

Special edition



**MEMBUAT WEB
TOKO ONLINE**
Metode Virtual Bank
dengan PHP - MySQL

Cocok buat TA



Bersama :
www.nanangkristanto.com
www.bunafit-komputer.com

Panduan Web Toko Online

Sistem Informasi Penjualan HaPe Secara Online – Virtual Bank

(Programmer : Bunafit Nugroho & Team)

Daftar Harga (SMS ke : **0838 6716 1314**)

- eBook versi Afiliasi : Rp. 50.000,-
<http://tokoonlinevirtualbank.nkmultimedia.com>
www.bisnisafiliasi.com
- Script Web Contoh v1 : Rp. 150.000,-
(versi phpTriad, dengan PHP 4 dan MySQL, register Global OFF)
- Script Web Contoh v2 : Rp. 500.000,-
(versi XAMPP, dengan PHP 5 dan MySQL, register Global ON)

planetKode.Com

Bunafit Komputer
www.bunafit-komputer.com

Adalah web dengan konsep sebagai bahan referensi/
contekan/ acuan Tugas Akhir (TA), Skripsi, Kerja
Praktek atau Tesis bidang Informatika Komputer.
Konsep ini gratis, tapi skrip web toko contohnya
dijual dengan harga **Rp. 150.000,- (promo untuk 10
orang pertama)**

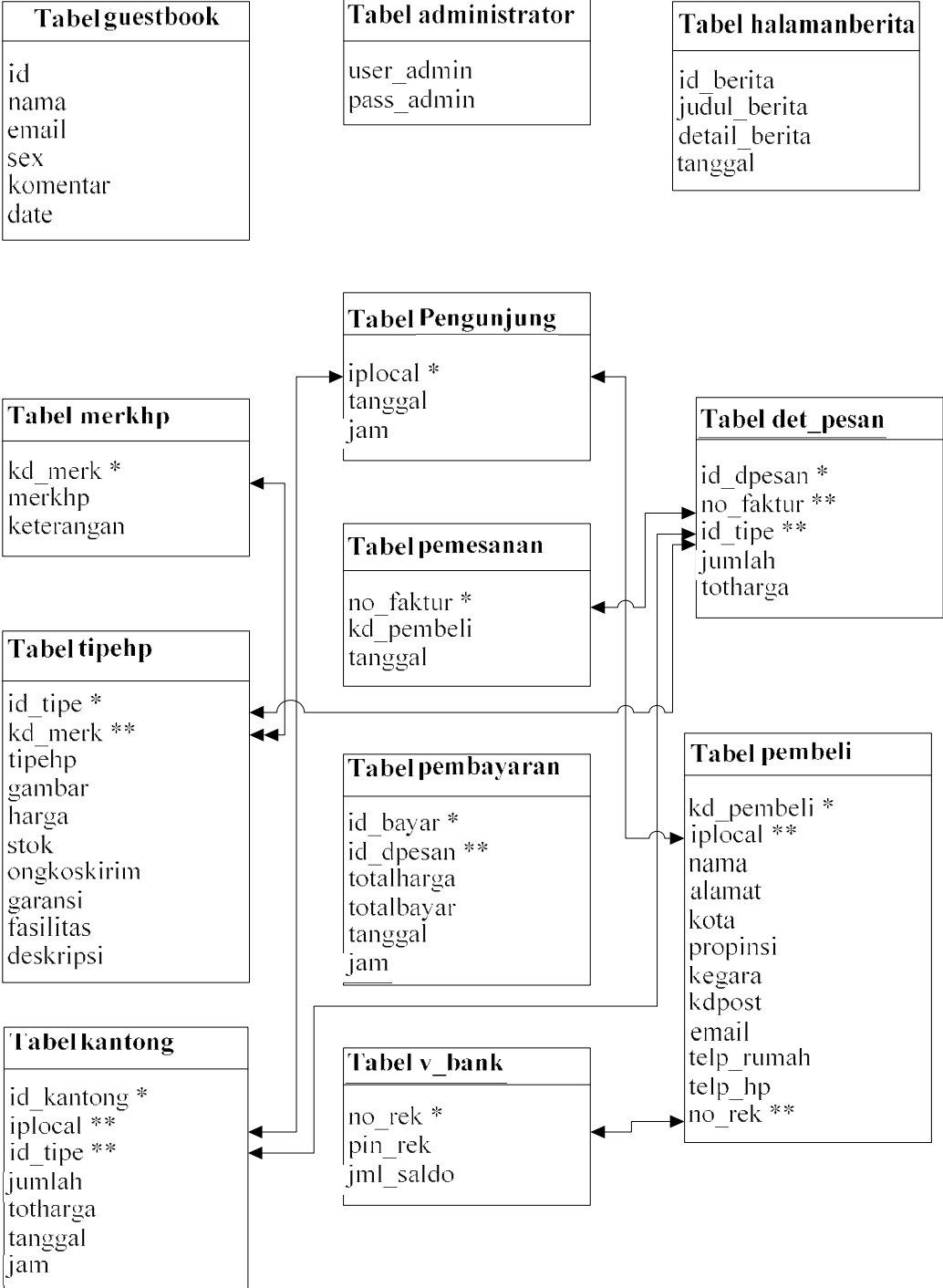
**Tersedia Source Code Perpustakaan dari program
Java, Visual Basic 6.0, Delphi dan C++ Builder, juga
tersedia dengan konsep Multi User dan
Client/Server**

Untuk mendapatkan No Rekening transfer, silahkan kirim email seperti
contoh di atas. Atau, kirim SMS ke nomor **0838 6716 1314.**

Contoh pesan SMS v1 : "*planetKode, minta No Rekening buat pembayaran
Sc PHP Web Toko HaPe v1 phpTriad*". Kami akan segera membalas.

Contoh pesan SMS v2 : "*planetKode, minta No Rekening buat pembayaran
Sc PHP Web Toko HaPe v2 XAMPP*". Kami akan segera membalas.

ERD / Relasi Tabel



Analisis Sistem Basisdata Toko Online untuk Counter Handphone

4.1 Pendahuluan

Pada bab ini Anda akan Kami tuntun untuk dapat membuat standar database yang digunakan untuk membuat Toko Online. Database ini tidak hanya dapat digunakan untuk aplikasi yang sedang Kita bangun saja, akan tetapi juga dapat digunakan untuk membuat aplikasi Toko Online lain yang memiliki prosedur dan sistem yang hampir sama.

Dengan adanya bab ini, maka Anda semua akan Kami tuntun untuk mampu melakukan perancangan pada sebuah aplikasi Toko Online. Sehingga, setelah menyelesaikan pembahasan bab ini, Anda akan mampu untuk menganalisis kebutuhan database dan mampu untuk mengembangkan kekurangan-kekurangan yang terdapat pada basisdata yang sedang dibangun.

4.2 Mendefinisikan Sistem dan Basisdata

Pada buku ini, studi kasus yang digunakan dalam pembuatan toko online men gacu pada objek counter handphone atau toko telpon cellular. Sehingga, sekarang persepsi Anda harus difokuskan pada toko tersebut. Walaupun objek yang Kita bahas adalah mengenai counter handphone, akan tetapi sistem dan alur program ini dapat Anda terapkan pada kasus yang Anda hadapi. Untuk lebih jelasnya, Anda dapat mengikuti pembahasan berikut ini.

4.2.1 Alasan Mendefinisikan Data

Sebelum Kita dapat mendefinisikan database dan melakukan coding program, maka sebagai pegangan Kita harus melakukan pencarian data pada objek yang bersangkutan. Sebagai contoh, apabila Kita hendak membangun sistem pemesanan online pada counter **Afit's Phone**, maka pada toko/counter tersebut Anda harus memburu data -data yang dibutuhkan

Sehingga dengan adanya data-data yang berkenaan dengan counter handphone tersebut, Kita dapat mendefinisikan data apasaja yang hendak ditampilkan. Selain itu, Kita juga harus mengumpulkan data-data mengenai penyediaan barang handphone yang akan ditawarkan secara online.

Dengan data-data tersebut, maka Kita sebagai pengembang dapat dengan mudah untuk mengimplementasikan sistem manual yang telah ada ke dalam sistem informasi secara online.

4.2.2 Mendefinisikan Objek Data

Setelah mengetahui alasan-alasan mengenai pencarian data, maka tugas Kita adalah mendefinisikan semua objek data yang ada di dalam sistem manual, tentunya objek -objek tersebut ada data yang akan diinformasikan pada sistem online nantinya.

