Bab 3

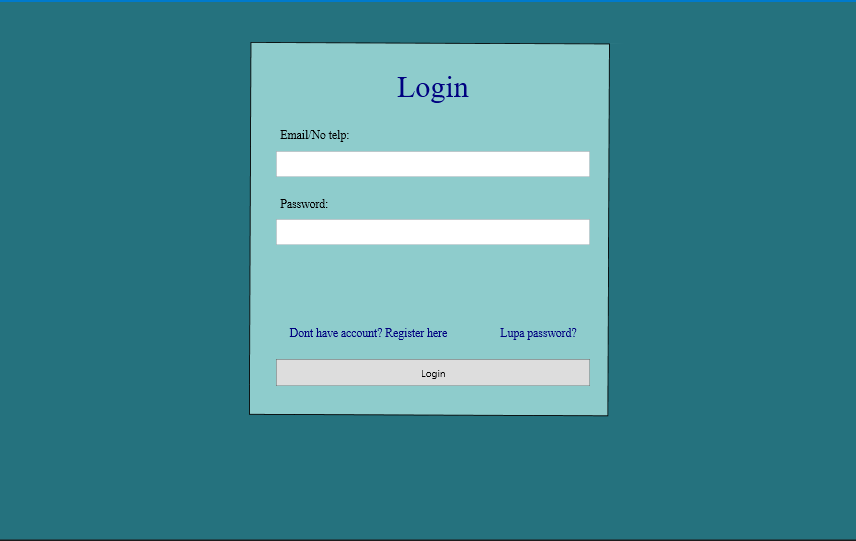
System Desain

Di bab ketiga ini, kami membuat berbagai desain Berisi; Sistem desain Database, Entity Relationship Diagram serta desain procedural. Proses mendefinisikan elemen dalam bab ini Sistem, arsitektur, komponen, dan datanya berbasis permintaan Didefinisikan dalam bab sebelumnya. Ini adalah proses definisi, Mengembangkan dan merancang sistem yang memenuhi kebutuhan

3.1 Desain Interface

Desain Interface adalah desain awal dari tampilan antarmuka Aplikasi untuk memudahkan pemrosesan lebih lanjut. Desain ini Dibuat dengan aplikasi Microsoft Visual 2019.berikut Ini adalah tampilan prototipe Aplikasi ALTERNATECH

