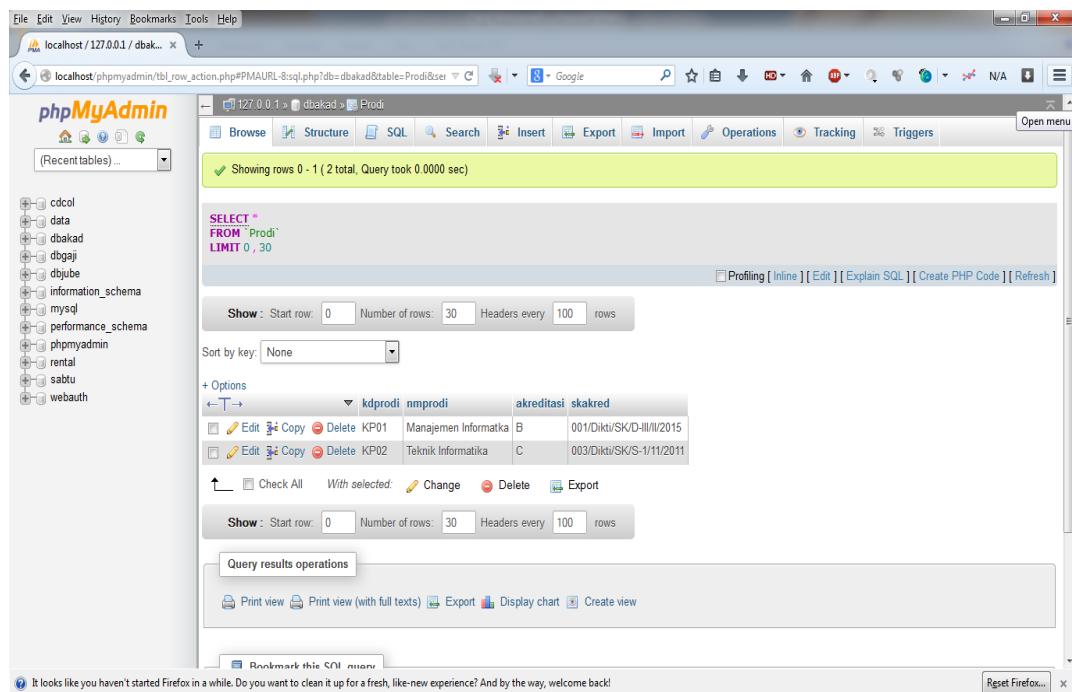
The screenshot shows the phpMyAdmin interface for the 'dbakad' database. The 'Structure' tab is selected. In the SQL panel, the following SQL code was run:

```
INSERT INTO `dbakad`.`Prodi` (
    `kdprodi`,
    `nmprodi`,
    `akreditasi`,
    `skakred`
)
VALUES (
    'KP01', 'Manajemen Informatika', 'B', '001/Dikti/SK/D-III/I/2015'
)
```

The status bar at the top indicates "2 rows inserted." Below the SQL panel is a query editor window titled "Run SQL query/queries on database dbakad:" containing the same SQL code. To the right of the editor is a "Columns" panel listing the columns of the 'Prodi' table: kdprodi, nmprodi, akreditasi, and skakred. At the bottom of the interface, there is a message: "It looks like you haven't started Firefox in a while. Do you want to clean it up for a fresh, like-new experience? And by the way, welcome back!"

## 6. Menampilkan Data

Untuk menampilkan data cukup mengklik tombol **Browse**, sehingga akan ditampilkan seperti gambar berikut.



The screenshot shows the phpMyAdmin interface for the 'dbakad' database. The 'Browse' tab is selected. The SQL panel contains the following query:

```
SELECT *
FROM `Prodi`
LIMIT 0 , 30
```

The results table shows two rows of data:

	kdprodi	nmprodi	akreditasi	skakred
<a href="#">Edit</a>	KP01	Manajemen Informatika	B	001/Dikti/SK/D-III/I/2015
<a href="#">Edit</a>	KP02	Teknik Informatika	C	003/Dikti/SK/S-1/11/2011

At the bottom of the interface, there is a message: "It looks like you haven't started Firefox in a while. Do you want to clean it up for a fresh, like-new experience? And by the way, welcome back!"

## 7. Merubah Data

Untuk merubah data, cukup mengklik tanda edit data (tanda pensil) pada baris (*record*) yang akan dirubah. Sebagai contoh akan dirubah data yang kedua (*record* ke-2) yaitu dari prodi Teknik Informatika dirubah menjadi Sistem Informasi.

The screenshot shows the phpMyAdmin interface for a database named 'dbakad'. On the left, there's a tree view of tables: cdcoll, data, dbakad, dbgaji, dbjube, information\_schema, mysql, performance\_schema, phpmyadmin, rental, sabtu, webauth. The main area is titled 'Prodi' and shows a form for editing a row. The columns are: kdprodi (char(4)), nmprodi (varchar(30)), akreditasi (varchar(1)), and skakred (varchar(30)). The current values are: kdprodi = KP01, nmprodi = Manajemen Informatika, akreditasi = B, and skakred = 001/Dikti/SK/D-III/II/2015. The 'nmprodi' field has been changed to 'Sistem Informasi'. At the bottom, there are buttons for 'Save' and 'Go'.

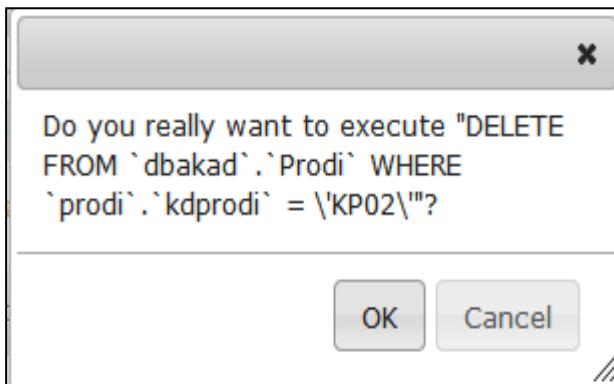
## 8. Menghapus Data

Untuk menghapus data cukup mengklik tanda **delete** data (tanda minus) pada baris (*record*) yang akan dihapus. Sebagai contoh akan dihapus data yang kedua (*record* ke-2) yaitu dari prodi Sistem Informasi (hasil perubahan).

The screenshot shows a MySQL query browser displaying a table with four columns: kdprodi, nmprodi, akreditasi, and skakred. There are two rows of data. The first row (kdprodi: KP01) has edit and delete buttons. The second row (kdprodi: KP02) also has edit and delete buttons. Below the table, there are buttons for 'Check All', 'With selected:', 'Change', 'Delete', and 'Export'. At the bottom, there are fields for 'Show', 'Start row', 'Number of rows', and 'Headers every'.

Merubah data      Menghapus data

Setelah diklik tombol **Delete**, maka akan ditampilkan kotak dialog untuk memastikan apakah penghapusan benar-benar mau dilakukan atau tidak. Jika yakin klik tombol **OK**.



## 9. Menampilkan Data Dengan SQL

SQL diperlukan jika ingin menampilkan data lebih kompleks lagi. Perintah yang diberikan dapat mengacu kepada perintah-perintah yang telah dicobakan pada materi praktikum dengan MySQL dengan **command line**.

- Klik tab SQL
- Tulis perintah SQL yang diinginkan (misal menggabung tabel Mahasiswa dengan tabel Prodi)
- Klik tombol **Go** untuk melihat hasil

```

SELECT m.NIM, m.namamhs, m.alamat, m.telp, p.nmprodi, p.akreditasi
FROM mahasiswa m, prodi p
WHERE m.kdprodi = p.kdprodi
LIMIT 0 , 30

```

NIM	namamhs	alamat	telp	nmprodi	akreditasi
14240001	Sri Pratiwi	Jl. Janti No. 43 Yogy	081234565456	Manajemen Informatika	B
14240002	Bambang Sungkono	Jl. Gejayan No. 87 Yogy	085889765546	Manajemen Informatika	B

Show : Start row: 0 Number of rows: 30 Headers every 100 rows

Query results operations

[Print view](#) [Print view \(with full texts\)](#) [Export](#) [Display chart](#) [Create view](#)

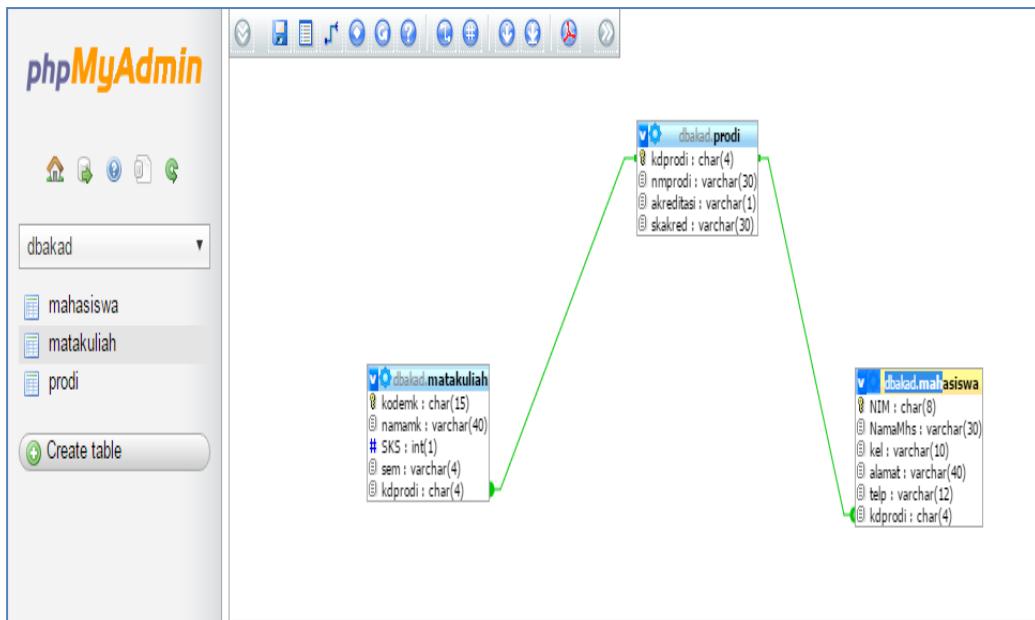
## 10. Membuat Relasi Antar Tabel

Tidak harus menggunakan software **Visio**, dengan **phpMyAdmin** juga dapat **membuat Relasi Antar Tabel pada Database MySQL** yang dikelola. Syarat membuat Relasi Antar Tabel adalah minimal ada 2 tabel yang datanya berhubungan, ditandai dengan adanya kolom kunci Primer yang menjadi tamu (kunci Tamu) di dalam tabel lainnya.

Sebagai contoh adalah *database dbakad* yang sudah dibuat sebelumnya. Pada *database* tersebut terdapat tabel **prodi**, **tabel mahasiswa** dan tabel **matakuliah**, ketiganya ada hubungan data (relasi data), yaitu kolom/*field* **kdprodi** menjadi kunci tamu di dalam tabel **mahasiswa** dan tabel **matakuliah**. Kolom/*field* **kdprodi** adalah kolom Kunci (*Primary Key*) yang dimiliki oleh tabel **prodi**, sehingga kolom tersebut dapat masuk ke dalam tabel **mahasiswa** dan tabel **matakuliah** sebagai kolom kunci tamu (*Foreign Key*). Berikut ini adalah cara membuat Relasi Tabel-nya:

- Klik nama tabel yang memiliki kolom tamu (misalnya tabel **mahasiswa**, memiliki kolom tamu **kdprodi** dari tabel **prodi**). Klik menu **structure**.
- Klik menu **Relation View**.
- Selanjutnya, relasinya nama kolom pada bagian **kdprodi**. Pilih kolom **kdprodi** dari tabel pemiliknya, yaitu tabel **prodi**, di sana tertulis **dbakad.prodi.kdprodi**.
- Ulangi langkah a sampai c untuk merelasikan tabel **matakuliah** dengan tabel **prodi**.
- Klik tombol **Save**.
- Untuk melihat hasilnya, klik pada nama *database* (**dbakad**), klik menu **Designer**.