Informasi yang berkenaan dengan counter tersebut sebaiknya didefinisikan dengan lengkapnya. Hal ini dapat mempermudah dan memberikan daya tarik pengunjung dalam

menjelajahi portal yang Kita miliki. Berkenaan dengan data counter handphone, beberapa hal yang mungkin akan Kita butuhkan adalah:

- **Profile counter handphone** : Profile merupakan informasi detail mengenai counter/toko handphone yang bersangkutan. Di dalam data ini menyangkut misi, visi ataupun sejarah berdirinya counter yang bersangkutan.
- **Data Merk yang Tersedia** : Ini merupakan data yang telah mengarah pada data handphone. Kita dapat mendaftar merek handphone apa saja yang dijual pada counter bersangkutan.
- **Data Tipe yang dijual** : Setelah merek, Anda dapat mendata jenis atau tipe handphone apa saja yang disediakan di dalam counter tersebut. Data -data yang berkaitan dengan tipe adalah keterangan lengkap tentang handphone tersebut, tentu saja beserta harga dan stok yang masih tersedia.
- **Alur Pemesanan** : Setelah beberapa data di atas Anda ketahui, maka Anda dapat mendiskusikan teknik pemesanannya dengan pemilik counter. Sehingga bentuk dan informasi tambahannya dapat disesuaikan dengan keinginan pemilik counter.

Kurang lebih beberapa informasi yang harus Anda kumpulkan seperti daftar di atas, Anda juga dapat mendefinisikannya dengan menyesuaikan counter handphone yang bersangkutan.

4.2.3 Alur Kerja Program

Pada study kasus toko online yang Kita bangun ini, program yang Kita bangun akan berbentuk portal (web dinamis). Jadi, selain menyediakan sistem pemesanan online, portal ini juga menyediakan web portal yang menampilkan informasi mengenai counter *Afit's Phone*.

Pada awal program dibuka, kepada seorang pengunjung akan disajikan sebuah halaman portal yang menyediakan informasi counter bersangkutan. Pada halaman tersebut pengunjung dapat mengisikan pesan atau kritik kepada pemilik counter. Selain itu, sebagai informasi juga akan Kita tampilkan halaman berita/news mengenai data -data handphone maupun berkenaan dengan informasi umum.

Untuk melakukan pemesanan, pengunjung dapat mengklik tombol/hyperlink *Pemesanan*, sehingga pengunjung akan dibawa pada program sistem pemesanan barang.

4.2.4 Alur Kerja Sistem Pemesanan

Pada waktu pertama kali pengunjung masuk pada halaman index pemesanan, maka program akan melakukan pencatatan dengan membaca nomor IP Address dari komputer klien yang digunakan untuk program pemesanan. Nomor IP inilah yang digun akan untuk memisahkan data pilihan barang handphone dengan data yang juga dipilih oleh pengunjung lain pada tempat yang berbeda dan waktu yang sama.

Dengan menggunakan pencatatan nomor IP ini, maka setiap pemesanan yang dilakukan pada komputer klien A, datanya tidak akan campur dengan data handphone yang sedang dipesan oleh klien B, klien C atau klien D. Penggunaan metode inilah yang menjadikan program pemesanan yang Kita bangun mampu bertindak sebagai aplikasi yang *Multi User* dan *Client Server*. Sehingga program dapat melayani banyak pengakses dalam waktu yang bersamaan dan dalam tempat yang berbeda -beda.

Beberapa hal yang dapat Anda ketahui mengenai jalannya pemesanan dapat Anda lihat pada daftar berikut:

1. Di dalam halaman program pemesanan, pengunjung dapat langsung memilih merek dan jenis handphone yang akan dibeli.
2. Pada saat memilih barang, pemesan dapat menentukan jumlah barang yang akan dipesan.

3. Setelah menekan tombol Pesan, maka informasi data yang dipilih akan disimpan pada tempat penyimpanan sementara. Dalam program yang Kita bangun, penyimpanan sementara diterapkan dengan tabel bernama kantong. Selain cara ini, Anda juga dapat menggunakan penyimpanan sementara dengan cookies.
4. Data-data yang ada di dalam halaman kantong pembelian akan dihapus apabila tanggal pemilihannya telah melewati satu hari dihitung dari hari ini. Jadi, setiap masuk pada kantong pembelian program akan selalu melakukan pembersihan data kantong dengan berdasarkan nomor IP-nya dan keterlambatan pemesanan. Dengan cara tersebut, misalnya pada tanggal 12-04-2005 melakukan pemilihan barang, akan tetapi tidak jadi dibayar, maka apabila pada tanggal 13-04-2005 mengunjungi portal ini data-datanya akan dihapus. Sehingga Kita tidak akan melihat tampilan pemesanan yang Kita gagalkan pada hari sebelumnya.
5. Dikatakan bahwa data-data yang ada di dalam kantong pembelian merupakan data sementara, sehingga sebelum barang tersebut dibayar, Kita dapat melakukan penghapusan.
6. Untuk menambah jumlah barang yang telah dipilih, Anda dapat mengubah datanya dengan cara memilih tombol **ubah**. Dengan fasilitas tersebut, maka calon pemesan dapat menambah atau mengurangi jumlah barang yang telah dipilih dan masuk pada kantong.
7. Setelah proses pemilihan data-data yang hendak dipesan selesai, maka Anda dapat melanjutkan proses pemesanan pada proses pembayaran.
8. Pada sistem pembayaran, seorang pemesan harus memiliki nomor rekening pada bank yang ditunjuk. Dalam sistem yang Kita bangun, data-data bank (nomor rekening, pin dan saldo) akan Kita simpan pada tabel bernama *v_bank*.
9. Pada proses pembayaran, Pemesan tidak harus menggunakan nomor rekening yang dimilikinya. Akan tetapi pemesan dapat menggunakan sembarang rekening yang ada. Jadi, apabila Anda mengetahui nomor rekening teman, maka Anda dapat menggunakannya.
10. Apabila proses autentifikasi rekening dan pin rekening dinyatakan berhasil, maka program akan segera mentransfer data-data yang dipilih (pada tabel *kantong*) ke dalam tabel *pemesanan*.
11. Pada proses transfer data, program juga akan melakukan pengurangan saldo pada rekening yang digunakan, dan menambahkannya pada nomor rekening yang dimiliki oleh counter (dalam hal ini nama rekeningnya adalah *centranet*).
12. Setelah pemesanan dinyatakan berhasil, maka pemesan akan segera diberikan nomor faktur atau nomor pemesanan yang dapat dijadikan sebagai tanda bukti telah melakukan pemesanan. Tanda bukti ini dapat didownload dan diprint oleh pemesan.

Selain kartu tanda pemesanan di atas, Anda juga dapat memperoleh deskripsi barang yang telah Anda pesan sebelumnya. Caranya adalah dengan masuk pada program informasi pemesanan. Pada program tersebut Anda dapat memasukkan nomor faktur dan kode pembeli yang sesuai dengan kartu tanda pesan yang telah Anda peroleh. Apabila benar, maka program akan segera menampilkan seluruh data handphone yang telah dipesan.

4.2.5 Database yang akan Digunakan

Karena Objek Toko yang akan dibangun adalah Toko Handphone, maka Kita dapat mempermudahnya dengan membuat database bernama *handphonedb*. Di dalam database

inilah nantinya Kita akan membuat berbagai jenis tabel yang dibu tuhkan dalam mendokumentasikan data-data handphone dan proses pemesanan. Pada database *handphonedb* tersebut akan Kita gunakan untuk merekam semua data -data yang berkenaan dengan sistem informasi counter *Afit's Phone* serta data handphone yang akan dijual.

4.3 Implementasi Pembuatan Database dan Tabel

Baiklah, sekarang mari Kita membuat perancangan database dan tabel yang akan digunakan sebagai basisdata untuk menyimpan data-data handphone yang akan Kita buat. Sebelumnya mungkin Anda harus memastikan bahwa server database Anda telah aktif sehingga siap untuk digunakan. Apabila belum, Anda dapat menjalankan daemونها seperti pembahasan **bab 2**. Untuk membuatnya, Anda dapat memperhatikan beberapa langkah berikut:

4.3.1 Membuat Database dan Mengaktifkannya

Sebelum melakukan perancangan dan menerapkan hasil rancangannya ke dalam database MySQL, maka pada langkah pertama Anda harus dalam kondisi aktif di dalam server MySQL. Sehingga sekarang Anda harus melakukan login pada server MySQL, perhatikan cara berikut:

```
bunafit@centranet:~$ mysql -u bunafit -p
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8 to server version: 4.1.7-max

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

mysql>
```

Setelah kondisi Anda berada pada Prompt MySQL, maka Anda dapat membuat database bernama *handphonedb* dan mengaktifkannya dengan cara berikut:

```
mysql> CREATE DATABASE handphonedb;
Query OK, 1 row affected (0.11 sec)
```

Pada query di atas, pernyataan “Query OK, 1 row affected (0.11 sec)” menandakan bahwa Kita sekarang telah berhasil membuat database baru bernama *handphonedb*. Di dalam database inilah semua informasi akan Kita dokumentasikan, sehingga dapat ditampilkan pada halaman program.

