

# **BAB 3**

## **MEMBUAT DATABASE INVENTORY BARANG**

Pada bab ini, Anda akan dipandu untuk bisa membangun database secara lengkap, mulai dari perancangan, pembuatan tabel, sampai dengan mengelola isi datanya. Semua langkah dipandu memakai phpMyAdmin, program aplikasi berbasis Web yang dapat dipakai untuk mengakses database MySQL. PhpMyAdmin sudah ada dalam satu paket dengan Xampp Server yang sudah terinstal, sehingga Anda sekarang tinggal memakainya.

### **3.1 Merancang Database Inventory**

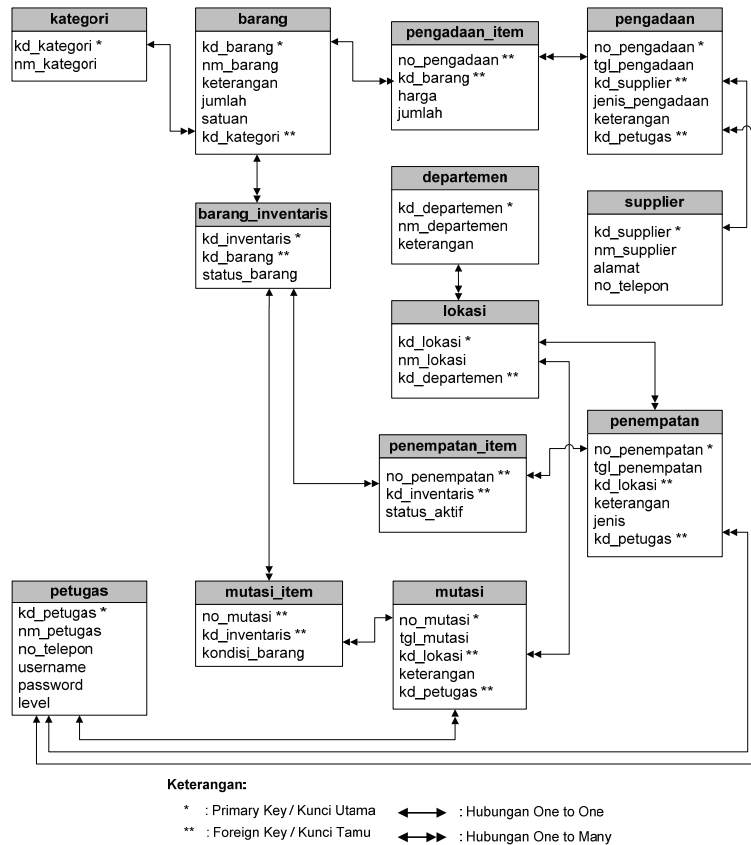
Sebelum masuk pada bagian membangun database dan tabel, ada baiknya Anda pahami dulu perancangan basis data (database) pada aplikasi Inventory Barang yang akan kita buat bersama. Pada aplikasi inventory ini, objek barang yang dikelola adalah barang di kantor, transaksi yang terjadi adalah pengadaan barang (barang masuk), transaksi penempatan, dan mutasi barang.

Pada transaksi barang masuk, maksudnya adalah proses diterimanya barang yang kita dapat dari supplier atau penyedia barang. Pada transaksi penempatan barang, artinya barang yang sudah kita inventarisasi akan kita tempatkan pada ruang atau lokasi yang ada di kantor, sedangkan mutasi adalah penarikan kembali barang yang sudah ada di ruang/lokasi ke gudang.

Dari gambaran aplikasi inventory yang akan kita buat di atas, maka setidaknya Anda membutuhkan beberapa tabel untuk menyimpan semua master data dan data transaksi. Berikut ini adalah tabel yang sudah berhasil kami analisis:

Tabel <b>kategori</b>	Tabel <b>pengadaan dan pengadaan_item</b>
Tabel <b>barang</b>	Tabel <b>penempatan dan penempatan_item</b>
Tabel <b>supplier</b>	Tabel <b>mutasi dan mutasi_item</b>
Tabel <b>departemen</b>	Tabel <b>petugas</b>
Tabel <b>lokasi</b>	

Dari daftar tabel di atas, Anda dapat memahami lebih lanjut lagi dengan melihat struktur relasi tabel, atau hubungan antar tabelnya yang sudah kami desain. Seperti gambar berikut:



**Gambar 3.1** Desain relasi antar tabel aplikasi Inventory Barang

## 3.2 Membuat Database Inventory

Semua langkah yang kami ajarkan dalam membangun database menggunakan tools phpMyAdmin, yaitu aplikasi berbasis web untuk manajemen database server MySQL.

### 3.2.1 Membuat Database

Struktur tabel bisa dibuat setelah nama database-nya dibuat. Pada proyek Inventory Barang yang kita bangun, database akan diberi nama **inventory\_barangdb**. Membuatnya dengan phpMyAdmin, berikut caranya:

1. Jalankan **XAMPP Control Panel**, klik **Start** pada servis **MySQL** dan **Apache**.
2. Buka browser Anda (misalnya; **Internet Explorer** atau **Firefox**), lalu buka aplikasi phpMyAdmin dengan alamat:  
`http://localhost/phpmyadmin/`
3. Klik menu **Server:Localhost / Database** yang ada di menu bagian atas.
4. Masukkan nama database yang akan dibuat (misalnya; **inventory\_barangdb**), lalu klik tombol **Create**.



*Gambar 3.2 Membuat database baru dari phpMyAdmin*

5. Klik nama database **inventory\_barangdb** untuk masuk dan mengaktifkan.

Setelah database **inventory\_barangdb** berhasil dibuat. Maka, Anda dapat lanjut untuk membuat struktur tabel di dalamnya.

### 3.2.2 Membuat Tabel Data Kategori

Tabel pertama yang akan kita buat adalah tabel **kategori**, yaitu tabel yang berisi daftar kategori barang. Di dalam tabel ini hanya ada 2 kolom, lebih jelasnya seperti desain berikut:

Field	Type(Length)	Keterangan
kd_kategori	Char(4)	Primary Key
nm_kategori	Varchar(100)	

*Tabel 3.1 Desain tabel kategori*

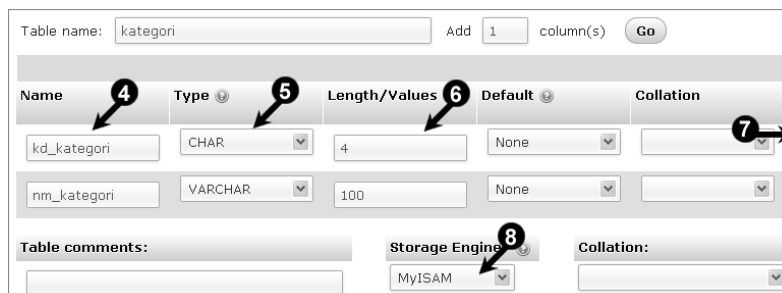
Dari desain struktur tabel **kategori** di atas, Anda dapat membuat tabelnya dari phpMyAdmin dengan mengikuti panduan berikut:

1. Buka database, dengan cara klik nama database **inventory\_barangdb**.
2. Dari kotak **Create table**, masukkan nama tabel **kategori** dan jumlah kolom 2.



