

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Peralatan Pendukung**

Peralatan pendukung sistem informasi manajemen terdiri dari perangkat keras (*hardware*) dan perangkat lunak (*software*). Perangkat yang digunakan dalam penelitian ini adalah sebagai berikut:

#### **3.2 Perangkat Keras**

Perangkat keras yang digunakan dalam mengembangkan sistem ini antara lain:

1. *Laptop*
2. *Mouse*

#### **3.3 Perangkat Lunak**

Adapun perangkat lunak yang dibutuhkan dalam membangun *website* penjualan Horn Music Merch adalah:

1. *Operating System Windows 8*
2. *MySQL 5.6.21*
3. *Text Editor Sublime Text 2*
4. *Microsoft Visio 2013*

#### **3.4 Bahan Penelitian**

##### **3.4.1 Studi Literatur**

Mempelajari buku, teori dan referensi yang berhubungan dengan manajemen berbasis *web* antara lain prinsip dan prosedur DBMS, pemodelan data meliputi *Flowchart*, DFD pemrograman dengan *PHP* dan database *MySQL*. *Internet* juga diperlukan untuk menunjang pencarian informasi yang berkaitan dengan objek penelitian.

##### **3.4.2 Wawancara**

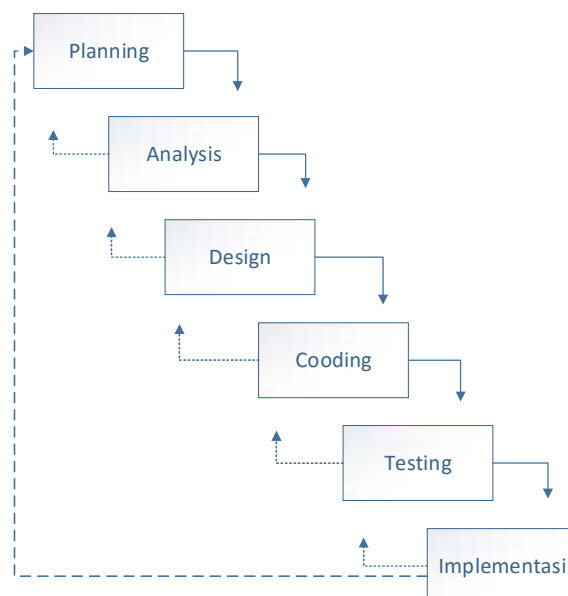
Wawancara dilakukan dengan mewawancarai secara langsung pihak terkait, yang berguna untuk mendapatkan informasi maupun data-data yang dibutuhkan untuk perancangan dan pembangunan yang akan dibuat. Wawancara terhadap pihak

HORN MUSIC MERCH DISTRO dengan melakukan wawancara kepada Saudara Muhammad Iqbal Risky sebagai pemilik usaha maupun penanggung jawab pemesanan yang berguna untuk mendapatkan data maupun informasi.

### 3.4.3 Observasi

Pada metode pengamatan (observasi), dilakukan peninjauan dan penelitian langsung di lapangan untuk memperoleh dan mengumpulkan data yang dibutuhkan. Pengamatan di HORN MUSIC MERCH DISTRO yang berlokasi di Yogyakarta. Adapun penelitian ini dilaksana tanggal 30 April 2016 sampai dengan 1 May 2016.

### 3.5 Langkah Penelitian



**Gambar 3.1.** Metode *Waterfall*

a. *Planning*

Tahap perancangan menyangkut studi tentang kebutuhan pengguna, kelayakan baik secara teknik maupun teknologi. Tahap ini dilakukan perancangan tentang sistem yang akan dibangun.

b. *Analysis*

Tahap analisis, merupakan proses pendalaman mengenai segala permasalahan dan resiko pada pengguna.

c. *Design*

Tahap perancangan menyangkut perancangan sistem dimana kita akan memberikan solusi dari masalah yang muncul pada tahap analisis.

d. *Cooding*

Tahap implementasi, adalah dimana perancangan sistem implementasikan ke situasi nyata dengan pemilihan perangkat keras dan penyusunan desain (*cooding*). Untuk implementasi yaitu dengan memasukkan desain yang sudah dibuat ke dalam *bootstrap* yang diinstal pada *server* yang disediakan oleh pihak Horn Music Merh Distro.

e. *Testing*

*Testing* adalah tahap yang menentukan apakah desain yang sudah dibuat sudah sesuai dengan kebutuhan pengguna atau belum. Tujuan *testing* ini adalah untuk meminimalisir cacat desain *website* sehingga sistem yang dikembangkan benar-benar dapat berjalan dengan sebaik mungkin.

f. *Implementasi*

*Implemtasi* adalah tahap dimana dilakukannya perawatan dan pemeliharaan *website*. Jika diperlukan akan dilakukan perbaikan kecil kemudian jika periode sistem sudah habis akan masuk lagi pada tahap perancangan.

### **3.6 Sistem Sedang Berjalan di DISTRO**

#### **3.6.1 Analisis Pembelian**

1. Calon *member* harus mempunyai akun.
2. Setelah calon *member* melakukan *registrasi* akan mendapatkan balasan konfirmasi di *email*.
3. Setelah *member* membuka *notifikasi* aktifasi pada *email member* sudah bisa melakukan pembelian produk secara *online*.
4. *Member* memilih produk yang sudah disediakan pada *website* HORN MUSIC MERCH DISTRO.
5. Jasa pengiriman barang dipilih oleh *member*.
6. Pembayaran dilakukan secara *online*.

7. Setelah *member* melakukan pembayaran, diwajibkan mengupload bukti *transfer*.
8. Jika *member* telah mengupload bukti pembayaran yang benar, maka *admin* akan melakukan konfirmasi pemesanan.
9. Barang yang dipesan oleh *member* maka akan segera dilakukan pengiriman oleh *admin*.

### **3.6.2 Analisis Proses Laporan**

1. *Admin* melakukan pengecekan terhadap laporan penjualan dan laporan stok barang.
2. *Admin* dapat melakukan penarikan laporan penjualan perminggu dan perbulan.