Sekarang Kita akan mencoba untuk mengaktifkan database *handphonedb* yang telah Kita buat, gunakanlah perintah berikut untuk mengaktifkannya.

```
mysql> USE handphonedb;
Database changed
```

Pada perintah di atas, pernyataan “Database changed” menandakan kondisi Kita sekarang telah berada di dalam database *handphonedb*. Berarti Kita sekarang dapat melakukan pembuatan tabel di dalam database tersebut.

4.3.2 Merancang Tabel Administrator

Untuk manajemen sistem basisdata yang Kita bangun ini, Kita akan membuat beberapa modul yang dapat melakukan operasi penambahan data merk, tipe maupun melakukan manipulasi dan informasi pada data -data lain di dalam database *handphonedb*.

Untuk mengamankan program pengelola database tersebut, Kita dapat membuat program autentifikasi yang fungsinya sebagai gerbang atau login untuk masuk pada sistem. Pada program autentifikasi yang Kita buat, username dan password yang digunakan akan disimpan pada tabel yang bernama *administrator*. Struktur rancangan dari tabel administrator dapat Anda lihat pada tabel berikut:

Tabel 4.1 Rancangan Tabel *Administrator*

Kolom	Tipe	Lebar	Keterangan
user_admin	VARCHAR	25	PRIMARY KEY
pass_admin	VARCHAR	25	

Untuk mengetahui lebih jelas dari rancangan tabel *administrator* di atas, maka Kami dapat mendefinisikan kolom dan fungsinya sebagai berikut:

- **User_admin** : adalah kolom yang digunakan untuk merekam data username dari seorang administrator.
- **Pass_admin** : adalah kolom yang digunakan untuk merekam data password dari username yang dimiliki oleh seorang administrator. Pada kolom ini, data yang digunakan akan Kita enkripsi dengan bentuk data berupa PASSWORD.

Setelah memahami beberapa deskripsi kolom dari tabel administrator di atas, maka sekarang Anda dapat membuat tabel tersebut dengan menggunakan perintah:

```
CREATE TABLE administrator (
  user_admin VARCHAR(25) NOT NULL,
  pass_admin VARCHAR(25) NOT NULL,
  PRIMARY KEY (user_admin) );
```

Dengan menggunakan perintah di atas, maka anda akan memiliki tabel administrator dengan bentuk deskripsi sebagai berikut:

```
mysql> DESC administrator;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| user_admin | varchar(25)   |      | PRI |          |       |
| pass_admin | varchar(25)   |      |     |          |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Pada program administrasi yang akan Kita bangun, Anda dapat masuk sebagai seorang admin dengan menggunakan username “admin” dan passwordnya adalah “centranet”. Maka dari itu, Anda dapat memasukkan datanya seperti tabel berikut:

Tabel 4.2 Data Login untuk administrator

User Login	Password Login
Admin	centranet

Perintah yang digunakan untuk memasukan data seperti tabel di atas adalah sebagai berikut:

```
INSERT INTO administrator
VALUES ('admin',PASSWORD('centranet'));
```

Setelah selesai, maka username dan password di atas harus Anda ingat, karena setiap kali akan masuk pada sistem administrasi harus memasukkan user dan password di atas.

4.3.3 Merancang Tabel Pengunjung

Tabel pengunjung ini akan Kita gunakan untuk mencatat nomor IP komputer klien yang sedang melakukan browsing pada sistem yang Kita bangun. Sehingga setiap orang yang melakukan browsing atau pemesanan akan memiliki nomor pengenal berupa IP Address sendiri-sendiri.

Nomor pengenal yang akan Kita simpan pada tabel ini berupa gabungan antara IP Address dan tanggal sekarang. Pengenal yang disimpan pada tabel ini juga akan digunakan sebagai pengenal data pemesanan. Sehingga setiap pengunjung yang melakukan pemilihan data akan dicatat dengan pengenal IP address dan tanggal. Struktur tabel *pengunjung* yang akan digunakan adalah:

Tabel 4.3 Rancangan Tabel *pengunjung*

Kolom	Tipe	Lebar	Keterangan
iplocal	VARCHAR	25	PRIMARY KEY
Tanggal	DATE		
Jam	CHAR	2	

Untuk mengetahui lebih jelas dari rancangan tabel *pengunjung* di atas, maka Kami dapat mendefinisikan kolom dan fungsinya sebagai berikut:

- **Iplocal** : kolom ini akan digunakan untuk mencatat data IP komputer klien dan data tanggal. Bentuk yang akan disimpan di dalam kolom tersebut adalah “*tahun-bulan-tanggal-nomorIP*”. Pada kolom ini dibentuk sebagai kunci primer karena tidak mungkin ada nomor IP yang sama dalam satu jaringan baik WAN (Wide Area Network) maupun LAN (Local Area Network) dan MAN (Medium Area Network).
- **Tanggal** : kolom ini akan digunakan untuk merekam tanggal saat pengunjung mengakses website ini.
- **Jam** : kolom ini akan digunakan untuk merekam jam berapa pengunjung mengakses website ini.

Dari bentuk rancangan di atas, Anda dapat merealisasikan ke dalam database MySQL dengan menggunakan perintah berikut:

```
CREATE TABLE pengunjung (
  iplocal varchar(25) NOT NULL default '',
  tanggal date NOT NULL default '0000-00-00',
  jam char(2) NOT NULL default '',
  PRIMARY KEY (iplocal)
);
```

Setelah query diatas dijalankan, maka Anda akan segera memiliki tabel *pengunjung* dengan bentuk deskripsi seperti tampilan berikut:

```
mysql> DESC pengunjung;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default    | Extra |
+-----+-----+-----+-----+-----+-----+
| iplocal | varchar(25)  |      | PRI |             |       |
| tanggal | date         |      |     | 0000-00-00 |       |
| jam     | char(2)      |      |     |             |       |
+-----+-----+-----+-----+-----+-----+
```

3 rows in set (0.02 sec)

4.3.4 Merancang Tabel Merk Handphone

Sistem yang Kita bangun adalah pemesanan barang elektronik berupa handphone atau telepon selular, maka untuk menyimpan data-data handphone Kita memerlukan tabel yang dapat mengelompokkan barangnya ke dalam beberapa merek.

Tabel yang akan Kita gunakan untuk merekamnya adalah tabel *merkhp*. Di dalam tabel ini semua data yang berkenaan dengan merek handphone yang disediakan pada counter akan disimpan. Berikut adalah rancangan tabel tersebut:

Tabel 4.4 Rancangan Tabel merkhp

Kolom	Tipe	Lebar	Keterangan
kd_merk	CHAR	4	PRIMARY KEY
merkhp	VARCHAR	35	
keterangan	VARCHAR	100	

Untuk mengetahui lebih jelas dari rancangan tabel *pengunjung* di atas, maka Kami dapat mendefinisikan kolom dan fungsinya sebagai berikut:

- **Kd_merk** : kolom ini berfungsi untuk mewakili tabel *merkhp* dalam relasi database. Sebagai kolom wakil, maka kolom ini dapat Kita pasang dengan tipe Primary Key. Sehingga data-data yang dimasukkan di dalam kolom ini bentuknya unik (tidak kembar). Kemungkinan data yang dimasukkan didalamnya misalnya seperti NOKI (Untuk merk Nokia), SEAM (Untuk merk Siemens) dll.
- **Merkhp** : kolom ini digunakan untuk menyimpan nama merk dari data handphone yang disediakan. Contoh datanya adalah Nokia, Siemens, Motorola, dan lain-lain.
- **Keterangan** : kolom ini digunakan untuk mendeskripsikan secara lengkap mengenai merek handphone yang dipasarkan. Pada kolom ini bisa saja tidak diisi dengan data, akan tetapi untuk kelengkapannya sebaiknya data yang Kita masukkan.

Setelah memahami struktur *merkhp* seperti tabel di atas, maka sekarang Anda dapat menerapkannya pada database MySQL. perintah yang dapat Anda gunakan adalah:

```
CREATE TABLE merkhp (  
  kd_merk CHAR(4) NOT NULL default '',  
  merkhp VARCHAR(35) NOT NULL default '',  
  keterangan VARCHAR(100) NOT NULL default '',  
  PRIMARY KEY (kd_merk)  
);
```

Setelah selesai menjalankan query di atas, maka Anda akan segera mendapatkan tabel bernama *merkhp*. Perhatikan kondisi strukturnya dengan perintah DESC berikut:

```
mysql> DESC merkhp;  
+-----+-----+-----+-----+-----+-----+  
| Field      | Type          | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| kd_merk    | char(4)       |      | PRI |          |       |  
| merkhp     | varchar(35)   |      |     |          |       |  
| keterangan | varchar(100)  |      |     |          |       |  
+-----+-----+-----+-----+-----+-----+  
3 rows in set (0.02 sec)
```

Sekarang, masukanlah data merk handphone yang disediakan oleh counter Afit's Phone. Beberapa contoh data yang dapat Anda masukkan adalah dengan menjalankan perintah berikut:

```
INSERT INTO merkhp VALUES ('SNER','Sony Ericson',
'Soni er....');
INSERT INTO merkhp VALUES ('NOKI','Nokia','');
INSERT INTO merkhp VALUES ('SEAM','Seamen','');
INSERT INTO merkhp VALUES ('ERIC','Ericson','');
INSERT INTO merkhp VALUES ('MOTO','Motorola','');
INSERT INTO merkhp VALUES ('SAMS','Samsung','');
```

Dengan menggunakan perintah di atas, maka pada tabel *merkhp* sekarang telah terdapat 6 baris data. Buktikan dengan menggunakan perintah SELECT seperti tampilan berikut:

```
mysql> SELECT * FROM merkhp;
+-----+-----+-----+
| kd_merk | merkhp      | keterangan |
+-----+-----+-----+
| SNER    | Sony Ericson | Soni er.... |
| NOKI    | Nokia       |              |
| SEAM    | Seamen       |              |
| ERIC    | Ericson      |              |
| MOTO    | Motorola     |              |
| SAMS    | Samsung      |              |
+-----+-----+-----+
6 rows in set (0.03 sec)
```

4.3.5 Merancang Tabel Tipe Handphone

Data terkecil setelah merek adalah tipe. Untuk setiap merek tentunya terdapat banyak sekali tipe handphone yang diproduksi. Pada sistem yang Kita bangun, data-data tipe handphone akan Kita simpan pada tabel yang bernama *tipehp*. Anda dapat melihat rancangannya seperti tabel berikut.

Tabel 4.5 Rancangan Tabel *tipehp*

Kolom	Tipe	Lebar	Keterangan
id_tipe	INT	4	PRIMARY KEY, AUTO INCREMENT, UNZIGNED ZEROFILL
kd_merk	CHAR	4	
tipehp	VARCHAR	45	
gambar	VARCHAR	60	
harga	INT	15	
stok	INT	4	
ongkos kirim	INT	15	
garansi	VARCHAR	25	
fasilitas	VARCHAR	100	
deskripsi	VARCHAR	225	

Untuk mengetahui lebih jelas dari rancangan tabel *tipehp* di atas, maka Kami dapat mendefinisikan kolom dan fungsinya sebagai berikut:

- **Id_tipe** : kolom ini merupakan kolom primer yang datanya dapat mewakili setiap baris pada tabel-tabel yang lain. Untuk mempermudah dalam penggunaannya, maka kolom ini dijadikan dengan bentuk kunci yang memiliki kemampuan sebagai auto increment data. Sehingga walaupun datanya tidak dimasukan, maka secara otomatis datanya diisi dengan data urut sebelumnya. Parameter UNZIGNED ZEROFILL berfungsi untuk membuat nomor dengan digit genap, seperti 0001, 0002, 0003, dll.

- **Kd_mer** : kolom ini merupakan kunci tamu yang berasal dari tabel merkhp. Dengan menggunakan bentuk relasi ini, maka hubungan antara tabel merkhp dengan tipehp adalah One to Many. Jadi, untuk satu merk memungkinkan memiliki tipe handphone yang lebih dari satu (banyak).
- **Tipehp** : kolom ini digunakan untuk merekam nama tipe handphone. Apabila dilihat dari kedudukannya, kolom ini menjadi informasi utama dari tabel tipehp.
- **Gambar** : kolom ini digunakan untuk merekam data path gambar yang sesuai dengan tipe handphone yang dimasukkan. Gambar ini berasal dari file yang masuk pada saat input dan upload gambar.
- **Garansi** : kolom ini digunakan untuk mencatat lama garansi dari setiap tipe handphone yang ada. Contoh data yang dimasukan adalah “12 bulan, 14 bulan, dll).
- **Fasilitas** : kolom ini digunakan untuk merekam data tentang fasilitas yang disediakan oleh tipe handphone. Karena data ini akan ditampilkan ke dalam bentuk web, maka dalam memasukan datanya harus diformat ke dalam bentuk script HTML.
- **Deskripsi** : hampir sama dengan kolom fasilitas. Kolom ini digunakan untuk memberikan keterangan secara lengkap pada tipe handphone yang bersangkutan. Bedanya dengan fasilitas adalah, bahwa pada kolom ini khusus menjelaskan masalah yang tidak dibahas pada kolom fasilitas. Misalnya sejarah, perkembangan atau perbandingan tipe handphone dengan tipe yang lainnya. Karena datanya sangat banyak, maka Kita dapat menyetingnya alam tipe data text.

Setelah memahami rancangan struktur tabel tipehp di atas, maka sekarang Anda dapat menerapkannya pada database MySQL. untuk lebih jelasnya, Anda dapat menggunakan perintah berikut di dalam prompt mysql.

```
CREATE TABLE tipehp (
  id_tipe INT(4) unsigned zerofill NOT NULL
  auto_increment,
  kd_merk CHAR(4) NOT NULL default '',
  tipehp VARCHAR(45) NOT NULL default '',
  gambar VARCHAR(60) NOT NULL default '',
  harga INT(15) NOT NULL default '0',
  stok INT(4) NOT NULL default '0',
  ongkos kirim INT(15) NOT NULL default '0',
  garansi VARCHAR(25) NOT NULL default '',
  fasilitas VARCHAR(200) NOT NULL default '',
  deskripsi VARCHAR(225) NOT NULL default '',
  PRIMARY KEY (id_tipe)
);
```

dengan perintah di atas, maka sekarang Anda akan memiliki tabel tipehp dengan bentuk struktur berikut:

```
mysql> DESC tipehp;
```

Field	Type	Null	Key	Default	Extra
id_tipe	int(4) unsigned zerofill		PRI	NULL	auto_increment
kd_merk	char(4)				
tipehp	varchar(45)				
gambar	varchar(60)				
harga	int(15)			0	
stok	int(4)			0	
ongkos kirim	int(15)			0	
garansi	varchar(25)				
fasilitas	varchar(200)				
deskripsi	varchar(225)				

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

Untuk kelengkapan datanya, sekarang Anda dapat memasukkan beberapa data tipe handphone pada tabel tipehp. Perhatikan perintah berikut:

```
INSERT INTO tipehp
VALUES (0001,'MOTO','Motorola v600 Camera Phone',
'Att600.gif',1670400,2,25000,'12 bulan',
'<UL>\r\n
<LI>Built-In VGA Camera With Zoom \r\n
<LI>Bluetooth® Connectivity \r\n
<LI>Speaker Phone With Voice Dialing \r\n
<LI>Personal Information Manager (PIM) Functionality \r\n
<LI>Situational Lights \r\n
<LI>MMS - Multi Media Messaging \r\n
<LI>Picture Caller ID \r\n
<LI>Color Display&nbsp;: \r\n
<LI>390 Min Talk/175 Hours Standby \r\n
<LI>Weight-4.40 Ounces \r\n
<LI>Dimensions-3.50\" x 1.90\" x 1.00\" \r\n
<LI>1000 Entry Phonebook \r\n
<LI>Ringtones & Vibrating Option \r\n
<LI>Games-Built-In and Downloadable / Supports Wireless
Multiplayer Gaming \r\n </LI>
</UL>\r\n','');

INSERT INTO tipehp
VALUES (0002,'MOTO','Motorola v265',
'motorola_v265.jpg',960000,5,20000,'10 bulan',
'<UL>\r\n
<LI>4x Digital Zoom With Self-timer \r\n
<LI>Advanced voice recognition for hands-free dialing \r\n
<LI>Vibrant color internal screen \r\n
<LI>Office-quality Speakerphone \r\n
<LI>BREW 2.1 for downloadable apps \r\n
<LI>200 Min Talk/150 Hours Standby \r\n
<LI>Weight-3.77 Ounces \r\n
<LI>Dimensions-3.60\" x 1.80\" x 0.90\"</< li> \r\n
<LI>500 Entry Phonebook \r\n
<LI>64 Chord Polyphonic Ring Tones \r\n
<LI>Vibrating Option \r\n
<LI>Built-in Alarm </LI>
</UL>\r\n','');

INSERT INTO tipehp
VALUES (0003,'MOTO','Motorola v551 Camera Phone',
'motorola_v551.jpg',672000,10,20000,'12 bulan',
'<UL>\r\n
<LI>Bluetooth Wireless Technology \r\n
<LI>Built in Digital VGA Camera \r\n
<LI>Record Video Clips \r\n
<LI>Quad-Band GSM&nbsp;: (World Phone) \r\n
<LI>Built In Speaker Phone \r\n
<LI>225 Min Talk/225 Hours Standby \r\n
```