Lihatlah hasilnya, akan terlihat gambar tabel **prodi** dan tabel **mahasiswa** akan berelasi, ada garis menghubungkan kedua tabel tersebut. Begitu juga dengan tabel prodi dan tabel matakuliah juga akan berelasi.



## 11. Membuat View

View adalah perintah *query* yang disimpan pada *database* dengan suatu nama tertentu, sehingga bisa digunakan setiap saat untuk melihat data tanpa menuliskan ulang query tersebut.

Perintah Pembuatan View

Perintah yang digunakan adalah **CREATE VIEW**.

Berikut adalah *syntax* lengkapnya :

```
CREATE
[OR REPLACE]
VIEW view_name [(column_list)]
AS select_statement
```

Dapat menggunakan opsi **OR REPLACE** jika ingin mengganti view dengan nama yang sama dengan perintah tersebut. Jika tidak, maka perintah CREATE VIEW akan menghasilkan error jika nama view yang ingin dibuat sudah ada sebelumnya.

Keterangan:

- *view\_name*= nama view.
- *column\_list*= daftar nama kolom dari hasil query select.. pada badan view. Jumlah kolom pada daftar nama kolom ini harus sama dengan hasil dari query select.
- *select\_statement*= query SQL select yang akan dilakukan.

## Contoh Penggunaan

1. Membuat view dari relasi antara tabel "mahasiswa", dengan tabel "prodi" dari database **dbakad** dengan nama "view\_mhs".

Perintahnya adalah sebagai berikut :

```
CREATE VIEW view_mhs
```

```
AS
```

```
SELECT m1.nim, m1.namamhs, m1.kdprodi, m2.nmprodi
```

```
FROM
```

```
mahasiswa m1 JOIN prodi m2 ON m1.kdprodi = m2.kdprodi;
```

2. Membuat view dari antara tabel "matakuliah", dengan tabel "prodi" dari database **dbakad** dengan nama "view\_mtkl".

Perintahnya adalah sebagai berikut :

```
CREATE VIEW view_mtkl
```

```
AS
```

```
SELECT m1.kodemk, m1.namamk, m1.SKS, m1.sem, m1.kdprodi,
```

```
m2.nmprodi
```

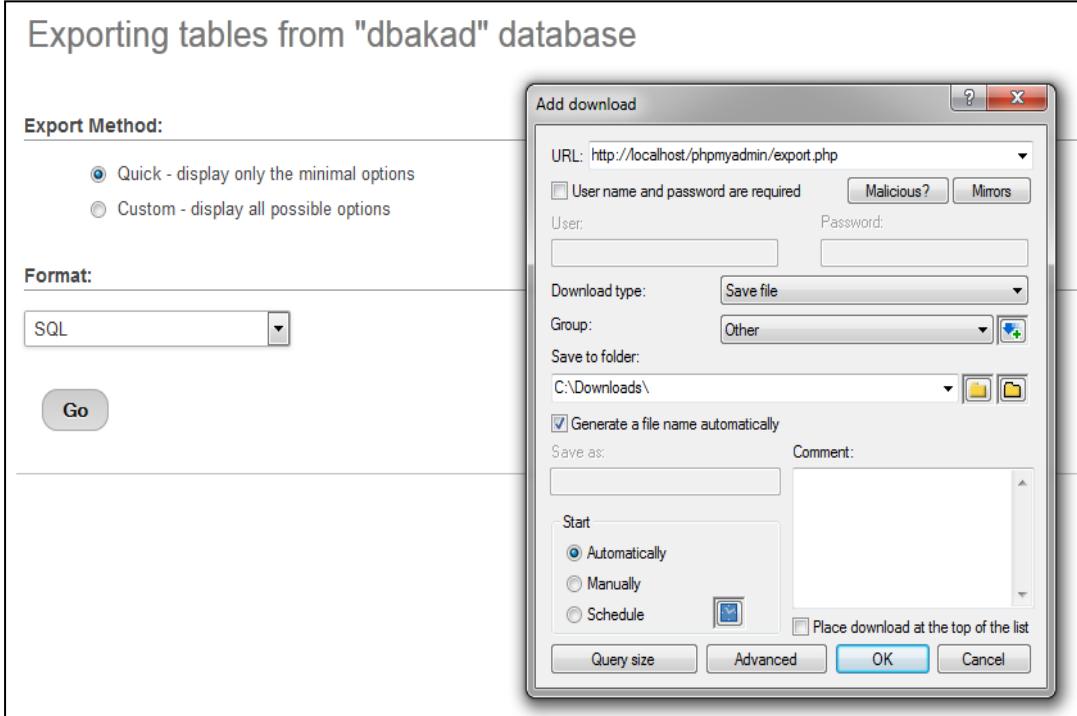
```
FROM
```

```
matakuliah m1 JOIN prodi m2 ON m1.kdprodi = m2.kdprodi;
```

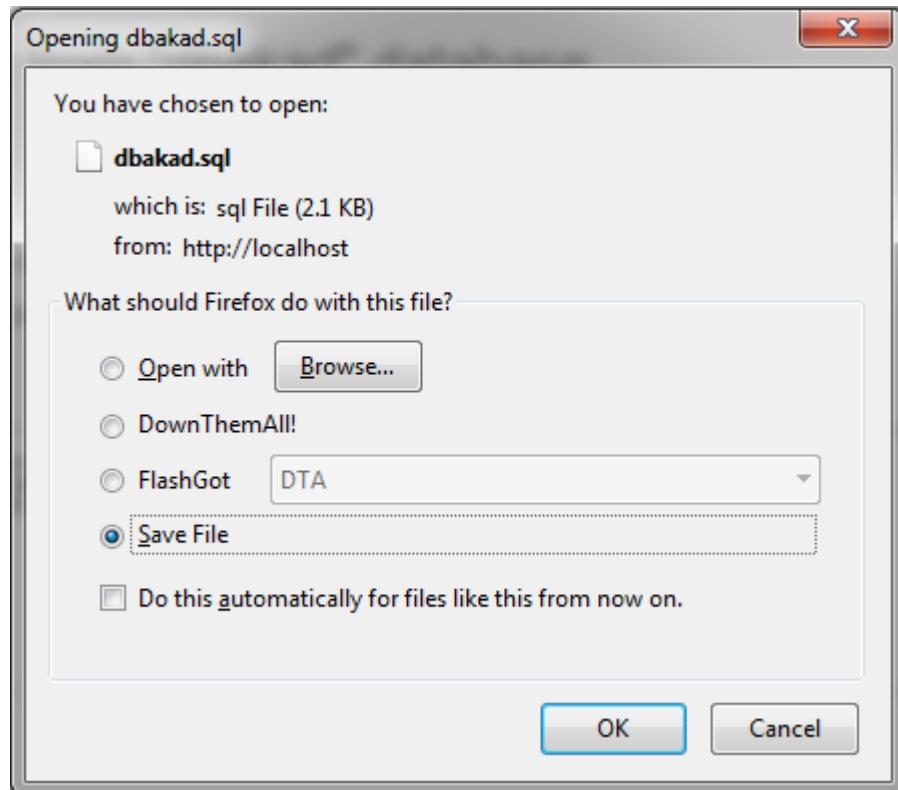
## 12. Export Database

Export *database* diperlukan jika diinginkan *database* tersimpan pada *directory* lain dalam bentuk .sql.

Untuk melakukan proses **export** dapat dilakukan dengan mengklik tab page **Export**, selanjutnya klik tombol **Go**, sehingga akan ditampilkan kotak dialog untuk menyimpan file *database*. Klik tombol **OK**.



Setelah diklik tombol **OK**, akan ditampilkan kotak dialog berikutnya seperti gambar di bawah ini. Klik tombol **OK** untuk mengakhiri.



### 13. Import Database

Proses *import database* diperlukan apabila *database* yang telah tersimpan pada directory tertentu dalam bentuk format .sql ingin digunakan kembali. Untuk melakukan proses import dapat dilakukan dengan mengklik tab page Import, selanjutnya klik tombol **Browse**. Setelah selesai klik tombol **Go**.

Importing into the current server

**File to Import:**

File may be compressed (gzip, bzip2, zip) or uncompressed.  
A compressed file's name must end in **[format].[compression]**. Example: **.sql.zip**

Browse your computer:  dbakad.sql (Max: 2,048KiB)

Character set of the file:

**Partial Import:**

Allow the interruption of an import in case the script detects it is close to the PHP timeout limit. (This might be a good way to import large files, however it can break transactions.)

Number of rows to skip, starting from the first row:

**Format:**

**Format-Specific Options:**

SQL compatibility mode:

Do not use **AUTO\_INCREMENT** for zero values

## B. Praktek

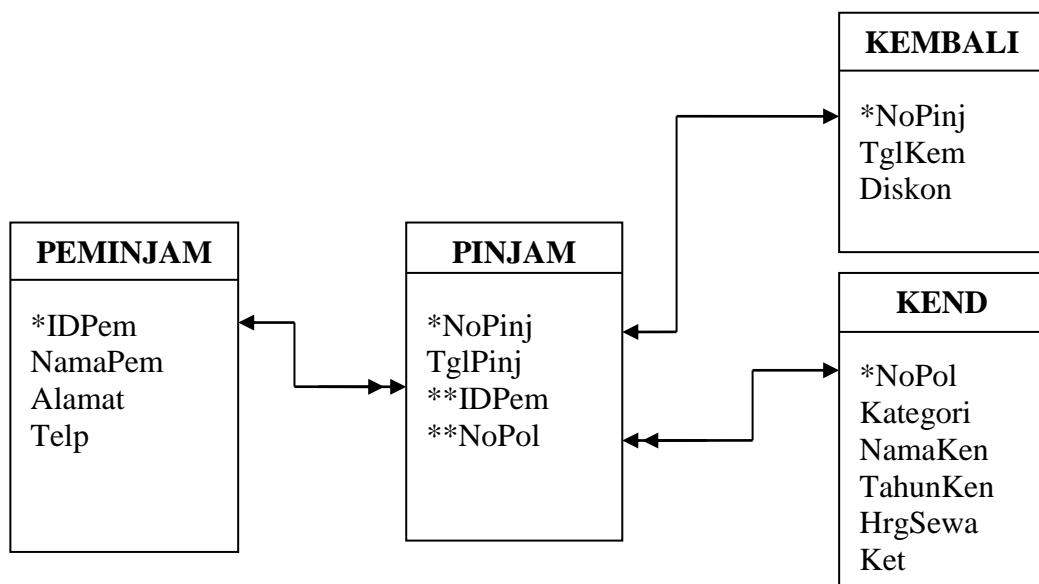
Kegiatan praktikum pertemuan ini adalah menggunakan PHPMyAdmin untuk mengelola *Database*.

### 7. Membuat *Database*

Topik yang dibahas pada pertemuan praktikum ini adalah pengelolaan *database rental kendaraan* yang diberi nama **DBREN**.

### 8. Membuat Tabel

Tabel yang dibuat ada sebanyak empat tabel yaitu tabel **Peminjam**, tabel **Kend**, tabel **Pinjam**, dan tabel **Kembali**. Untuk membuat tabel, mohon diperhatikan *field-field* yang akan diciptakan termasuk jenis data yang sesuai. Berdasarkan rancangan tabel, maka *field* yang diberi tanda satu bintang (\*) berarti *field* tersebut adalah *primary key*.



Silahkan membuat tabel dengan langkah yang telah dibahas di atas.

### 9. Mengisi Data

Setelah semua tabel dibuat, silahkan mengisi masing-masing tabel dengan data yang sesuai (dapat menggunakan contoh data yang ada pada modul sebelumnya).

### 10. Menampilkan data dengan SQL

Untuk menampilkan data menggunakan SQL, silahkan dicobakan perintah-perintah SQL yang telah dicobakan pada modul-modul sebelumnya

### C. Tugas

Untuk memperlancar materi pada praktikum ini maka:

1. Buatlah *database* baru mengenai gaji pegawai
2. Buatlah tabel-tabel yang dibutuhkan
3. Isilah data minimal 10 *record* untuk masing-masing tabel.
4. Lakukan proses menampilkan data dengan SQL (catat perintah yang dibuat dan dipindah ke MS Word. Selanjutnya hasil SQL yang dibuat dikirim bersama-sama dengan *database* yang dibuat.