## **3.7 Analisis Kebutuhan Fungsional dan Non Fungsional**

### **3.7.1 Kebutuhan Fungsional**

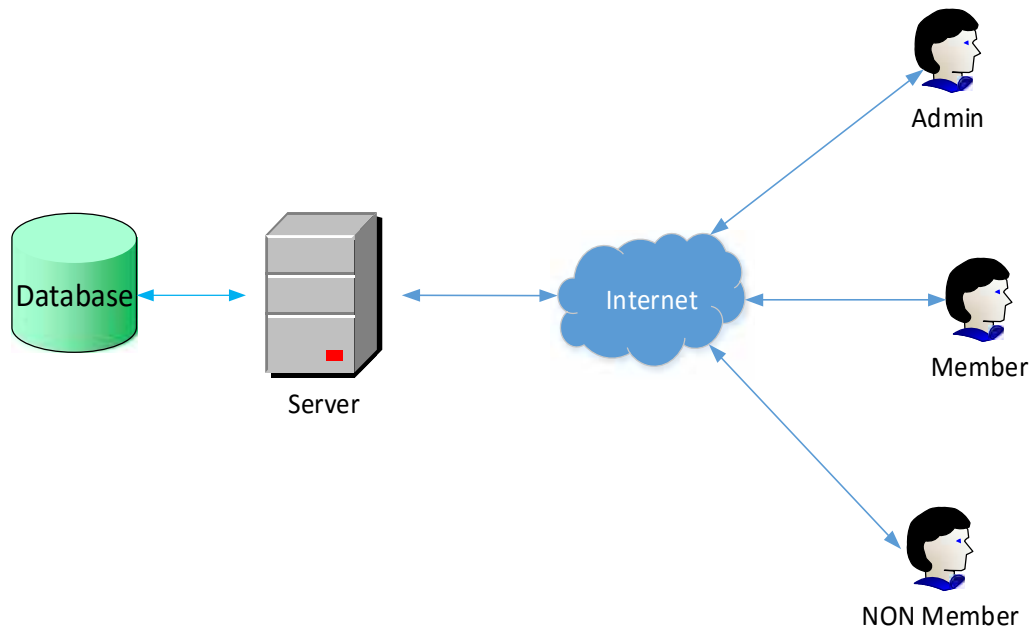
1. Data barang.
2. Kategori barang.
3. *List* barang.
4. Data pesanan.
5. *Slide benner*.
6. Data member.
7. Laporan perbarang.
8. Laporan pertransaksi.

### **3.7.2 Kebutuhan Non Fungsional**

1. Proses pembayaran dilakukan maksimal 1x24jam.
2. Laporan penjualan dan laporan stok barang ditampilkan dengan ketentuan pilihan.

## **3.8 Perancangan Sistem**


Pada tahap ini dilakukan penentuan perancangan sistem yang akan digunakan. Berikut ini merupakan gambar perancangan sistem yang diusulkan:




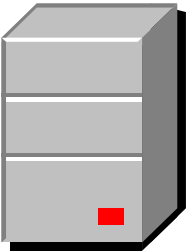
**Gambar 3.2.** Arsitektur Sistem yang Diusulkan

*Database server* yang digunakan menggunakan pada aplikasi *MySql* dan menggunakan *xampp* sebagai *web server*. Komunikasi antara pengguna dan *web server* menggunakan *internet* dan *web browser* pada perangkat pengguna. Saat pengguna mengakses aplikasi, *web server* memuat antar muka dan melakukan pengambilan data yang dilakukan dari *database server*. Melalui antar muka yang dimuat *web server* sebagai pengguna bisa menyimpan *database server*.

**Tabel 3.1.** Keterangan komponen Arsitektur.

	<p>Admin Member Non Member</p>
---	--

**Tabel 3.1.** Keterangan Komponen Arsitektur (Lanjutan)

 <p style="text-align: center;">Database</p>	<p>Berisi data-data sebagai berikut:</p> <ol style="list-style-type: none"> <li>1. Data barang.</li> <li>2. Kategori barang.</li> <li>3. <i>List</i> barang .</li> <li>4. Data pesanan .</li> <li>5. <i>Slide benner</i>.</li> <li>6. Data <i>member</i>.</li> <li>7. Laporan perbarang.</li> <li>8. Laporan pertransaksi.</li> </ol>
 <p style="text-align: center;">Server</p>	<p><i>Server</i> digunakan untuk penyimpanan data-data tersebut.</p>

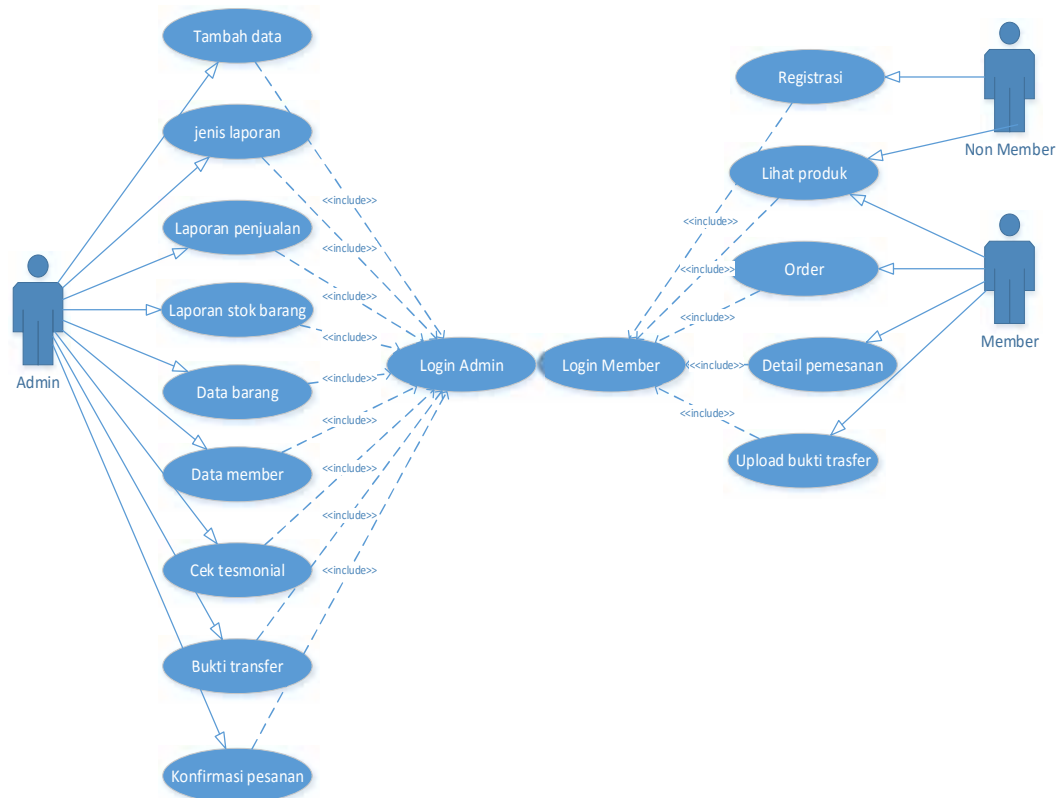
### 3.9 Perancangan Proses

Metode perancangan yang digunakan dalam aplikasi adalah *United Markup Language* (UML). Seperti yang sudah dijelaskan sebelumnya, metode UML yang dipakai pengembangan aplikasi yaitu model *Use Case Diagram*, *Activity Diagram*, dan *Class Diagram*.