1. Tampilan Awal



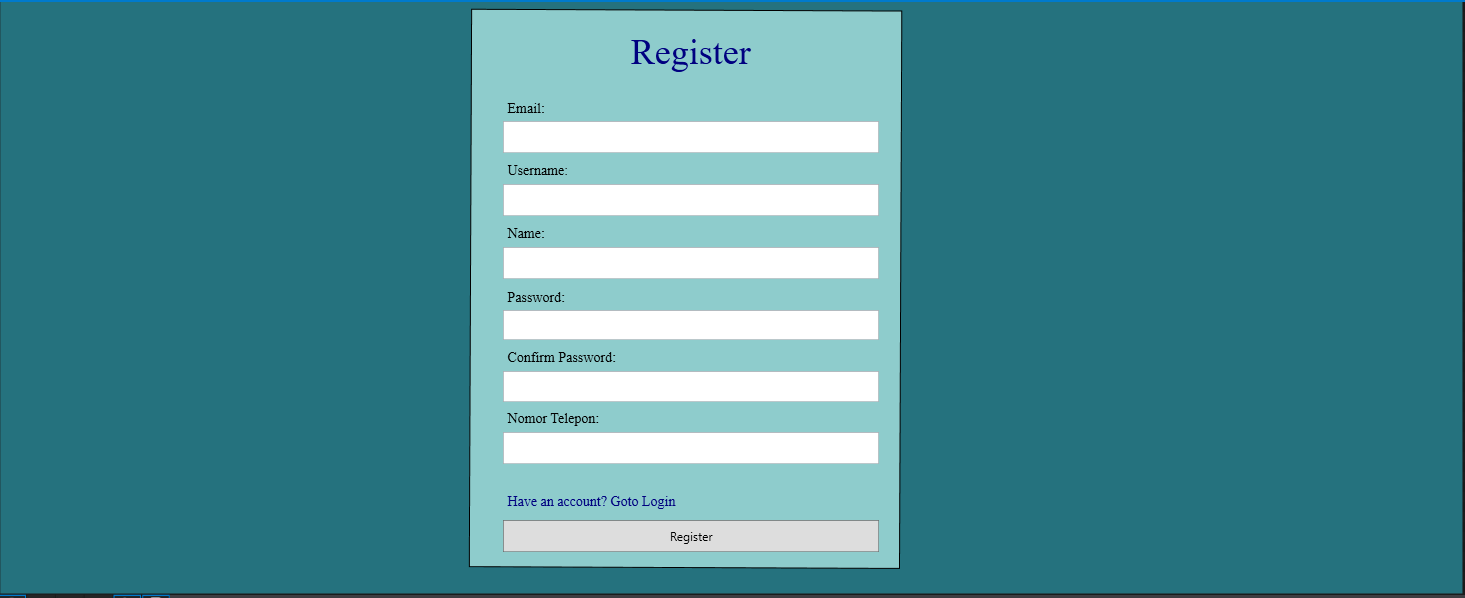
**Gambar 3.1**

**Login**

Gambar 3.1 adalah tampilan yang menunjukkan layar saat memasuki program ALTERNATECH.Direncanakan untuk masuk ke bagian user page, harus masuk dengan nama pengguna: <Email> dan password: <password>.

Untuk login sebagai Admin maka pada awal pembuatan client akan kami mengatur username dan password untuk mengakses Admin sesuai keinginan client. Setiap User memiliki role masing-masing yang dapat dibuat ketika membuat akun user itu sendiri pada halaman register.

1. Tampilan Register

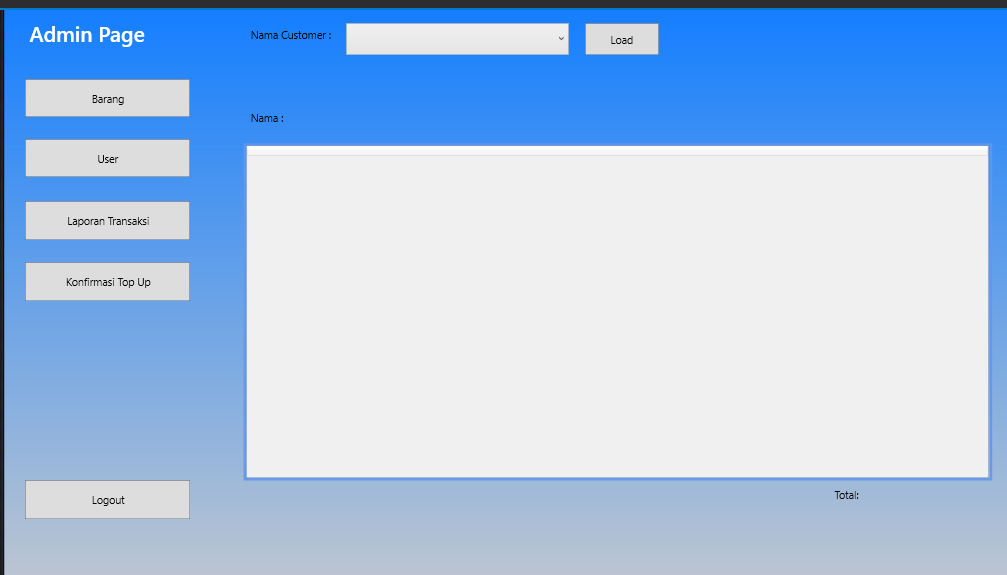


**Gambar 3.2**

**Register**

Gambar 3.2 adalah tampilan yang menunjukkan layer saat memasuki halaman register. Untuk mendaftarkan user agar bisa login dibutuhkan Email ,username ,nama ,password dan no telp untuk data diri. Data diri tersebut akan tercatat dan tersimpan dalam database untuk proses login serta pengubahan data diri user seperti role user,Untuk default role user akan diarahkan ke buyer.