Kirimkan tugas yang dibuat ke email: [zaidirtan@gmail.com](mailto:zaidirtan@gmail.com) paling lambat 3 hari sejak hari praktikum.

## BAB 11

### MENGELOLA DATABASE DENGAN PHPMYADMIN EXPAND

#### A. Contoh Studi Kasus Database Rumah Sakit

##### 1. Pengantar

**Rumah sakit** adalah tempat yang digunakan untuk memeriksa maupun merawat orang-orang yang sedang sakit. Dalam rumah sakit ini pasti terdapat banyak sekali orang yang memeriksakan kesehatannya setiap hari. Tentunya rumah sakit ini butuh sebuah pengelolaan untuk semua data-data tersebut. Data-data tersebut antara lain data pasien, data dokter, data petugas jaga, data pendaftaran, data pemeriksaan maupun data pasien yang melakukan pembayaran.

Maka, dibuatlah **sistem informasi rumah sakit**. **Sistem informasi rumah sakit** digunakan untuk mempermudah dalam pengelolaan data pada rumah sakit. Sistem ini tentunya sudah menggunakan metode komputerisasi. Karena dengan menggunakan metode komputerisasi, proses penginputkan data, proses pengambilan data maupun proses pengupdate data sangat mudah, cepat dan akurat.

##### 2. Objek yang dipakai

Berdasarkan analisis untuk pembuatan sistem informasi rumah sakit rawat jalan, maka objek yang dipakai yaitu:

- a. Petugas
- b. Dokter
- c. Pasien

##### 3. Penentuan Entitas

Berdasarkan objek di atas dan kebutuhan dalam pengolahan data rawat jalan, maka entitas yang diperlukan yaitu:

- a. **Petugas**: menyimpan data petugas jaga atau pengguna sistem
- b. **Dokter**: menyimpan data dokter yang akan melayani pasien
- c. **Pasien**: menyimpan data identitas/data induk pasien
- d. **Daftar**: menyimpan data pendaftaran pasien
- e. **Periksa**: menyimpan data pemeriksaan pasien
- f. **Bayar**: menyimpan data administrasi pembayaran pasien

##### 4. Penentuan Atribut

- a. **Petugas**: kdpet char(4) *primary key*, nmpet varchar(40), alpet varchar(50), telpet varchar(15), jabatan varchar(20), nmuser varchar(20), pasw varchar(20), levelp varchar(20).
- b. **Dokter**: kddok char(4) *primary key*, nmdok varchar(40), aldok varchar(50), teldok varchar(15), kategori varchar(10), tarif int(10).
- c. **Pasien**: idpas int(10) *primary key*, nmpas varchar(40), kelamin varchar(10), tglh date(10), tb int(3), bb int(3), alamat varchar(50), telp varchar(15), pekerjaan varchar(15).
- d. **Daftar**: menyimpan data pendaftaran pasien
- e. **Periksa**: menyimpan data pemeriksaan pasien
- f. **Bayar**: menyimpan data administrasi pembayaran pasien

5. Hubungan atau Relasi
  - a. Tabel Petugas memiliki hubungan satu ke banyak (1:M) dengan tabel Daftar.
  - b. Tabel Pasien memiliki hubungan satu ke banyak (1:M) dengan tabel Daftar.
  - c. Tabel Daftar memiliki hubungan satu ke satu (1:1) dengan tabel Periksa.
  - d. Tabel Dokter memiliki hubungan satu ke banyak (1:M) dengan tabel Periksa.
  - e. Tabel Periksa memiliki hubungan satu ke satu (1:1) dengan tabel Bayar.
  - f. Tabel Petugas memiliki hubungan satu ke banyak (1:M) dengan tabel Bayar.

## B. Praktek

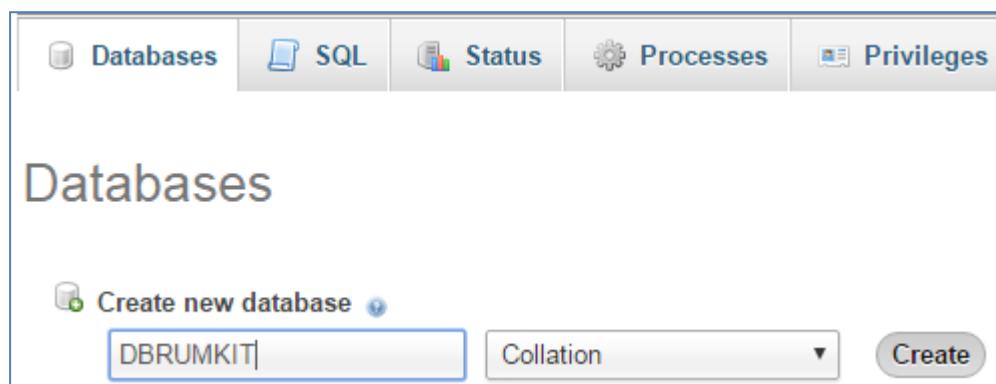
Kegiatan praktikum pertemuan ini adalah menggunakan PHPMyAdmin untuk mengelola Database berdasarkan studi kasus di atas.

- Aktifkan Apache dan MySQL pada XAMPP tersebut dengan mengklik button '**Start**' pada kolom **Action** di tiap modulennya
- Buka browser (Mozilla atau Google Crome) lalu ketik **localhost** lanjutkan dengan menekan ENTER, kemudian pilih phpMyAdmin pada tools yang disediakan

### 1. Membuat Database

Topik yang dibahas pada pertemuan praktikum ini adalah pengelolaan database pengelolaan rawat jalan di rumah sakit yang diberi nama **DBRUMKIT**.

- Klik tab **Databases**
- Ketik nama database yang akan dibuat yaitu: **DBRUMKIT**
- Klik tombol **Create**



## 2. Membuat Tabel

Jumlah tabel yang akan dibuat adalah sesuai dengan jumlah entitas yang telah dibahas di atas. Untuk memulai membuat tabel, maka klik terlebih dahulu **database DBRUMKIT** yang telah dibuat sebelumnya.

### a. Membuat tabel Petugas

- Klik tombol **Create Table**
- Ketik nama tabel yang akan dibuat (contoh ini tabel petugas)
- Masukkan jumlah kolom yang akan dibuat (contoh ini 8 kolom)
- Isikan/ketik nama *field/kolom* yang akan dibuat, tentukan jenis datanya, tentukan ukuran datanya, tentukan juga apakah *field* tersebut sebagai kunci utama (*primary key*) atau hanya *field* biasa.

Field	Type	Length/Values	Index
Kdpet	Char	4	Primary
Nmpet	Varchar	40	
Alpet	Varchar	50	
Telpet	Varchar	15	
Jabatan	Varchar	20	
Nmuser	Varchar	20	
Pasw	Varchar	20	
Levelp	Varchar	20	

- Jika sudah selesai membuat semua *field*, klik tombol **Save** untuk mengakhirinya.

### b. Membuat tabel Dokter

- Klik tombol **Create Table**
- Ketik nama tabel yang akan dibuat (contoh ini tabel dokter)
- Masukkan jumlah kolom yang akan dibuat (contoh ini 6 kolom)
- Isikan/ketik nama *field/kolom* yang akan dibuat, tentukan jenis datanya, tentukan ukuran datanya, tentukan juga apakah *field* tersebut sebagai kunci utama (*primary key*) atau hanya *field* biasa.

Field	Type	Length/Values	Index
Kddok	Char	4	Primary
Nmdok	Varchar	40	
Aldok	Varchar	50	
Teldok	Varchar	15	
Kategori	Varchar	10	
Tarif	Integer	10	

- Jika sudah selesai membuat semua *field*, klik tombol **Save** untuk mengakhirinya.

c. Membuat tabel Pasien

- Klik tombol **Create Table**
- Ketik nama tabel yang akan dibuat (contoh ini tabel pasien)
- Masukkan jumlah kolom yang akan dibuat (contoh ini 9 kolom)
- Isikan/ketik nama *field/kolom* yang akan dibuat, tentukan jenis datanya, tentukan ukuran datanya, tentukan juga apakah *field* tersebut sebagai kunci utama (*primary key*) atau hanya *field* biasa.

Field	Type	Length/Values	Index
Idpas	Int	10	Primary
Nmpas	Varchar	40	
Kelamin	Varchar	10	
Tglh	Date	10	
Tb	Int	3	
Bb	Int	3	
Alamat	Varchar	50	
Telp	Varchar	15	
Pekerjaan	Varchar	15	

- Jika sudah selesai membuat semua *field*, klik tombol **Save** untuk mengakhirinya.

d. Membuat tabel Daftar

- Klik tombol **Create Table**
- Ketik nama tabel yang akan dibuat (contoh ini tabel daftar)
- Masukkan jumlah kolom yang akan dibuat (contoh ini 5 kolom)
- Isikan/ketik nama *field/kolom* yang akan dibuat, tentukan jenis datanya, tentukan ukuran datanya, tentukan juga apakah *field* tersebut sebagai kunci utama (*primary key*) atau hanya *field* biasa.

Field	Type	Length/Values	Index
Nodaftar	Int	10	Primary
Tgldaftar	Date	10	
Idpas	Int	10	Index
Keluhan	Varchar	75	
Kdpet	Char	4	Index

- Jika sudah selesai membuat semua *field*, klik tombol **Save** untuk mengakhirinya.

e. Membuat tabel Periksa

- Klik tombol **Create Table**
- Ketik nama tabel yang akan dibuat (contoh ini tabel periksa)
- Masukkan jumlah kolom yang akan dibuat (contoh ini 4 kolom)

- Isikan/ketik nama *field/kolom* yang akan dibuat, tentukan jenis datanya, tentukan ukuran datanya, tentukan juga apakah *field* tersebut sebagai kunci utama (*primary key*) atau hanya *field* biasa.

Field	Type	Length/Values	Index
Nodaftar	Int	10	Primary
Kddok	Char	4	Index
Diagnosa	Varchar	50	
Catatan	Varchar	75	

- Jika sudah selesai membuat semua *field*, klik tombol **Save** untuk mengakhirinya.

f. Membuat tabel Bayar

- Klik tombol **Create Table**
- Ketik nama tabel yang akan dibuat (contoh ini tabel bayar)
- Masukkan jumlah kolom yang akan dibuat (contoh ini 7 kolom)
- Isikan/ketik nama *field/kolom* yang akan dibuat, tentukan jenis datanya, tentukan ukuran datanya, tentukan juga apakah *field* tersebut sebagai kunci utama (*primary key*) atau hanya *field* biasa.

Field	Type	Length/Values	Index
Nodaftar	Int	10	Primary
Tglbayar	Date	10	
Beaadm	Int	10	
Beadokter	Int	10	
Beabat	Int	10	
Bealain	Int	10	
Kdpet	Char	4	Index

- Jika sudah selesai membuat semua *field*, klik tombol Save untuk mengakhirinya.

Catatan:

Entitas dan struktur dari masing-masing entitas/tabel seperti yang diuraikan di atas, bisa berubah sesuai dengan hasil analisis dan kebutuhan pengguna. Contoh entitas, atribut/*field*, jenis data serta ukuran masing-masing bisa menyesuaikan berdasarkan pertimbangan tertentu. Semua komponen pembentuk database yang diuraikan di atas hanya mengambil asumsi dan perkiraan secara sederhana. Sebagai pembanding silahkan melihat contoh *database* untuk kasus yang sama melalui referensi yang lain seperti buku atau melalui internet.

### 3. Mengisi Data

Setelah semua tabel dibuat, silahkan mengisi masing-masing tabel dengan data yang sesuai. Supaya informasi lebih bervariasi, usahakan data yang diisikan untuk masing-masing tabel minimal 10 record.

