



PRAKTIKUM PERANCANGAN BASIS DATA

[Modul Praktikum STRATA I]

PROGRAM SISTEM INFORMASI STMIK NUSA MANDIRI

**Tim Penyusun Modul STMIK NUSA
MANDIRI 2013**

Basis Data dan Normalisasi

☑ Konsep Dasar Basis Data

Basis Data (*Database*) dapat dibayangkan sebagai sebuah lemari arsip. Jika kita memiliki sebuah lemari arsip dan berwenang untuk mengelolanya. Atau kumpulan informasi yang terorganisasi dan disajikan untuk tujuan khusus. Prinsip utama basis data adalah pengaturan data atau arsip. Sedangkan tujuan utamanya adalah kemudahan dan kecepatan dalam pengambilan data atau arsip. Sedangkan sistem basis data merupakan perpaduan antara basis data dan sistem manajemen basis data (DBMS). DBMS (*Database Management System*) adalah *software* yang menangani semua akses ke basis data. Contoh dari DBMS yaitu Microsoft Access, MySQL, Oracle, Server 2000, Interbase, Paradox, dan Lain-Lain.

Komponen-komponen basis data antara lain :

1. Entitas

Entitas digunakan untuk menerapkan integritas pada tingkat Entity (Tabel), agar setiap Instances (Record/Baris) pada suatu Entity bersifat unique yang disebut sebagai Primary Key sehingga dapat dibedakan antara yang satu dengan yang lain.

Contoh:

- a. Semua pelanggan, atau pelanggan saja dengan entitas Adi, Ryan, Endah dan seterusnya.
- b. Semua Mobil atau mobil apa saja dengan entitas mobil Kijang, Starlett dan lain-lain.

2. Atribut

Setiap entitas pasti memiliki atribut yang mendeskripsikan karakteristik dari entitas tersebut. Penentuan atau pemilihan atribut-atribut yang relevan bagi sebuah entitas merupakan hal penting lainnya dalam pembentukan model data. Penetapan atribut bagi sebuah entitas umumnya memang didasarkan pada fakta yang ada. Tetapi tidak selalu seperti itu. Istilah atribut sebenarnya identik dengan pemakaian kolom data.

Contoh :

- a. Entitas pelanggan

Atributnya kd_pelanggan, nm_pelanggan, alamat, notelpon.

☑ **Normalisasi**

Dalam perancangan sebuah basis data perlu dilakukan secara cermat agar dihasilkan data yang kompak dan efisien dalam penggunaan ruang penyimpanan, cepat dalam pengaksesan dan mudah dalam manipulasi data. Salah satu cara yang dapat dilakukan dalam merancang basis data seperti ini adalah dengan melakukan normalisasi.

Normalisasi adalah proses penyusunan tabel-tabel yang tidak redudan (doubel), yang dapat menyebabkan anomali pada saat terjadi operasi manipulasi data seperti tambah, ubah, dan hapus. Anomali yaitu proses basis data yang memberikan efek samping yang tidak diharapkan (misalnya menyebabkan ketidakkonsistenan data atau membuat sesuatu data menjadi hilang ketika data lain dihapus). Adapun bentuk-bentuk dalam normalisasi:

1. Bentuk Tidak Normal

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti format tertentu, dapat saja tidak lengkap dan terduplikasi, data dikumpulkan apa adanya sesuai keadaannya. Data didapat dari bentuk dokumen yang ada.

2. Bentuk Normal Pertama (1st Normal Form)

Sebuah tabel disebut 1NF jika :

- a. Tidak ada baris yang duplikat dalam tabel tersebut.
- b. Masing-masing cell bernilai tunggal.

3. Bentuk Normal Kedua (2nd Normal Form)

Bentuk Normal Kedua(2NF) terpenuhi jika pada sebuah tabl semua atribut yang tidak termasuk dalam primary key memiliki ketergantungan fungsional pada primary key secara utuh.

4. Bentuk Normal Ketiga (3rd Normal Form)

Menghilangkan anomali-anomali hasil dari ketergantungan fungsional.

5. Bentuk Normal Keempat (4th Normal Form)

6. BCNF (*Boyce Code Normal Form*)

Berikut ini adalah contoh kasus mengenai pembelian barang yang digambarkan dalam normalisasi:

1. Dokumen Dasar

Faktur Pembelian Barang				
PT.XYZ				
Jl.Bekasi Timur 24				
Kode supplier :A01			tanggal:9 /09/2009	
NamaSupplier :bbb			Nomor:002	
Kode	Nama barang	Qty	Harga	jml
T01	TV 20 inchi	10	1200000	12.000.000
T02	TV 29 inchi	5	2400000	120.000.000
Total Faktur				132.000.000
Jatuh tempo 12/09/2009				

Faktur Pembelian Barang				
PT.XYZ				
Jl.Bekasi Timur 24				
Kode supplier :A03			tanggal:10 /09/2009	
Nama Supplier :ccc			Nomor :005	
Kode	Nama barang	Qty	Harga	jml
K01	Kulkas 1 pintu	6	1.000.000	6.000.000
K02	Kulkas 2 pintu	4	1.400.000	5.600.000
Total Faktur				11.600.000
Jatuh tempo 14/09/2009				

2. Bentuk Tidak Normal (UNF)

Bentuk Tidak Normal (UNF)

Kd_fak
Kd_supp
Nm_supp
Kd_brg
Nm_brg
Tgl
J_tempo
Qty
Hrg
Jml
Tot

3. Bentuk Normal Pertama (1NF)

Bentuk Normal Pertama (1NF)

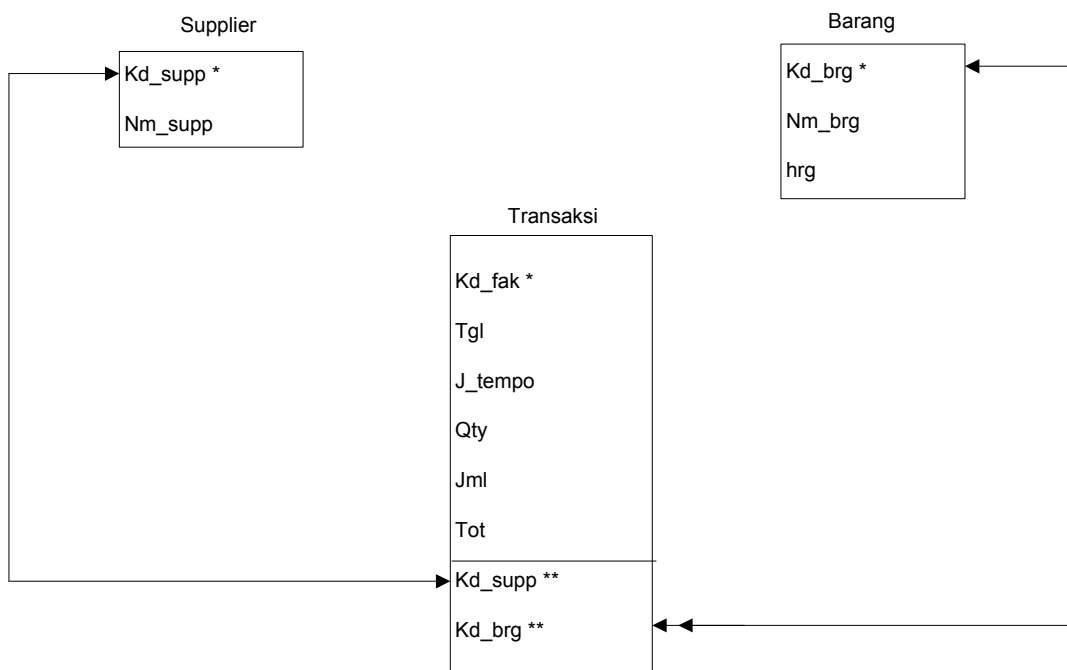
Kd_fak *
Tgl
J_tempo
Qty
Jml
Tot
Kd_supp *
Nm_supp
Kd_brg *
Nm_brg
Hrg

Keterangan :

* = Candidate Key

4. Bentuk Normal Kedua (2NF)

Bentuk Normal Kedua (2NF)



Keterangan:

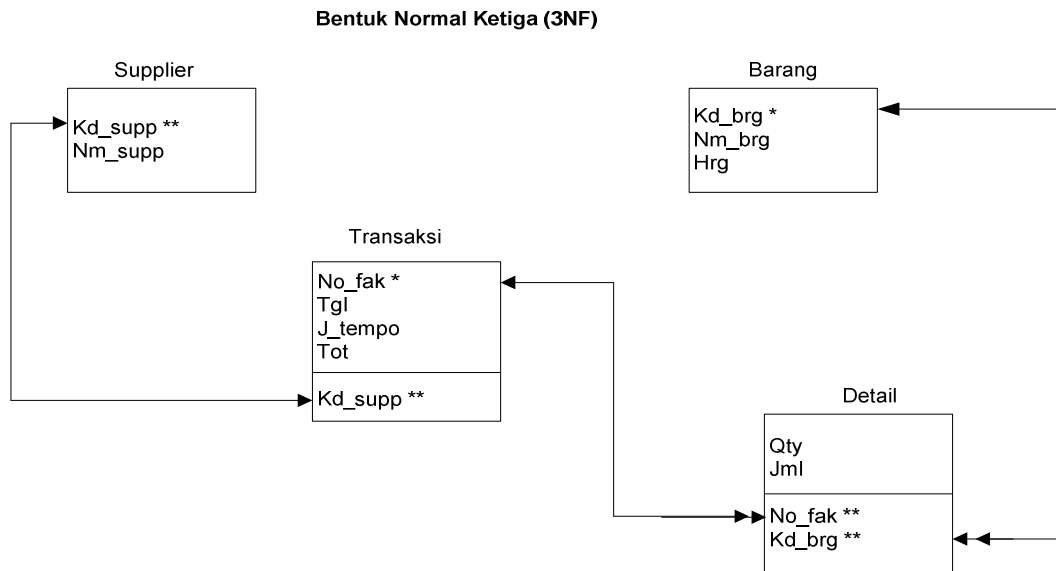
* : Primary Key

↔ : One to One Relationships

** : Foreign Key

↔ : One to Many Relationships

5. Bentuk Normal Ketiga (3NF)



Keterangan:

* : Primary Key

** : Foreign Key

↔ : One to One Relationships

↔ : One to Many Relationships

Database, Tabel dan Relasi

☑ Mengenal Lingkungan Kerja Microsoft Access 2010

Microsoft Access adalah sebuah program aplikasi basis data dengan model relasional. Aplikasi ini menggunakan mesin basis data Microsoft Jet Database Engine, dan juga menggunakan tampilan grafis yang intuitif sehingga memudahkan pengguna. Versi yang akan kita gunakan adalah Microsoft Office Access 2010.

Microsoft Access dapat menggunakan data yang disimpan di dalam format Microsoft Access, Microsoft Jet Database Engine, Microsoft SQL Server, Oracle Database, atau semua container basis data yang mendukung standar ODBC.