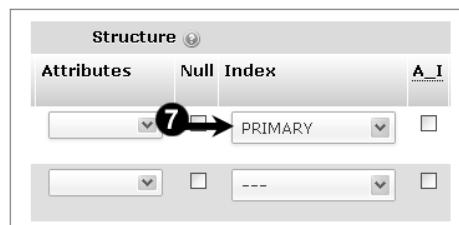
*Gambar 3.3 Dari phpMyAdmin, membuat tabel baru bernama kategori*

3. Klik tombol **Go**.
4. Desain struktur kolom dari tabel **kategori** seperti tabel di atas.



*Gambar 3.4 Mendesain struktur kolom tabel*

5. Pada nama kolom **kd\_kategori**, pilih Index: **PRIMARY**, cara ini untuk membuat **Primary Key**.



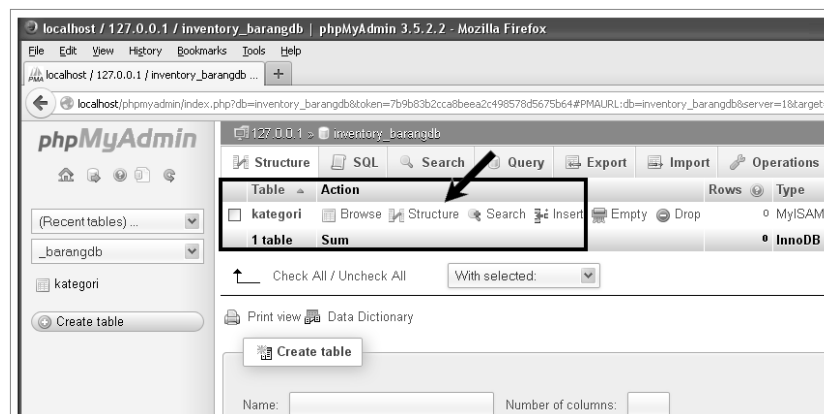
*Gambar 3.5 Membuat kunci primer (Primary Key)*

6. Pada bagian **Storage Engine**, pilih **MyISAM**.
7. Jika desain kolom selesai, klik tombol **Save**.



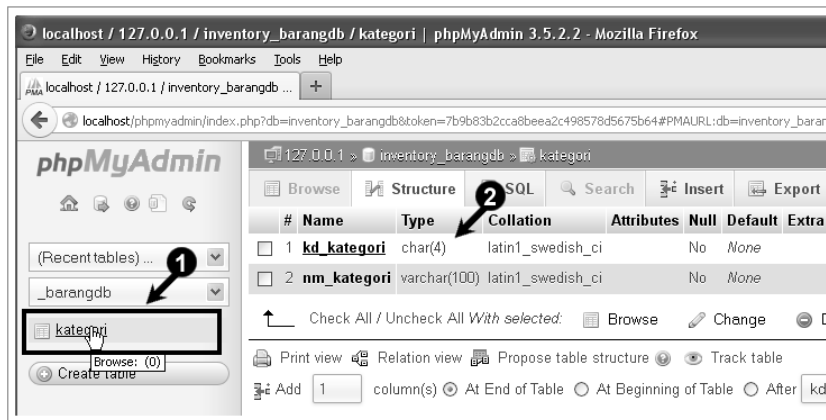
*Gambar 3.6 Tombol Save, untuk menyimpan desain struktur tabel*

Satu tabel berhasil dibuat. Di dalam database **inventory\_barangdb**, saat ini sudah ada 1 tabel bernama **kategori**. Berikut contohnya, daftar tabel terlihat pada database yang sedang dibuka.



*Gambar 3.7 Daftar tabel di dalam database yang aktif*

Klik pada nama tabelnya, jika Anda ingin melihat struktur desain tabel tersebut. Contoh:



Gambar 3.8 Struktur tabel kategori

### 3.2.3 Membuat Tabel Data Barang dan Inventaris

Pada bagian ini, kita akan buat 2 tabel untuk menyimpan data koleksi barang dan kode inventaris barang, tabelnya diberi nama **barang** dan **barang\_inventaris**. Kedua tabel ini saling berhubungan.

Sebagai contoh, kantor memiliki 3 unit laptop, maka nama dan spesifikasi laptop disimpan di dalam tabel **barang** (misal kodenya **B0001**), sedangkan identitas kode inventaris dari ke-3 barang tersebut disimpan di dalam tabel **barang\_inventaris**, misalnya diberi kode (**B0001.01** untuk unit laptop ke-1, **B0001.02** untuk unit ke-2, dan **B0001.03** untuk unit ke-3).

Untuk tabel barang, Anda dapat melihat desainnya seperti berikut:

Field	Type(Length)	Keterangan
kd_barang	Char(5)	Primary Key
nm_barang	Varchar(100)	
keterangan	Varchar(100)	
jumlah	Int(6)	
satuan	Varchar(20)	
kd_kategori	Char(4)	

Tabel 3.2 Desain tabel barang

Dengan melihat desain di atas, buat tabel **barang** dengan menggunakan phpMyAdmin. Caranya:

1. Klik nama database **inventory\_barangdb**.

2. Dari kotak **Create table**, masukkan nama tabel **barang** dan jumlah kolom 6.

*Gambar 3.9* Dari phpMyAdmin, membuat tabel baru bernama **barang**

3. Klik tombol **Go**.
4. Desain struktur kolom dari tabel **barang** seperti tabel di atas.

Name	Type	Length/Values	Default	Colla
kd_barang	CHAR	5	None	
nm_barang	VARCHAR	100	None	
keterangan	VARCHAR	100	None	
jumlah	INT	6	None	
satuan	VARCHAR	20	None	
kd_kategori	CHAR	4	None	

*Gambar 3.10* Mendesain struktur kolom tabel **barang**

5. Pada nama kolom **kd\_barang**, pilih **Index: PRIMARY** untuk membuat kunci primer (**Primary Key**).
6. Pada nama kolom **kd\_kategori**, pilih **Index: INDEX**, untuk membuat kunci tamu (**Foreign Key**).
7. Pada bagian **Storage Engine**, pilih **MyISAM**.
8. Jika desain kolom selesai, klik tombol **Save**.