#### 3.9.1 *Use Case Diagram*

*Use case* diagram merupakan salah satu bagian dari UML, *use case* diagram adalah suatu diagram yang menggambarkan suatu sistem dan bagaimana sistem tersebut bekerja. Perancangan *use case* digunakan untuk memodelkan proses berdasarkan perspektif *user*. *Use case* diagram terdiri atas diagram untuk *use case*

dan *actor*. *Use case* diagram menggambarkan orang yang akan mengoperasikan atau berinteraksi dengan sistem. Berikut ini adalah *use case* sistem di DISTRO:



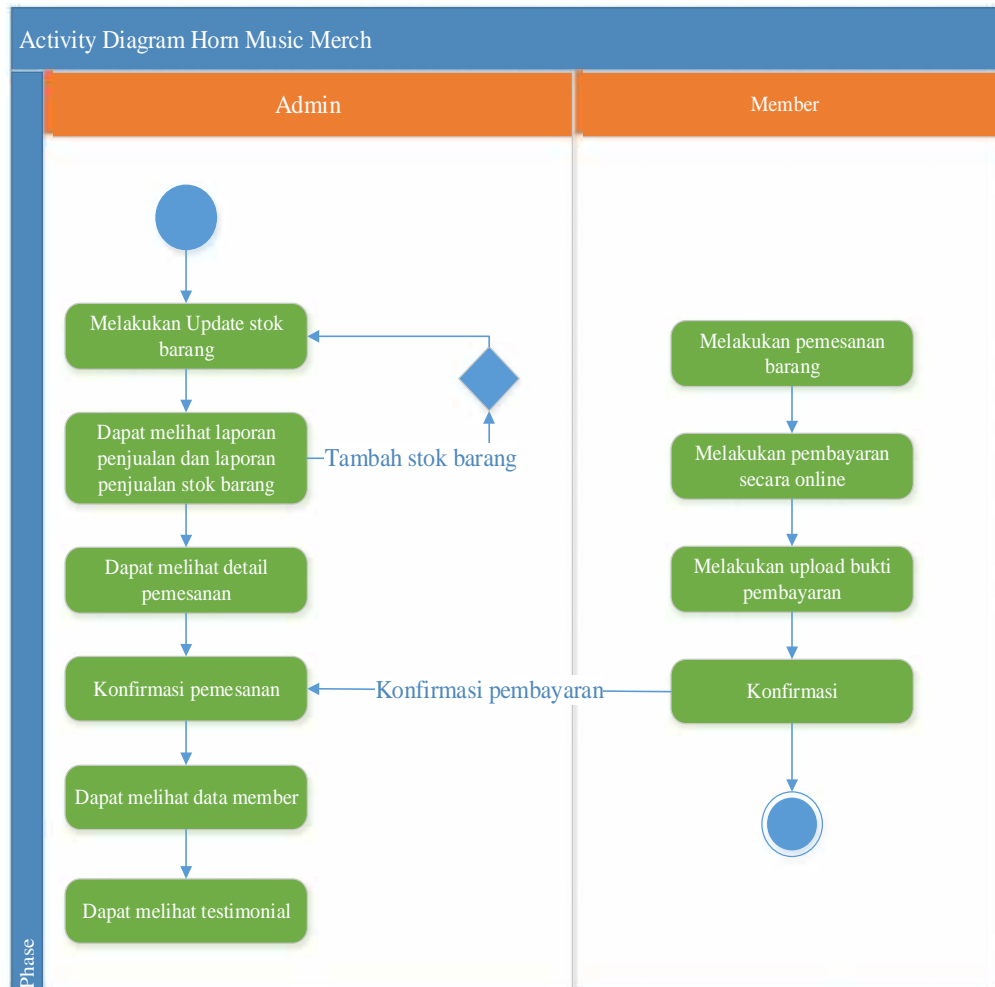
**Gambar 3.3.** *Use Case* Diagram

Berikut penjelasan tentang Gambar3.3:

1. Terdapat 3 aktor pada *use case* diagram aplikasi yakni *admin*, *member*, dan *non member*.
2. *Admin* berhubungan langsung pada *update* data barang seperti *list* barang dan stok barang sehingga dapat melakukan pengelolaan data produk.
3. Berdasarkan barang yang dibeli *member*, maka dapat melakukan pembayaran secara *transfer* dan melakukan *upload* bukti *transfer*.
4. *Non member* hanya dapat melihat produk yang ada pada *website* dan dapat registrasi *member* baru.

### 3.9.2 Activity Diagram

Gambar *activity* diagram yang digunakan dalam aplikasi dapat dilihat pada Gambar 3.4.



**Gambar 3.4.** Activity diagram Aplikasi

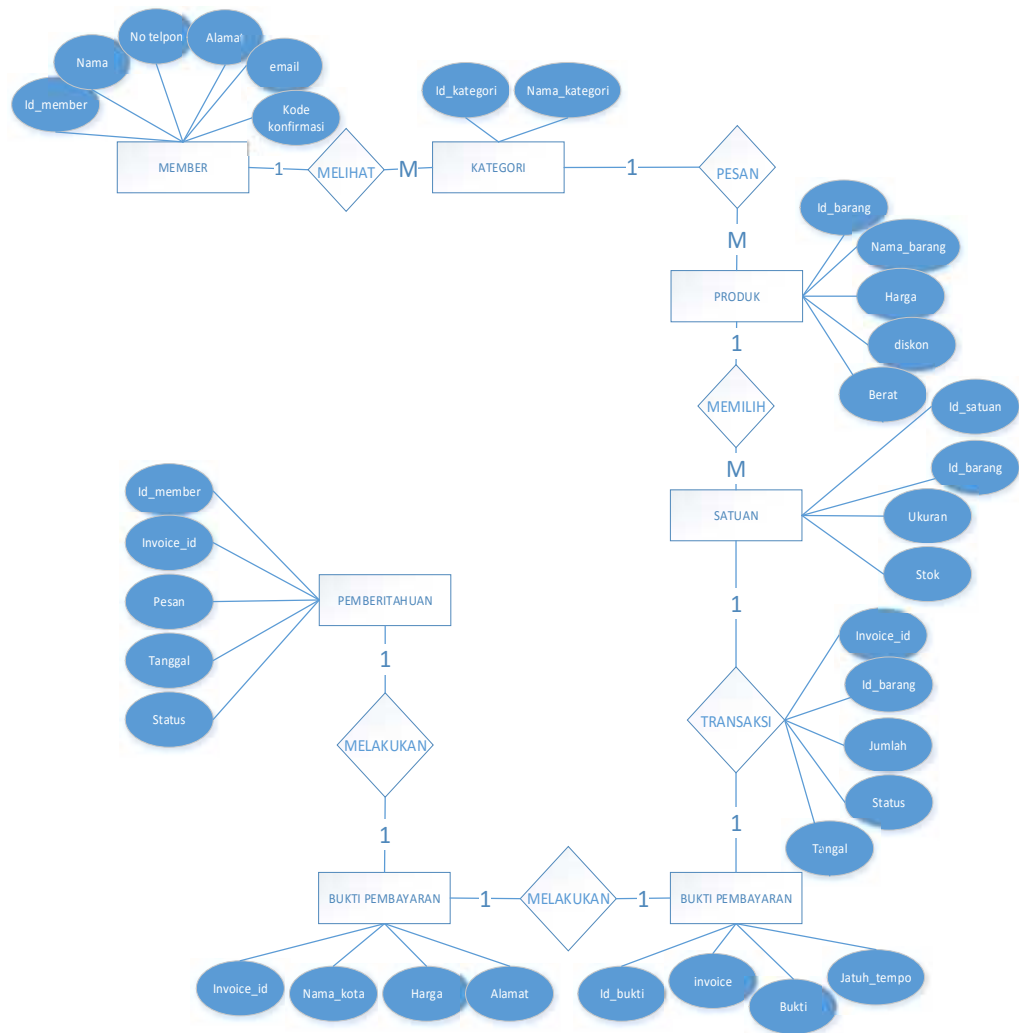
Gambar 3.4 menunjukkan *Activity diagram* pada kegiatan alur proses penjualan. *Admin* dapat melihat stok barang yang tersedia maupun sudah stok habis, jika stok barang habis maka *admin* dapat melakukan *update* stok barang, jika stok barang masih ada maka tidak ada perubahan.