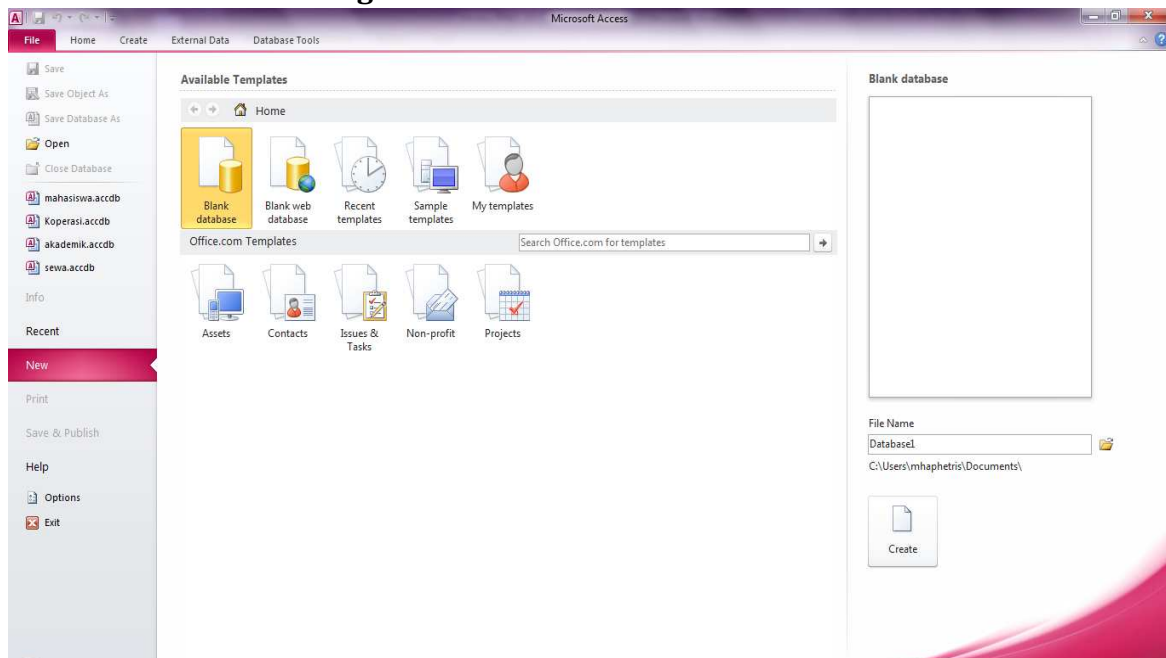
1. Susunan atau Hierarki pada Database

Character → Field → Record → File → Database

☑ Membuat Basis Data di Microsoft Access 2010

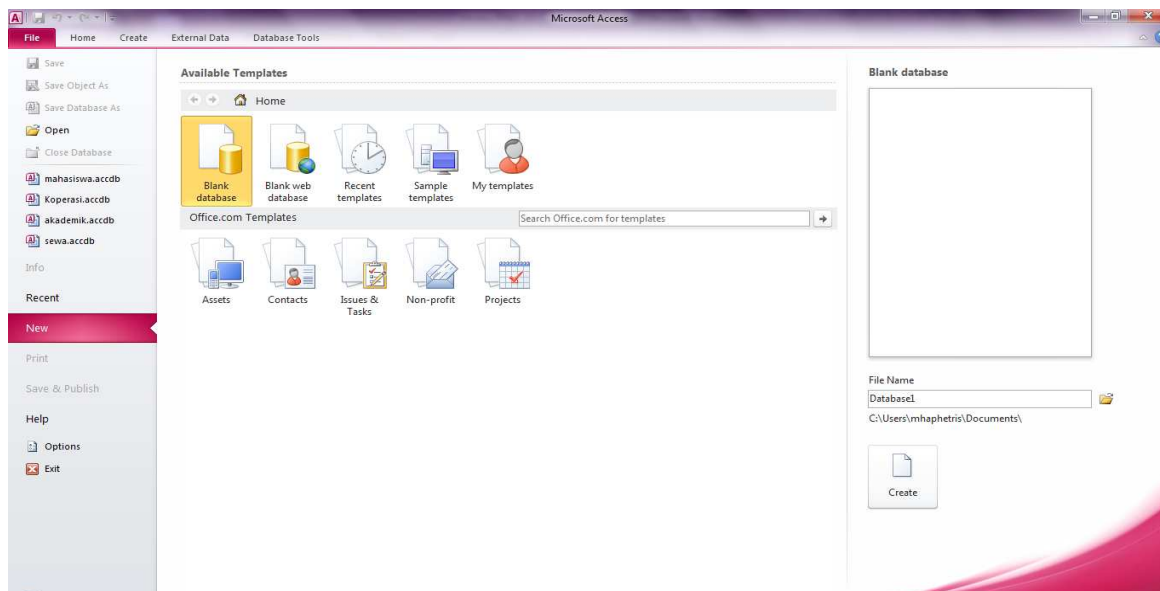
1. Mengaktifkan Ms. Access 2010

Klik menu **Start → All Program → Microsoft Office → Microsoft Access 2010**



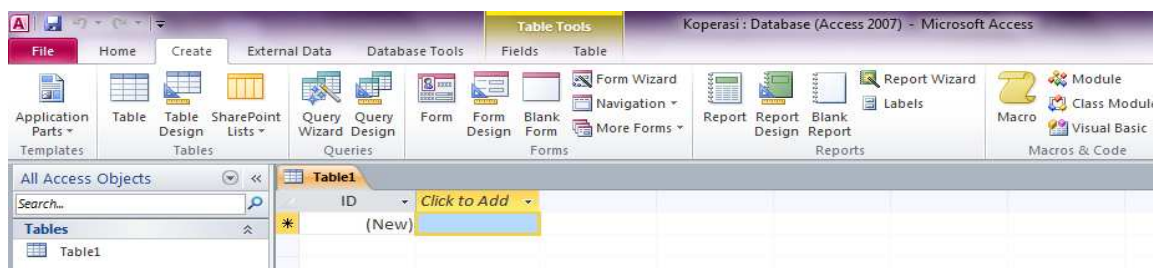
2. Membuat Database

Klik **Blank Database** maka akan tampil jendela Blank Database →



→ Isi *File Name* sesuai dengan nama Database → Klik **icon folder** untuk memilih atau mengubah lokasi penyimpanan → lalu Klik **Create** untuk membuat database.

3. Objek yang ada pada Microsoft Access 2010



a. Tables

Merupakan objek utama dalam sebuah file database sekaligus objek pertama yang harus dibuat karena didalam tabel inilah seluruh data-data yang dibutuhkan disimpan.

b. Query

Merupakan objek yang berfungsi untuk menampilkan, mengubah dan menganalisa data.

c. Form

Merupakan objek yang digunakan untuk menampilkan data yang tersimpan dalam tabel (untuk pembuatan *user interface*).

d. *Reports*

Merupakan objek yang digunakan untuk merangkum dan mencetak data atau suatu laporan secara efektif.

e. *Macros*

Merupakan objek yang digunakan untuk mengoptimasi sekaligus menghemat waktu yang diperlukan dalam pembuatan aplikasi database. Dengan kata lain Macro adalah kumpulan beberapa perintah, yang tersusun dalam sebuah daftar.

f. *Modules*

Merupakan objek yang digunakan untuk menuliskan kode program atau prosedur yang dapat digunakan untuk program aplikasi.

Latihan membuat database:

Buatlah sebuah database dengan nama **koperasi.accdb**

4. Pembuatan Tabel

Sebelum membuat tabel, kita diharuskan membuat database terlebih dahulu seperti pembahasan sebelumnya. Bukalah database koperasi.accdb pada pembahasan sebelumnya. Langkah-langkah membuat tabel :

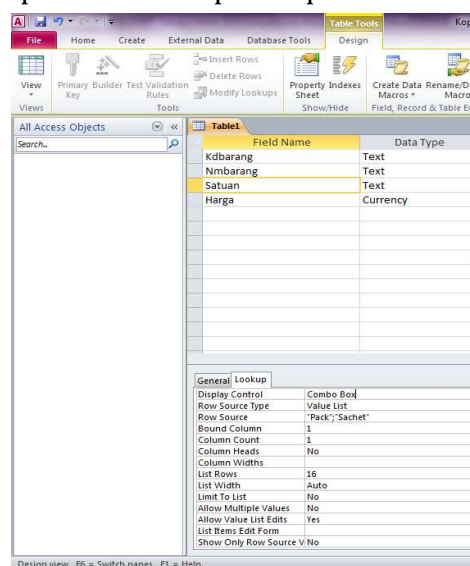
- Klik Menu **Create** Pilih **Table Design**
- Buat tabel sesuai dengan strukturnya.

Tabel : Barang

Field Name	Type	Size	Keterangan
Kdbarang	Text	4	Primary Key
Nmbarang	Text	30	
Satuan	Text	6	Combo Box
Harga	Currency	-	

1) Pembuatan *Combo Box*

Klik Field Satuan → Klik tab **Lookup** pada **Field Properties** → Lalu ubah setingan menjadi seperti dibawah ini:



Display Control : Combo Box

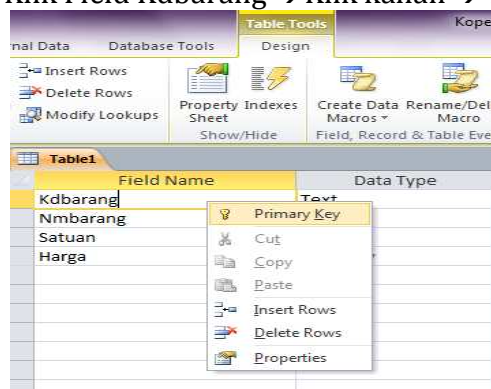
Row Source Type : Value List

Row Source : "Pack" ; "Sachet"

General	Lookup
Display Control	Combo Box
Row Source Type	Value List
Row Source	"Pack"; "Sachet"
Bound Column	1
Column Count	1
Column Heads	No
Column Widths	
List Rows	16
List Width	Auto
Limit To List	No
Allow Multiple Values	No
Allow Value List Edits	Yes
List Items Edit Form	
Show Only Row Source Values	No

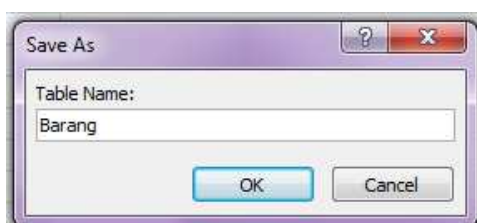
2) Membuat *Primary Key*

Klik Field Kdbarang → Klik kanan → Pilih **Primary Key**



c. Setelah itu simpan tabel dengan nama Barang

Klik Menu **File** → Klik **Save** → Ketikkan nama tabel yang akan disimpan → **Ok**



d. Lalu isikan data pada tabel barang sebagai berikut:

Kdbarang	Nmbarang	Satuan	Harga	Click to Add
B001	Nescafe Cream Isi 30	Pack	Rp20.000	
B002	Coffemix 50 gr	Sachet	Rp1.500	
B003	Kapal Api Spesial Isi 10	Pack	Rp7.500	
B004	Torabika Mocca 50 gr	Sachet	Rp1.200	
B005	Good Day Moccacino Isi 5	Pack	Rp45.000	
*				

Latihan membuat tabel

Pada Pembahasan sebelumnya kita telah mempelajari Konsep Dasar Basis Data. Kita juga telah mempelajari cara membuat database serta mendesain tabel di *Microsoft Access 2010*. Buka kembali database **Koperasi.accdb** yang telah kita buat pada pertemuan sebelumnya dan tambahkan 3 buah tabel baru dengan struktur sebagai berikut:

1. Nama tabel: **Karyawan** (diisi sebanyak 3 record)

Field Name	Data Type	Size	Keterangan
NIK	Text	9	Primary Key
Nmkaryawan	Text	30	
Bagian	Text	15	Combo Box → Administrasi, Kasir