```
<LI>Weight-4.23 Ounces \r\n
<LI>Dimensions-3.50\" x 1.93\" x 0.97\" \r\n
<LI>1,000 Entry Phonebook \r\n
<LI>Alert Ringer Options-Polyphonic, Customized For Each
Caller \r\n
<LI>Vibrating Option \r\n
<LI>Games </LI>
</UL>\r\n','');
```

INSERT INTO tipehp

```
VALUES (0004,'SAMS','Samsung A740 Camera Phone',
'samsung_a740.jpg',580000,15,30000,'12 bulan',
'<UL>\r\n
<LI>Digital Camera w/ Flash & Self-portrait \r\n
<LI>Beautiful Color Displays, Inside and Out \r\n
<LI>To-Do List and Voice Memo \r\n
<LI>Large Phonebook \r\n
<LI>Speaker-independent Hands-free Dialing \r\n
<LI>AOL Instant Messenger Built-in \r\n
<LI>32 Ringtones & Vibrate Mode \r\n
<LI>Picture Caller ID & Ringer Profiles \r\n
<LI>Alarm, Calculator & Calendar \r\n
<LI>300 Entry Phonebook \r\n
<LI>3 Hours Talk/8 Days Standby \r\n
<LI>Built-in Memory 1.5 MB \r\n
<LI>Dimensions- 3.50\" x 1.80\" x 1.00\" \r\n
<LI>Weight-3.46 oz. </LI>
</UL>\r\n','');
```

INSERT INTO tipehp

```
VALUES (0005,'SAMS','Samsung e715 Camera Phone',
'samsung_e715.gif',960000,5,25000,'12 bulan',
'<UL>\r\n
<LI>MMS-with maximum message size of 50KB / Supports Voice
Messages \r\n
<LI>Built-in Camera with flash, self-timer, multi-shot, CMOS
sensor \r\n
<LI>GPRS High-speed data \r\n
<LI>Java capable \r\n
<LI>Color Display \r\n
<LI>Wireless Internet Ready \r\n
<LI>Battery Type-LiIon 900 mAh \r\n
<LI>240 Min Talk/300 Hours Standby \r\n
<LI>1000 Entry Phone Book \r\n
<LI>40-chord polyphonic ringtones \r\n
<LI>Vibrating Option \r\n
<LI>Weight-3.35 Ounces \r\n
<LI>SMS Messaging-2-Way with T9@ Predictive Text
Messaging</LI>
</UL>\r\n','');
```

INSERT INTO tipehp

```
VALUES (0006,'NOKI','Nokia 6600 Camera Phone',
'nokia-6600_large.jpg',2550000,4,30000,'12 bulan',
'<ul>\r\n
<li> Dukungan OS Symbian versi 7.0 </li>\r\n
<li>Memory internal sebesar 6MB</li>\r\n
<li>Dukungan GPRS dan HSCSD</li>\r\n
<li>Mampu merekam momen-momen penting lewat video
recordernya</li>\r\n
<li>Hasil rekaman lewat video maupun jepretan kamera dapat
disimpan langsung \r\n pada mdeia gallery yang sudah
disiapkan</li>\r\n
<li>Menggunakan WAP versi 2.0 dan JAVA MIDP 2.0</li> \r\n
</ul>\r\n',
'Bagaimana sumber air yang tak pernah kering. Vendor ponsel
asal Finlandia ini terus menggulirkan satu demi satu produk
ponselnya dengan segudang keragaman. Boleh dibbilang hampir
```

setiap bulannya selalu ada produk baru yang dilempar ke pasar. Seperti kali ini Nokia kembali menggerojoki pasar dengan seri terbarunya Nokia 6600. \r\n\r\nTampil sangat lengkap dan merupakan ponsel pertama Nokia yang mengadopsi layar warna dengan teknologi TFT (Thin Film Transistor) berke dalaman 65 ribu warna. Selain layar warna, teknologi yang telah dibenamkan didalamnya juga termasuk kelas atas dengan versi baru. Sebut saja XHTML, Java MIDP 2.0, Mobile Wallet versi 2.0 dan WAP versi 2.0. Ditunjang software Symbian versi 7.0 menjadikan ponsel ini semakin terlihat bobotnya. Di jajaran keluarga Nokia, seri 6600 merupakan ponsel pertama yang menggunakan sistem operating Symbian versi 7.0. \r\n\r\n');

Pada query masukan data tipe handphone di atas, datanya telah Kita format dengan bentuk kode HTML. Sehingga dengan format di atas, halaman web dapat menampilkan datanya sesuai dengan format data yang dimasukan. Untuk mempermudah dalam memasukan data di atas, sebaiknya Anda desain dahulu data masukannya menggunakan editor web seperti Front Pages ataupun Dreamweaver.

4.3.6 Merancang Tabel Pembeli

Dalam setiap kali pemesanan dinyatakan berhasil, maka seseorang yang melakukan transaksi harus memasukan data-data pribadinya beserta alamat tempat tinggalnya atau alamat pengiriman datanya. Semua data-data yang dimasukkan oleh pelanggan tersebut akan disimpan pada tabel pembeli. Rancangan tabel pembeli dapat Anda lihat pada tabel berikut.

Tabel 4.6 Rancangan Tabel *pembeli*

Kolom	Tipe	Lebar	Keterangan
kd_pembeli	CHAR	4	PRIMARY KEY
iplocal	VARCHAR	25	
nama	VARCHAR	35	
alamat	VARCHAR	60	
kota	VARCHAR	60	
propinsi	VARCHAR	60	
negara	VARCHAR	60	
kdpost	INT	5	
email	VARCHAR	40	
telp_rumah	VARCHAR	15	
telp_hp	VARCHAR	15	
no_rek	VARCHAR	15	

Untuk mengetahui lebih jelas dari rancangan tabel *pembeli* di atas, maka Kami dapat mendefinisikan kolom dan fungsinya sebagai berikut:

- **Kd_pembeli:** kolom ini digunakan untuk merekam kode pengenal dari setiap pembeli/pelanggan yang telah melakukan transaksi. Kode yang disimpan di dalam kolom ini akan dibuat secara otomatis. Jadi, Kita akan membuat script untuk otomatisasi kode pembelian.
- **Iplocal** : adalah kolom yang digunakan untuk merekam data (tanggal+IP Address) yang didapatkan dari tabel pengunjung. Pada tabel pengunjung, data iplocal berasal dari penggabungan antara tanggal pemesanan dan nomor IP komputer client yang digunakan.
- **Nama** : kolom ini digunakan untuk merekam data nama dari pengunjung yang telah berhasil melakukan pembelian.

- **Alamat** : data yang disimpan pada kolom ini akan dijadikan sebagai alamat tujuan pengiriman barang yang dipesan. Jadi, apabila Anda melakukan pemesanan, maka alamat yang akan dijadikan sebagai tujuan pengiriman harus dimasukkan pada kolom ini.
- **Kota** : Anda dapat menyimpan data kota tempat pembeli tinggal. Sehingga dapat melakukan pencarian alamat yang dimasukkan dapat diketahui dengan cepat.
- **Propinsi** : data lokasi yang kondisinya di atas kota adalah propinsi. Data yang dimasukkan di dalam tabel ini untuk menjelaskan di propinsi mana letak dari alamat yang akan menjadi target pengiriman.
- **Negara** : kolom ini digunakan untuk merekam data pengenalan dari nama negara tempat orang melakukan pemesanan dan target alamat yang dituju untuk pengiriman.
- **E_mail** : kolom ini digunakan sebagai media penyimpanan data konfirmasi berupa alamat E-mail.
- **Telp_rumah**: kolom ini digunakan untuk merekam data alamat konfirmasi berupa nomor telepon rumah.
- **Telp_hp** : merupakan kolom yang digunakan untuk merekam nomor handphone.
- **No_rek** : kolom ini digunakan untuk merekam data nomor rekening yang digunakan untuk melakukan transaksi pembayaran.

Setelah memahami bentuk rancangan dari tabel pembeli, maka Anda dapat mengimplementasikannya dengan cara menuliskan perintah berikut pada prompt MySQL.