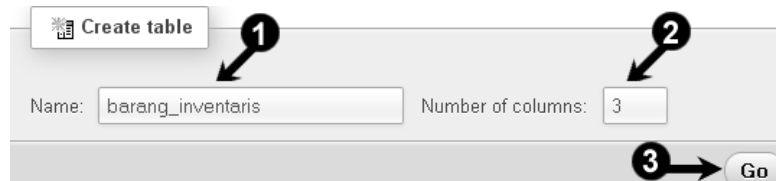
Setelah tabel barang selesai dibuat, sekarang Anda dapat membuat tabel **barang\_inventaris**. Di dalam tabel ini juga memiliki 1 kunci utama (*Primary Key*), yaitu **kd\_inventaris**, dan memiliki kunci tamu (*Foreign Key*) dari tabel **barang**.

Field	Type(Length)	Keterangan
kd_inventaris	Char(9)	Primary Key
kd_barang	Char(5)	
status_barang	ENUM	'Tersedia', 'Terpakai', 'Dipinjam'

*Tabel 3.3 Desain tabel barang\_inventaris*

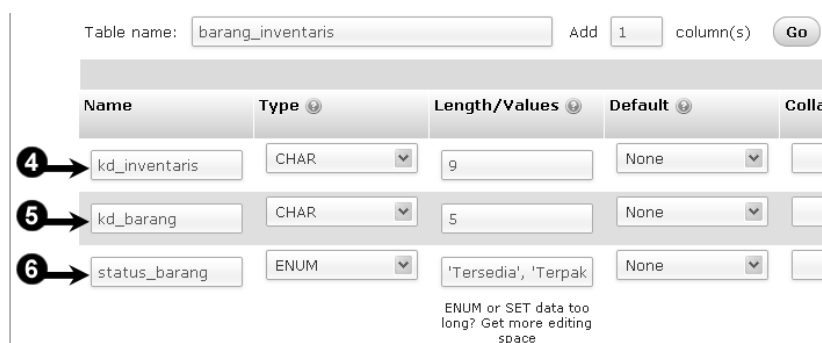
Ikuti langkah berikut untuk membuat tabel **barang\_inventaris**:

1. Klik nama database **inventory\_barangdb**.
2. Dari kotak **Create table**, masukkan nama tabel **barang\_inventaris** dan jumlah kolom 3.



*Gambar 3.11 Dari phpMyAdmin, membuat tabel barang\_inventaris*

3. Klik tombol **Go**.
4. Desain struktur kolom dari tabel **barang\_inventaris** seperti tabel di atas.



*Gambar 3.12 Mendesain struktur kolom tabel barang\_inventaris*



5. Pada nama kolom **kd\_inventaris**, pilih Index: **PRIMARY**.
6. Pada nama kolom **kd\_barang**, pilih Index: **INDEX**.
7. Pada nama kolom **status\_barang** tipe datanya adalah **ENUM**, isi pilihannya dibuat pada kotak **Length/Values**: 'Tersedia', 'Terpakai', 'Dipinjam'.
8. Pada bagian **Storage Engine**, pilih **MyISAM**.
9. Jika desain kolom selesai, klik tombol **Save**.

### 3.2.4 Membuat Tabel Data Supplier

Supplier adalah bagian yang menjadi tujuan kita memperoleh barang, atau pihak yang menjadi asal barang saat ada transaksi pengadaan barang. Saat pengadaan, barang bisa saja kita beli, atau juga diberikan oleh pihak yang menjadi donatur, pihak ini kita simpan dalam tabel **supplier**. Desain tabel dapat dilihat di bawah:

Field	Type(Length)	Keterangan
kd_supplier	Char(4)	Primary Key
nm_supplier	Varchar(100)	
alamat	Varchar(200)	
no_telepon	Varchar(20)	

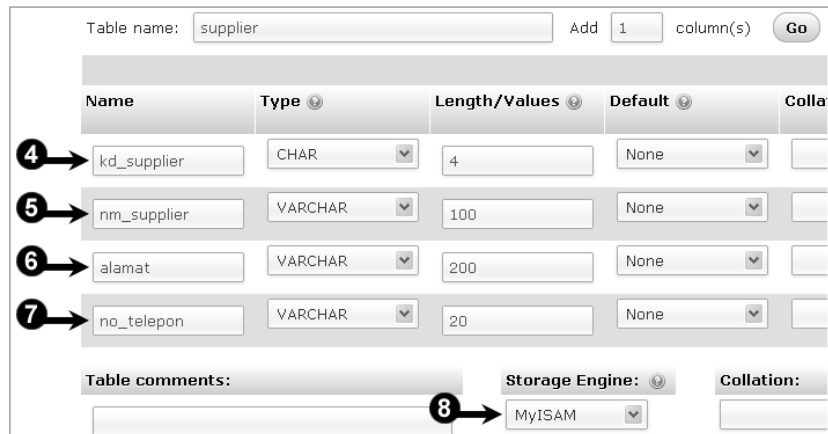
*Tabel 3.4 Desain tabel supplier*

Ikuti langkah berikut untuk membuat tabel **supplier**. Caranya:

1. Klik nama database **inventory\_barangdb**.
2. Dari kotak **Create table**, masukkan nama tabel **supplier** dan jumlah kolom 4.

*Gambar 3.13 Dari phpMyAdmin, membuat tabel supplier*

3. Klik tombol **Go**.
4. Desain struktur kolom dari tabel **supplier** seperti tabel di atas.



*Gambar 3.14 Mendesain struktur kolom tabel supplier*

5. Pada nama kolom **kd\_supplier**, pilih **Index: PRIMARY** untuk membuat kunci primer (**Primary Key**).
6. Pada bagian **Storage Engine**, pilih **MyISAM**.
7. Klik tombol **Save**.