### 3.10 Perancangan Basis Data

#### 3.10.1 Entity Relationship Diagram (ERD)

Gambar berikut ini menjelaskan hubungan relasi antar data dalam basis data yang terdapat di dalam sistem DISTRO:





**Gambar 3.5.** Entity Relationship Diagram (ERD)

Berikut penjelasan tentang gambar 3.5:

1. Pada entitas *member* memiliki relasi dengan entitas kategori *one to many*, artinya *member* dapat memesan banyak kategori yang tersedia di *website*.
2. Pada entitas kategori memiliki relasi dengan entitas produk *one to many*, artinya dapat melakukan banyak transaksi pembelian, dan sebaliknya banyak barang yang dapat dibeli oleh kategori.
3. Pada entitas produk memiliki relasi dengan entitas satuan *many to many*, artinya produk dapat banyak memilih dan membeli ukuran dan stok barang.

4. Pada entitas satuan memiliki relasi dengan entitas transaksi *one to one*, artinya satu pembelian memiliki satu *invoice id* beserta satu *id* bukti pembayaran.
5. Pada entitas produk memiliki relasi dengan entitas bukti pembayaran *one to one*, artinya satu transaksi memiliki satu produk.
6. Pada entitas pengiriman memiliki relasi dengan entitas pemberitahuan *one to one*, artinya dalam satu pesanan barang akan mendapat konfirmasi pengiriman barang.

**Tabel 3.2.** Struktur Tabel *Admin*

No	Name	Type	Null	Extra
1.	Id_admin	Int(11)	NN	<i>Auto_Increment</i>
2.	Nama	Varchar(30)	NN	
3.	<i>Username</i>	Varchar(30)	NN	
4.	<i>Password</i>	Varhcar(40)	NN	
5.	<i>Email</i>	Varhcar(40)	NN	
6.	Foto	Varchar(30)	NN	

**Tabel 3.3.** Struktur Tabel *Member*

No	Name	Type	Null	Extra
1.	Id_member	Int(11)	NN	Auto_Increment
2.	Nama	Varhcar(40)	NN	
3.	Hp	Varhcar(14)	NN	
4.	Alamat	<i>Text</i>	NN	
5.	<i>Username</i>	Varchar(30)	NN	
6.	<i>Email</i>	Varchar(40)	NN	
7.	<i>Password</i>	Varchar(40)	NN	
8.	Show_password	Varchar(30)	NN	
9.	Tanggal_daftar	Date	NN	
10.	Kode_konfirmasi	Varchar(100)	NN	
11.	Status	Int(1)	NN	

**Tabel 3.4.** Struktur Tabel *Barang*

No	Name	Type	Null	Extra
1.	Id_barang	Int(11)	NN	Auto_Increment
2.	Nama_barang	Varchar(40)	NN	
3.	Harga	Double	NN	
4.	Kategori	Int(11)	NN	
5.	Dilihat	Int(11)	NN	
6.	Diskon	Int(3)	NN	
7.	Harga_baru	Double	NN	
8.	Berat	Double	NN	

**Tabel 3.5.** Struktur Tabel Kategori

No	Name	Type	Null	Extra
1.	Id_kategori	Int(11)	NN	Auto_Increment
2.	Nama_kategori	Varchar(40)	NN	

**Tabel 3.6.** Struktur Tabel Satuan

No	Name	Type	Null	Extra
1.	Id_satuan	Int(11)	NN	Auto_Increment
2.	Id_barang	Int(11)	NN	
3.	Ukuran	Varchar(4)	NN	
4.	Stok	Int(4)	NN	

**Tabel 3.7.** Struktur Tabel Order

No	Name	Type	Null	Extra
1.	Id_order	Int(11)	NN	Auto_Increment
2.	Invoice_id	Varchar(30)	NN	
3.	Id_barang	Int(11)	NN	
4.	Id_satuan	Int(11)	NN	
5.	Jumlah	Int(3)	NN	
6.	Diskon	Int(3)	NN	
7.	Id_user	Int(3)	NN	
8.	Status	Int(11)	NN	
9.	Dilihat	Int(1)	NN	

**Tabel 3.7.** Struktur Tabel Order

No	Name	Type	Null	Extra
10.	Tanggal_terkonfirmasi	Date	NN	

**Tabel 3.8.** Struktur Tabel Pengiriman

No	Name	Type	Null	Extra
1.	Invoice_id	Varchar(30)	NN	Auto_Increment
2.	Nama_kota	Varchar(30)	NN	
3.	Paket	Varchar(50)	NN	
4.	Harga_ongkir	Double	NN	
5.	Alamat	Text	NN	
6.	Id	Int(11)	NN	

**Tabel 3.9.** Struktur Tabel Gambar

No	Name	Type	Null	Extra
1.	Id_gambar	Int(11)	NN	Auto_increment
2.	Gambar	Varchar(50)	NN	
3.	Id_gambar	Int(11)	NN	

### 3.10.2 Class Diagram

Gambaran *class diagram* yang digunakan dalam aplikasi dapat dilihat pada Gambar 3.6, dan berikut penjelasannya:

1. Kelas *login* terdapat fungsi untuk memasukkan *username* dan *password* agar *user* dapat masuk kedalam sistem.
2. Kelas *member* memiliki *association* dengan kelas kategori. Artinya setiap *member* boleh tidak melakukan pembelian atau melakukan pembelian lebih dari beberapa kali.
3. Kelas kategori memiliki *association* dengan kelas produk. Artinya setiap *member* melakukan pembelian bisa memilih kategori produk.
4. Kelas produk memiliki *association* dengan kelas satuan. Artinya *member* bisa melakukan pemilihan satuan lebih dari satu.
5. Kelas produk memiliki *association* dengan kelas transaksi. Artinya setiap *member* melakukan pembelian barang akan mendapatkan satu *invoice id*.