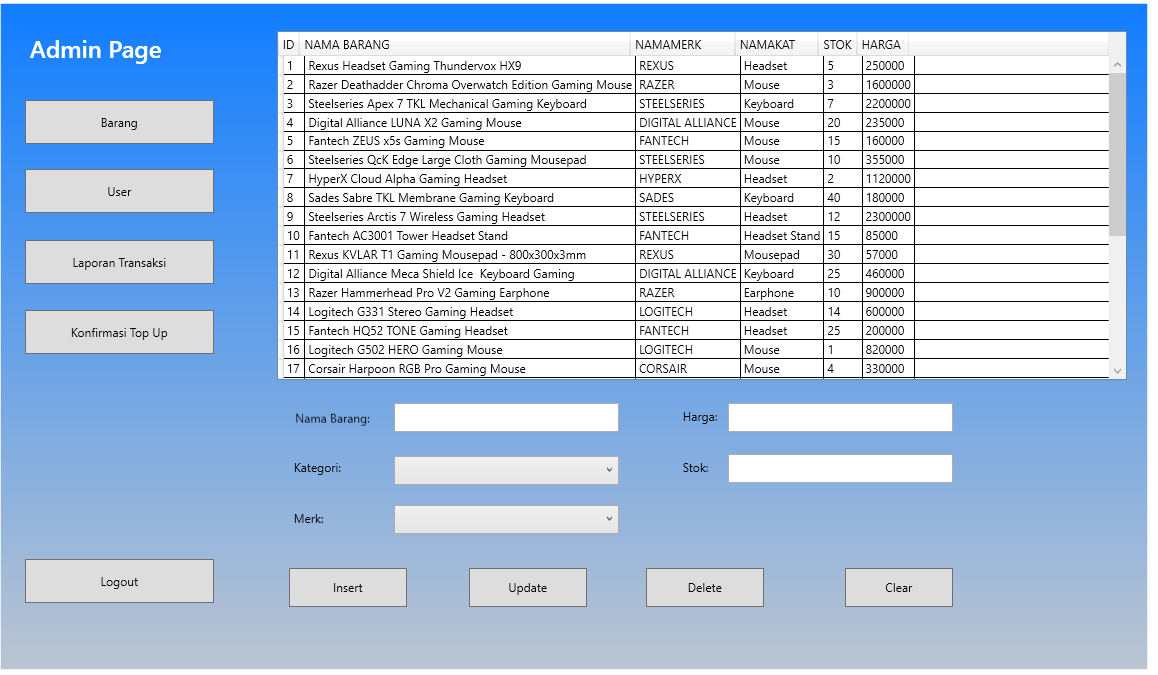
1. Tampilan Admin



**Gambar 3.3**

**Tampilan Awal Admin**

Gambar 3.3 adalah tampilan awal saat client memasuki halaman admin, disini terdapat 5 tombol navigasi yang digunakan untuk melihat list serta laporan user dan barang, list dan laporan tersebut ditampilkan di kotak besar tersebut dan terdapat fitur search yang hanya tampil Ketika client menekan button User. saat button Barang ditekan maka tampilan tersebut menjadi seperti gambar 3.4 dibawah ini