```
CREATE TABLE pembeli (
  kd_pembeli CHAR(4) NOT NULL default '',
  iplocal VARCHAR(25) NOT NULL default '',
  nama VARCHAR(35) NOT NULL default '',
  alamat VARCHAR(60) NOT NULL default '',
  kota VARCHAR(60) NOT NULL default '',
  propinsi VARCHAR(60) NOT NULL default '',
  negara VARCHAR(60) NOT NULL default '',
  kdpost INT(5) NOT NULL default '0',
  email VARCHAR(40) NOT NULL default '',
  telp_rumah VARCHAR(15) NOT NULL default '',
  telp_hp VARCHAR(15) NOT NULL default '',
  no_rek VARCHAR(15) NOT NULL default '0',
  PRIMARY KEY (kd_pembeli)
);
```

Setelah perintah di atas berhasil dibuat, maka Anda akan segera mendapatkan bentuk tampilan tabel pembeli seperti struktur berikut:

```
mysql> DESC pembeli;
```

Field	Type	Null	Key	Default	Extra
kd_pembeli	char(4)		PRI		
iplocal	varchar(25)				
nama	varchar(35)				
alamat	varchar(60)				
kota	varchar(60)				
propinsi	varchar(60)				
negara	varchar(60)				
kdpost	int(5)			0	
email	varchar(40)				
telp_rumah	varchar(15)				
telp_hp	varchar(15)				

```
| no_rek      | varchar(15) |          | 0          |          |
+-----+-----+-----+-----+-----+
12 rows in set (0.09 sec)
```

4.3.7 Merancang Tabel Pemesanan

Tabel pemesanan digunakan untuk merekam semua data yang sebelumnya berada di dalam tabel kantong. Data yang ditransfer ke dalam tabel pemesanan memiliki syarat apabila proses validasi nomor rekening telah dilakukan dengan sukses. Pada tabel ini, setiap kali melakukan pemesanan pelanggan akan diberikan nomor faktur pemesanan yang menjadi bukti dari pembelian.

Secara fungsi, tabel ini hanya akan menyimpan data faktur pembelian, kode pembelinya serta tanggal pembelian saja, sedangkan untuk data barangnya secara lengkap akan disimpan pada tabel *det_pesanan*. Untuk lebih jelasnya, Anda dapat melihat rancangan tabel data pemesanan seperti pada tabel berikut:

Tabel 4.7 Rancangan Tabel *Pemesanan*

Kolom	Tipe	Lebar	Keterangan
No_faktur	CHAR	7	PRIMARY KEY
kd_pembeli	CHAR	4	
tanggal	DATE		

Untuk mengetahui lebih jelas dari rancangan tabel *pemesanan* di atas, maka Kami dapat mendefinisikan kolom dan fungsinya sebagai berikut:

- **No_faktur**: kolom ini digunakan untuk mencatat nomor faktur atau nomor pembelian yang diberikan kepada pembeli yang telah berhasil melakukan pemesanan. Nomor faktur yang dimasukkan ke dalam tabel ini berasal dari script yang mampu menciptakan nomor pembelian secara otomatis. Misalnya FAK0001, FAK0002, dll.
- **Kd_pembeli** : kolom ini digunakan untuk merekam data kode pembeli yang telah melakukan transaksi pemesanan. Kolom ini berelasi dengan *kd_pembeli* yang ada di dalam tabel *pembeli*.
- **Tanggal** : kolom ini digunakan untuk mencatat data tanggal saat melakukan pemesanan barang.

Dengan melihat struktur rancangan tabel di atas, Kita dapat menerapkan tabel pemesanan pada database MySQL. berikut adalah perintahnya:

```
CREATE TABLE pemesanan (
  id_pesan INT(4) unsigned zerofill NOT NULL
  auto_increment,
  kd_pembeli CHAR(4) NOT NULL default '',
  tanggal DATE NOT NULL default '0000-00-00',
  PRIMARY KEY (id_pesan)
);
```

Setelah Anda menjalankan perintah di atas, maka pada database *handphonedb* telah bertambah satu tabel dengan nama *pemesanan*. Perhatikan deskripsi tabel tersebut.

```
mysql> DESC pemesanan;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| no_faktur  | char(7)   |      | PRI |          |       |
```

```

| kd_pembeli | char(4) |          |          |          |          |
| tanggal   | date    |          |          | 0000-00-00 |          |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.58 sec)

```

4.3.8 Merancang Tabel Detail Pesan

Tabel detail pesan (*det_pesan*) digunakan untuk mendeskripsikan data lengkap handphone yang telah dipesan. Fungsi utama dari tabel ini adalah menerangkan nomor faktur yang ada di dalam tabel pemesanan. Keterangan lengkap mengenai semua barang yang telah dipesan akan disimpan pada tabel *det_pesan* ini. Berikut adalah rancangan tabelnya.

Tabel 4.8 Rancangan Tabel *det_pesan*

Kolom	Tipe	Lebar	Keterangan
id_dpesan	INT	4	PRIMARY KEY, AUTO INCREMENT, UNZIGNED ZEROFILL
No_faktur	CHAR	7	
Id_tipe	CHAR	4	
Jumlah	INT	3	
Totalharga	INT	12	

Untuk mengetahui lebih jelas dari rancangan tabel *det_pesan* di atas, maka Kami dapat mendefinisikan kolom dan fungsinya sebagai berikut:

- **Id_dpesan:** kolom ini digunakan untuk kunci utama dari tabel *det_pesan*. Untuk mempermudah memasukkan data, kolom ini dibuat dalam bentuk penambahan secara otomatis (auto increment).
- **No_faktur:** kolom ini berasal dari tabel *pemesanan*, sehingga sifat hubungan antara tabel pemesanan dan tabel *det_pesan* adalah One to Many, artinya dalam satu kali pemesanan (satu nomor faktur) pembeli dapat membeli barang lebih dari satu handphone.
- **Id_tipe** : kolom ini berasal dari tabel *tipehp*. Fungsi keberadaan kolom ini adalah digunakan untuk mencatat tipe handphone apa yang telah dipesan.
- **Jumlah** : kolom ini digunakan untuk merekam data jumlah handphone yang dipesan. Karena data jumlah, maka kolom ini dibuat dalam bentuk integer.
- **Totharga** : kolom ini digunakan untuk menjumlahkan harga total dari barang yang dibeli, total ini didapatkan dari perkalian antara harga barang x dengan jumlah barang yang dipesan.

Sekarang Anda dapat merealisasikan desain tabel *det_pesan* di atas dengan menggunakan perintah berikut:

```

CREATE TABLE det_pesan (
  id_dpesan INT(6) NOT NULL auto_increment,
  id_pesan CHAR(4) NOT NULL default '',
  id_tipe CHAR(4) NOT NULL default '',
  jumlah INT(3) NOT NULL default '0',
  totalharga INT(12) NOT NULL default '0',
  PRIMARY KEY (id_dpesan)
);

```

setelah perintah di atas dijalankan, maka Anda akan segera mendapatkan tabel *det_pesan* dengan struktur seperti deskripsi tabel berikut:

```
mysql> DESC det_pesanan;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id_pesanan | int(6)    |      | PRI | NULL    | auto_increment |
| no_faktur  | char(7)   |      |     |         |                |
| id_tipe    | char(4)   |      |     |         |                |
| jumlah     | int(3)    |      |     | 0       |                |
| totharga   | int(12)   |      |     | 0       |                |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.06 sec)
```

4.3.9 Merancang Tabel Pembayaran

Tabel ini digunakan untuk mendata pembayaran setiap kali pemesan berhasil melakukan transaksi pemesanan. Data-data yang dimasukkan di dalam tabel ini berupa total uang yang dibayarkan dan waktu saat melakukan transaksi pemesanan. Berikut adalah rancangan dari tabel *pembayaran*.

Tabel 4.9 Rancangan Tabel *Pembayaran*

Kolom	Tipe	Lebar	Keterangan
id_bayar	INT	5	PRIMARY KEY, AUTO INCREMENT
Id_pesanan	CHAR	4	
Totalharga	INT	12	
Totalbayar	INT	12	
Tanggal	DATE		
Jam	TIME		

Untuk mengetahui lebih jelas dari rancangan tabel *pembayaran* di atas, maka Kami dapat mendefinisikan kolom dan fungsinya sebagai berikut:

- **Id_bayar** : kolom ini merupakan kode kunci dari tabel pembayaran, data yang ada di dalam ini dibentuk dengan metode *auto increment*. Sehingga secara otomatis datanya akan masuk dengan cara urut data.
- **Id_pesanan** : karena tabel ini berfungsi untuk merekam data pembayaran saat pemesanan, maka kolom ini digunakan untuk merelasikan antara proses transaksi *pemesanan* dengan *pembayaran*.
- **Totharga** : kolom ini digunakan untuk mencatat berapa jumlah harga total dari semua barang yang telah dipesan untuk satu kali pemesanan yang telah dilakukan.
- **Totbayar** : kolom ini datanya sama dengan kolom totharga, yaitu jumlah uang yang telah dibayarkan .
- **Tanggal** : kolom ini digunakan untuk merekam tanggal saat proses pembayaran berlangsung. Data yang ada di dalam kolom ini akan sama dengan kolom tanggal yang ada di dalam tabel *pemesanan*.
- **Jam** : kolom ini digunakan untuk merekam jam pembayaran berlangsung.

Dari deskripsi rancangan tabel di atas, maka Anda dapat menerapkannya pada database MySQL dengan menggunakan query berikut:

```
CREATE TABLE pembayaran (
  id_bayar INT(5) NOT NULL auto_increment,
  id_pesanan CHAR(4) NOT NULL default '',
  totalharga INT(12) NOT NULL default '0',
```