### 3.2.5 Membuat Tabel Data Departemen

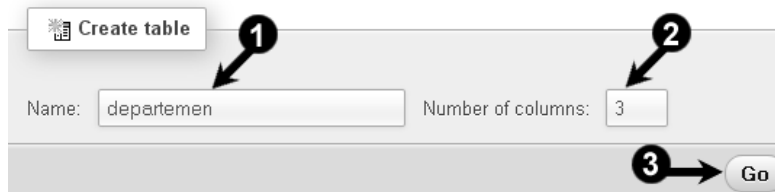
Di dalam perusahaan atau kantor memiliki beberapa departemen, atau bisa disebut dengan bagian, di dalamnya mungkin masih terdapat sub departemen. Untuk informasi data ini, kita perlu satu tabel bernama **departemen**. Desainnya adalah:

Field	Type(Length)	Keterangan
kd_departemen	Char(4)	Primary Key
nm_departemen	Varchar(100)	
keterangan	Varchar(100)	

*Tabel 3.5 Desain tabel departemen*

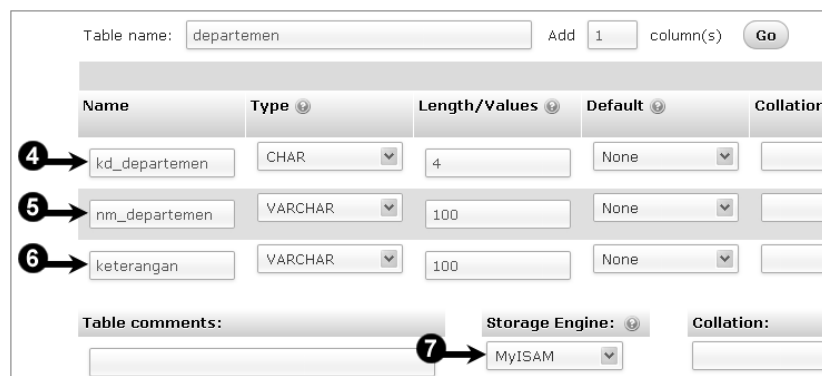
Ikuti langkah berikut untuk membuat tabel **departemen**. Caranya:

1. Klik nama database **inventory\_barangdb**.
2. Dari kotak **Create table**, masukkan nama tabel **departemen** dan jumlah kolom 3.



*Gambar 3.15 Dari phpMyAdmin, membuat tabel departemen*

3. Klik tombol **Go**.
4. Desain struktur kolom tabel **departemen** seperti tabel di atas.



*Gambar 3.16 Mendesain struktur kolom tabel departemen*

5. Pada nama kolom **kd\_departemen**, pilih **Index: PRIMARY** untuk membuat kunci primer (**Primary Key**).
6. Pada bagian **Storage Engine**, pilih **MyISAM**.
7. Klik tombol **Save**.

### 3.2.6 Membuat Tabel Data Lokasi

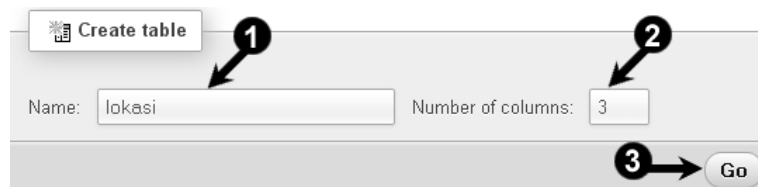
Pengelompokkan tempat terkecil di dalam satu kantor dapat kita sebut dengan lokasi, di dalamnya bisa berarti ruang atau sub-bagian yang ada di bawah departemen. Fungsi dari data lokasi adalah untuk informasi lokasi tempat pada saat transaksi penempatan barang, misalnya saja lokasi ruang dosen, ruang wakil, atau ruang direktur, ruang sekretaris dan lainnya. Berikut desain tabelnya:

Field	Type(Length)	Keterangan
kd_lokasi	Char(5)	Primary Key
nm_lokasi	Varchar(100)	
kd_departemen	Char(4)	

Tabel 3.6 Desain tabel lokasi

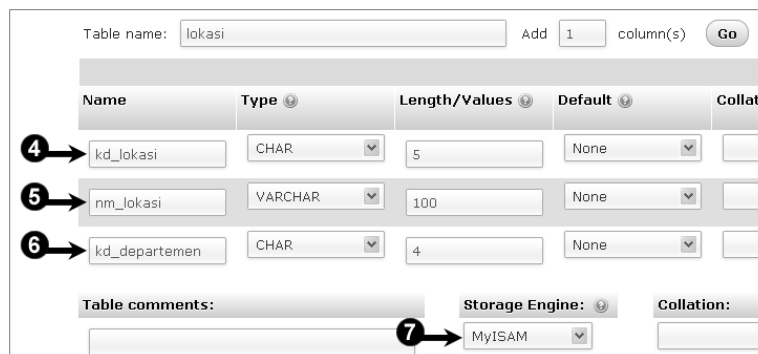
Ikuti langkah berikut untuk membuat tabel lokasi. Caranya:

1. Klik nama database **inventory\_barangdb**.
2. Dari kotak **Create table**, masukkan nama tabel **lokasi** dan jumlah kolom 3.



Gambar 3.17 Dari phpMyAdmin, membuat tabel lokasi

3. Klik tombol **Go**.
4. Desain struktur kolom dari tabel lokasi seperti tabel di atas.



Gambar 3.18 Mendesain struktur kolom tabel lokasi

5. Pada nama kolom **kd\_lokasi**, pilih Index: **PRIMARY**.
6. Pada nama kolom **kd\_departemen**, pilih Index: **INDEX**.
7. Pada bagian **Storage Engine**, pilih **MyISAM**.
8. Klik tombol **Save**.

### 3.2.7 Membuat Tabel Transaksi Pengadaan

Salah satu metode pengadaan barang adalah dengan cara pembelian barang baru dari supplier. Cara lain, pengadaan juga dapat dilakukan dengan sistem hibah atau sumbangan dari pihak lain, misalnya saja dari pemerintah. Dari manapun asalnya, baik pembelian atau dari hasil hibah dan sumbangan, maka saat barang masuk akan kita catat dalam pencatatan transaksi pengadaan.

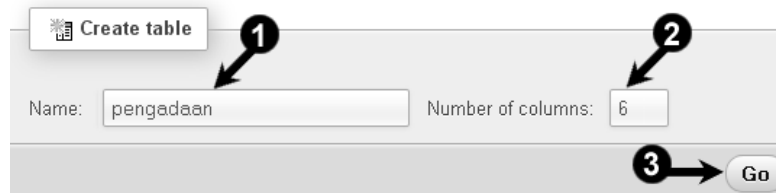
Pada transaksi pengadaan, kita memerlukan 2 tabel penyimpanan database, yaitu tabel **pengadaan** sebagai utama, dan tabel **pengadaan\_item** untuk menyimpan daftar item barang yang diterima. Di bawah ini adalah desainnya.