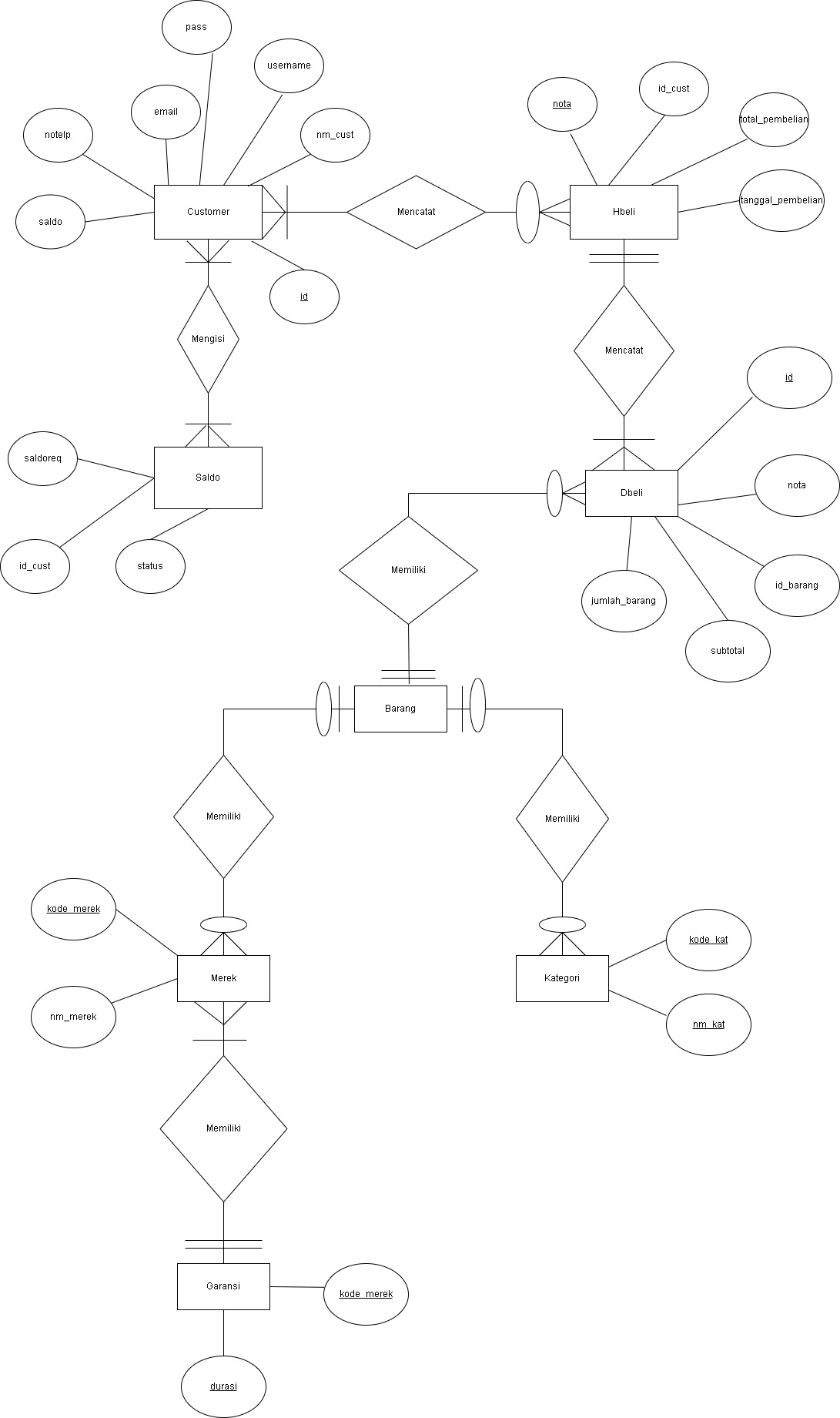
**Gambar 3.4**

**Tampilan Admin Barang**

Ditampilan barang ini terdapat fitur Create , update , delete barang yang digunakan untuk mengubah status barang , jika client ingin memasuki barang hanya mengisi field nama barang , kategori , merek serta harganya , jika client ingin mengubah atau update , client harus menekan dua kali ke salah satu barang didalam kotak besar maka akan tertampil secara langsung di dalam field serta tombon insert menjadi disable lalu fitur delete ini juga serupa , lalu client hanya menekan tombol update untuk mengubah data barang serta delete untuk menghapus barang.