2. Nama tabel: **Penjualan** (diisi sebanyak 3 record)

Field Name	Data Type	Size	Keterangan
Nofaktur	Text	5	Primary Key
Tglfaktur	Date/Time	8	
NIK	Text	9	Combo Box → Tabel Karyawan

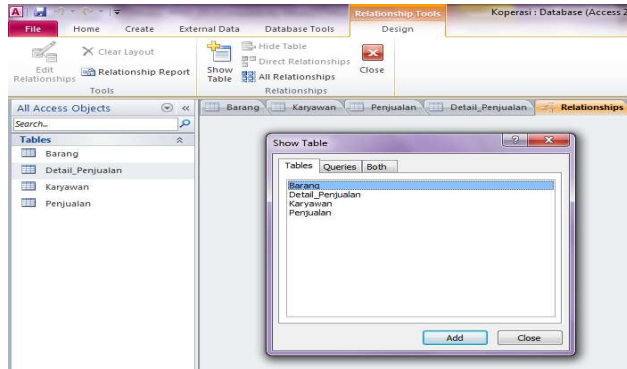
3. Nama tabel: **Detail_Penjualan** (diisi sebanyak 3 record)

Field Name	Data Type	Size	Keterangan
Nofaktur	Text	5	Combo Box → Tabel Penjualan
Kdbarang	Text	4	Combo Box → Tabel Barang
Jumbel	Number		

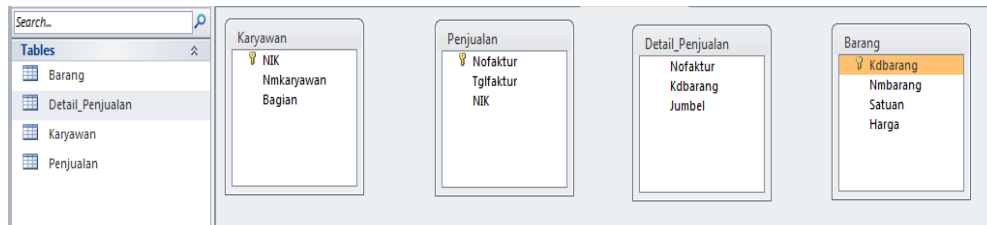
🔐 Selamat Mengerjakan 🔐

5. Membuat *Relationship*

Untuk membuat *relationship*, buka salah satu tabel dalam keadaan *design* → lalu klik **icon Relationship**, maka akan tampil jendela **Show Table** →



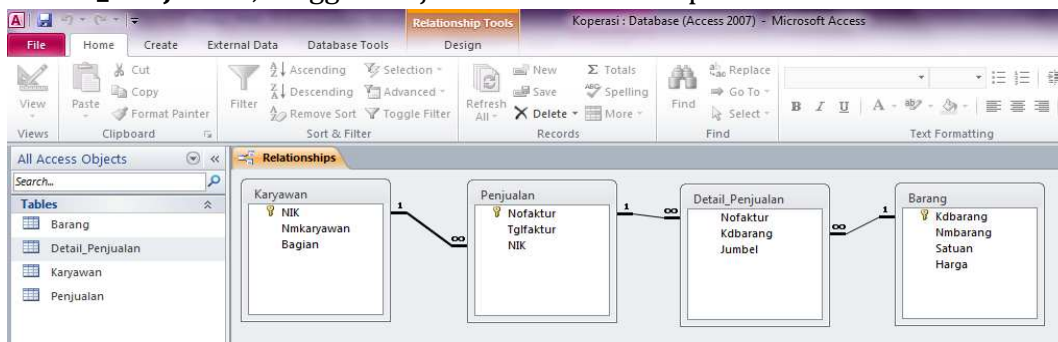
→ Pilih tabel satu persatu → kemudian Klik **Add** dan susun seperti berikut :



Untuk membuat relasi antar tabel, tarik garis diagonal dari *field* Kdbarang (*primary key*) dari tabel **Barang** ke *field* Kdbarang di tabel **Detail_Penjualan**, hingga tampil jendela **Edit Relationship** seperti berikut ini:



Lakukan hal yang sama untuk *field* NIK (*primary key*) pada tabel karyawan ke tabel **Penjualan**, dan *field* Nofaktur (*primary key*) pada tabel **Penjualan** ke tabel **Detail_Penjualan**, hingga menjadi sebuah relasi seperti berikut:

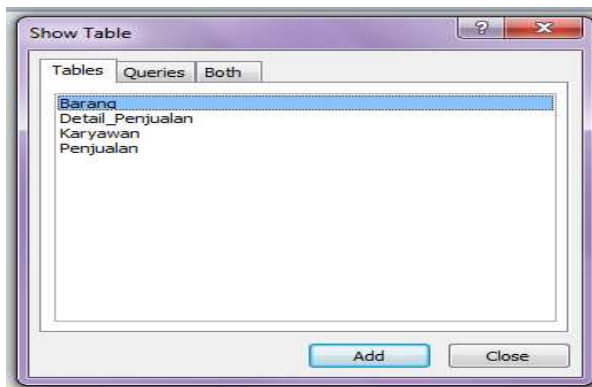


Data Definition Query

☑ Query

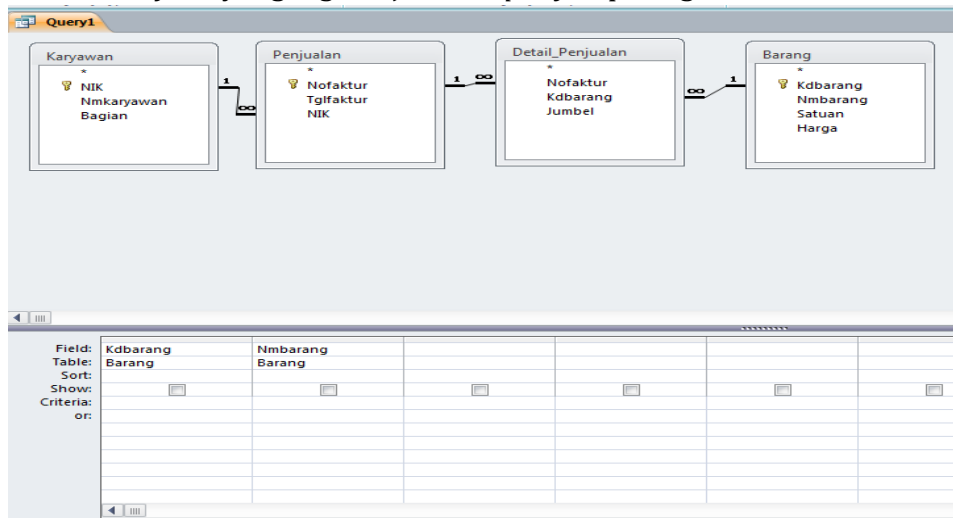
Query adalah suatu objek database yang dapat digunakan untuk menampilkan, menyunting atau menganalisa suatu data dengan memberikan baris-baris perintah tertentu. Pada aplikasi ini sudah dilengkapi dengan *Structured Query Language (SQL)* yang menjadi standard bahasa dalam mengolah database. Dengan *SQL* kita dapat mempersingkat penulisan listing program yang panjang karena pada setiap perintahnya dapat mencakup banyak perintah sekaligus. Langkah-langkah membuat *query*:

1. Klik Tab **Create**
2. Pilih icon **Query Design**, muncul jendela **Show Tables**.



3. Pilih nama tabel kemudian Klik **Add**, contohnya kita pilih tabel Barang.

3. Double klik *field* yang ingin dijadikan *query* seperti gambar dibawah ini:



Keterangan

- Field* : Nama *field* yang ingin ditampilkan
Table : Nama tabel dari *field* yang ditampilkan
Sort : Mengurutkan data hasil *query*
Show : Mengatur *field* ditampilkan atau tidak
Criteria : Syarat dari data yang ingin ditampilkan

Jika ingin menggunakan semua *field* pada tabel maka kita hanya klik dan drop tanda * pada tabel barang ke baris *field*. Jika tidak kita dapat memilih salah satu *field* dengan cara mendouble klik *field* yang diinginkan.

Terdapat 3 pilihan pada *toolbars View*, yaitu :

1. *Design View* : untuk menampilkan desain *tabel query*.
2. *Datasheet View* : untuk menampilkan data hasil *query*.
3. *SQL View* : untuk membuat *query* dengan *script SQL*.

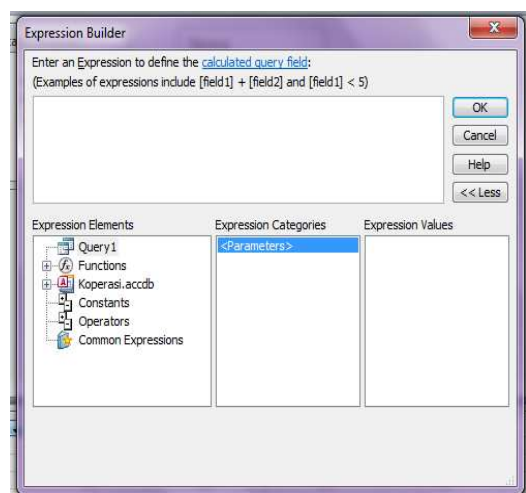
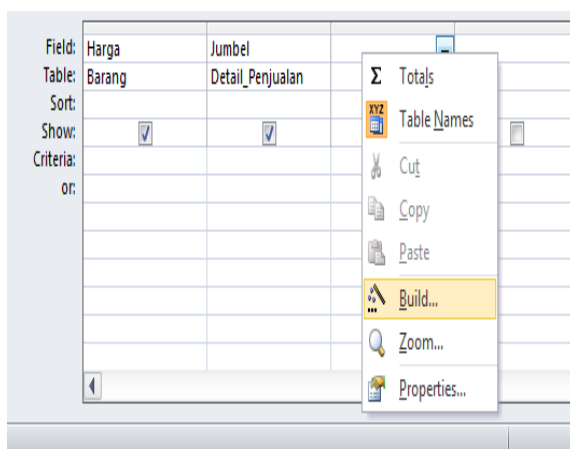
Pada bagian ini, jika kita sudah membuat desain *query* menggunakan *design view*, kita bisa melihat langsung *script SQL*nya pada *SQL View*.