6. Kelas transaksi memiliki *association* dengan kelas biaya pengiriman. Artinya setiap *member* melakukan pembelian barang akan mendapatkan biaya pengiriman yang telah ditentukan.
7. Kelas biaya pengiriman memiliki *composition* dengan kelas bukti pembayaran. Artinya kelas bukti pembayaran merupakan bagian dari kelas biaya pengiriman.
8. Kelas bukti pembauaran memiliki *association* dengan kelas pemberitahuan. Artinya apabila barang telah selesai di proses, maka akan mendapatkan notifikasi.

*Class* kategori memiliki *association* dengan *class* produk, *class member* dapat mengakses apa saja yang yang terdapat di dalam *website* melalui *method* *Getidmember()*.

*Class* satuan memiliki *composition* dengan *class* produk, artinya *class* satuan merupakan bagian dari *class* produk. *Class* satuan tidak dapat berdiri sendiri apabila *class* produk tidak ada.

*Class* produk memiliki *association* dengan *class* transaksi, *class* transaksi boleh melakukan akses transaksi melalui *method* *Getinvoiceid()* dan *class* transaksi memiliki *association* dengan *class* biaya pengiriman, *class* transaksi boleh melakukan akses biaya pengiriman melalui *method* *Getinvoiceid()*.

*Class* biaya pengiriman memiliki *association* dengan *class* bukti pembayaran, *class* biaya pengiriman boleh melakukan bukti pembayaran melalui *method* *Getbukti()*.

*Class* bukti pembayaran memiliki *association* dengan *class* pemberitahuan, *class* bukti pembayaran boleh melakukan pemberitahuan melalui *method* *Getstatus()*.



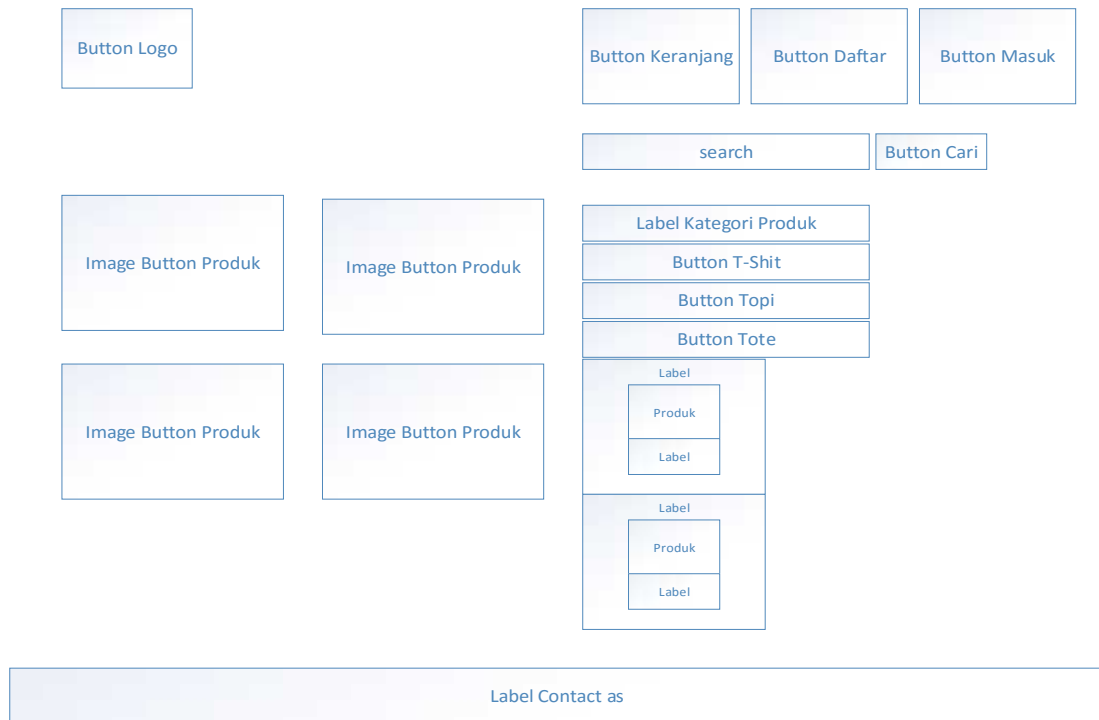
Gambar 3.6. Class Diagram Aplikasi

### 3.11 Perancangan Interface

Perancangan *interface* sistem diperlukan untuk memudahkan *user* dalam melakukan proses interaksi terhadap sistem. *Interface* menyediakan tampilan halaman sebuah sistem yang digunakan untuk proses *input* hingga menghasilkan *output* yang sesuai dengan kebutuhan. *Interface* untuk sistem yang akan dibuat adalah sebagai berikut:

### 3.11.1 Rancangan Antar Muka Halaman Utama

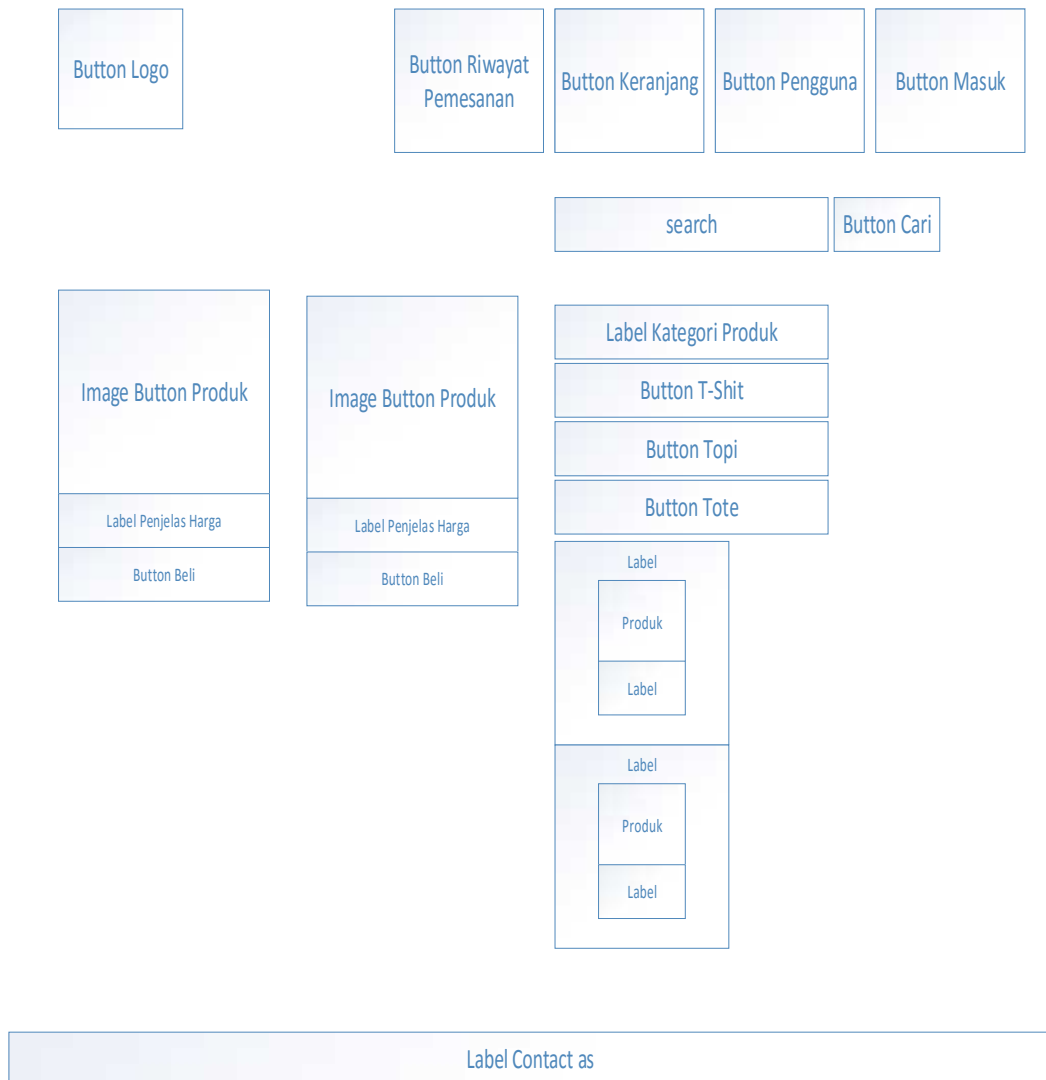
Rancangan antarmuka halaman utama merupakan tampilan utama dari aplikasi yang dapat dilihat oleh *member*, *nonmember* dan *admin*. Gambaran rancangan antarmuka halaman utama aplikasi dapat dilihat pada Gambar 3.7.



**Gambar 3.7.** Rancangan Halaman Utama

### 3.11.2 Rancangan Antarmuka Halaman Produk

Rancangan antarmuka halaman produk berisi tentang produk-produk barang yang berkaitan dengan produk barang. Dimana calon *member* dapat melihat beberapa jenis kategori produk barang yang mereka inginkan dapat dilihat pada Gambar 3.8.



**Gambar 3.8.** Halaman Utama



### 3.11.3 Rancangan Antarmuka Halaman *Member Baru*

Rancangan antarmuka halaman *member* baru berisi halaman yang berkaitan dengan ketentuan pendaftaran yang dapat dilihat oleh calon *member*. Gambaran rancangan antarmuka halaman *member* baru terdapat pada Gambar 3.9.

The diagram illustrates the layout of a new member registration page. It features a top navigation bar with buttons for 'Logo', 'Keranjang', 'Pengguna', and 'Masuk'. Below this is a search bar with a 'search' input and a 'Button Cari'. The main content area is divided into two columns. The left column contains a registration form with text fields for 'Nama Lengkap', 'Nomor Handphone', 'Nama Pengguna', 'Alamat Rumah', 'Alamat Email', 'Kata Sandi', and 'Ulangi Kata Sandi'. It also includes a 'Check Button' and a 'Label Keterangan' section, and a 'Button Daftar' at the bottom. The right column displays a list of product categories: 'Label Kategori Produk', 'Button T-Shirt', 'Button Topi', and 'Button Tote'. Each category is followed by a 'Label' containing 'Produk' and another 'Label'. At the bottom of the page is a wide 'Label Contact as' section.

**Gambar 3.9.** Rancangan Antarmuka Halaman *Member Baru*

### 3.11.4 Rancangan Antarmuka Halaman Registrasi *Aktifasi*

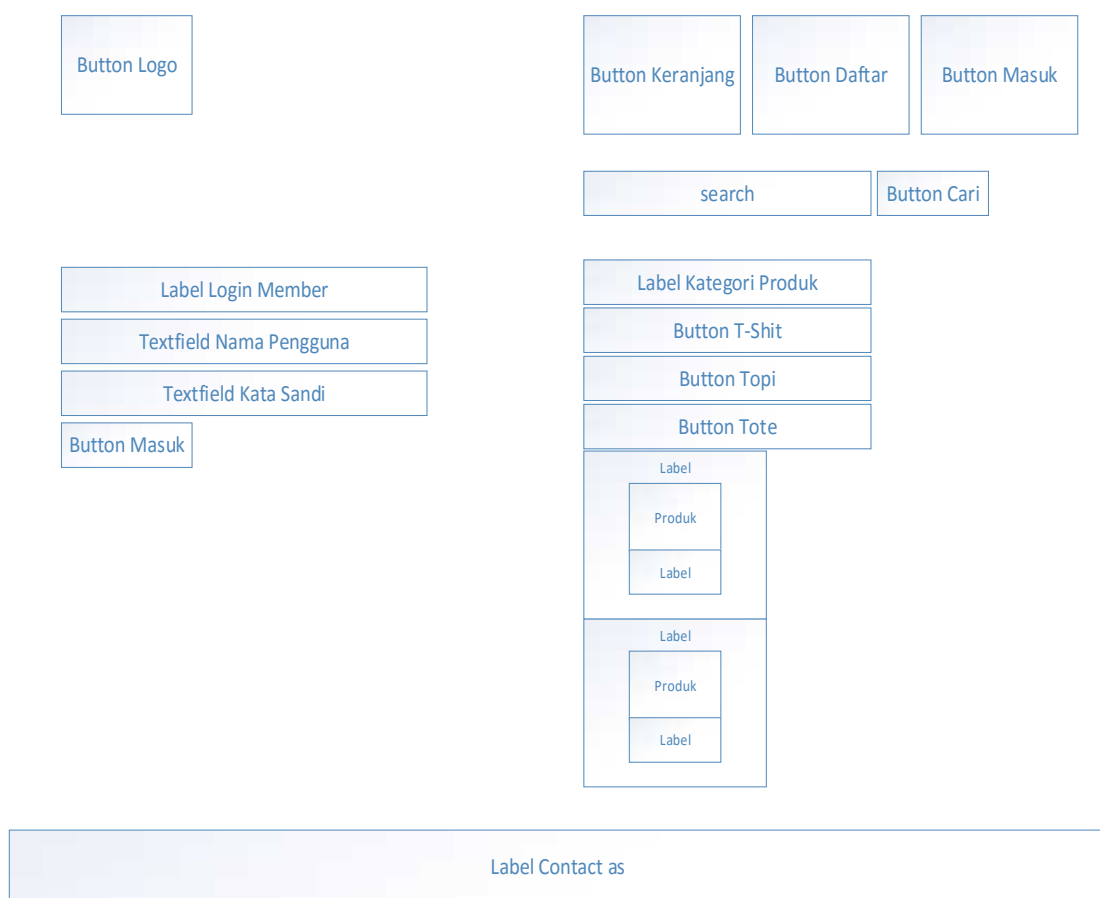
Rancangan antarmuka halaman registrasi *aktifasi* yang dapat dilihat oleh *member* ketika telah selesai melakukan pendaftaran *member* baru. Terdapat langkah selanjutnya *member* diwajibkan untuk membuka *email* agar akun bisa terverifikasin. Gambaran rancangan antarmuka halaman *member* baru dapat dilihat pada Gambar 3.10.