Field	Type(Length)	Keterangan
no_pengadaan	Char(7)	Primary Key
tgl_pengadaan	Date	
kd_supplier	Char(4)	
jenis_pengadaan	Varchar(40)	
kd_petugas	Char(4)	

*Tabel 3.7 Desain tabel pengadaan*

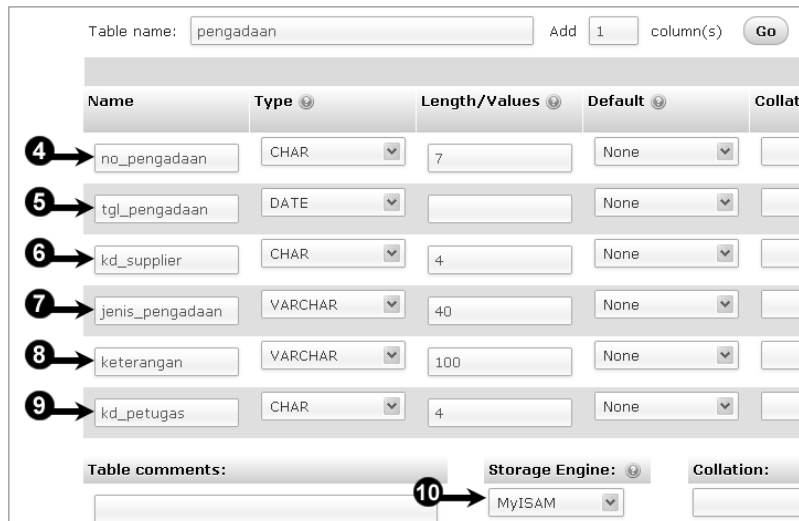
Pada desain **pengadaan** tabel di atas, kita memasukkan **kd\_supplier** sebagai kolom tamu (**Foreign Key**), sehingga dalam satu transaksi barangnya hanya berasal dari satu Supplier penyedia barang. Ikuti langkah berikut untuk membuat tersebut:

1. Klik nama database **inventory\_barangdb**.
2. Dari kotak **Create table**, masukkan nama tabel **pengadaan** dan jumlah kolom 6.



*Gambar 3.19 Dari phpMyAdmin, membuat tabel pengadaan*

3. Klik tombol **Go**.
4. Desain struktur kolom dari tabel **pengadaan** seperti tabel di atas.



Gambar 3.20 Mendesain struktur kolom tabel pengadaan

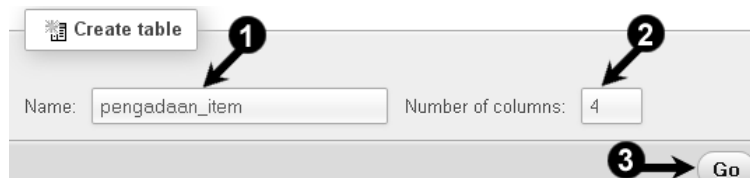
5. Pada nama kolom **no\_pengadaan**, pilih Index: **PRIMARY** untuk membuat kunci primer (**Primary Key**).
6. Pada nama kolom **kd\_supplier**, pilih Index: **INDEX**.
7. Pada nama kolom **kd\_petugas**, pilih Index: **INDEX**.
8. Pada bagian **Storage Engine**, pilih **MyISAM**.
9. Klik tombol **Save**.

Field	Type(Length)	Keterangan
no_pengadaan	Char(7)	Primary Key
kd_barang	Char(5)	
harga	Int(12)	
Jumlah	Int(4)	

Tabel 3.8 Desain tabel pengadaan\_item

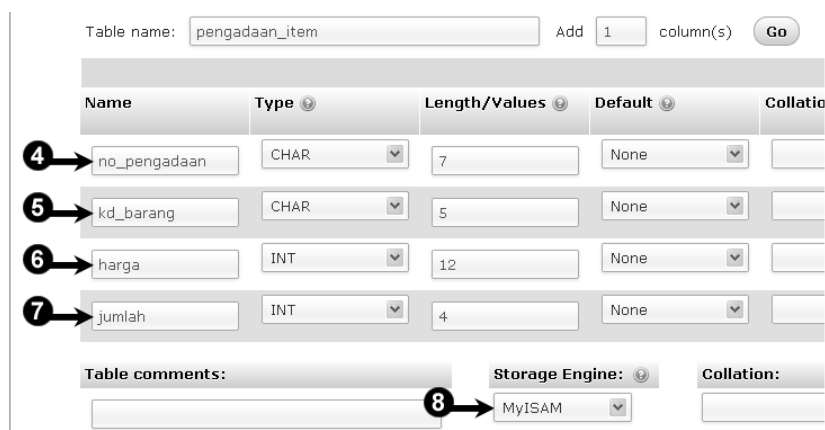
Selanjutnya tabel **pengadaan\_item**, digunakan untuk menyimpan detail daftar barang yang diterima, termasuk harga dan jumlah barang yang diterima. Ikuti langkah berikut untuk membuatnya:

1. Klik nama database **inventory\_barangdb**.
2. Dari kotak **Create table**, masukkan nama tabel **pengadaan\_item** dan jumlah kolom 4.



Gambar 3.21 Dari phpMyAdmin, membuat tabel `pengadaan_item`

3. Klik tombol **Go**.
4. Desain struktur kolom dari tabel `pengadaan_item` seperti tabel di atas.



Gambar 3.22 Mendesain struktur kolom tabel `pengadaan_item`

5. Pada nama kolom `no_pengadaan`, pilih **Index: INDEX**.
6. Pada nama kolom `kd_barang`, pilih **Index: INDEX**.
7. Pada bagian **Storage Engine**, pilih **MyISAM**.
8. Klik tombol **Save**.

### 3.2.8 Membuat Tabel Transaksi Penempatan

Hasil dari transaksi pengadaan, barang akan masuk ke gudang. Dari gudang, barang akan ditempatkan ke ruang-ruang atau lokasi yang ada di kantor bersangkutan. Misalnya saja, 2 unit komputer dan 1 laptop akan ditempatkan pada Ruang Kepala Sekolah, atau Ruang Direktur.

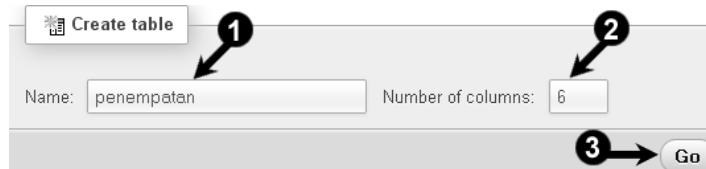
Sebagai pencatatan, kita akan menggunakan 2 tabel, yaitu tabel **penempatan** sebagai tabel utama transaksi, yang mencatat tanggal dan nama lokasi penempatan. Tabel kedua adalah **penempatan\_item**, sebagai penyimpan daftar barang. Di bawah ini adalah desain tabelnya:

Field	Type(Length)	Keterangan
no_penempatan	Char(7)	Primary Key
tgl_penempatan	Date	
kd_lokasi	Char(5)	
keterangan	Varchar(100)	
jenis	ENUM	'Baru','Mutasi'
kd_petugas	Char(4)	

*Tabel 3.9 Desain tabel penempatan*

Dalam satu kali transaksi penempatan barang, program hanya mencatat 1 tempat atau satu lokasi saja, sedangkan barang yang bisa ditempatkan bisa lebih dari 1 unit barang. Ikuti langkah berikut untuk membuat tabel **penempatan**. Caranya:

1. Klik nama database **inventory\_barangdb**.
2. Dari kotak **Create table**, masukkan nama tabel **penempatan** dan jumlah kolom 6.