#### a. Mengisi data Petugas

Data petugas piket atau yang menggunakan sistem informasi layanan rawat jalan di rumah sakit dapat buat seperti gambar di bawah ini.

kdpet	nmpet	alpet	telpet	jabatan	nmuser	pasw	levelp
KP01	Zaidir	Jl. Gejayan No. 87 Yogyakarta	81328330538	Manajer IT	Zaidir	Zaidir	Admin
KP02	Rahmat	Jl. Gejayan No. 33 Yogyakarta	85776556431	Staf Admin	Rahmat	Rahmat	User
KP03	Fadli	Jl. Gejayan No. 67 Yogyakarta	87987321211	Staf Admin	Fadli	Fadli	User
KP04	Bambang	Jl. Gejayan No. 12 Yogyakarta	81328330111	Staf Admin	Bambang	Bambang	User
KP05	Umi Kalsum	Jl. Gejayan No. 87 Yogyakarta	81328330987	Staf Admin	Umi	Umi	User
KP06	Ratna Pertiwi	Jl. Gejayan No. 12 Yogyakarta	81328330098	Staf Admin	Ratna	Ratna	User
KP07	Kumala Sari	Jl. Gejayan No. 98 Yogyakarta	81328331234	Staf Admin	Kumala	Kumala	User
KP08	Jajang Zaini	Jl. Gejayan No. 54 Yogyakarta	81328333344	Staf Admin	Jajang	Jajang	User
KP09	Fitria Dewi	Jl. Gejayan No. 11 Yogyakarta	81328338876	Staf Admin	Fitria	Fitria	User
KP10	Hermawan Z	Jl. Gejayan No. 54 Yogyakarta	81328332132	Staf Admin	Hermawan	Hermawan	User

#### b. Mengisi data Dokter

Data dokter yang melayani pasien dapat diisikan seperti gambar di bawah ini.

kddok	nmdok	aldok	teldok	kategori	tarif
KD01	Hartono	Jl. Sisingamangaraja No. 11 Yogyakarta	085443232467	Umum	60000
KD02	Wibowo	Jl. Godean No. 76 Yogyakarta	085443232111	Spesialis	100000
KD03	Susanti	Jl. C Simanjuntak No. 54 Yogyakarta	085443232879	Spesialis	100000
KD04	Dewi Ariani	Jl. A Yani No. 09 Yogyakarta	085443232444	Umum	60000
KD05	Vivi Artanti	Jl. Kaliurang No. 87 Yogyakarta	085443236578	Umum	60000
KD06	Wiguna S	Jl. MT Haryono No. 19 Yogyakarta	085443231212	Spesialis	100000
KD07	Ahmad Subari	Jl. Sultan Agung No. 55 Yogyakarta	085443237766	Umum	60000
KD08	Lilik P	Jl. Kusumanegara No. 43 Yogyakarta	085443238899	Spesialis	100000
KD09	Zubardi	Jl. Janti No. 04 Yogyakarta	085443239887	Umum	60000
KD10	Dedi Wahyudi	Jl. Tendean No. 79 Yogyakarta	085443230998	Umum	60000

c. Mengisi data Pasien

Data pasien yang diisikan bisa menggunakan contoh data seperti gambar di bawah ini.

<b>idpas</b>	<b>nmpas</b>	<b>kelamin</b>	<b>tgih</b>	<b>tb</b>	<b>bb</b>	<b>alamat</b>	<b>telp</b>	<b>pekerjaan</b>
1	Dadang S	L	1967-02-03	166	56	Jl. S Parman No. 87 Yogyakarta	85667878345	Petani
2	Wahyu Ananto	L	1965-11-11	172	62	Jl. Pancasila No. 3 Yogyakarta	85667875456	Wiraswasta
3	Ferr Sulistyo	L	1970-04-03	178	68	Jl. Sudirman No. 54 Yogyakarta	85667871111	Peg Negeri
4	Fahril S	L	1969-08-04	156	46	Jl. MT Haryono No. 87 Yogyakarta	85667872234	Peg Swasta
5	Vivi Safitri	P	1976-10-12	167	57	Jl. Herman Yohanes No. 87 Yogyakarta	85667875467	Petani
6	Weni Sujimin	P	1978-09-08	168	58	Jl. Cut Nyak Dien No. 23 Yogyakarta	85667876655	Wiraswasta
7	Linawati	P	1978-07-10	155	45	Jl. Kartini No. 49 Yogyakarta	85667878790	Peg Swasta
8	Ambarawati	P	1972-06-11	165	55	Jl. Sisingamangaraja No. 5 Yogyakarta	85667875456	Wiraswasta
9	Mahmud Al Fajri	L	1980-02-09	172	62	Jl. Sultan Agung No. 8 Yogyakarta	85667871211	Petani
10	Guntur	L	1976-10-10	174	64	Jl. Adisucipto No. 54 Yogyakarta	85667874345	Wiraswasta
11	Yasril	L	1975-03-02	158	48	Jl. A Yani No. 98 Yogyakarta	85667877687	Peg Swasta
12	Iqbal	L	1970-11-10	164	54	Jl. Pancasila No. 41 Yogyakarta	85667873333	Petani
13	Susanti	P	1977-11-10	176	66	Jl. Kolombo No. 21 Yogyakarta	85667877891	Wiraswasta
14	Cindy	P	1969-09-12	165	55	Jl. S Parman No. 54 Yogyakarta	85667873324	Peg Swasta
15	Asrul	L	1973-10-10	163	53	Jl. Mawar No. 32 Yogyakarta	85667877656	Petani
16	Endang	P	1974-08-11	165	55	Jl. Sukun No. 77 Yogyakarta	85667878998	Petani
17	Rini Cahyani	P	1971-10-12	169	59	Jl. Sultan Agung No. 32 Yogyakarta	85667874444	Peg Swasta
18	Nani S	P	1973-07-05	177	67	Jl. Kolombo No. 45 Yogyakarta	85667871213	Peg Swasta
19	Zubaidi	L	1968-09-11	174	64	Jl. Gejayan No. 54 Yogyakarta	85667873423	Wiraswasta
20	Tuti S	P	1980-11-10	166	56	Jl. Kartini No. 89 Yogyakarta	85667877786	Petani
21	Ferdinand	L	1977-09-12	169	59	Jl. Empu Tantular No. 3 Yogyakarta	85667874534	Wiraswasta
22	Sulaiman	L	1978-09-12	170	60	Jl. Sisingamangaraja No. 45 Yogyakarta	85667875645	Peg Swasta
23	Agus Riyadi	L	1976-04-03	175	65	Jl. Tendean No. 32 Yogyakarta	85667878988	Peg Swasta

d. Mengisi data Daftar

Data pendaftaran pasien bisa diisikan seperti gambar di bawah ini

<b>nodaftar</b>	<b>tgldaftar</b>	<b>idpas</b>	<b>keluhan</b>	<b>kdpet</b>
1	2015-01-01	4	Pusing	KP01
2	2015-01-01	7	Mual	KP01
3	2015-01-01	12	Sakit perut	KP01
4	2015-01-01	3	Lemas	KP01
5	2015-01-01	11	Muntah	KP01
6	2015-01-01	5	Mencoret	KP01
7	2015-02-01	6	Ngilu-ngilu	KP01
8	2015-02-01	8	Sakit Gigi	KP02
9	2015-02-01	1	Sembelit	KP02
10	2015-02-01	2	Pusing	KP02
11	2015-02-01	4	Panas	KP02
12	2015-03-01	5	Panas	KP02
13	2015-03-01	3	Ngilu-ngilu	KP02
14	2015-03-01	9	Muntah	KP02
15	2015-03-01	16	Mencoret	KP02
16	2015-03-01	17	Panas	KP02
17	2015-03-01	14	Ngilu-ngilu	KP02
18	2016-01-01	20	Ngilu-ngilu	KP02
19	2016-01-01	25	Panas	KP02
20	2016-01-01	13	Pusing	KP02
21	2016-01-01	10	Muntah	KP01
22	2016-01-01	11	Mencoret	KP01
23	2016-01-01	24	Panas	KP01

e. Mengisi data Periksa

Data pemeriksaan pasien bisa diisikan seperti gambar di bawah ini

Nodaftarn	Iddok	Diagnosa	Catatan
1	KD03	Gangguan syaraf kepala dan radang	Banyak olah raga
2	KD03	Gangguan metabolisme	Hindari makanan pedas dan minuman dingin
3	KD03	Gangguan metabolisme	Hindari makanan pedas dan minuman dingin
4	KD03	Gangguan metabolisme	Diperbanyak minuman madu
5	KD03	Terjadi radang pada usus	Hindari makanan pedas, minuman dingin dan banyak i...
6	KD03	Terjadi gangguan pencernaan	Hindari makanan pedas
7	KD03	Terjadi pengapuran dan penyempitan pembuluh darah	Hindari makanan tinggi kalsium, perbanyak makan bu...
8	KD03	Adanya gangguan syaraf gigi	Banyak istirahat
9	KD03	Gangguan metabolisme	Hindari makanan keras
10	KD03	Gangguan syaraf kepala dan radang	Banyak olah raga
11	KD03	Ada radang tenggorokan, gangguan metabolisme	Hindari makanan keras, perbanyak istirahat
12	KD03	Ada radang tenggorokan, gangguan metabolisme	Hindari makanan keras, perbanyak istirahat
13	KD03	Terjadi pengapuran dan penyempitan pembuluh darah	Hindari makanan tinggi kalsium, perbanyak makan bu...
14	KD03	Terjadi radang pada usus	Hindari makanan pedas, minuman dingin dan banyak i...
15	KD03	Terjadi gangguan pencernaan	Hindari makanan pedas
16	KD01	Ada radang tenggorokan, gangguan metabolisme	Hindari makanan keras, perbanyak istirahat
17	KD01	Terjadi pengapuran dan penyempitan pembuluh darah	Hindari makanan tinggi kalsium, perbanyak makan bu...
18	KD01	Terjadi pengapuran dan penyempitan pembuluh darah	Hindari makanan tinggi kalsium, perbanyak makan bu...
19	KD01	Ada radang tenggorokan, gangguan metabolisme	Hindari makanan keras, perbanyak istirahat
20	KD01	Gangguan syaraf kepala dan radang	Banyak olah raga
21	KD01	Terjadi radang pada usus	Hindari makanan pedas, minuman dingin dan banyak i...
22	KD01	Terjadi gangguan pencernaan	Hindari makanan pedas
23	KD01	Ada radang tenggorokan, gangguan metabolisme	Hindari makanan keras, perbanyak istirahat

f. Mengisi data Bayar

Data pembayaran pasien bisa diisikan seperti gambar di bawah ini

nodaftarn	tglbayar	beaadm	beadokter	beabobat	bealain	kdpet
1	2015-01-01	10000	100000	25000	5000	KP02
2	2015-01-01	10000	100000	25000	5000	KP02
3	2015-01-01	10000	100000	25000	5000	KP02
4	2015-01-01	10000	100000	25000	5000	KP02
5	2015-01-01	10000	100000	25000	5000	KP02
6	2015-01-01	10000	100000	25000	5000	KP02
7	2015-02-01	10000	100000	25000	5000	KP02
8	2015-02-01	10000	100000	25000	5000	KP02
9	2015-02-01	10000	100000	25000	5000	KP02
10	2015-02-01	10000	100000	25000	5000	KP02
11	2015-02-01	0	100000	25000	5000	KP02
12	2015-03-01	0	100000	25000	5000	KP02
13	2015-03-01	0	100000	25000	5000	KP02
14	2015-03-01	10000	100000	25000	5000	KP02
15	2015-03-01	10000	100000	25000	5000	KP02
16	2015-03-01	10000	60000	25000	5000	KP02
17	2015-03-01	10000	60000	25000	5000	KP02
18	2016-01-01	10000	60000	25000	5000	KP02
19	2016-01-01	10000	60000	25000	5000	KP02
20	2016-01-01	10000	60000	25000	5000	KP02
21	2016-01-01	10000	60000	25000	5000	KP05
22	2016-01-01	0	60000	25000	5000	KP05
23	2016-01-01	10000	60000	25000	5000	KP05

#### 4. Menampilkan data dengan SQL

Untuk menampilkan data menggunakan SQL, silahkan dicobakan perintah-perintah SQL yang telah dicobakan pada modul-modul sebelumnya

- a. Menampilkan data dari 3 tabel (tabel daftar terkait dengan tabel pasien dan tabel petugas. Data yang ditampilkan adalah secara secara keseluruhan.

```
SELECT daftar.Nodaftar, daftar.Tgldaftar,
daftar.Idpas, pasien.nmpas, pasien.kelamin,
daftar.Keluahan, daftar.Kdpet, petugas.nmpet
FROM (daftar INNER JOIN pasien ON daftar.Idpas =
pasien.idpas) INNER JOIN petugas ON daftar.Kdpet =
petugas.kdpet;
```