**Gambar 3.10.** Rancangan Antarmuka Halaman Registrasi *aktifasi*

### 3.11.5 Rancangan Antarmuka Halaman *Login Member*

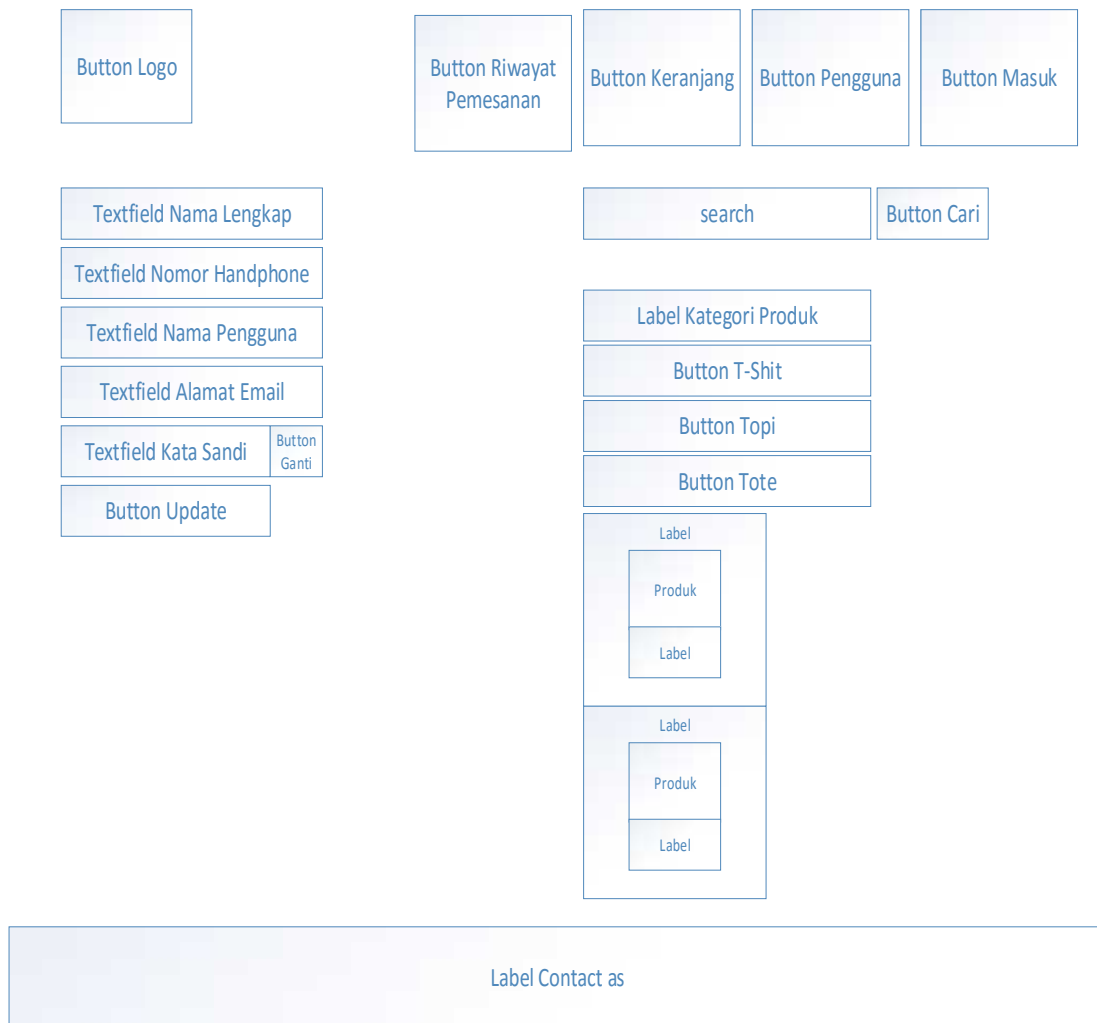
Rancangan antarmuka halaman *login member* terdapat dua kolom dan satu *button* yaitu kolom *username*, *password* dan *button* masuk. *Member* dapat *login* sesuai *username* dan *password* yang telah dibuat sebelumnya, jika *member* lupa *password member* akan di arahkan ke menu *reset password*. Gambaran rancangan antarmuka halaman *login member* terdapat pada Gambar 3.11.