3.2 Desain Database

Pada Software ELREORS, akan dilakukan pencatatan pemesanan dan pembayaran, daftar menu, dan juga pegawai yang melayani setiap pemesanan customer. Entity pada ELREORS ada delapan buah, yakni Customer, Hbeli, Dbeli , Kategori, Merk, Barang, Saldo dan juga Garansi. Berikut adalah Diagram ERD ELREORS



Gambar 3.2

Entity Relationship Diagram

Ketika customer melakukan pembelian, pembelian akan dicatat dan ditampung di detail pembelian. Barang yang sudah dibeli oleh customer akan mendapatkan nota. Barang-barang tersebut memiliki banyak kategori dan merek. Berikut adalah detail dari tabel-tabel yang dihasilkan dari ERD tersebut

Tabel 3.1

Customer

|  |  |  |
| --- | --- | --- |
| Nama | Data Type | Keterangan |
| Id | INT | Primary Key, Auto increment |
| nm\_cust | Varchar(255) |  |
| username | Varchar(255) |  |
| email | Varchar(255) |  |
| pass | Varchar(255) |  |
| no telp | INT |  |
| saldo | INT |  |

Tabel 3.1 Merupakan Tabel Customer yang digunakan untuk menyimpan data customer. Tabel ini berisikan ID yang menjadi Primary Key serta auto increment, username,email,password,notelp serta saldo

Tabel 3.2

Barang

|  |  |  |
| --- | --- | --- |
| Nama | Data Type | Keterangan |
| id | INT | Primary Key, Auto increment |
| nm\_barang | Varchar(255) |  |
| stock | Varchar(255) |  |
| harga | Varchar(255) |  |
| merk(ref.merk) | INT | Foreign Key (Merek) |
| kategori(ref.kategori) | INT | Foreign Key (Kategori) |

Tabel 3.2 Merupakan Tabel Barang yang digunakan sebagai database dari data-data

sebuah barang. Tabel ini berisikan ID yang menjadi Primary Key serta auto increment , nm\_barang, stock, harga,merek yang merupakan foreign key yang digunakan untuk relasi Tabel barang dengan tabel merek dan kategori juga memiliki foreign key yang digunakan untuk relasi Tabel kategori dengan Tabel Barang

Tabel 3.3

Merek

|  |  |  |
| --- | --- | --- |
| Nama | Data Type | Keterangan |
| kode merek | Varchar(5) | Primary Key |
| nm\_merk | Varchar(255) |  |

Tabel 3.3 merupakan tabel merek yang digunakan sebagai database Relasi dari barang dan garansi. Tabel ini berisikan kode merek yang menjadi Primary Key serta nm\_merk.

Tabel 3.4

Dbeli

|  |  |  |
| --- | --- | --- |
| Nama | Data Type | Keterangan |
| id | Varchar(255) | Primary Key |
| nota | Varchar(255) | Foreign Key(Hbeli) |
| Id\_barang(fk barang) | INT | Foreign Key(Barang) |
| jumlah barang | INT |  |
| subtotal | INT |  |

Tabel 3.4 merupakan table Dbeli yang digunakan menampilkan detail seperti barang apa saja yang dibeli beserta jumlah dan harganya. Tabel ini berisikan id sebagai primary key , nota merupakan foreign key yang digunakan untuk relasi antara tabel Dbeli dengan tabel hbeli , id\_barang merupakan foreign key yang digunakan untuk relasi antara tabel Dbeli dengan tabel barang , jumlah barang serta sub\_total

Tabel 3.5

Hbeli

|  |  |  |
| --- | --- | --- |
| Nama | Data Type | Keterangan |
| nota | Varchar(255) | Primary Key |
| id\_cust(fk customer) | INT | Foreign Key(Customer) |
| total pembelian | INT |  |
| tanggal pembelian | Date |  |