*Gambar 3.23 Dari phpMyAdmin, membuat tabel penempatan*

3. Klik tombol **Go**.
4. Desain struktur kolom tabel **penempatan** seperti tabel di atas.
5. Pada nama kolom **no\_penempatan**, pilih **Index: PRIMARY** untuk membuat kunci primer (**Primary Key**).
6. Pada nama kolom **kd\_lokasi**, pilih **Index: INDEX**.
7. Pada nama kolom **kd\_petugas**, pilih **Index: INDEX**.
8. Pada bagian **Storage Engine**, pilih **MyISAM**.
9. Klik tombol **Save**.



Table name:  Add  column(s)

Name	Type	Length/Values	Default	Collation
4 → no_penempatan	CHAR	7	None	
5 → tgl_penempatan	DATE		None	
6 → kd_lokasi	CHAR	5	None	
7 → keterangan	VARCHAR	100	None	
8 → jenis	ENUM	'Baru','Mutasi'	None	
ENUM or SET data too long? Get more editing space				
9 → kd_petugas	CHAR	4	None	

*Gambar 3.24 Mendesain struktur kolom tabel penempatan*

Field	Type(Length)	Keterangan
no_penempatan	Char(7)	Primary Key
kd_inventaris	Char(9)	
status_aktif	Enum	'Aktif','Tidak'

*Tabel 3.10 Desain tabel penempatan\_item*

Tabel **penempatan\_item** dipakai untuk mencatat daftar unit barang dalam satu transaksi penempatan barang. Kode barang yang dicatat adalah kode inventaris, sehingga kita bisa tahu persis peta koleksi barang saat ini ada di lokasi/ruang mana. Berikut caranya:

1. Klik nama database **inventory\_barangdb**.
2. Dari kotak **Create table**, masukkan nama tabel **penempatan\_item** dan jumlah kolom 3.

Name:  Number of columns:

*Gambar 3.25 Dari phpMyAdmin, membuat tabel penempatan\_item*

3. Klik tombol **Go**.

- Desain struktur kolom dari tabel **penempatan\_item** seperti tabel di atas.

Name	Type	Length/Values	Default	Collation
no_penempatan	CHAR	7	None	
kd_inventaris	CHAR	9	None	
status_aktif	ENUM	'Aktif','Tidak'	None	

*Gambar 3.26 Mendesain struktur kolom tabel penempatan\_item*

- Pada nama kolom **no\_penempatan**, pilih Index: **INDEX**.
- Pada nama kolom **kd\_inventaris**, pilih Index: **INDEX**.
- Pada bagian **Storage Engine**, pilih **MyISAM**.
- Klik tombol **Save**.

### 3.2.9 Membuat Tabel Transaksi Mutasi

Mutasi artinya perpindahan barang, dari lokasi A ke lokasi B. Pada aplikasi yang kita bangun ini, transaksi mutasi adalah proses penarikan barang dari lokasi A ke gudang, yang kemudian barang tersebut dapat kita pindahkan ke tempat (lokasi B) lain dengan cara membuat transaksi penempatan baru.

Untuk membuat transaksi mutasi, kita bisa menggunakan 2 tabel, yaitu tabel **mutasi** dan tabel **mutasi\_item**. Desain tabelnya adalah:

Field	Type(Length)	Keterangan
no_mutasi	Char(7)	Primary Key
tgl_mutasi	Date	
kd_lokasi	Char(5)	
keterangan	Varchar(100)	
kd_petugas	Char(4)	

*Tabel 3.11 Desain tabel mutasi*

Tabel **mutasi** dipakai untuk mencatat informasi transaksi utama, yaitu tanggal mutasi dan lokasi asal (lokasi lama) dari barang yang ditarik/diambil. Untuk membuatnya, ikuti langkah berikut:

1. Klik nama database **inventory\_barangdb**.
2. Dari kotak **Create table**, masukkan nama tabel **mutasi** dan jumlah kolom 5.

*Gambar 3.27 Dari phpMyAdmin, membuat tabel mutasi*

3. Klik tombol **Go**.
4. Desain struktur kolom dari tabel **mutasi** seperti tabel di atas.

Name	Type	Length/Values	Default	Collation
no_mutasi	CHAR	7	None	
tgj_mutasi	DATE		None	
kd_lokasi	CHAR	5	None	
keterangan	VARCHAR	100	None	
kd_petugas	CHAR	4	None	

Table comments:

Storage Engine: MyISAM

Collation:

*Gambar 3.28 Mendesain struktur kolom tabel mutasi*

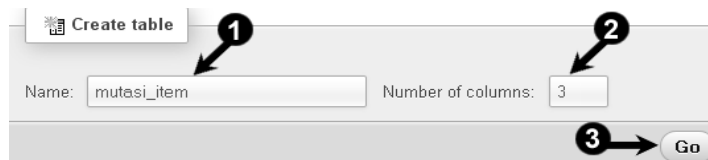
5. Pada nama kolom **no\_mutasi**, pilih **Index: PRIMARY** untuk membuat kunci primer (**Primary Key**).
6. Pada nama kolom **kd\_lokasi**, pilih **Index: INDEX**.
7. Pada nama kolom **kd\_petugas**, pilih **Index: INDEX**.
8. Pada bagian **Storage Engine**, pilih **MyISAM**.
9. Klik tombol **Save**.

Field	Type(Length)	Keterangan
no_mutasi	Char(7)	Primary Key
kd_inventaris	Char(9)	
kondisi_barang	Varchar(100)	

*Tabel 3.12 Desain tabel mutasi\_item*

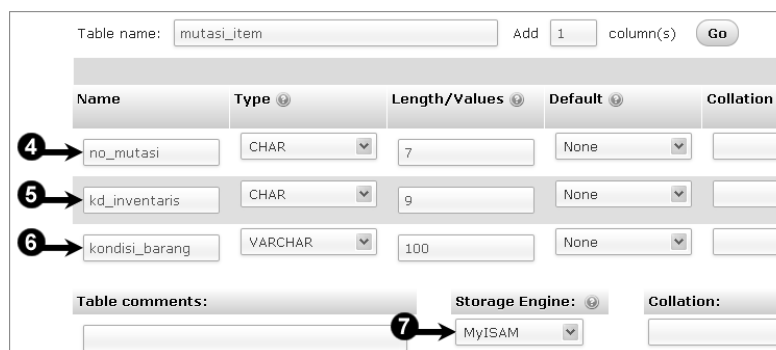
Daftar item barang yang dimutasi disimpan dalam tabel **mutasi\_item**. Setiap item barang akan diberi komentar tentang kondisi barang saat penarikan. Cara membuatnya adalah:

1. Klik nama database **inventory\_barangdb**.
2. Dari kotak **Create table**, masukkan nama tabel **mutasi\_item** dan jumlah kolom 3.