```

totalbayar INT(12) NOT NULL default '0',
tanggal DATE NOT NULL default '0000-00-00',
jam TIME NOT NULL default '00:00:00',
PRIMARY KEY (id_bayar)
);

```

hasil dari query di atas akan tampak seperti bentuk deskripsi tabel berikut:

```
mysql> DESC pembayaran;
```

Field	Type	Null	Key	Default	Extra
id_bayar	int(5)		PRI	NULL	auto_increment
id_pesanan	char(4)				
totalharga	int(12)			0	
totalbayar	int(12)			0	
tanggal	date			0000-00-00	
jam	time			00:00:00	

```
6 rows in set (0.05 sec)
```

4.3.10 Membuat Tabel Kantong Pembelian

Sebelum semua data yang dipesan dimasukkan ke dalam tabel pemesanan, maka data-data barang yang dipilih akan disimpan pada tabel *kantong*. Dalam hal ini, tabel kantong berfungsi sebagai keranjang belanja yang dapat merekam data-data sementara sebelum data barang tersebut dibayar. Rancangan tabel tersebut adalah.

Tabel 4.10 Rancangan Tabel *kantong*

Kolom	Tipe	Lebar	Keterangan
id_kantong	INT	5	PRIMARY KEY, AUTO INCREMENT
Iplocal	VARCHAR	25	
Id_tipe	CHAR	4	
Jumlah	INT	3	
Tanggal	DATE		
Jam	CHAR	2	

Untuk mengetahui lebih jelas dari rancangan tabel *kantong* di atas, maka Kami dapat mendefinisikan kolom dan fungsinya sebagai berikut:

- **Id_kantong** : kolom ini akan menjadi kunci utama dari tabel *kantong*. Untuk mempermudah pemasukan data, maka kolom ini dibuat dengan tipe Auto Increment.
- **Iplocal** : kolom ini digunakan untuk merekam pengenalan dari komputer yang digunakan untuk melakukan transaksi. Sehingga server dapat membedakan setiap data yang dipilih adalah data milik siapa. Hal ini karena menyangkut sistem yang Kita bangun dapat digunakan dalam waktu yang bersamaan oleh banyak pengguna.
- **Id_tipe** : adalah kolom yang digunakan untuk merekam data kode tipe barang yang dipilih. Kolom ini berasal dari tabel *tipehp*.
- **Jumlah** : kolom ini berfungsi untuk merekam data jumlah barang yang telah dipesan. Untuk data jumlah, Kita dapat memasangnya dengan bentuk Integer.
- **Tanggal** : kolom ini digunakan untuk mencatat waktu tanggal saat Kita memilih barang untuk dipesan
- **Jam** : digunakan untuk merekam waktu berupa jam saat barang dipilih.

Setelah memahami struktur tabel di atas, maka Anda dapat menerapkan desainnya pada database MySQL. Berikut adalah querinya:

```
CREATE TABLE kantong (
  id_kantong INT(5) NOT NULL auto_increment,
  iplocal VARCHAR(25) NOT NULL default '',
  id_tipe CHAR(4) NOT NULL default '',
  jumlah INT(3) NOT NULL default '0',
  tanggal DATE NOT NULL default '0000-00-00',
  jam CHAR(2) NOT NULL default '',
  PRIMARY KEY (id_kantong)
);
```

Hasil dari query di atas dapat Anda lihat seperti deskripsi tabel berikut:

```
mysql> DESC kantong;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_kantong | int(5)    |      | PRI | NULL    | auto_increment |
| iplocal   | varchar(25) |      |     |         |               |
| id_tipe   | char(4)   |      |     |         |               |
| jumlah    | int(3)    |      |     | 0       |               |
| tanggal  | date      |      |     | 0000-00-00 |           |
| jam       | char(2)   |      |     |         |               |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

4.3.11 Membuat Tabel Virtual Bank

Untuk mempermudah membangun sistem pemesanan ini, Kita akan menggunakan tabel bank virtual sebagai media penyimpanan data rekening dan saldonya. Sehingga pada saat melakukan validasi rekening, program akan melakukan pengecekan data rekening dan saldo pada tabel ini. Rancangan dari tabel *v_bank* dapat Anda lihat pada tabel berikut:

Tabel 4.11 Rancangan Tabel *v_bank*

Kolom	Tipe	Lebar	Keterangan
No_rek	VARCHAR	15	PRIMARY KEY
Pin_rek	VARCHAR	15	
Jml_saldo	INT	12	

Untuk mengetahui lebih jelas dari rancangan tabel *v_bank* di atas, maka Kami dapat mendefinisikan kolom dan fungsinya sebagai berikut:

- **No_rek** : kolom ini digunakan untuk merekam data nomor rekening. Pemesan dapat menggunakan semua nomor rekening yang disimpan pada tabel ini apabila pinnya juga mengetahuinya.
- **Pin_rek** : kolom ini digunakan untuk merekam data pin rekening dari nomor rekening. Untuk dapat menggunakan nomor rekening, maka Kita juga harus mengetahui nomor pin yang sesuai.
- **Jml_saldo** : kolom ini digunakan untuk merekam jumlah finansial saldo yang dimiliki oleh rekening bersangkutan. Setiap kali pemesanan berlangsung, maka jumlah saldo yang ada di dalam kolom ini akan selalu dikurangi dan datanya akan ditambahkan pada rekening “centranet” milik *Afit's Phone*.

Sekarang Anda dapat menerapkan deskripsi tabel *v_bank* di atas pada database MySQL. Perintah yang dapat Anda gunakan adalah:

```
CREATE TABLE v_bank (  
  no_rek VARCHAR(15) NOT NULL default '',  
  pin_rek VARCHAR(15) NOT NULL default '',  
  jml_saldo INT(10) NOT NULL default '0',  
  PRIMARY KEY (no_rek)  
);
```

Setelah perintah di atas dijalankan, maka Anda akan memiliki tabel *v_bank* dengan struktur seperti deskripsi berikut:

```
mysql> DESC v_bank;  
+-----+-----+-----+-----+-----+-----+  
| Field      | Type          | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| no_rek     | varchar(15)   |      | PRI |          |       |  
| pin_rek    | varchar(15)   |      |     |          |       |  
| jml_saldo  | int(10)       |      |     | 0        |       |  
+-----+-----+-----+-----+-----+-----+  
3 rows in set (0.03 sec)
```

Untuk mempermudah Anda dalam menggunakan tabel ini, maka sekarang masukkanlah beberapa rekening dan jumlah saldo masing-masing rekening. Berikut adalah perintahnya:

```
INSERT INTO v_bank VALUES ('centranet','bunafit',0);  
INSERT INTO v_bank VALUES ('123456','654321',10000000);  
INSERT INTO v_bank VALUES ('234567','765432',10000000);
```

Setelah query input di atas dijalankan, maka kondisi tabel *v_bank* Anda sekarang akan berisi 3 baris data rekening. Perhatikan hasil seleksi berikut:

```
mysql> select * from v_bank;  
+-----+-----+-----+  
| no_rek  | pin_rek | jml_saldo |  
+-----+-----+-----+  
| centranet | bunafit | 0          |  
| 123456   | 654321 | 10000000  |  
| 234567   | 765432 | 10000000  |  
+-----+-----+-----+  
3 rows in set (0.01 sec)
```

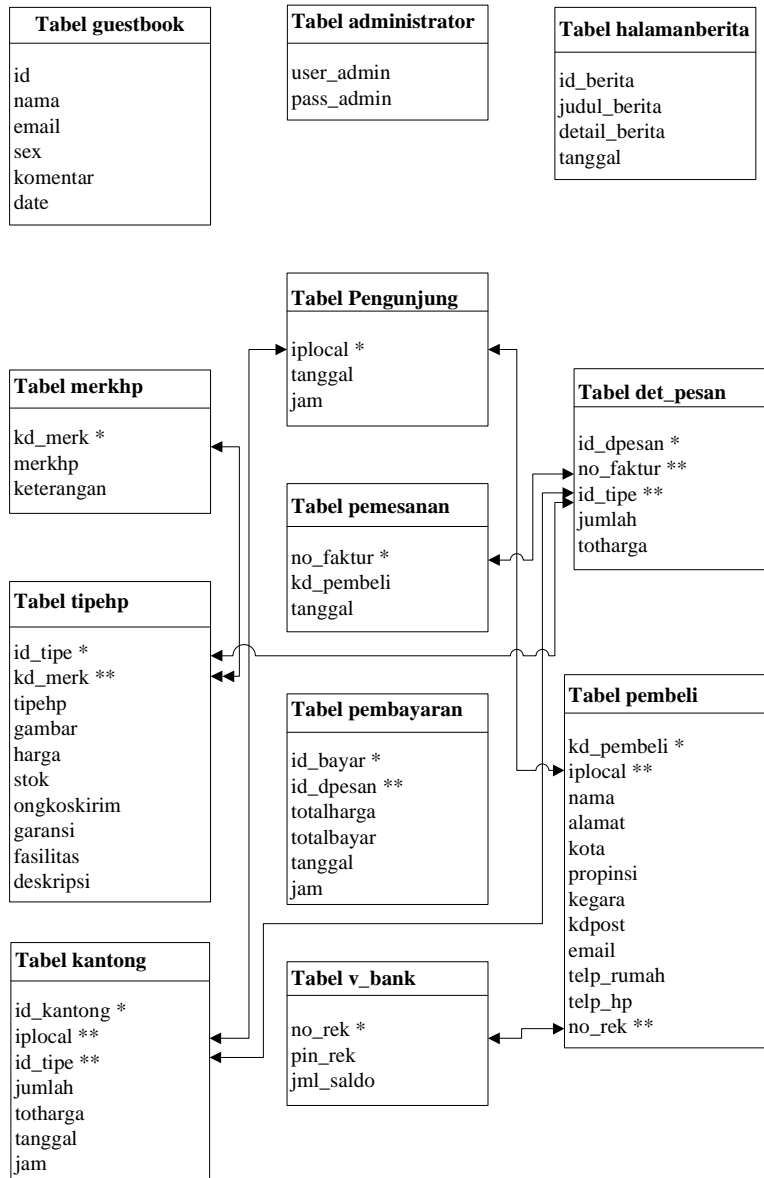
Pada seleksi data di atas, nomor rekening “centranet” merupakan nomor rekening pusat yang dimiliki oleh Afit’s Phone. Pada saat melakukan pemesanan, pelanggan dapat menggunakan nomor rekening selain “centranet”, yaitu 123456 atau 23456.