☑ Membuat ekspresi pada query

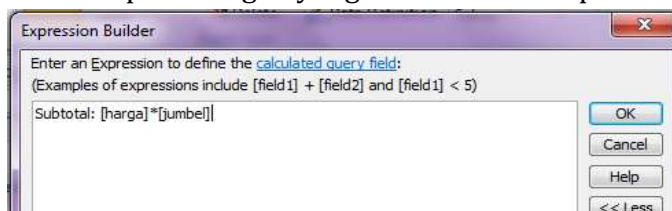
Pembuatan ekspresi pada *query* dilakukan, apabila ada sebuah penambahan *field* yang bersifat sementara (tidak bergabung dengan tabel) yang berfungsi untuk membantu dalam pembuatan *query* yang lebih kompleks. Cara pembuatan *query* antara lain:

1. Klik kanan pada *field* → lalu pilih **Build**
2. Maka akan tampil jendela **Ekspression**

Builder



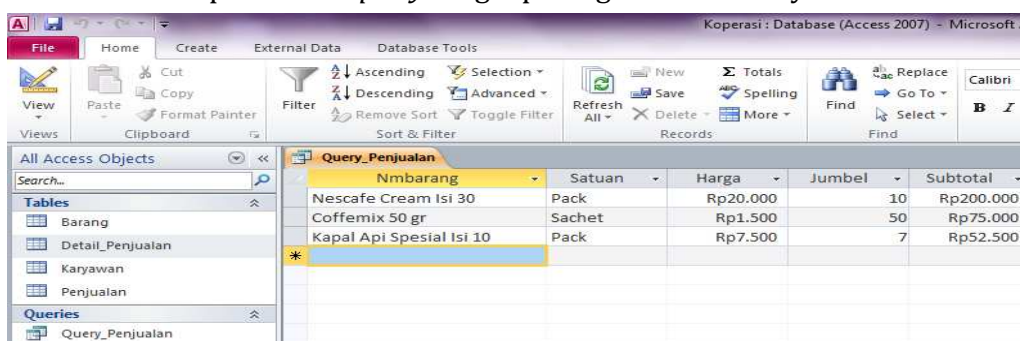
3. Ketikkan perhitungan yang akan dibuat Ekspresi → **Ok**



4. Setelah itu *running query* yang sudah dibuat dengan cara mengklik *icon run*



5. Maka akan tampil sebuah *query* lengkap dengan isi *recordnya*



Latihan membuat Query

1. Berdasarkan database **Koperasi.accdb** yang telah dibuat, buatlah **Query_Penjualan** berdasarkan struktur query dibawah ini:

Field	Asal Tabel
Nofak	Detail_penjualan
Kdbarang	Detail_penjualan
Nmbarang	Barang
Satuan	Barang
Harga	Barang
Jumlah	Detail_Penjualan
Subtotal	Ekspresi
Diskon	Ekspresi
Bonus	Ekspresi

Keterangan:

- ☒ Subtotal = jumlah * harga
- ☒ Diskon 15% , jika subtotal lebih dari Rp. 50.000,-
Diskon 10%, jika subtotal lebih dari Rp. 30.000,-
Diskon 7%, jika subtotal lebih dari Rp. 20.000,-
Selain itu diskon 5%
- ☒ Bonus 1 Pcs Kopi ABC Susu, jika kode barang yang dibeli adalah B002

🔧 Selamat Mengerjakan 🖋️

Data Definition Language (DDL)

☑ Data Definition Language (DDL)

Merupakan suatu sub bahasa SQL yang berfungsi mendefinisikan database dan tabel. Perintah yang digunakan diantaranya : **CREATE**, **ALTER**, dan **DROP**. Dengan menggunakan ketiga perintah tersebut kita sudah dapat membangun struktur data dengan lengkap.

1. Perintah **CREATE** : digunakan untuk membuat tabel baru

```
CREATE TABLE nama_tabel (field-1 tipe(panjang), field-2 tipe(panjang), field-n tipe(Panjang));
```

```
CREATE TABLE mahasiswa (nim text(8) not null primary key,nama text(30) not null, alamat text(50));
```

2. Perintah **ALTER** : digunakan untuk merubah, menambah(ADD), menghapus(DROP) struktur tabel.

```
ALTER TABLE nama_tabel [spesifikasi perubahan];
```

```
ALTER TABLE mahasiswa ADD jenis_kel text(1);
```

Beberapa perintah *alter* yang dapat digunakan antara lain:

a. Menambahkan *Primary key* pada *field* NIM di tabel mahasiswa

```
ALTER TABLE mahasiswa ADD CONSTRAINT NIM Primary Key(NIM);
```

b. Merubah ukuran dan tipe dari *field* NIM pada tabel mahasiswa

```
ALTER TABLE mahasiswa ALTER COLUMN NIM Number;
```

c. Menghapus *field* Jenis kelamin dari tabel mahasiswa

```
ALTER TABLE mahasiswa DROP COLUMN jenis_kel;
```

3. Perintah **DROP** : digunakan untuk menghapus tabel

```
DROP TABLE nama_tabel;
```

```
DROP TABLE mahasiswa;
```

Data Manipulation Language (DML)

☑ **Data Manipulation Language (DML)**

Merupakan sub bahasa SQL yang berfungsi memanipulasi data yang ada di dalam basisdata atau database. Perintah yang termasuk didalam DML adalah **INSERT**, **SELECT**, **UPDATE** dan **DELETE**.

1. Perintah **INSERT** : berfungsi untuk menyisipkan, memasukkan, atau menyimpan data dari luar sistem kedalam database.

```
INSERT INTO nama_tabel (field-1, field-2....field-n) VALUES (value-1, value-2,...,value-n);
```

atau

```
INSERT INTO nama_tabel VALUES (value-1, value-2,...,value-n);
```

```
INSERT INTO mahasiswa (nim,nama,tgl_lahir,alamat) VALUES
```

```
"12130001","Normah","Cipulir");
```

atau

```
INSERT INTO mahasiswa VALUES ("12130002","Risma"," Ciledug");
```

2. Perintah **SELECT** : berfungsi untuk membentuk tabel baru dengan cara mengcopy (*backup*) seluruh data dari tabel aktif.

```
SELECT field-1, field-2....,field-n INTO nama_tabel_baru FROM nama_tabel_aktif;
```

atau

```
SELECT * INTO nama_tabel_baru FROM nama_tabel_aktif;
```

```
SELECT * INTO mahasiswa_baru FROM mahasiswa;
```

3. Perintah **UPDATE**: berfungsi untuk mengubah satu atau lebih data *field* yang terdapat pada satu atau lebih *record*.

```
UPDATE nama_tabel SET field=value WHERE kriteria;
```

atau

```
UPDATE nama_tabel SET field-1=value-1, field-1=value-1,... ,field-n=value-n WHERE kriteria;
```

```
UPDATE mahasiswa SET nama="Nurika" WHERE NIM="12120002";
```

4. Perintah **DELETE** : berfungsi untuk menghapus satu atau beberapa *record* dari suatu

DELETE * FROM nama_tabel **WHERE** kriteria;

DELETE * FROM mahasiswa **WHERE**
nim="12120002";

Inner Join

Inner Join merupakan operasi yang digunakan untuk mendapatkan data gabungan dari dua tabel atau lebih yang akan menghasilkan data yang berpasangan saja. *Inner Join* dipakai untuk memperoleh data-data yang lebih detail dari tabel-tabel yang saling memiliki hubungan. Operasi *inner join* ini biasanya digunakan dalam perintah **Select**. Contoh penggunaan sintak innerjoin dalam pembuatan query sebagai berikut:

Bentuk dasar :

```
SELECT * FROM nama_tabel;
```

Keterangan:

- a. Tanda * berarti menampilkan keseluruhan *field* yang berada pada suatu tabel.

```
SELECT * FROM barang;
```

- b. Tanda * dapat diganti dengan **nama_tabel.nama_field** : untuk menampilkan *field* tertentu yang berada pada tabel tertentu.

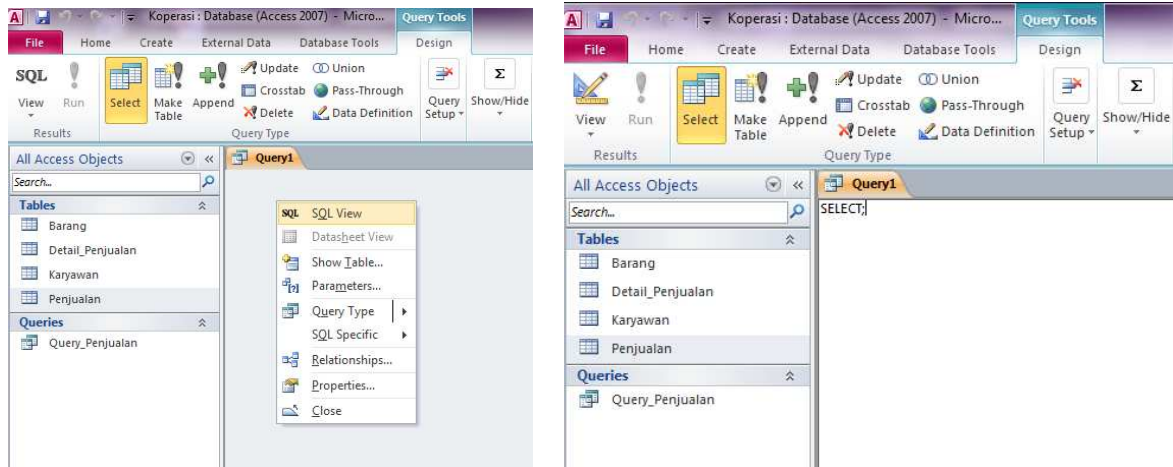
```
SELECT Barang.Kdbarang, Barang.Nmbarang, Barang.Harga FROM barang;
```

- c. Tanda * dapat digantikan juga dengan **prosesperhitungan as nama_field** : untuk menampilkan *field* baru yang sebelumnya tidak berada pada tabel.

```
SELECT Barang.Kdbarang, Barang.Nmbarang, Barang.Harga , 10 as jumbel, harga*jumbel  
as subtotal FROM barang;
```

Didalam Microsoft Access 2010, kita dapat mengaplikasikan *inner join* dengan menggunakan salah *toolbars view* yaitu *SQL View*. Buka kembali database **Koperasi.accdb** yang telah dibuat dipembahasan sebelumnya.

1. Pilih menu **Create** → Pilih **icon Query Design** → Maka akan muncul jendela **Show tables** → Klik tombol **Close**, maka akan tampil jendela **Query**.
2. Klik kanan pada jendela *query* → Pilih **SQL View**, maka akan tampil jendela **SQL View** seperti dibawah ini:



3. Ketikkan sintaks *inner join* didalam jendela *query*.

Sumber → Latihan halaman 14


SELECT Barang.Nmbarang, Barang.Satuan, Barang.Harga, Detail_Penjualan.Jumbel,

[harga]*[jumbel] AS Subtotal,

IIf([Subtotal]>50000,15/100*[subtotal],IIf([Subtotal]>30000,10/100*[subtotal],IIf([Subtotal]>20000,7/100*[subtotal]*5/100*[Subtotal]))) AS Diskon,

IIf([Detail_Penjualan].[Kdbarang]="B002","1 Pcs Kopi ABC Susu","Tidak dapat") AS Bonus

FROM (Karyawan INNER JOIN Penjualan ON Karyawan.NIK = Penjualan.NIK) INNER JOIN (Barang INNER JOIN Detail_Penjualan ON Barang.Kdbarang = Detail_Penjualan.Kdbarang) ON Penjualan.Nofaktur = Detail_Penjualan.Nofaktur;

4. Klik **icon run**  apabila ingin melihat hasilnya.
5. Klik **File** → **Save**, apabila ingin menyimpan *query*.

Latihan PPBD UTS

Sebuah Toko Alat Musik di Jakarta dalam melakukan transaksi penjualan alat musiknya, ingin merubah sistemnya ke komputerisasi. Anda sebagai seorang programmer diminta merancang database menggunakan Microsoft Access 2010. Pembuatan tabel dan pengisian recordnya dapat anda kerjakan secara wizard yang tersimpan dalam sebuah database bernama **studiomusik.accdb**.