*Gambar 3.29 Dari phpMyAdmin, membuat tabel mutasi\_item*

3. Klik tombol **Go**.
4. Desain struktur kolom tabel **mutasi\_item** seperti tabel di atas.



*Gambar 3.30 Mendesain struktur kolom tabel mutasi\_item*

5. Pada nama kolom **no\_mutasi**, pilih Index: **INDEX**.
6. Pada nama kolom **kd\_inventaris**, pilih Index: **INDEX**.
7. Pada bagian **Storage Engine**, pilih **MyISAM**.
8. Klik tombol **Save**.

### 3.2.10 Membuat Tabel Data Petugas

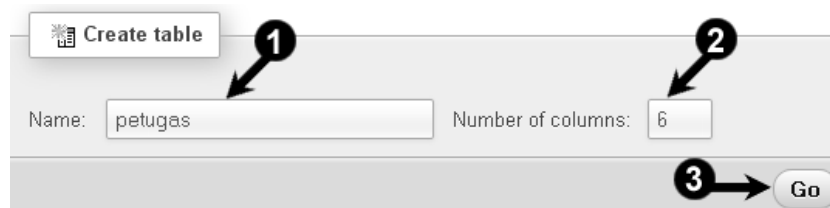
Petugas adalah orang yang menjalankan program aplikasi Inventory, bisa disebut User. Setiap ada transaksi (pengadaan, penempatan dan mutasi), maka siapa Petugas yang melakukan transaksi harus dicatat ke dalam database, sehingga suatu saat jika ada kejanggalan, kita bisa menelusuri siapa petugas yang memasukkan datanya. Berikut adalah desain tabel untuk menyimpan data petugas.

Field	Type(Length)	Keterangan
kd_petugas	Char(4)	Primary Key
nm_petugas	Varchar(100)	
no_telepon	Varchar(20)	
username	Varchar(20)	
password	Varchar(100)	
level	Varchar(20)	

Tabel 3.13 Desain tabel petugas

Ikuti langkah berikut untuk membuat tabel **petugas**. Caranya:

1. Klik nama database **inventory\_barangdb**.
2. Dari kotak **Create table**, masukkan nama tabel **lokasi** dan jumlah kolom 6.



Gambar 3.31 Dari phpMyAdmin, membuat tabel petugas

3. Klik tombol **Go**.
4. Desain struktur kolom dari tabel **petugas** seperti tabel di atas.
5. Pada nama kolom **kd\_petugas**, pilih **Index: PRIMARY** untuk membuat kunci primer (**Primary Key**).
6. Pada bagian **Storage Engine**, pilih **MyISAM**.
7. Klik tombol **Save**.

Table name:  Add  column(s)

Name	Type	Length/Values	Default	Collat
<b>4</b> → kd_petugas	CHAR	4	None	
<b>5</b> → nm_petugas	VARCHAR	100	None	
<b>6</b> → no_telepon	VARCHAR	20	None	
<b>7</b> → username	VARCHAR	20	None	
<b>8</b> → password	VARCHAR	100	None	
<b>9</b> → level	VARCHAR	20	None	

Table comments:

Storage Engine:  Collation:

**10** →

Gambar 3.32 Mendesain struktur kolom tabel petugas

### 3.3 Manajemen Data

Manajemen data secara penuh dapat kita lakukan dengan mudah dari phpMyAdmin, yaitu: memasukkan data, menampilkan data, memperbaiki data, dan menghapus data. Sebagai latihan, Anda dapat mencobanya pada tabel **kategori**, **barang**, dan tabel **barang\_inventaris**.

#### 3.3.1 Memasukkan Data

Memasukan data dalam perintah SQL adalah **INSERT**. Dalam program nanti saat Anda akan menyimpan data dari form ke database, maka perintah utama yang dipakai adalah SQL, yaitu menggunakan **INSERT**.

Sebagai latihan, pertama Anda dapat memasukkan data kategori barang ke dalam tabel **kategori**. Kode dan nama kategori di-input dari phpMyAdmin. Caranya:

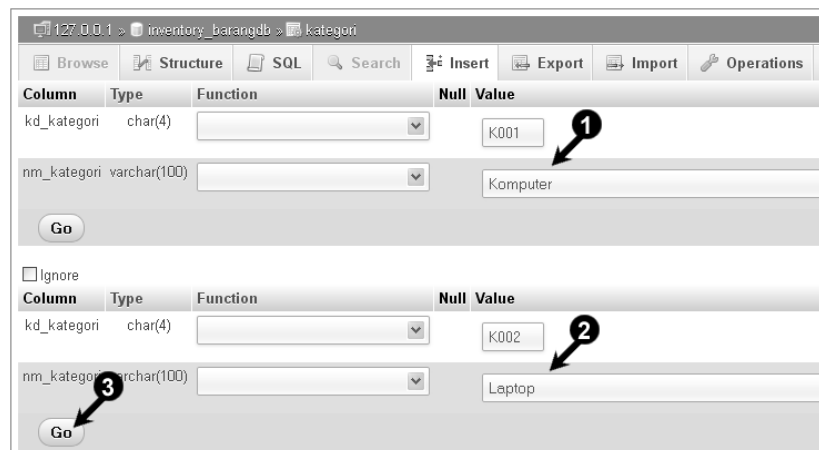
1. Jalankan lagi aplikasi phpMyAdmin.
2. Buka database **inventory\_barangdb**, klik dari daftar database yang ada di sebelah kiri.

3. Buka tabel **kategori**, lalu klik dari daftar tabel yang ada di sebelah kiri.



*Gambar 3.33 Membuka tabel kategori*

4. Klik menu **Insert**.
5. Dari form data, Anda dapat memasukkan data kategori dari sana. Masukkan kode dan nama kategori, kodenya tidak boleh sama, karena kode merupakan kunci utama (Primer).



*Gambar 3.34 Memasukan 2 data kategori dengan kode berbeda*

6. Setelah semua data terisi, klik tombol **Go**.

Setelah klik tombol **Go**, maka data yang Anda masukkan tadi akan tersimpan ke dalam database MySQL, yaitu pada tabel **kategori** di dalam nama database **inventory\_barangdb**. Jika Anda ingin memasukkan data lagi, klik menu **Insert**.