- b. Menampilkan data dari 4 tabel (tabel periksa terkait dengan tabel daftar, tabel pasien dan tabel dokter. Data yang ditampilkan adalah secara secara keseluruhan.

```
SELECT periksa.Nodaftar, daftar.Tgldaftar,
pasien.nmpas, daftar.Keluahan, dokter.nmdok,
periksa.Diagnosa, periksa.Catatan
FROM (periksa INNER JOIN dokter ON periksa.Kddok =
dokter.kddok) INNER JOIN (daftar INNER JOIN pasien
ON daftar.Idpas = pasien.idpas) ON periksa.Nodaftar
= daftar.Nodaftar;
```

- c. Menampilkan data dari 5 tabel (tabel bayar terkait dengan tabel periksa, tabel daftar, tabel pasien dan tabel petugas. Data yang ditampilkan adalah secara secara keseluruhan.

```
SELECT bayar.Nodaftar, pasien.nmpas,
bayar.tglbayar, bayar.beaadm, bayar.beadokter,
bayar.beaobat, bayar.bealain,
beaadm+beadokter+beaobat+bealain AS JmlTotal,
petugas.nmpet
FROM ((bayar INNER JOIN petugas ON bayar.kdpet =
petugas.kdpet) INNER JOIN periksa ON bayar.Nodaftar
= periksa.Nodaftar) INNER JOIN (daftar INNER JOIN
pasien ON daftar.Idpas = pasien.idpas) ON
periksa.Nodaftar = daftar.Nodaftar;
```

- d. Menampilkan data dari beberapa tabel dan metode menampilkan dengan group (laporan dalam bentuk rekapitulasi pendaftaran).

```
SELECT Year(Tgldaftar) AS Tahun,
Sum(If(kelamin="L",1,0)) AS Lakilaki,
Sum(If(kelamin="P",1,0)) AS Perempuan,
Count(daftar.Nodaftar) AS JumlahSemua
FROM daftar INNER JOIN pasien ON daftar.Idpas =
pasien.idpas
GROUP BY Year(Tgldaftar);
```

- e. Menampilkan data dari beberapa tabel dan metode menampilkan dengan group (laporan dalam bentuk rekapitulasi pembayaran).

```
SELECT Year(tglbayar) AS Tahun,
Count(bayar.Nodaftar) AS JmlTransaksi,
Sum(bayar.beaadm) AS TotBeaadm,
Sum(bayar.beadokter) AS TotBeadokter,
Sum(bayar.beaobat) AS TotBeaobat,
Sum(bayar.bealain) AS TotBealain,
Sum(Beaadm+Beadokter+Beaobat+Bealain) AS TotalSemua
FROM bayar
GROUP BY Year(tglbayar);
```

## 5. Membuat Relasi Antar Tabel

- a. Menghubungkan tabel bayar dengan tabel yang terkait

- Klik tabel bayar
- Klik menu structure
- Klik Relation view
- Pilih field kunci yang terkait dengan tabel yang lain, dan tentukan field terkait dengan tabel yang mana.

Column	Internal relation	Foreign key constraint (INNODB)
nodaftar		'dbrumkit'.`periksa`.`Nodaftar` ON DELETE RESTRICT ON UPDATE RESTRICT
tglbayar		No index defined!
beaadm		No index defined!
beadokter		No index defined!
beaobat		No index defined!
bealain		No index defined!
kdpet		'dbrumkit'.`petugas`.`kdpet` ON DELETE RESTRICT ON UPDATE RESTRICT

- Klik tombol Save jika sudah selesai.

b. Menghubungkan tabel periksa dengan tabel terkait

- Klik tabel bayar
- Klik menu structure
- Klik Relation view
- Pilih field kunci yang terkait dengan tabel yang lain, dan tentukan field terkait dengan tabel yang mana.

Column	Internal relation	Foreign key constraint (INNODB)
Nodaftar		'dbrumkit'.`daftar`. `nodaftar` ON DELETE RESTRICT ON UPDATE RESTRICT
kddok		'dbrumkit'.`dokter`. `kddok` ON DELETE RESTRICT ON UPDATE RESTRICT
Diagnosa		No index defined!
Catatan		No index defined!

- Klik tombol Save jika sudah selesai.

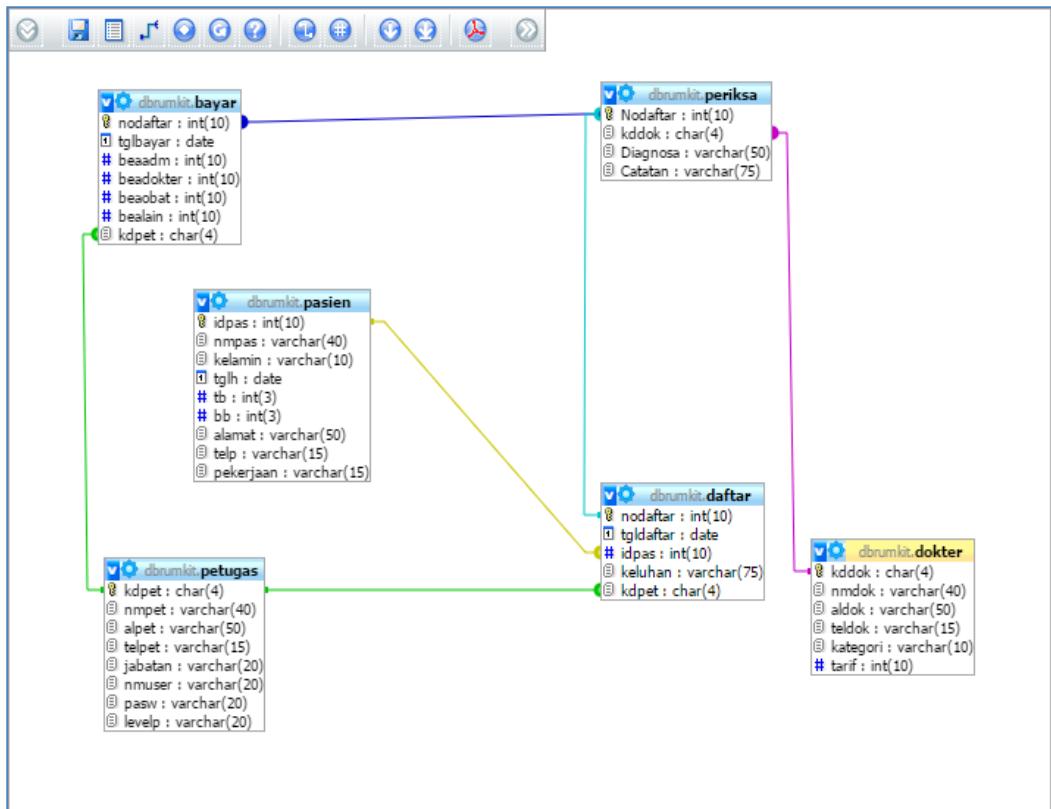
c. Menghubungkan tabel daftar dengan tabel terkait

- Klik tabel bayar
- Klik menu structure
- Klik Relation view
- Pilih field kunci yang terkait dengan tabel yang lain, dan tentukan field terkait dengan tabel yang mana.

Column	Internal relation	Foreign key constraint (INNODB)
nodaftar		'dbrumkit'.`periksa`. `Nodaftar` ON DELETE RESTRICT ON UPDATE RESTRICT
tglbayar		No index defined!
beaadm		No index defined!
beadokter		No index defined!
beabobat		No index defined!
bealain		No index defined!
kdpet		'dbrumkit'.`petugas`. `kdpet` ON DELETE RESTRICT ON UPDATE RESTRICT

- Klik tombol Save jika sudah selesai.

Apabila semua tahapan relasi sudah selesai, maka klik nama database, selanjutnya klik menu Designer. Dengan demikian akan ditampilkan hasil relasi antar tabel dari masing-masing tabel yang telah dihubungkan.



## 6. Membuat View

- View untuk menampilkan informasi pendaftaran

```

CREATE VIEW view_daftar
AS
SELECT daftar.Nodaftar, daftar.Tgldaftar,
daftar.Idpas, pasien.nmpas, pasien.kelamin,
daftar.Keluahan, daftar.Kdpet, petugas.nmpt
FROM (daftar INNER JOIN pasien ON daftar.Idpas =
pasien.idpas) INNER JOIN petugas ON daftar.Kdpet =
petugas.kdpet;
  
```

- View untuk menampilkan informasi pemeriksaan

```

CREATE VIEW view_periksa
AS
SELECT periksa.Nodaftar, daftar.Tgldaftar,
pasien.nmpas, daftar.Keluahan, dokter.nmdok,
periksa.Diagnosa, periksa.Catatan
FROM (periksa INNER JOIN dokter ON periksa.Kddok =
dokter.kddok) INNER JOIN (daftar INNER JOIN pasien ON
daftar.Idpas = pasien.idpas) ON periksa.Nodaftar =
daftar.Nodaftar;
  
```

- c. View untuk menampilkan informasi pembayaran

```
CREATE VIEW view_bayar
AS
SELECT bayar.Nodaftar, pasien.nmpas, bayar.tglbayar,
bayar.beaadm, bayar.beadokter, bayar.beaobat,
bayar.bealain, beaadm+beadokter+beaobat+bealain AS
JmlTotal, petugas.nmpet
FROM ((bayar INNER JOIN petugas ON bayar.kdpet =
petugas.kdpet) INNER JOIN periksa ON bayar.Nodaftar =
periksa.Nodaftar) INNER JOIN (daftar INNER JOIN
pasien ON daftar.Idpas = pasien.idpas) ON
periksa.Nodaftar = daftar.Nodaftar;
```

- d. View untuk menampilkan rekapitulasi pendaftaran

```
CREATE VIEW view_rekdaftar
AS
SELECT Year(Tgldaftar) AS Tahun,
Sum(If(kelamin="L",1,0)) AS Lakilaki,
Sum(If(kelamin="P",1,0)) AS Perempuan,
Count(daftar.Nodaftar) AS JumlahSemua
FROM daftar INNER JOIN pasien ON daftar.Idpas =
pasien.idpas
GROUP BY Year(Tgldaftar);
```

- e. View untuk menampilkan rekapitulasi pembayaran

```
CREATE VIEW view_rekbayar
AS
SELECT Year(tglbayar) AS Tahun,
Count(bayar.Nodaftar) AS JmlTransaksi,
Sum(bayar.beaadm) AS TotBeaadm, Sum(bayar.beadokter)
AS TotBeadokter, Sum(bayar.beaobat) AS TotBeaobat,
Sum(bayar.bealain) AS TotBealain,
Sum(Beaadm+Beadokter+Beaobat+Bealain) AS TotalSemua
FROM bayar
GROUP BY Year(tglbayar);
```

### C. Tugas

Untuk memperlancar materi pada praktikum ini maka:

1. Buatlah *database* baru mengenai gaji pegawai
2. Buatlah tabel-tabel yang dibutuhkan
3. Isilah data minimal 10 record untuk masing-masing tabel.
4. Lakukan proses menampilkan data dengan SQL (catat perintah yang dibuat dan dipindah ke MS Word. Selanjutnya hasil SQL yang dibuat dikirim bersama-sama dengan *database* yang dibuat.

Kirimkan tugas yang dibuat ke email: [zaidirtan@gmail.com](mailto:zaidirtan@gmail.com) paling lambat 3 hari sejak hari praktikum.

## BAB 12

### SERBA SERBI

Pada akhir materi modul praktikum ini akan dibahas mengenai permasalahan pengelolaan data yang cukup besar namun menginginkan waktu yang cepat untuk menyelesaiannya.