1. Buatlah tabel seperti struktur dibawah ini:

a. Nama tabel: **alatmusik**

Field Name	Data Type	Field Size	Keterangan
kdalat	Text	3	Primary Key
nmalat	Text	15	
harga	Currency	-	

Isikan tabel alatmusik seperti berikut ini:

kdalat	nmalat	harga
BIO	Biola	Rp 3.000.000
GTR	Gitar	Rp 2.000.000
PIO	Piano	Rp 5.000.000

b. Nama tabel: **kasir**

Field Name	Data Type	Field Size	Keterangan
kdakasir	Text	3	Primary Key
nmkasir	Text	20	
jabatan	Text	10	

Isikan tabel kasir seperti berikut ini:

kdkasir	nmkasir	jabatan
SHR	Sherina	Administrator
DRB	Derby	Kasir
TSY	Tasya	Kasir

c. Nama tabel: **penjualan**

<i>Field Name</i>	<i>Data Type</i>	<i>Field Size</i>	<i>Keterangan</i>
nonota	<i>Text</i>	5	<i>Primary Key</i>
tgl	<i>Date</i>	-	
kdkasir	<i>Text</i>	3	
kdalat	<i>Text</i>	3	
jumbel	<i>Number</i>	-	

Isikan tabel penjualan seperti berikut ini:

nonota	tgl	kdkasir	kdalat	jumbel
NT001	16/07/2010	SHR	GTR	7
NT002	16/07/2010	TSY	PIO	4
NT003	17/07/2010	DRB	BIO	2

2. Buatlah **QueryPenjualanAlat** dengan menggunakan **SQL** dengan struktur *query* sebagai berikut:

<i>Field</i>	<i>Asal Tabel</i>
nonota	penjualan
tgl	penjualan
kdkasir	penjualan
nmkasir	kasir
kdalat	penjualan
nmalat	alat
harga	alat
jumbel	penjualan
subtotal	<i>ekspresi</i>
ppn	<i>ekspresi</i>
diskon	<i>ekspresi</i>
total	<i>ekspresi</i>

Ketentuan Soal

- ☒ Subtotal = harga x jumbel
- ☒ PPn = 10% dari harga
- ☒ Diskon diberikan 10 % jika jumlah beli lebih dari 5 buah, 7% jika jumlah beli lebih dari 3 buah, selain itu tidak dapat

📄 Selamat Mengerjakan 📄



☑ **Pengenalan MySql dan Lingkungan Kerja**

MySql adalah sebuah program database server yang mampu menerima dan mengirimkan datanya dengan sangat cepat, multi user serta menggunakan perintah standar SQL (*Structure Query Language*). MySql merupakan sebuah database server yang *free* artinya kita bebas menggunakan database ini untuk keperluan pribadi atau usaha tanpa membeli atau membayar lisensiya.

MySQL merupakan hasil ciptaan dari **Michael "Monty" Widenius, David Axmark,** dan **Allan Larson**. Pada tahun 1995 mereka lalu mendirikan perusahaan bernama MySQL AB di Swedia. Tujuan awal diciptakannya MySQL yaitu untuk mengembangkan aplikasi web yang akan digunakan oleh salah satu klien MySQL AB. Database MySQL, biasa kita baca mai-es-ki-el merupakan perangkat lunak database yang berbentuk database relasional atau dalam bahasa basis data sering disebut dengan *Relation Database Management System* (RDBMS) yang menggunakan suatu bahasa permintaan bernama SQL. Perbedaan antara MySQL dan SQL, MySQL adalah program database server sedangkan SQL adalah bahasa yang digunakan didalamnya.

☑ **Kelebihan MySQL**

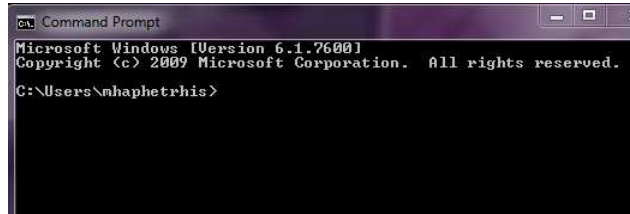
MySQL adalah sebuah database *server* dapat juga berperan sebagai *client* sehingga disebut database *client / server* yang *open source* dapat berjalan pada OS manapun dengan *platform* Windows maupun Linux.

☑ **Penggunaan MySQL dengan Command Prompt**

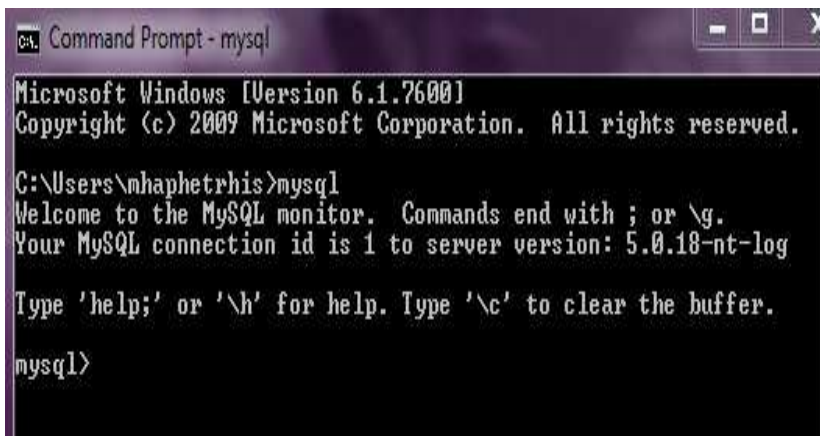
Dalam pembahasan ini kita akan menggunakan paket program XAMPP versi 1.7.3 dengan sistem operasi Windows XP. Paket XAMPP biasanya di-instal pada direktori atau folder C:\XAMPP. Sedangkan untuk databasenya akan tersimpan di direktori C:\xampp\mysql\data.

Untuk dapat menggunakan mysql terlebih dahulu aktifkan dengan cara sebagai berikut :

1. Pilih menu **Start** → **All Programs** → **Accessories** Kemudian Pilih **Command Prompt**. Selanjutnya akan tampil jendela sebagai berikut:



2. Kemudian ketikkan **mysql** pada command prompt tersebut untuk menjalankan program mysql: **C:\Users\mhaphetrhis>mysql**. Bila perintah tersebut berhasil, maka pada layar akan tampil ucapan selamat datang seperti tampilan dibawah ini:



☒ Pengantar Sintak Dasar

MySQL merupakan bahasa pemrograman database dimana penulisan sintaknya tidak serumit bahasa pemrograman lainnya seperti Java, C++, dan sebagainya. Satu hal yang perlu diingat bahwa setiap penulisan script MySQL di Dos-Prompt harus selalu diakhiri dengan tanda titik koma (;). Didalam *source* MySQL yang terinstal secara *default* telah terisi sebuah database yang bernama *mysql* dan *information_schema*.

1. Sintak untuk menampilkan semua nama database yang telah ada adalah:

mysql>show databases;

```
C:\Windows\system32\cmd.exe - mysql
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\mhaphetnhis>mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1 to server version: 5.0.18-nt-log

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

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
+-----+
2 rows in set (0.05 sec)

mysql>
```

Keterangan: Contoh diatas menunjukkan bahwa dengan menggunakan sintak show databases; berarti kita dapat menampilkan seluruh nama database yang ada.

```
C:\Windows\system32\cmd.exe - mysql
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\mhaphetnhis>mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1 to server version: 5.0.18-nt-log

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

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
+-----+
2 rows in set (0.05 sec)

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

2. Sintak untuk bisa masuk kedalam salah satu sistem database adalah:

Mysql>use nama_database;

Keterangan: dengan mengetikkan sintak use **mysql**; akan tampil kalimat database change, berarti kita telah masuk ke dalam database mysql yang telah ada.

```
C:\Windows\system32\cmd.exe - mysql
mysql> use mysql;
Database changed
mysql> show tables;
+-----+
| Tables_in_mysql |
+-----+
| columns_priv |
| db |
| func |
| help_category |
| help_keyword |
| help_relation |
| help_topic |
| host |
| proc |
| procs_priv |
| tables_priv |
| time_zone |
| time_zone_leap_second |
| time_zone_name |
| time_zone_transition |
| time_zone_transition_type |
| user |
+-----+
17 rows in set (0.00 sec)

mysql>
```

3. Di setiap database tentu ada terdapat beberapa tabel yang menjadi komponen dasar dari sebuah database. Sintak untuk menampilkan seluruh tabel yang telah ada di dalam database mysql adalah:

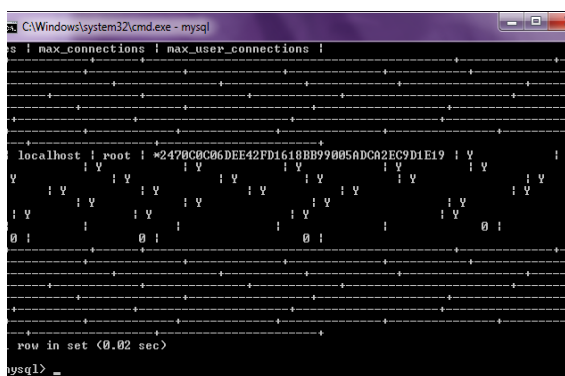
Mysql>show tables;

Dari gambar disamping , kita dapat melihat bahwa didalam database mysql sudah ada beberapa tabel yang telah default dari mysql nya sendiri. Didalam database tersebut ada table columns_priv

(columns_privilege), db, host, tables_privilege, dan user dimana tiap tabelnya memiliki fungsi tertentu yang dapat diakses oleh tiap user. Database mysql ini biasanya sering digunakan bagi webmaster sebagai tempat penyimpanan data, karena tabel-tabel yang telah ada dan table yang dibutuhkan sebagian besar telah terpenuhi.

4. Untuk bisa melihat isi seluruhnya dari salah satu tabel seperti user, maka gunakan sintak sebagai berikut ini:

```
Mysql>select * from nama_tabel;
```



Maka akan tampil suatu kolom-kolom dan baris-baris yang telah ada seperti gambar disamping ini:

Dari gambar di atas akan terlihat kolom host, user dan password yang telah terisi. Isi dalam kolom tersebut seperti gambar dibawah ini:

Host	User	Password
Localhost	Root	1fb744464c9bba54

Localhost merupakan nama komputer kita sendiri apabila komputer tersebut kita jadikan server. Sedangkan **root** adalah nama user yang berhak penuh untuk mengatur database yang telah ada seperti menambah user, memberi fasilitas kepada user lain agar user lain dapat akses ke database tersebut, bahkan juga root bisa membuat database baru yang dapat digunakan kapanpun. Kolom password yang terisi oleh kombinasi angka dan huruf diatas yaitu **1fb744464c9bba54** merupakan hasil password yang telah terenkripsi pada sintak terdahulu yaitu dengan menambahkan sintak password itu sendiri.

Data Definition Language

☑ **Data Definition Language**

Skema basis data dispesifikasikan oleh sekumpulan definisi dengan sebuah bahasa khusus yang disebut *Data Definiton Language* (DDL). Hasil kompilasi DDL berupa tabel-tabel yang disimpan dalam sebuah file, disebut data *dictionary* (Kamus Data) atau data *directory*. Kamus data adalah sebuah *file* yang berisi metadata. *File* ini yang dikonsultasi sebelum data yang sebenarnya dibaca atau dimodifikasi oleh sistem basis data.

☑ **Penggunaan sintak DDL dalam membuat database**

Database adalah sebuah media utama yang harus dibuat dalam membangun sebuah basis data agar nantinya dapat diletakkan beberapa tabel dengan field-fieldnya. Bila kita lihat dalam mysql database hanyalah semacam direktori saja, berbeda jika dibandingkan dengan program database yang berbentuk visual seperti access ataupun paradox. Pada mysql database akan disimpan dan dikumpulkan pada sebuah direktori khusus yang bernama data. Dalam direktori **C:\Apache2triad\mysql\data**, semua sub direktori yang ada didalamnya disebut database.

1. Sintak untuk menciptakan database baru

```
Mysql>create database nama_database;
```

```
Mysql>create db_jual;
```

2. Sintak untuk menghapus database.