Pada saat proses validasi rekening dinyatakan berhasil, maka jumlah saldo pada rekening yang digunakan akan dikurangi dengan total harga yang harus dibayarkan. Jumlah uang pengurangan tersebut selanjutnya dimasukkan ke dalam nomor rekening pusat dengan alamat “centranet”.

4.3.12 Merancang Relasional Tabel

Relasional merupakan hubungan antara beberapa tabel yang saling memiliki keterkaitan data dan tidak dapat dipisah-pisahkan. Tidak ada sebuah sistem informasi yang dalam pemakaian database yang tidak membutuhkan berelasi dalam basisdatanya. Begitu juga dengan sistem Informasi Toko Online, Kita pasti akan memerlukan sebuah sistem basisdata yang saling berelasi dengan tabel yang lain.

Dari berbagai objek data yang didapatkan pada subbab sebelumnya, maka, maka kurang lebih Kita akan memerlukan database dengan relasional tabel seperti yang tampak pada gambar berikut:



Gambar 4.1 Relasi antar tabel

Dengan menggunakan struktur tabel yang terbagi-bagi seperti yang tampak pada gambar di atas, maka setiap atribut data dapat dipisah-pisahkan menurut tabel. Sehingga redundansi (perulangan data) atau kesalahan rekaman data tidak terjadi.

4.4 Kebutuhan Database di Luar Sistem

Selanjutnya semua tabel inti di atas, Kita masih memerlukan beberapa tabel yang berkenaan dengan informasi dan untuk konfirmasi pengunjung kepada counter atau admin pengelola. Sehingga di dalam website yang Kita bangun dapat memiliki informasi berupa news/berita

serta kolom pengisian data konfirmasi atau kritik saran. Berikut adalah yang akan dibutuhkan.

4.4.1 Membuat Tabel Guestbook

Tabel questbook digunakan untuk membuat kolom kritik dan saran. Dengan adanya fasilitas ini, maka seorang pengunjung dapat mengirimkan kritik, saran atau konfirmasi kepada pengelola berkenaan dengan sistem atau masalah yang dihadapi dalam menggunakan program yang ada. Struktur tabel guestbook yang dapat dibuat adalah:

Tabel 4.12 Rancangan Tabel *guestbook*

Kolom	Tipe	Lebar	Keterangan
Id	INT	3	PRIMARY KEY, AUTO INCREMENT
Nama	VARCHAR	35	
Email	VARCHAR	60	
Sex	ENUM('P','W')		
Komentar	VARCHAR	200	
Date	DATE		

Dari struktur tabel di atas, Anda dapat menerapkannya pada database MySQL dengan menggunakan perintah berikut:

```
CREATE TABLE guestbook (
  id INT(3) NOT NULL auto_increment,
  nama VARCHAR(35) NOT NULL default '',
  email VARCHAR(60) NOT NULL default '',
  sex ENUM('P','W') NOT NULL default 'P',
  komentar VARCHAR(200) NOT NULL default '',
  date DATE NOT NULL default '0000-00-00',
  PRIMARY KEY (id)
);
```

Setelah perintah di atas dijalankan, maka Anda dapat melihat hasil pembuatannya dengan menampilkan deskripsinya. Berikut adalah hasilnya:

```
mysql> DESC guestbook;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id    | int(3)       |      | PRI | NULL    | auto_increment |
| nama  | varchar(35)  |      |     |         |                |
| email | varchar(60)  |      |     |         |                |
| sex   | enum('P','W')|      |     | P       |                |
| komentar | varchar(200) |      |     |         |                |
| date  | date         |      |     | 0000-00-00 |                |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

4.4.2 Membuat Tabel Berita

Program yang akan Kita buat akan menampilkan beberapa berita. Di dalam kolom berita, Kita dapat menginformasikan beberapa berita yang berkaitan dengan perkembangan handphone, informasi internal sistem maupun berita umum yang sedang terjadi. Rancangan tabel tersebut adalah:

Tabel 4.13 Rancangan Tabel *halamanberita*

Kolom	Tipe	Lebar	Keterangan
id_berita	INT	3	PRIMARY KEY, AUTO INCREMENT
Judul_berita	VARCHAR	100	
Detail_berita	TEXT		
Tanggal	DATE		

Untuk merealisasikan rancangan tabel di atas, Anda dapat menggunakan perintah berikut:

```
CREATE TABLE halamanberita (
  id_berita int(3) NOT NULL auto_increment,
  judul_berita varchar(100) NOT NULL default '',
  detail_berita text NOT NULL,
  tanggal date NOT NULL default '0000-00-00',
  PRIMARY KEY (id_berita)
)
```

Hasil dari query di atas dapat Anda lihat dengan menggunakan perintah DESC. Berikut adalah tampilan deskripsinya:

```
mysql> DESC halamanberita;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default        | Extra          |
+-----+-----+-----+-----+-----+-----+
| id_berita      | int(3)        |      | PRI | NULL           | auto_increment |
| judul_berita   | varchar(100)  |      |     |                 |                |
| detail_berita  | text          |      |     |                 |                |
| tanggal       | date          |      |     | 0000-00-00     |                |
+-----+-----+-----+-----+-----+
4 rows in set (0.05 sec)
```

Sebagai contoh beritanya, sekarang Anda dapat memasukan beberapa baris berita pada tabel *halamanberita* dengan menggunakan query berikut:

```
INSERT INTO halamanberita
VALUES (1,'Gempa Bumi Melanda Nias',
'Baru beberapa hari masyarakat di Pulau Nias merasakan
ketenangan, sekarang ALLAH telah menguji kembali dengan
terguncangnya Gempa yang bertenaga 8,2 SL',
'2005-03-30');

INSERT INTO halamanberita
VALUES (2,'Keterlambatan Bantuan',
'Sampai berita ini dionlinekan, bantuan yang berasal dari
pemerintah masih belum dapat didistribusikan. Hal ini karena
tidak adanya transportasi disana',
'2005-03-30');

INSERT INTO halamanberita
VALUES (7,'INDOSAT meluncurkan Mentari Hebat',
'Mentari hebat merupakan sebuah terobosan baru yang dikeluarkan
oleh mentari dalam melayani pelanggannya. Dengan
menggunakan mentari hebat, maka pelanggan dapat melakukan
call dengan biaya hemat sampai dengan 60%.',
'2005-04-07');
```

CONTOH EMAIL PEMESANAN	
TUJUAN (To)	: bunafit_komputer@yahoo.com
SUBJEK (Subject)	: Pesan SC PHP : Toko Herbal
ISI SURAT	: SURAT PESANAN DI SINI IDENTITAS PEMESAN (Untuk Arsip) : 1. Nama lengkap : ? 2. Kuliah di mana : ? 3. Nomor HP aktif (Untuk unik transfer) : ?

Untuk mendapatkan No Rekening transfer, silahkan kirim email seperti contoh di atas. Atau, kirim SMS ke nomor [0838 6716 1314](tel:083867161314).

Contoh pesan SMS v1 : "planetKode, minta No Rekening buat pembayaran Sc PHP Web Toko HaPe v1 phpTriad ". Kami akan segera membalas.

Contoh pesan SMS v2 : "planetKode, minta No Rekening buat pembayaran Sc PHP Web Toko HaPe v2 XAMPP ". Kami akan segera membalas.