Sebelum lebih lanjut mengenai proses pengelolaan data tersebut, akan diberikan ilustrasi sebagai berikut:

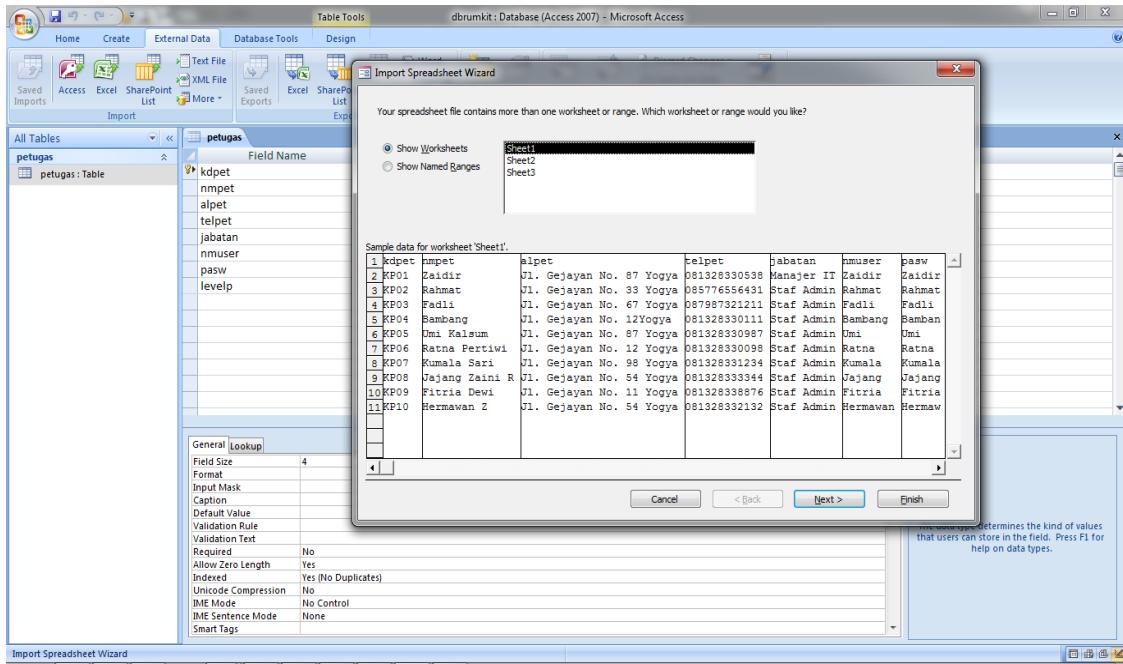
1. Sebuah usaha yang sedang berkembang, ingin melakukan proses administrasi secara terkomputerisasi dengan program khusus. Saat ini pengolahan data sudah dilakukan dengan komputer, tetapi hanya menggunakan aplikasi umum yaitu MS Excel. Data yang sudah diolah mencapai 5000 data. Jika saudara diminta untuk membuat program dengan dukungan *database*, apakah saudara kan memasukan data satu per satu? Berapa waktu yang dibutuhkan untuk pekerjaan itu?
2. Sebuah perusahaan yang sudah cukup lama berkiprah di dunia usaha, telah memiliki program aplikasi dengan menggunakan *database MS Access*. Karena data cukup banyak dan menyesuaikan dengan perkembangan teknologi khususnya DBMS yang memiliki kemampuan lebih besar seperti MySQL. Bagaimana caranya data dari **MS Access** bisa dimasukan ke *database MySQL* dengan cepat?. Jawabannya adalah dilakukan proses konversi, dari **MS Access** ke **MS Excel**, kemudian dari **MS Excel** di masukkan ke **MySQL**.

Berdasarkan ilustrasi di atas, tentu akan menarik jika memiliki pengetahuan untuk melakukan hal-hal yang bermanfaat.

#### A. Mengisi Database dari Excel

Ada dua hal yang akan dibahas pada materi ini yaitu konversi data dari **Excel** ke **MS Access** dan konversi data dari **Excel** ke **MySQL**.

1. Data Excel ke MS Access
  - Buat *database* baru atau buka **database** yang sudah ada dengan menggunakan **MS Access**.
  - Saat *database* sudah terbuka, klik menu **External Data**
  - Pilih **Excel**
  - Tentukan sumber data **Excel** yang akan dimasukkan ke tabel **MS Access**
  - Pilih *Import the source data into a new table in the current database*, kemudian klik tombol **OK**
  - Beri tanda centang pada “*First Row Contains Column Headings*”, kemudian klik tombol **Next**.
  - Tentukan tipe *index (field)* yang dijadikan *primary key* dengan isian **Yes (No Duplicate)** kolom paling kiri, kemudian klik tombol **Next**.
  - Klik pilihan *Choose my own primary key* dan pilih *field* yang dijadikan kunci, kemudian klik tombol **Next**.
  - Ketik nama tabel yang akan digunakan untuk menyimpan data (defaultnya ditawarkan Sheet1), klik tombol **Finish**.
  - Klik tombol **Close**.

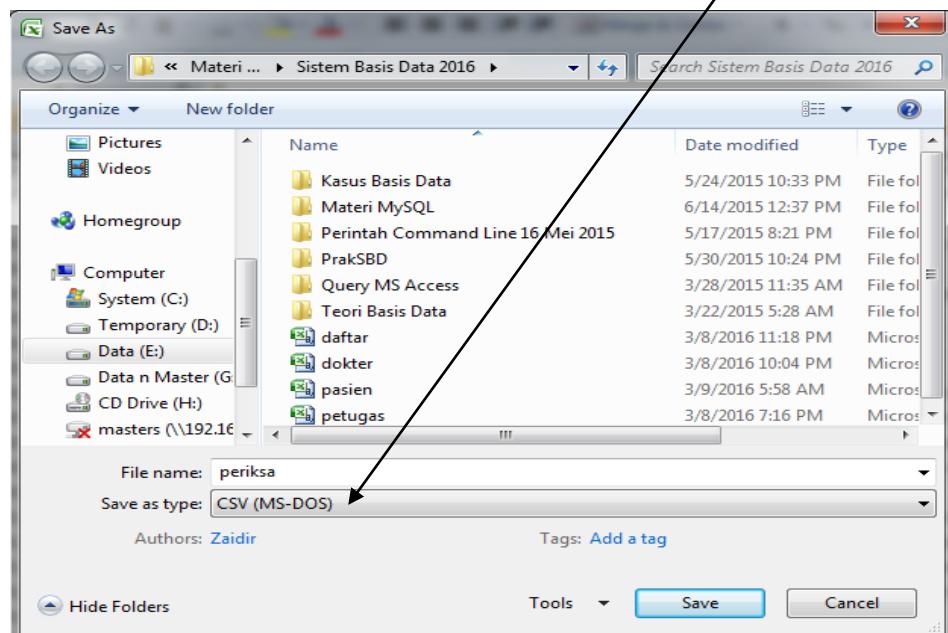


## 2. Data excel ke MySQL

- Buat *database* baru atau buka *database* yang sudah ada dengan menggunakan **MySQL**.
- Buka file Excel yang akan dimasukkan ke *database* MySQL. Pada file tersebut tidak perlu ada judul kolom.

	A	B	C	D
1	1 KD03	Gangguan syaraf kepala dan radang	Banyak olah raga	
2	2 KD03	Gangguan metabolisme	Hindari makanan pedas dan minuman dingin	
3	3 KD03	Gangguan metabolisme	Hindari makanan pedas dan minuman dingin	
4	4 KD03	Gangguan metabolisme	Diperbanyak minuman madu	
5	5 KD03	Terjadi radang pada usus	Hindari makanan pedas, minuman dingin dan bar	
6	6 KD03	Terjadi gangguan pencernaan	Hindari makanan pedas	
7	7 KD03	Terjadi pengapuran dan penyempitan pembuluh darah	Hindari makanan tinggi kalsium, perbanyak maka	
8	8 KD03	Adanya gangguan syaraf gigi	Banyak istirahat	
9	9 KD03	Gangguan metabolisme	Hindari makanan keras	
10	10 KD03	Gangguan syaraf kepala dan radang	Banyak olah raga	
11	11 KD03	Ada radang tenggorokan, gangguan metabolisme	Hindari makanan keras, perbanyak istirahat	
12	12 KD03	Ada radang tenggorokan, gangguan metabolisme	Hindari makanan keras, perbanyak istirahat	
13	13 KD03	Terjadi pengapuran dan penyempitan pembuluh darah	Hindari makanan tinggi kalsium, perbanyak maka	
14	14 KD03	Terjadi radang pada usus	Hindari makanan pedas, minuman dingin dan bar	
15	15 KD03	Terjadi gangguan pencernaan	Hindari makanan pedas	
16	16 KD01	Ada radang tenggorokan, gangguan metabolisme	Hindari makanan keras, perbanyak istirahat	
17	17 KD01	Terjadi pengapuran dan penyempitan pembuluh darah	Hindari makanan tinggi kalsium, perbanyak maka	

- File data dari Excel harus disimpan dengan format CSV (MS-DOS).



- Aktifkan tabel pada MySQL yang akan diisi datanya dengan file MS Excel.
- Klik tab Import
- Pilih file Excel yang telah disimpan dalam format CSV

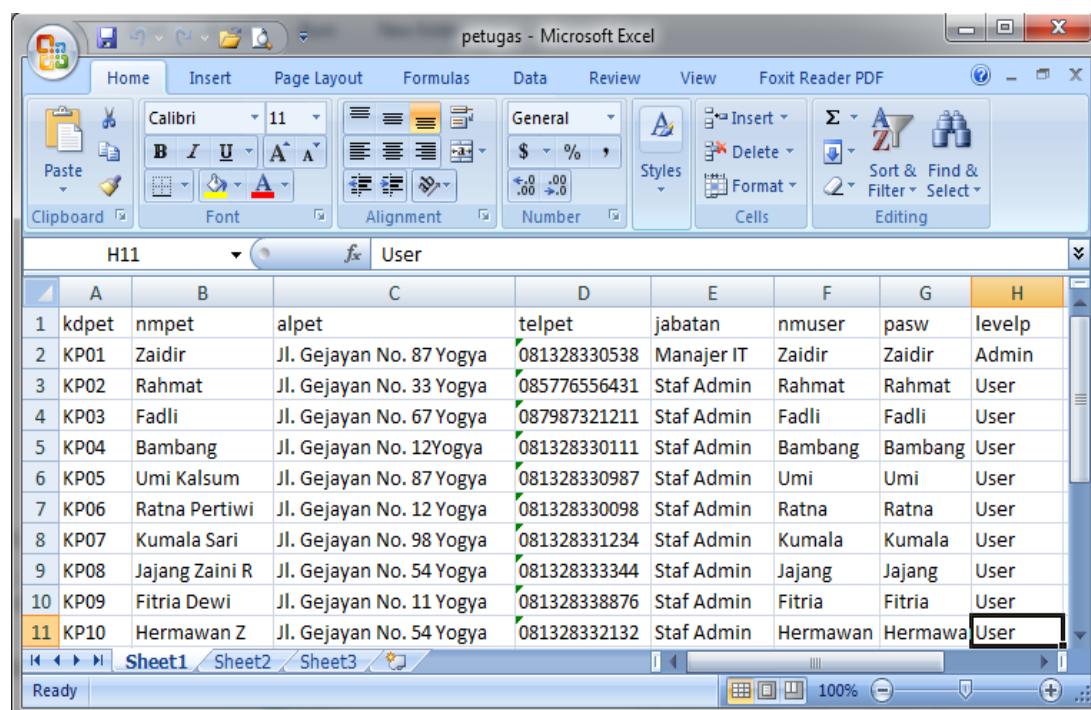
The screenshot shows the phpMyAdmin interface for importing data into the 'periksa' table of the 'dbrumkit' database. The 'Import' tab is selected. The 'File to Import:' section shows a 'Choose File' button with 'periksa.csv' selected. The 'Character set of the file:' dropdown is set to 'utf-8'. Below this, the 'Partial Import:' section has a checked checkbox for allowing interruptions. The 'Format:' dropdown is set to 'CSV'. A note at the bottom states: 'Note: If the file contains multiple tables, they will be combined into one'.