```
Mysql>drop database nama_database;
```

```
Mysql>drop database db_jual;
```

☑ Penggunaan sintak DDL dalam membuat tabel

Tabel adalah obyek utama yang harus ada pada sebuah basis data karena didalamnya semua data akan disimpan. Tabel terletak pada sebuah database, sehingga pembuatan table dilakukan setelah database dibuat. Untuk membuat tabel kita harus terlebih dahulu **mengaktifkan** database, perintah yang digunakan adalah **use**. Syntak penulisannya adalah : **Mysql>use namadatabase;** Setelah masuk kedalam database, anda telah diijinkan melakukan operasi-operasi menyangkut tabel dan isinya.

1. Sintak untuk membuat sebuah tabel

```
Mysql>create table nama_tabel(field-1 type(lenght), field-2 type(lenght),..... field-  
n type(lenght),(.....));
```

```
Mysql>create table tb_pel (kd_pel char(5), nm_pel varchar(20), alamat varchar(50),  
telp varchar(15));
```

2. Sintak untuk melihat suatu struktur/field tabel

```
Mysql>desc nama_tabel;
```

```
Mysql>desc tb_pel;
```

3. Sintak untuk membuat *primary key*

Dalam membuat sebuah database, kita pasti akan menemui sebuah *record* yang datanya tidak boleh sama dengan *record* yang lain. Agar data tidak sama kita dapat membuat sebuah kolom data dengan isi yang berbeda dengan kolom lain. Kita dapat mendeklarasikan sebuah kolom sebagai kunci primer atau *primary key*.

```
Mysql >alter table nama_table add primary key (nama_field_primary);
```

```
Mysql >alter table tb_pel add primary key (kd_pel);
```

4. Sintak untuk menghapus *Primary Key*

```
Mysql>alter table nama_table drop primary key;
```

```
Mysql>alter table tb_pel drop primary key;
```

5. Sintak untuk menghapus tabel

```
Mysql>drop table nama_tabel;
```

```
Mysql>drop table tb_pel;
```

☑ **Perintah-Perintah DDL yaitu:**

No.	Keterangan	Perintah	Contoh
1.	Mengganti nama tabel	Rename table tabel_lama to tabel_baru;	Rename table pelanggan to pelanggan_baru;
2.	Perubahan field table	Alter table nama_table alter spesifikasi kondisi;	Alter table pelanggan add status varchar(10); Alter table pelanggan add status varchar(10) after nama;
3.	Menambah field unik	Alter table nama_table add index (nama_field);	Alter table pelanggan add index (nm_pel);
4.	Mengubah nama field	Alter table nama_table change field_lama field_baru type(length);	Alter table pelanggan change telp telp_pel varchar(15);
5.	Menghapus field	Alter table nama table drop nama_field;	Alter table pelanggan drop status;
6.	Menghapus index	Alter table nama_table drop index nama_field;	Alter table pelanggan drop index nm_pel;
7.	Membuat Validasi	Create table nama_tabel(field-1 type(lenght), field-2 type(lenght), ... field-n enum ("pilihan1","pilihan2"));	Create table pelanggan(kd_pel char(5), nm_pel varchar(20), alamat varchar(50), telp varchar(15), status enum ("member","nonmember"));

Data Manipulation Language

☑ **Data manipulation Language (DML)**

DML atau *Data Manipulation Language* merupakan perintah-perintah yang berfungsi untuk melakukan manipulasi data ataupun objek-objek yang ada didalam tabel. Definisi lainnya tentang DML adalah perintah-perintah untuk memanipulasi data pada basis data, misalnya perintah untuk memilih data (*query*), menyisipkan, mengubah dan menghapus data dalam basis data. Bentuk manipulasi yang dapat dilakukan oleh DML diantaranya adalah untuk keperluan pencarian kembali data lama, penyisipan data baru, penghapusan data, pengubahan data dll. Sedangkan DML sendiri menurut jenisnya dapat dibagi menjadi 2 Jenis yaitu *procedural* dan *non procedural*. Beberapa manfaat atau kegunaan dari DML diantaranya adalah sebagai berikut :

1. Pengambilan informasi yang disimpan dalam basis data.
2. Penyisipan informasi baru ke basis data.
3. Penghapusan informasi dari basis data.
4. Modifikasi informasi yang disimpan dalam basis data.
5. *Query* adalah perintah yang ditulis untuk mengambil informasi. Bagian dari DML yang menangani pengambilan informasi ini disebut bahasa *query*.

☑ **Penggunaan Data Manipulation Language (DML)**

1. Perintah Insert

Perintah INSERT digunakan untuk menambah data baru kedalam tabel. Insert mempunyai 2 format penulisan. Berikut ini contoh-contoh penulisan perintah INSERT dalam proses penambahan data kedalam tabel pelanggan:

```
mysql>insert into nama_tabel (daftar_field) values (daftar_nilai);
```

atau

```
Mysql>insert into nama_tabel set nama_field=nilai_field;
```

```
mysql>insert into tb_pel (kd_pel, nm_pel, alamat, telp) values ("P001","Risma","Ciledug","08561234567);
```

2. Perintah Select

Berfungsi untuk menampilkan data yang terdapat didalam database atau tabel.

```
Mysql>select * from nama_tabel;
```

```
Mysql>select * from tb_pel;
```

3. Perintah Update

Berfungsi untuk melakukan perubahan terhadap sejumlah data yang ada didalam tabel yang telah dibuat.

```
Mysql>update nama_tabel set field-2=nilai_field-2, field-n=nilai_field-n where field-1=nilai_field-1;
```

```
Mysql>update tb_pel set nm_pel="Normah", alamat="Kebayoran" where kd_pel="P001";
```

4. Perintah Delete

Berfungsi untuk menghapus sejumlah data yang ada didalam table yang telah dibuat.

```
Mysql> delete from nama_tabel where field=nilai_field;
```

```
Mysql> delete from tb_pel where kd_pel="P001";
```

Latihan:

1. Buat database dengan nama **db_barang**.
2. Buat tabel dengan nama **tb_barang**.

<i>Field</i>	<i>Type</i>	<i>Size</i>	<i>Keterangan</i>
kobar	<i>Char</i>	5	<i>Primary key</i>
nabar	<i>Varchar</i>	25	
hrng	<i>Int</i>		
stok	<i>Int</i>		

3. Isilah record pada tabel **tb_barang** seperti dibawah ini:

kobar	nabar	hrng	stok
B001	Pulpen	3500	20
B002	Pensil	2000	30
B003	Penghapus	1000	25

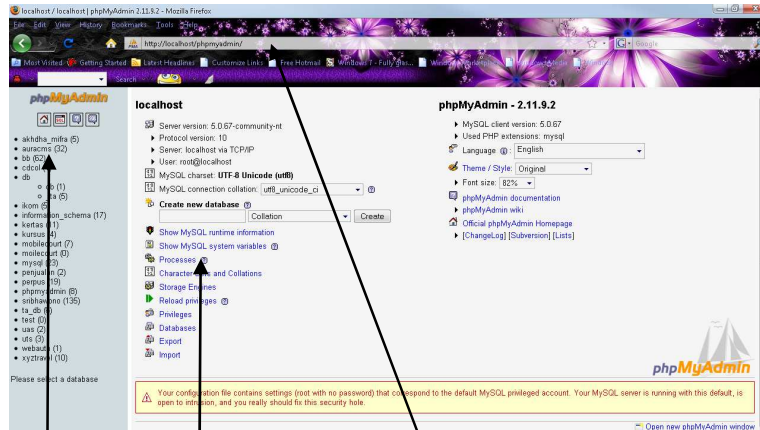
4. Tampilkanlah isi *record* dari tabel **barang**.
5. Gantilah **nabar** pada kobar **B001** menjadi **penggaris**.
6. Hapuslah **record** pada kobar **B003**.
7. Ganti nama *field* **hrng** menjadi **harga**.
8. Ganti stok= **20** menjadi stok =**10** untuk *record* dengan kobar "**B001**".
9. Ganti nama tabel **tb_barang** menjadi **tb_brg**.

1. Mengaktifkan PHPMyadmin

-

-

2. IDE *phpmyadmin*



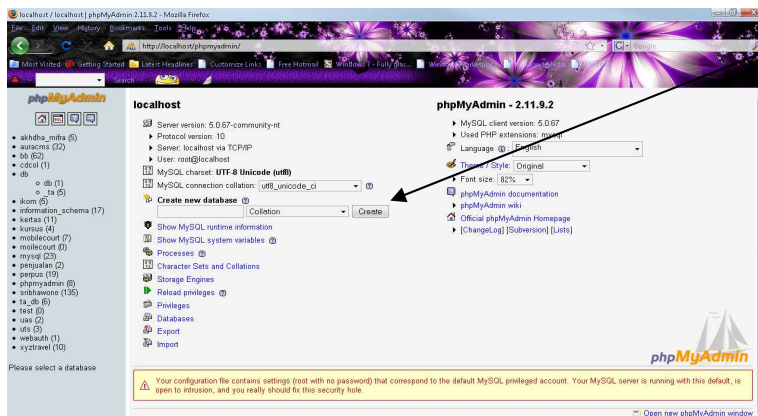
Database

Menu *mysql*

Address Bar

3. Membuat Database

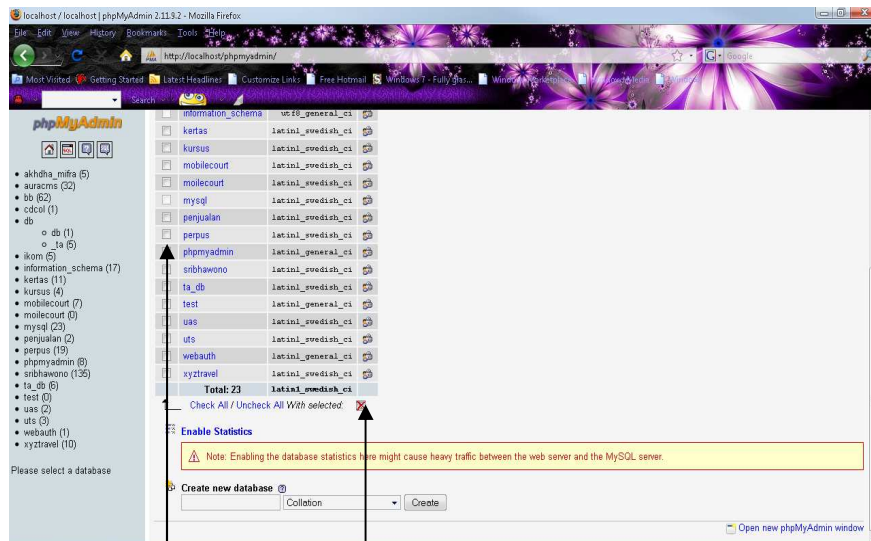
- Ketikan nama database pada textbox ***create new database***
- Kemudian klik ***create***



Klik ***Create***

4. Menghapus database

- Pilih menu database pada tampilan *Home* Phpmyadmin.
- Checklist*** pada database yang akan dihapus kemudian pilih ***drop***.

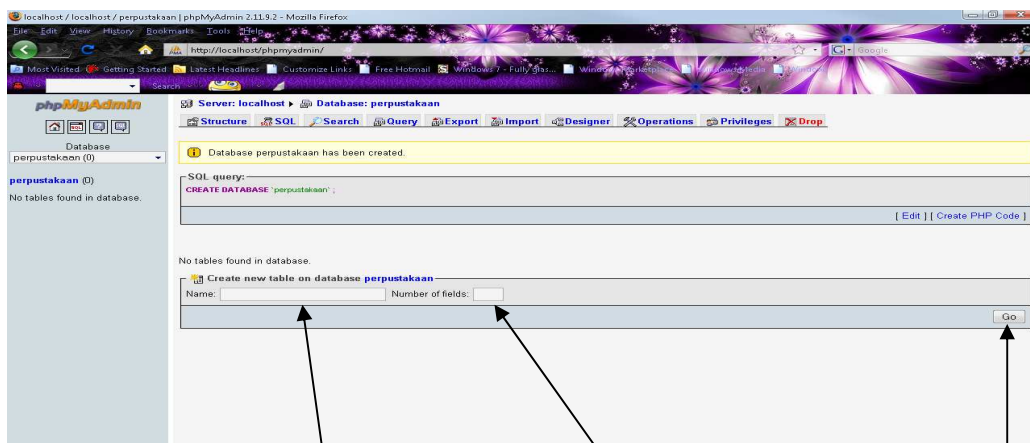


CheckList
database

Klik Drop

4. Pembuatan Tabel

- Ketikkan nama tabel pada **textbox name**, kemudian ketikkan jumlah **field** yang akan dibuat pada **number of field** tersebut setelah itu klik **Go** untuk memulai pengetikan struktur datanya.

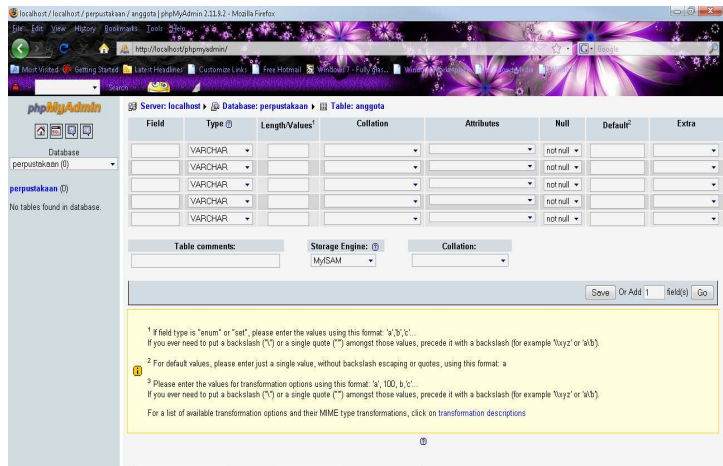


Ketikkan Nama Tabel

Ketikkan Jumlah Field

Klik Go

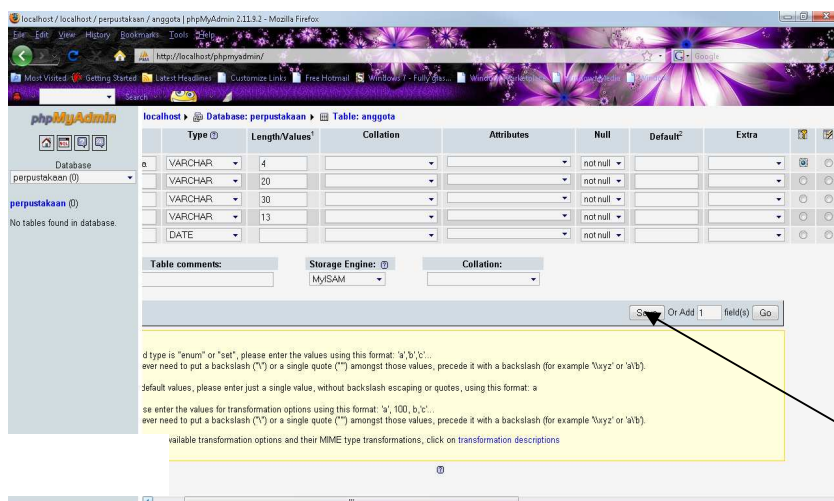
- b. Maka tampil layar seperti gambar di samping.



- c. Kemudian ketikkan struktur tabel seperti contoh dibawah ini

Nama Field	Type	Size	Keterangan
Id_anggota	Varchar	4	Primary Key
Nama	Varchar	20	
Alamat	Varchar	30	
Notelp	Varchar	13	
Tgl_lahir	Date		

- d. Setelah kita masukkan *field-field* yang dibutuhkan kemudian klik *save*



Klik Save

e. Apabila telah sukses maka akan tampil tampilan seperti dibawah ini

SQL query:

```
CREATE TABLE `perpustakaan`.`anggota` (
  `id_anggota` VARCHAR(4) NOT NULL,
  `nama` VARCHAR(20) NOT NULL,
  `alamat` VARCHAR(30) NOT NULL,
  `notelp` VARCHAR(13) NOT NULL,
  `tgl_lahir` DATE NOT NULL,
  PRIMARY KEY (`id_anggota`)
) ENGINE = MYISAM
```

[Edit]

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	id_anggota	varchar(4)	latin1_swedish_ci		No			
<input type="checkbox"/>	nama	varchar(20)	latin1_swedish_ci		No			
<input type="checkbox"/>	alamat	varchar(30)	latin1_swedish_ci		No			
<input type="checkbox"/>	notelp	varchar(13)	latin1_swedish_ci		No			
<input type="checkbox"/>	tgl_lahir	date			No			

↑ Check All / Uncheck All With selected:

Print view Relation view Propose table structure

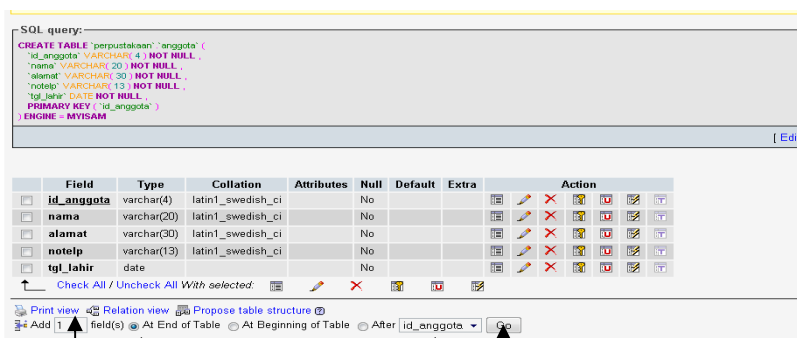
Add 1 field(s) At End of Table At Beginning of Table After id_anggota Go

Mengelola Tabel di PhpMyadmin

☑ Mengelola Tabel di PHPMyadmin

1. Menambah *Field*

- Buka tabel yang ingin diubah fieldnya.
- Klik menu tab **structure**
- Masukkan jumlah *field* yang ingin ditambah.
- Pilih posisi *field at end of table/at beginning of table/after dari field* yang sudah ada.
- Setelah itu klik **Go**

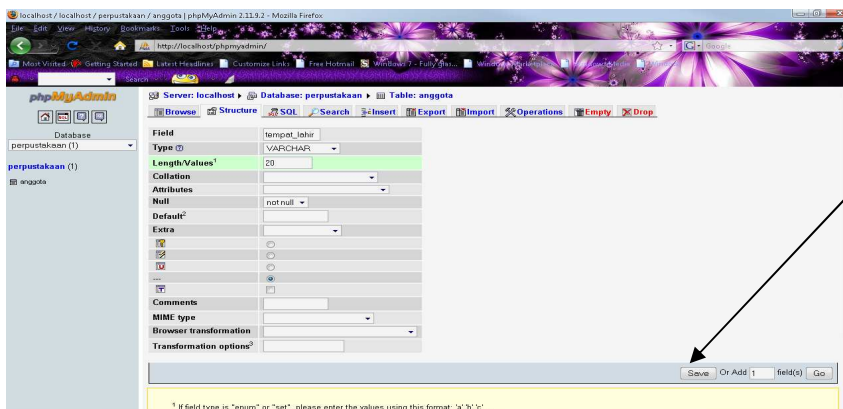


Jumlah *field*
yang baru

Posisi *field* yang
ditambah

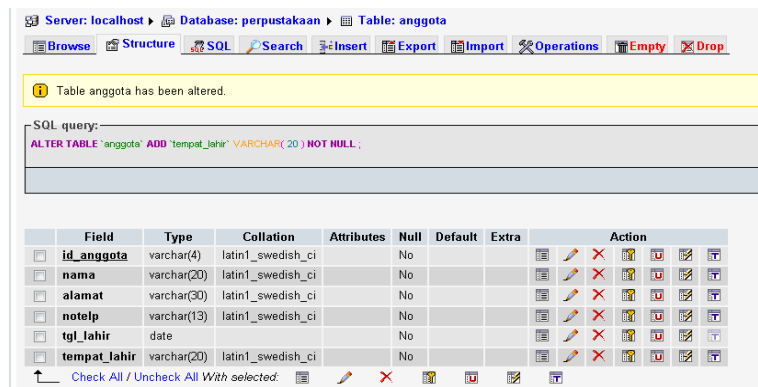
Klik GO

- Masukan nama *field* yang akan ditambahkan → kemudian klik **save**



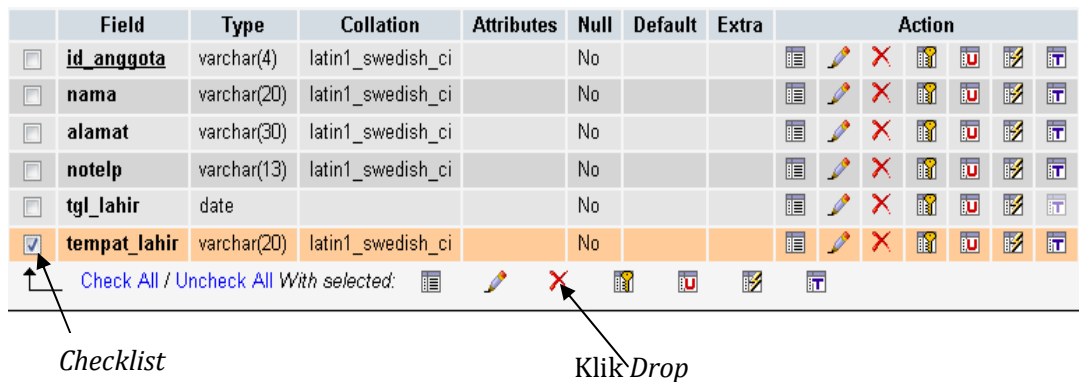
Klik Save

- g. Maka apabila telah di klik **save** akan tampil seperti dibawah ini:



2. Menghapus Field

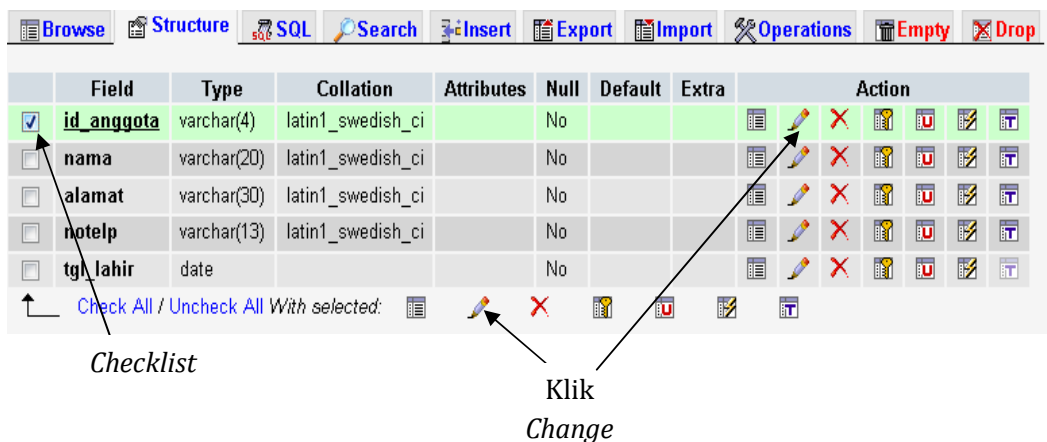
- a. *Checklist field* yang ingin dihapus kemudian klik **drop**



- b. Setelah klik **drop** maka akan tampil pertanyaan, pilih **yes**. Kemudian akan tampil pesan sukses.

3. Merubah Field

- a. *Checklist field* yang ingin dirubah, kemudian klik **change**



- b. Ubahlah nama *field* **id_anggota** menjadi *field* **kode**, setelah itu klik **save**

Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> kode	varchar(4)	latin1_swedish_ci		No			
<input type="checkbox"/> nama	varchar(20)	latin1_swedish_ci		No			
<input type="checkbox"/> alamat	varchar(30)	latin1_swedish_ci		No			
<input type="checkbox"/> notelp	varchar(13)	latin1_swedish_ci		No			
<input type="checkbox"/> tgl_lahir	date			No			
<input type="checkbox"/> Check All / <input type="checkbox"/> Uncheck All With selected:							

[Print view](#) [Relation view](#) [Propose table structure](#)
 Add 1 field(s) At End of Table At Beginning of Table After kode Go

4. Menambah Isi Record Tabel

- a. Klik database perpustakaan, kemudian *checklist* tabel yang ingin kita isikan data setelah itu klik **insert**