Contoh selanjutnya, kita akan membuat data barang, data barang memiliki hubungan data dengan data kategori yang sudah dibuat sebelumnya. Misalnya saja, kita membuat data unit **TOSHIBA Satellite C800D-1003**, maka kategorinya bisa dimasukkan dalam kategori **Laptop**, data kategori yang dimasukkan ke dalam tabel barang adalah kodenya. Contoh:

1. Buka tabel **barang**, lalu klik dari daftar tabel yang ada di sebelah kiri.
2. Klik menu **Insert**.
3. Dari form data, Anda dapat memasukan data seperti contoh berikut.

Column	Type	Function	Null	Value
kd_barang	char(5)		B0001	
nm_barang	varchar(100)		TOSHIBA Satellite C800D-1003 - Black	
keterangan	varchar(100)		Notebook / Laptop 13 inch - 14 inch AMD Dual Core E1-1200, 2Gf	
jumlah	int(6)		3	
satuan	varchar(20)		Unit	
kd_kategori	char(4)		K002	

*Gambar 3.35 Memasukan data barang dengan kategori Laptop (K002)*

4. Setelah semua data terisi, klik tombol **Go**.

Perhatikan contoh di atas. Jadi, karena tabel **barang** dengan tabel **kategori** adalah relasi, maka datanya juga harus berelasi. Jika Anda memasukkan data barang dengan kategori Laptop, maka data yang diisi dalam kolom **kd\_kategori** adalah kode kategori yang dimiliki dalam tabel **kategori**. Misalnya kategori Laptop, maka dalam tabel kategori yang diambil adalah kode **K002**.



### 3.3.2 Menampilkan Data

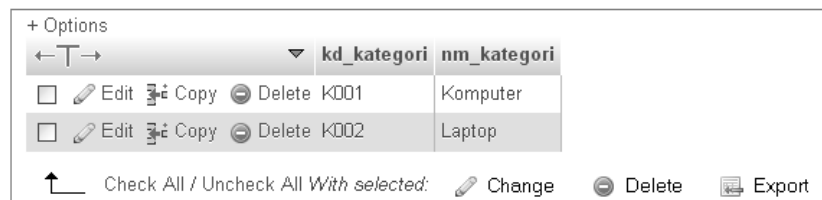
Perintah yang dipakai dalam database MySQL untuk menampilkan data adalah **SELECT**, kemudian diikuti parameter data dari kolom dan tabel apa yang akan ditampilkan. Pada phpMyAdmin, Anda dapat melihat isi data dari tabel dengan mengklik menu **Browse**.

1. Buka database **inventory\_barangdb**.
2. Dari daftar tabel yang terlihat, klik menu Browse pada nama tabel yang akan diakses.



*Gambar 3.36* Daftar tabel yang ada di database, klik Browse

3. Atau, buka tabel yang akan diakses dari daftar nama tabel yang ada di halaman kiri, misalnya tabel **kategori**.



*Gambar 3.37* Isi dari tabel kategori, berisi 2 baris data

Berikut ini adalah contoh lain dari isi tabel barang, di sana sudah kami berikan contoh beberapa baris data. Perhatikan, kode barang (**kd\_barang**) tidak ada yang kembar, karena kode barang harus unik, bersifat **Primary Key**.

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

Sort by key: None

+ Options

	kd_barang	nm_barang	keterangan	jumlah	satuan	kd_kategori
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	E0001	TOSHIBA Satellite C800D-1003 - Black	Notebook / Laptop 13 inch - 14 inch AMD Dual Core ...	3	Unit	K002
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	E0002	TOSHIBA Satellite C40-A106 - Black	Notebook / Laptop 13 inch - 14 inch Intel Core i3-...	5	Unit	K002
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	E0003	Printer Canon LBP 5100 Laser Jet	Canon LBP 5100 Laser Jet	3	Unit	K003
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	E0004	Printer Canon IP 2770	Canon IP 2770	0	Unit	K003
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	E0005	Printer Brother Colour Laser HL-2150N Mono	Brother Colour Laser HL-2150N Mono Laser Printer, ...	0	Unit	K003
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	E0006	UPS Prolink Pro 700	Prolink Pro 700	0	Unit	K004

Gambar 3.38 Isi dari tabel barang

### 3.3.3 Memperbaiki Data

MySQL memiliki dasar perintah SQL untuk memperbaiki data lama dengan perintah **UPDATE**. Pada aplikasi phpMyAdmin, menu yang dapat Anda gunakan adalah **Edit**, tersedia pada semua baris data yang sedang ditampilkan.

1. Buka database `inventory_barangdb`.
2. Buka tabel (misalnya tabel `kategori`), lalu tampilkan isinya (klik **Browse**).
3. Klik menu **Edit** dari baris data yang akan diperbaiki.



Gambar 3.39 Halaman tampil data, menu Edit untuk memperbaiki data

4. Perbaiki data lama yang tampil pada form. Untuk kode tidak boleh diubah.

127.0.0.1 -> inventory\_barangdb -> kategori

Browse Structure SQL Search Insert Export Import Operations

Column	Type	Function	Null	Value
kd_kategori	char(4)			K002
nm_kategori	char(100)			Laptop / Notebook

Go

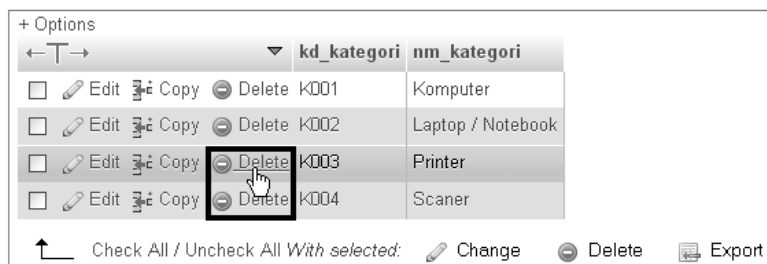
Gambar 3.40 Halaman untuk memperbaiki data lama

5. Setelah selesai, klik tombol **Go**.
6. Klik lagi menu **Browse** untuk melihat hasilnya.

### 3.3.4 Menghapus Data

MySQL memiliki perintah dasar **DELETE** untuk menghapus data. Pada phpMyAdmin, Anda juga dapat memakai menu **Delete** untuk menghapus data yang dipilih.

1. Buka database **inventory\_barangdb**.
2. Buka tabel (misalnya tabel **kategori**), lalu tampilkan isinya (klik **Browse**).
3. Klik menu **Delete** dari baris data yang akan dihapus.



*Gambar 3.41* Halaman tampil data, menu **Delete** untuk menghapus data

4. Pada jendela konfirmasi, klik tombol **Ok**.

###