- Klik Menu **GO**

## B. Praktek

### 1. Konversi data Excel ke MS Access

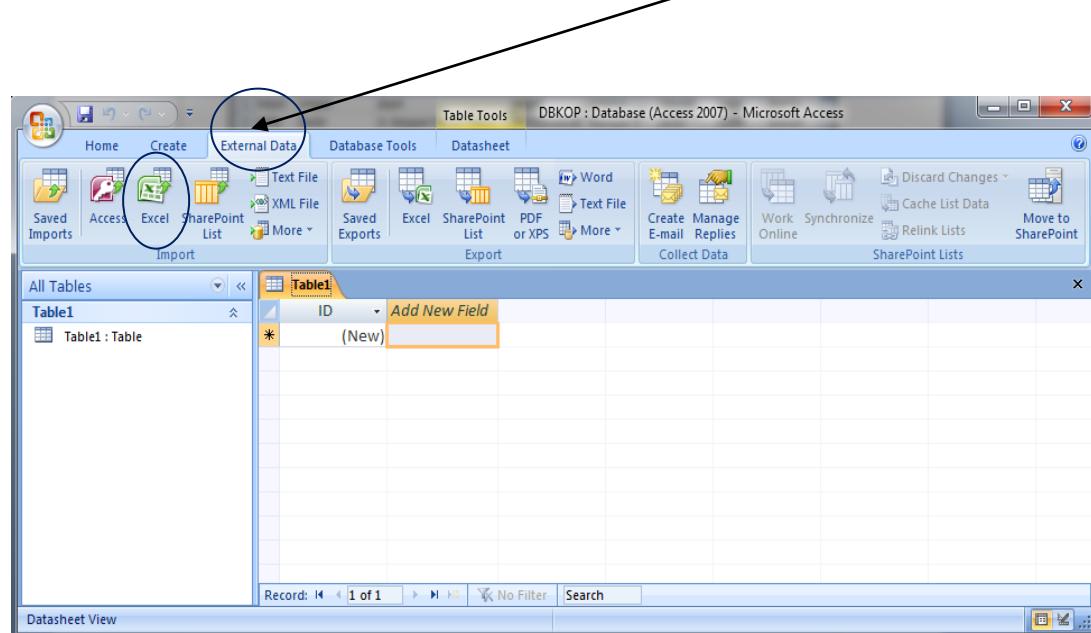
- Buat data dengan menggunakan MS Excel, dan simpan dengan nama petugas.xlsx, seperti gambar di bawah ini



A screenshot of Microsoft Excel showing a table titled "User". The table contains 11 rows of data with columns A through H. The data includes fields like kdpet, nmpet, alpet, telpet, jabatan, nmuser, pasw, and levelp. Row 11 is highlighted in yellow.

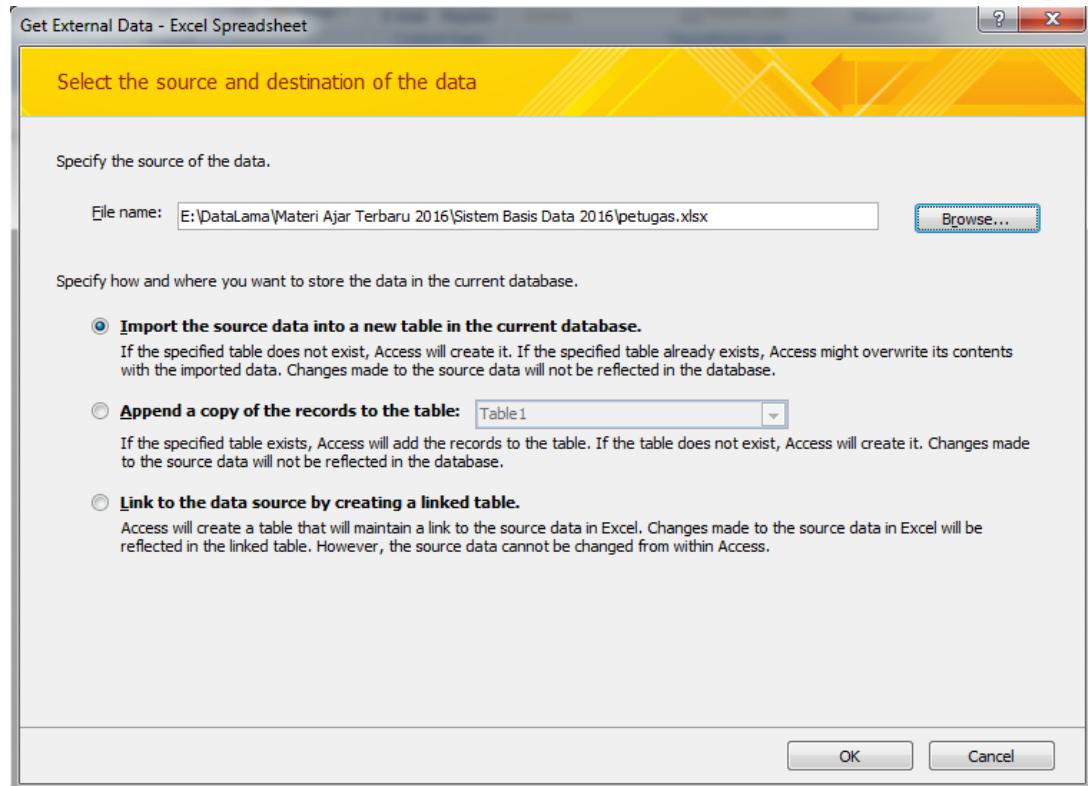
	A	B	C	D	E	F	G	H
1	kdpet	nmpet	alpet	telpet	jabatan	nmuser	pasw	levelp
2	KP01	Zaidir	Jl. Gejayan No. 87 Yogyo	081328330538	Manajer IT	Zaidir	Zaidir	Admin
3	KP02	Rahmat	Jl. Gejayan No. 33 Yogyo	085776556431	Staf Admin	Rahmat	Rahmat	User
4	KP03	Fadli	Jl. Gejayan No. 67 Yogyo	087987321211	Staf Admin	Fadli	Fadli	User
5	KP04	Bambang	Jl. Gejayan No. 12Yogyo	081328330111	Staf Admin	Bambang	Bambang	User
6	KP05	Umi Kalsum	Jl. Gejayan No. 87 Yogyo	081328330987	Staf Admin	Umi	Umi	User
7	KP06	Ratna Pertiwi	Jl. Gejayan No. 12 Yogyo	081328330098	Staf Admin	Ratna	Ratna	User
8	KP07	Kumala Sari	Jl. Gejayan No. 98 Yogyo	081328331234	Staf Admin	Kumala	Kumala	User
9	KP08	Jajang Zaini R	Jl. Gejayan No. 54 Yogyo	081328333344	Staf Admin	Jajang	Jajang	User
10	KP09	Fitria Dewi	Jl. Gejayan No. 11 Yogyo	081328338876	Staf Admin	Fitria	Fitria	User
11	KP10	Hermawan Z	Jl. Gejayan No. 54 Yogyo	081328332132	Staf Admin	Hermawan	Hermawan	User

- Buat database baru dengan menggunakan **MS Access** dan diberi nama **DBKOP**.
- Saat database sudah terbuka, klik menu **External Data**

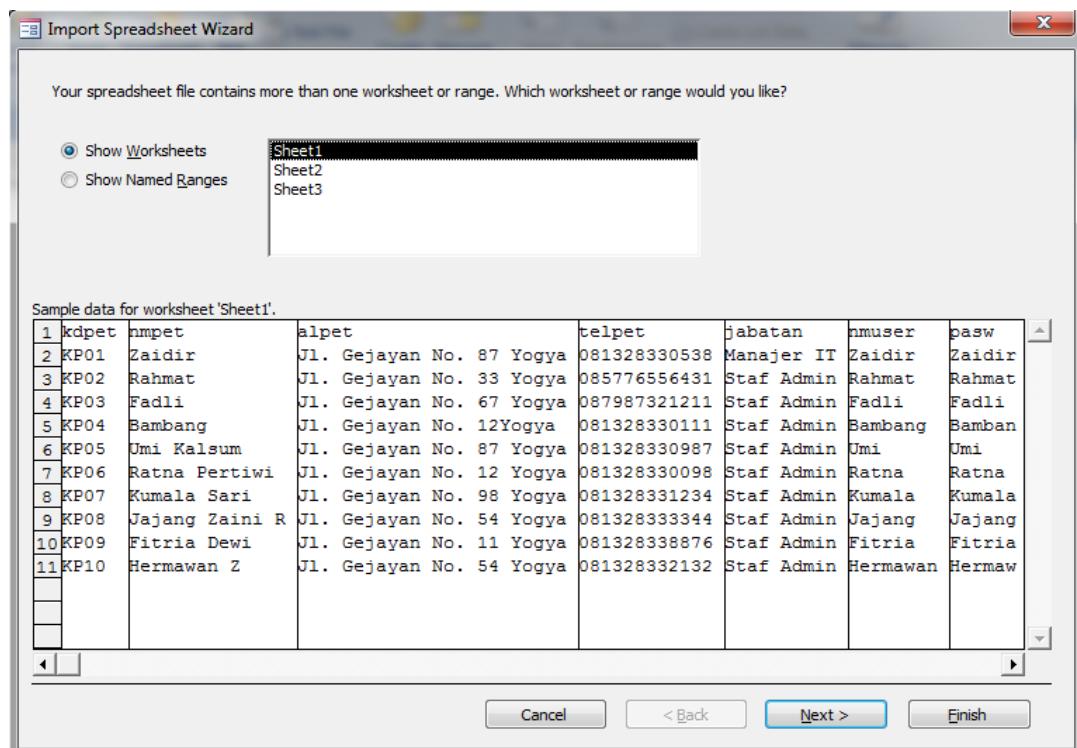


A screenshot of Microsoft Access showing the ribbon with the "External Data" tab selected. The "Import" icon in the "Import & Export" group is also circled. The "Datasheet" view is active, showing a table named "Table1" with one record. An arrow points from the "External Data" tab to the "Import" icon.

- Pilih **Excel**
- Tentukan sumber data **Excel** yang akan dimasukkan ke tabel **MS Access**

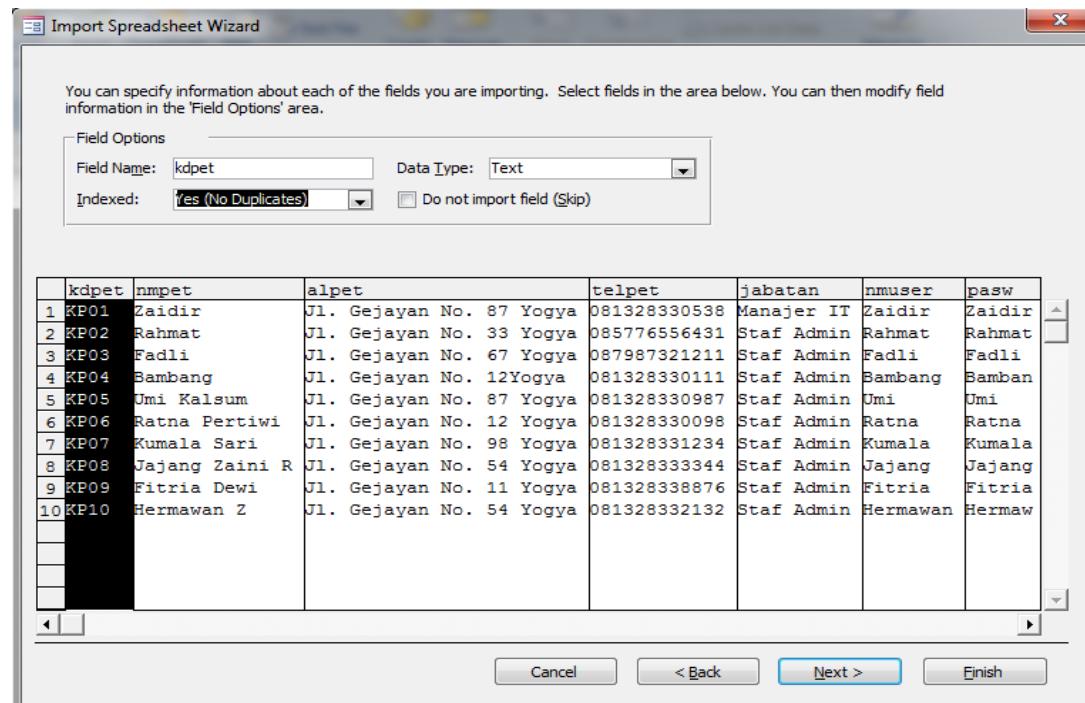


- Pilih *Import the source data into a new table in the current database*, kemudian klik tombol **OK**, sehingga akan ditampilkan kotak dialog sebagai berikut:

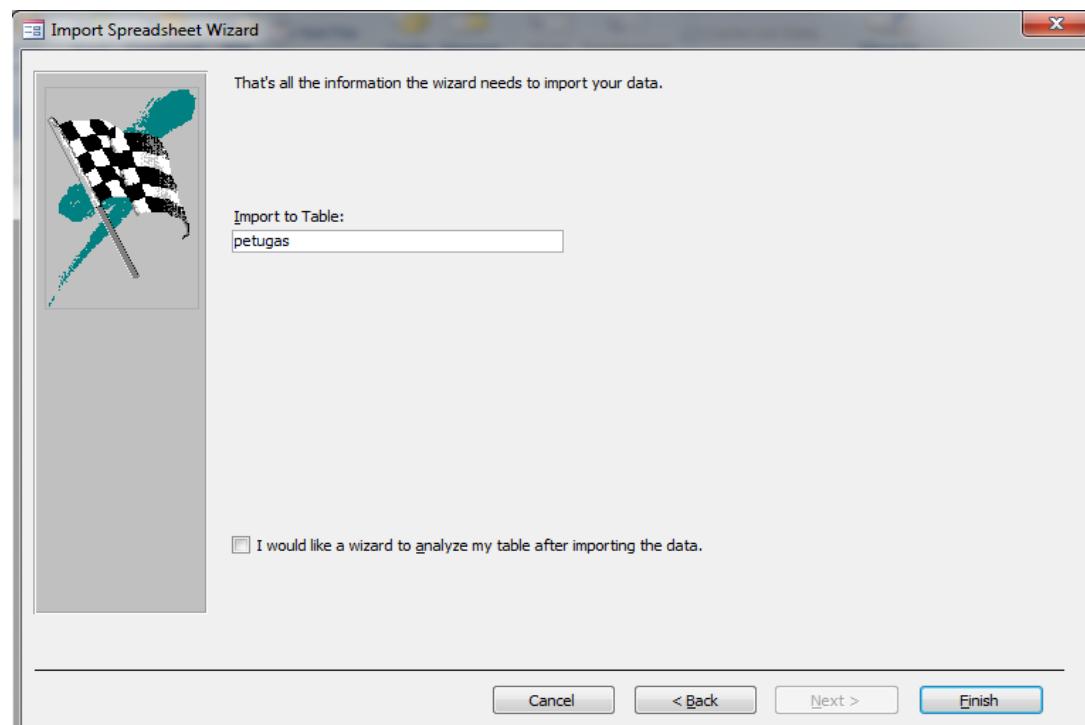


- Karena data memang berada di Sheet1 maka klik tombol **Next**.

- Beri tanda centang pada “First Row Contains Column Headings”, kemudian klik tombol **Next**.
- Tentukan tipe *index (field)* yang dijadikan *primary key* dengan isian **Yes (No Duplicate)** kolom paling kiri, kemudian klik tombol **Next**.



- Klik pilihan *Choose my own primary key* dan pilih *field* yang dijadikan kunci, kemudian klik tombol **Next**.
- Ketik nama tabel yang akan digunakan untuk menyimpan data (ketik **petugas**), klik tombol **Finish**.



- Klik tombol **Close**.

Sekarang database akan berubah seperti tampilan berikut ini

	kdpet	nmpet	alpet	telpet	jabatan	nmuser	pasw	levelp
KP01	Zaidir	Jl. Gejayan No. 87 Yogy			Manajer IT	Zaidir	Zaidir	Admin
KP02	Rahmat	Jl. Gejayan No. 33 Yogy			Staf Admin	Rahmat	Rahmat	User
KP03	Fadli	Jl. Gejayan No. 67 Yogy			Staf Admin	Fadli	Fadli	User
KP04	Bambang	Jl. Gejayan No. 12Yogy			Staf Admin	Bambang	Bambang	User
KP05	Umi Kalsum	Jl. Gejayan No. 87 Yogy			Staf Admin	Umi	Umi	User
KP06	Ratna Pertwi	Jl. Gejayan No. 1081328330987			Staf Admin	Ratna	Ratna	User
KP07	Kumala Sari	Jl. Gejayan No. 9081328331234			Staf Admin	Kumala	Kumala	User
KP08	Jajang Zaini R	Jl. Gejayan No. 5081328333344			Staf Admin	Jajang	Jajang	User
KP09	Fitria Dewi	Jl. Gejayan No. 1081328338876			Staf Admin	Fitria	Fitria	User
KP10	Hermawan Z	Jl. Gejayan No. 5081328332132			Staf Admin	Hermawan	Hermawan	User

## 2. Konversi data Excel ke MySQL

- Buat file excel atau buka file petugas.xlsx yang sudah ada. Modifikasi file tersebut dengan menghilangkan baris judul kolomnya.

	A	B	C
1	KP01	Zaidir	Jl. Gejayan No. 87 Yogy
2	KP02	Rahmat	Jl. Gejayan No. 33 Yogy
3	KP03	Fadli	Jl. Gejayan No. 67 Yogy
4	KP04	Bambang	Jl. Gejayan No. 12Yogy
5	KP05	Umi Kalsum	Jl. Gejayan No. 87 Yogy
6	KP06	Ratna Pertwi	Jl. Gejayan No. 12 Yogy
7	KP07	Kumala Sari	Jl. Gejayan No. 98 Yogy
8	KP08	Jajang Zaini R	Jl. Gejayan No. 54 Yogy
9	KP09	Fitria Dewi	Jl. Gejayan No. 11 Yogy
10	KP10	Hermawan Z	Jl. Gejayan No. 54 Yogy
11			

- Lakukan penyimpanan dengan **Save As**, dan beri nama petugas dan tipenya adalah CSV(MS-DOS).
- Buat database baru dengan menggunakan MySQL dan beri nama DBKOP.
- Buat tabel petugas dengan struktur sebagai berikut:

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0005 sec)

```
SELECT * FROM petugas LIMIT 0 , 30
```

#	Column	Type	Collation	Attributes	Null	Default	Extra	Action
1	kdpet	char(4)	latin1_swedish_ci		No	None		<a href="#">Change</a> <a href="#">Drop</a> More ▾
2	nmpet	varchar(40)	latin1_swedish_ci		No	None		<a href="#">Change</a> <a href="#">Drop</a> More ▾
3	alpet	varchar(50)	latin1_swedish_ci		No	None		<a href="#">Change</a> <a href="#">Drop</a> More ▾
4	telpet	varchar(15)	latin1_swedish_ci		No	None		<a href="#">Change</a> <a href="#">Drop</a> More ▾
5	jabatan	varchar(20)	latin1_swedish_ci		No	None		<a href="#">Change</a> <a href="#">Drop</a> More ▾
6	nmuser	varchar(20)	latin1_swedish_ci		No	None		<a href="#">Change</a> <a href="#">Drop</a> More ▾
7	pasw	varchar(20)	latin1_swedish_ci		No	None		<a href="#">Change</a> <a href="#">Drop</a> More ▾
8	levelp	varchar(20)	latin1_swedish_ci		No	None		<a href="#">Change</a> <a href="#">Drop</a> More ▾

Check All / Uncheck All With selected: [Browse](#) [Change](#) [Drop](#) Primary Unique Index

[Print view](#) [Relation view](#) [Propose table structure](#) [Track table](#)

[Add](#)  column(s)  At End of Table  At Beginning of Table  After kdpet [Go](#)

- Klik menu Import, selanjutnya pilih file petugas.CSV yang telah dibuat.

Importing into the table "petugas"

File to Import:

File may be compressed (gzip, bzip2, zip) or uncompressed.  
A compressed file's name must end in .[format].[compression]. Example: .sql.zip

Browse your computer:  petugas.csv (Max: 8,192KiB)

Character set of the file:

Partial Import:

- Klik menu GO yang ada pada bagian bawah

### 3. Data lain

Untuk memperlancar materi konversi data, berikut ini data anggota koperasi yang sudah ada di MS Excel. Silahkan dimasukkan ke database **MS ACCESS** dan **MySQL** yang sudah tercipta dari proses praktikum tadi.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	idangt	nmagt	kelamin	alamat	telp	pekerjaan											
2	1 Dadang S	L	Jl. S Parman No. 87 Yogy	085667878345	Petani												
3	2 Wahyu Ananto	L	Jl. Pancasila No. 3 Yogy	085667875456	Wiraswasta												
4	3 Ferr Sulisty	L	Jl. Sudirman No. 54 Yogy	085667871111	Peg Negeri												
5	4 Fahril S	L	Jl. MT Haryono No. 87 Yogy	085667872234	Peg Swasta												
6	5 Vivi Safitri	P	Jl. Herman Yohanes 87 Yogy	085667875467	Petani												
7	6 Weni Sujimin	P	Jl. Cut Nyak Dien No. 23 Yogy	085667876655	Wiraswasta												
8	7 Linawati	P	Jl. Kartini No. 49 Yogy	085667878790	Peg Swasta												
9	8 Ambarawati	P	Jl. Sisingamangaraja No. 5 Yogy	085667875456	Wiraswasta												
10	9 Mahmud Al Fajri	L	Jl. Sultan Agung No.8 Yogy	085667871211	Petani												
11	10 Guntur	L	Jl. Adisucipto No. 54 Yogy	085667874345	Wiraswasta												
12	11 Yasril	L	Jl. A Yani No. 98 Yogy	085667877687	Peg Swasta												
13	12 Iqbal	L	Jl. Pancasila No. 41 Yogy	085667873333	Petani												
14	13 Susianti	P	Jl. Kolombo No. 21 Yogy	085667877891	Wiraswasta												
15	14 Cindy	P	Jl. S Parman No. 54 Yogy	085667873324	Peg Swasta												
16	15 Asrul	L	Jl. Mawar No. 32 Yogy	085667877656	Petani												
17	16 Endang	P	Jl. Sukun No. 77 Yogy	085667878998	Petani												
18	17 Rini Cahyani	P	Jl. Sultan Agung No. 32 Yogy	085667874444	Peg Swasta												
19	18 Nani S	P	Jl. Kolombo No. 45 Yogy	085667871213	Peg Swasta												
20	19 Zubaidi	L	Jl. Gejayan No. 54 Yogy	085667873423	Wiraswasta												
21	20 Tuti S	P	Jl. Kartini No. 89 Yogy	085667877786	Petani												
22	21 Ferdinand	L	Jl. Empu Tantular No. 3 Yogy	085667874534	Wiraswasta												
23	22 Sulaiman	L	Jl. Sisingamangaraja No. 45 Yogy	085667875645	Peg Swasta												
24	23 Agus Riyadi	L	Jl. Tendean No. 32 Yogy	085667878988	Peg Swasta												
25	24 Kasdiyono	L	Jl. Herman Yohanes 32 Yogy	085667879999	Petani												
26	25 Aniek Sasmita	P	Jl. C Simanjuntak No. 23 Yogy	085667878452	Wiraswasta												
27																	

### C. Tugas

5. Kumpulkan semua hasil praktikum dari awal sampai akhir.
6. Masukkan dalam satu *folder* yang diberi nama sesuai nomor mahasiswa masing-masing.
7. Kumpulkan dengan menggunakan CD (dikumpulkan bersama-sama dengan teman yang lain). Cukup 1 CD saja untuk semua.

## **DAFTAR PUSTAKA**

- Kadir, A., 2010, Mudah Mempelajari Database MySQL, ANDI, Yogyakarta
- Mustakini, Jogyianto, H., 2009, Sistem Informasi Teknologi, ANDI, Yogyakarta
- Nugroho, A., 2011, Perancangan dan Implementasi Sistem Basis Data, Media Komputindo, Jakarta.
- Nugroho, B., 2005, Database Relasional dengan MySQL, ANDI, Yogyakarta.
- Permana,B., 2005, 36 Jam Belajar computer Microsoft Access 2003, PT. Elex Media Komputindo, Jakarta
- Riyanto Tosin, R. dan Hokki Puji Haryadi, H.P., 1997, Cara Mudah Belajar Access 97, Penerbit Dimastindo, Jakarta
- Waljiyanto, 2000, Sistem Basis Data : Analisis dan Pemodelan Data, Edisi Pertama, J&J Learning, Yogyakarta