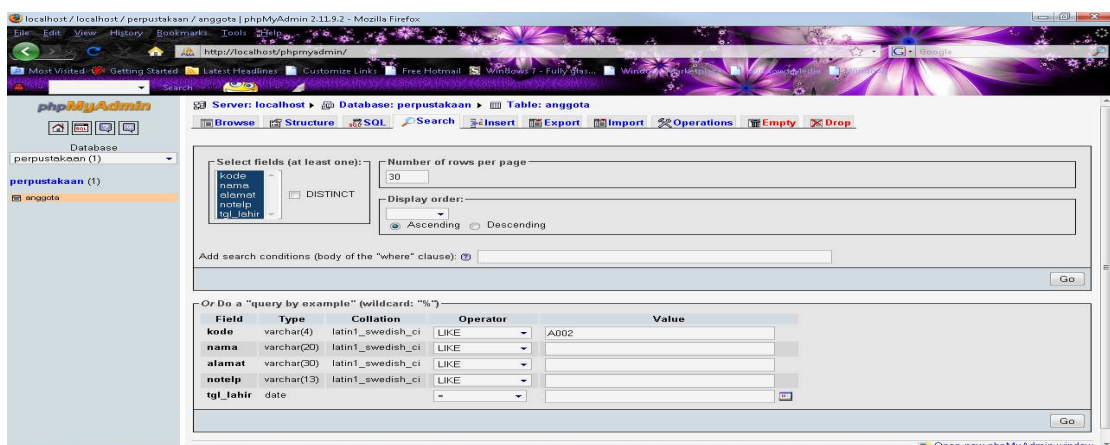
Checklist

Klik Insert

- b. Maka akan tampil layar seperti dibawah ini, kemudian isikan 2 record, setelah itu klik **Go**

5. Mencari Data

- a. Klik **Search** pada menu tab, kemudian masukkan kriteria yang ingin dicari di **value**, kemudian klik **Go**

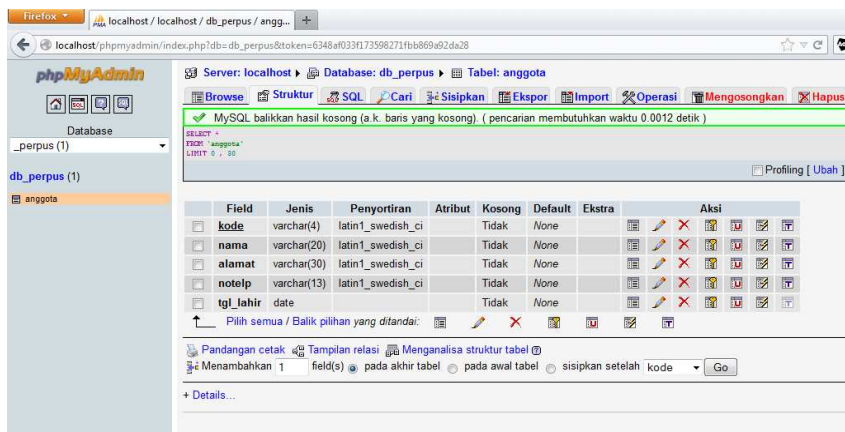


Export-Import Database

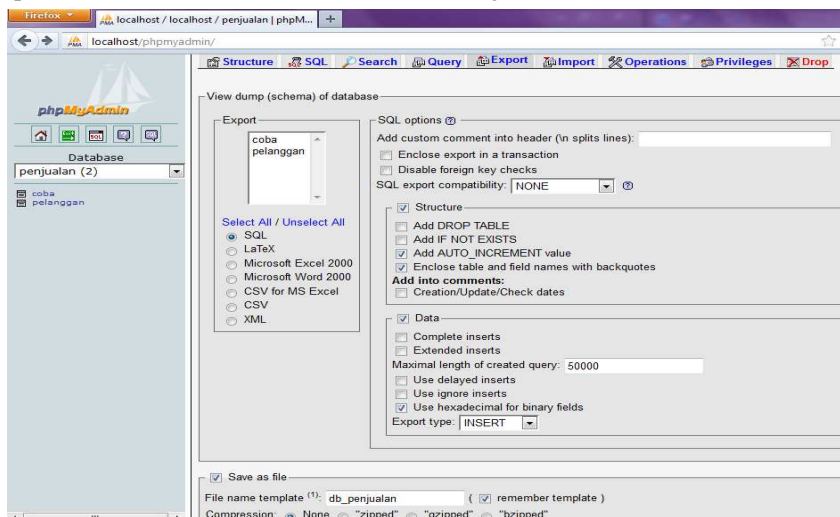
PhpMyAdmin memiliki fungsi-fungsi untuk proses pengelolaan database, dimana dengan PhpMyAdmin kita bisa menghapus tabel, import database maupun export database.

1. *Eksport Database*

- Masuk kembali ke halaman PhpMyAdmin.
- Pilih database yang hendak di-**Export**.
- Pada tab menu database pilih **Export**.

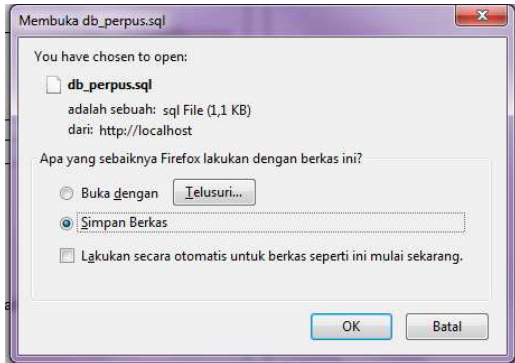


- Setelah itu akan tampil halaman *Export*, lalu pada opsi *Export* pilih SQL dan pada opsi **Save as File** ketikkan nama filenya.



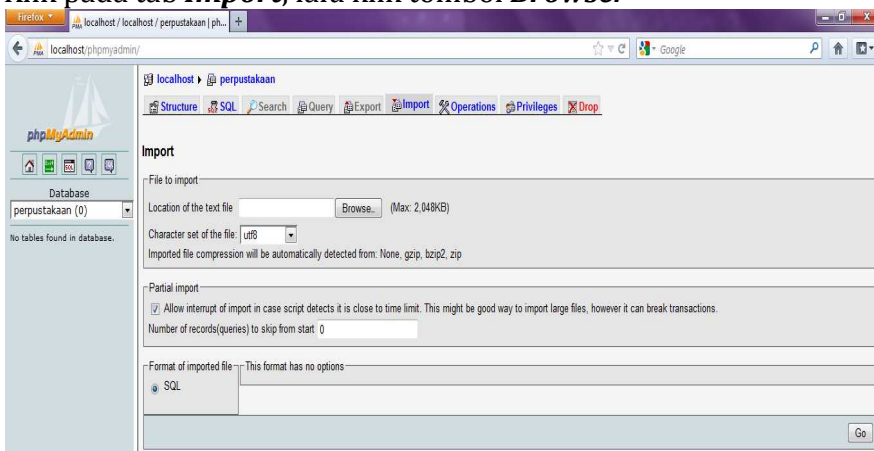
- Lalu klik tombol **Go**.

- f. Simpan dan download database tersebut. Lama proses download database tergantung besar atau tidaknya database yang tersimpan.



2. Import Database

- Buat database baru.
- Masuk kembali ke PhpMyAdmin.
- Pilih database yang hendak di **import** (dengan asumsi belum ada tabel didalamnya).
- Klik pada tab **Import**, lalu klik tombol **Browse**.



- Pilih database yang hendak di **import**, lalu klik **Open**.
- Lalu klik tombol **Go**.
- Tunggulah beberapa saat, proses **import** database sedang dilakukan.
- Jika **import** database sudah dilakukan, maka akan ditampilkan tabel-tabelnya.

Latihan PPBD UAS

1. Kerjakalah soal dibawah ini dengan menggunakan perintah SQL di *Command Prompt*.
 - a. Buatlah sebuah database dengan nama “**db_perpustakaan**”.
 - b. Buatlah sebuah tabel dengan nama “**tb_anggota**” dengan struktur dibawah ini:

Field Name	Type	Size	Keterangan
id_agt	Char	5	Primary Key
nama_agt	Varchar	25	
alamat	Varchar	50	
telp	Varchar	15	

- c. Isilah tabel “**tb_anggota**” sebanyak 5 *record*.
 - d. Gantilah nama *field* **telp** menjadi **telepon**.
 - e. Gantilah alamat “**Kalibata**” menjadi alamat “**Kaliabang**” untuk record dengan id_agt “**AGT02**”.
 - f. Gantilah nama “**tb_anggota**” menjadi “**tb_agt**”.
2. Kerjakalah soal dibawah secara wizard ini di *PhpMyadmin*.
 - a. Buatlah sebuah tabel dengan nama “**tb_buku**” didalam database “**db_perpustakaan**” dengan struktur sebagai berikut:

Field Name	Type	Size	Keterangan
kd_buku	Char	5	Primary Key
judul	Varchar	30	
pengarang	Varchar	20	
penerbit	Varchar	30	

- b. Isilah tabel “**tb_buku**” sebanyak 10 *record*.
 - c. Export database “**db_perpustakaan**” dan simpan di C:/LatihanUAS/NIM/db_perpustakaan.sql.