Tabel 3.5 merupakan table Hbeli yang digunakan untuk menampilkan detail total pembelian user dan tanggal pembelian barang tersebut. Tabel ini berisikan nota sebagai primary key , id\_cust merupakan foreign key yang digunakan untuk relasi antara tabel Hbeli dengan tabel Customer, total\_pembelian, tanggal\_pembelian

Tabel 3.6

Garansi

|  |  |  |
| --- | --- | --- |
| Nama | Data Type | Keterangan |
| kode merk(fk merk) | Varchar(255) | Foreign Key(Merek) |
| durasi(hari) | INT |  |

Tabel 3.6 merupakan table Garansi yang digunakan untuk

Tabel 3.7

Kategori

|  |  |  |
| --- | --- | --- |
| Nama | Data Type | Keterangan |
| kode\_kat | Varchar(255) | Primary Key |
| nm\_kat | Varchar(255) |  |

Tabel 3.7 merupakan table Kategori yang digunakan untuk menyimpan data kategori serta database Relasi dari tabel kategori dan tabel barang untuk mengkategorikan barang. Tabel ini berisikan kode\_kat sebagai primary key , nm\_kat

Tabel 3.8

Saldo

|  |  |  |
| --- | --- | --- |
| Nama | Data Type | Keterangan |
| id\_cust(fk customer) | INT | Foreign Key(Customer) |
| saldoreq | INT |  |
| status | INT | (1pending 2accepted 3rejected) |

Tabel 3.8 merupakan table saldo yang digunakan untuk melihat data saldo user serta menyetujui atau menolak request saldo user. tabel ini berisikan id\_cust sebagai foreign key yang digunakan untuk relasi antara tabel Saldo dengan tabel Customer,saldoreq serta status yang dibagi menjadi 3 jenis yaitu 1 = pending , 2 = accepted , 3 = rejected

3.3. Desain Procedural

Pada point ini kita menjelaskan algoritma pada aplikasi yang kita kembangkan pada aplikasi ini,berikut adalah algoritma yang kami kembangkan pada aplikasi kami

Algoritma 3.1 Contoh Algoritma Filtering barang

1. If kategori[] is equal “all” then
2. Print filter barang untuk all success
3. Elseif kategori[] is not equal “all”
4. Print filter barang untuk kategori[] success
5. End if

Algoritma 3,2 Contoh Algoritma Login

1. If email is not Null and password is not null then
2. If email is equal admin and password is equal admin then
3. Print login sebagai admin
4. Elseif email is equal user and password is equald user then
5. Print login sebagai User
6. End if
7. Else then
8. Print “data fill blank”
9. End if

Algoritma 3.3 contoh Algoritma Register

1. If email is not null and username is not null and pass is not null then
2. If pass is equal passConfirm then
3. Print Register Berhasil
4. Else then
5. Print Password not match with Confirm Password
6. End if
7. Else then
8. Print data is fill blank
9. End if

Algoritma 3.4 contoh Algoritma CRUD

Insert

1. query = “”;
2. If kat is not null or merk is not null then
3. Call getKodekat()
4. Call getKodemerk()
5. Query = “insert into BARANG(NAMA\_BARANG, MERK, KATEGORI,STOK,HARGA)”+“values('{nm\_barang}','{kodemerk}','{kodekat}',{stock},{harga})”
6. Print Update barang berhasil
7. Else then
8. Print fill blank !!
9. End if

Delete

1. Query = “”
2. If kat is not null or merk is not null then
3. Query = “delete from barang where ID = {index+1}”
4. Print Delete barang berhasil
5. Else then
6. Print fill blank !!
7. End if

Update

1. query = “”;
2. If kat is not null or merk is not null then
3. Query = “Update Barang SET NAMA\_BARANG = '{ nm\_barang }',MERK = '{ kodemerk }',kategori = '{ kodekat }',STOK = {stock},HARGA = {harga} Where id = " + (index+1)”
4. Print Update barang berhasil
5. Else then
6. Print fill blank !!
7. End if