**Gambar 3.11.** Rancangan Antarmuka Halaman *Login Member*

### 3.11.6 Rancangan Antarmuka Akun *Member*

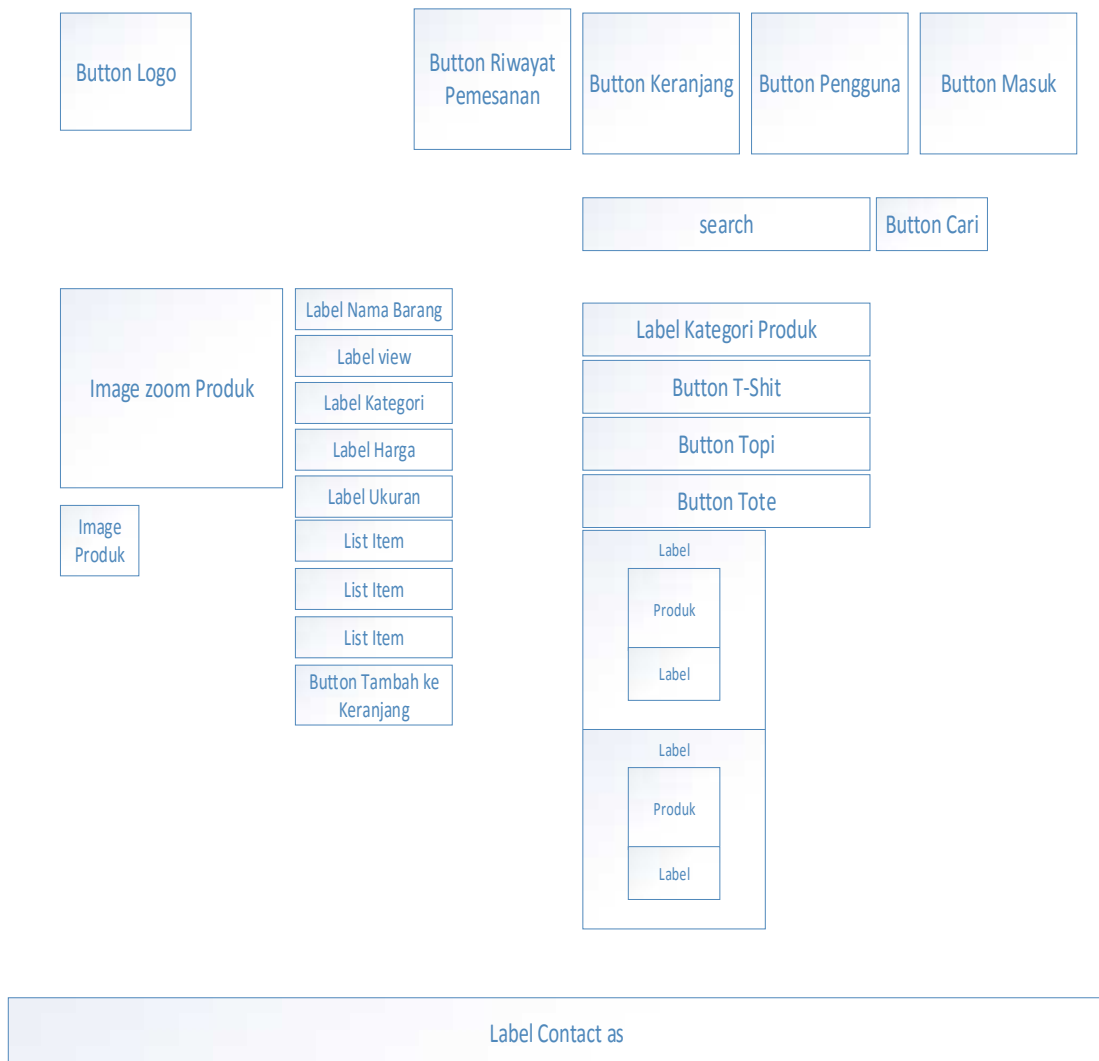
Rancangan antarmuka akun *member* terdapat pada profil *member* mengubah data yang ada sebelumnya. Gambar rancangan antarmuka akun *member* terdapat pada Gambar 3.12.



**Gambar 3.12.** Rancangan Antarmuka akun *member*

### 3.11.7 Rancangan Antarmuka Halaman Barang

Rancangan antarmuka halaman barang merupakan tampilan barang yang akan dibeli oleh *member*, tampilan barang terdapat *detail* barang dan ukuran barang yang akan dipesan. Gambar rancangan antarmuka halaman barang terdapat pada Gambar 3.13.

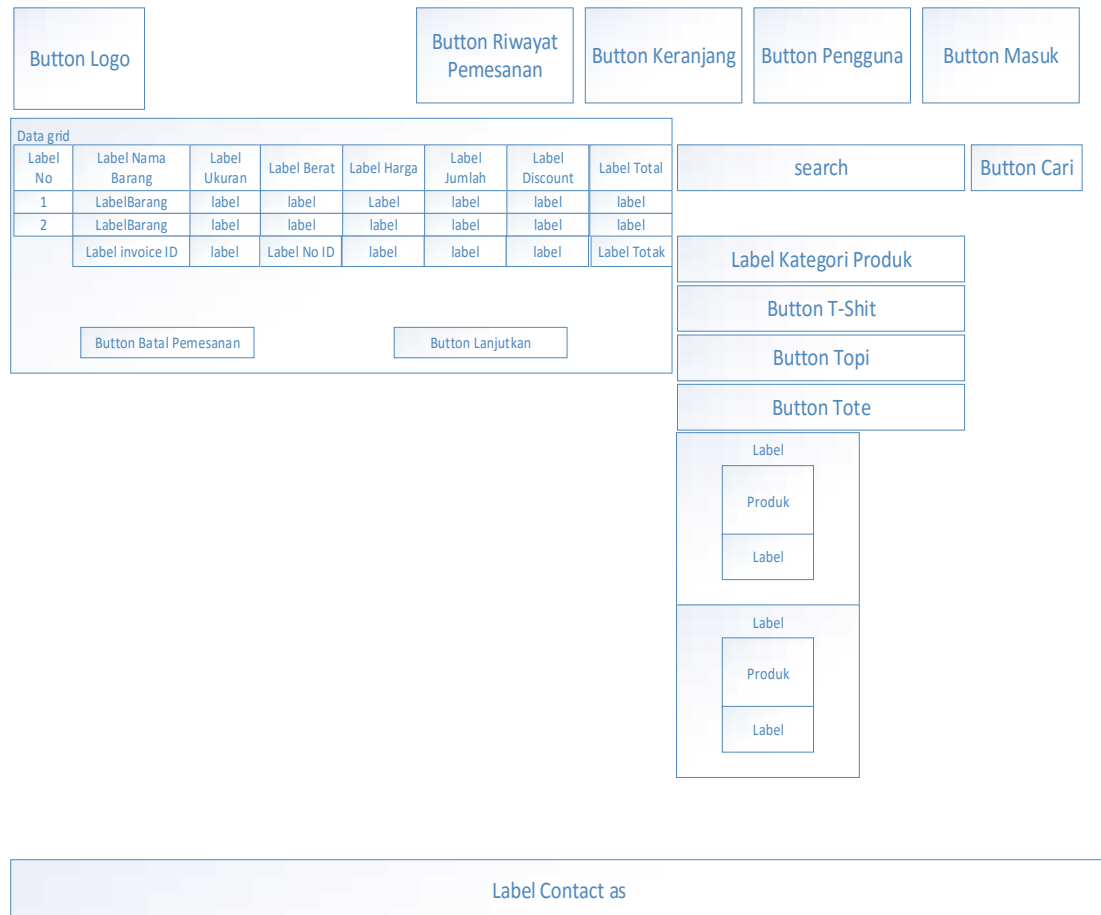


**Gambar 3.13.** Rancangan Antarmuka Halaman Barang

### 3.11.8 Rancangan Antarmuka Halaman Keranjang Belanja

Rancangan antarmuka halaman keranjang belanja terdapat data barang yang dibeli oleh *member*, terdapat dua *button* yaitu *button* batalkan pemesanan dan

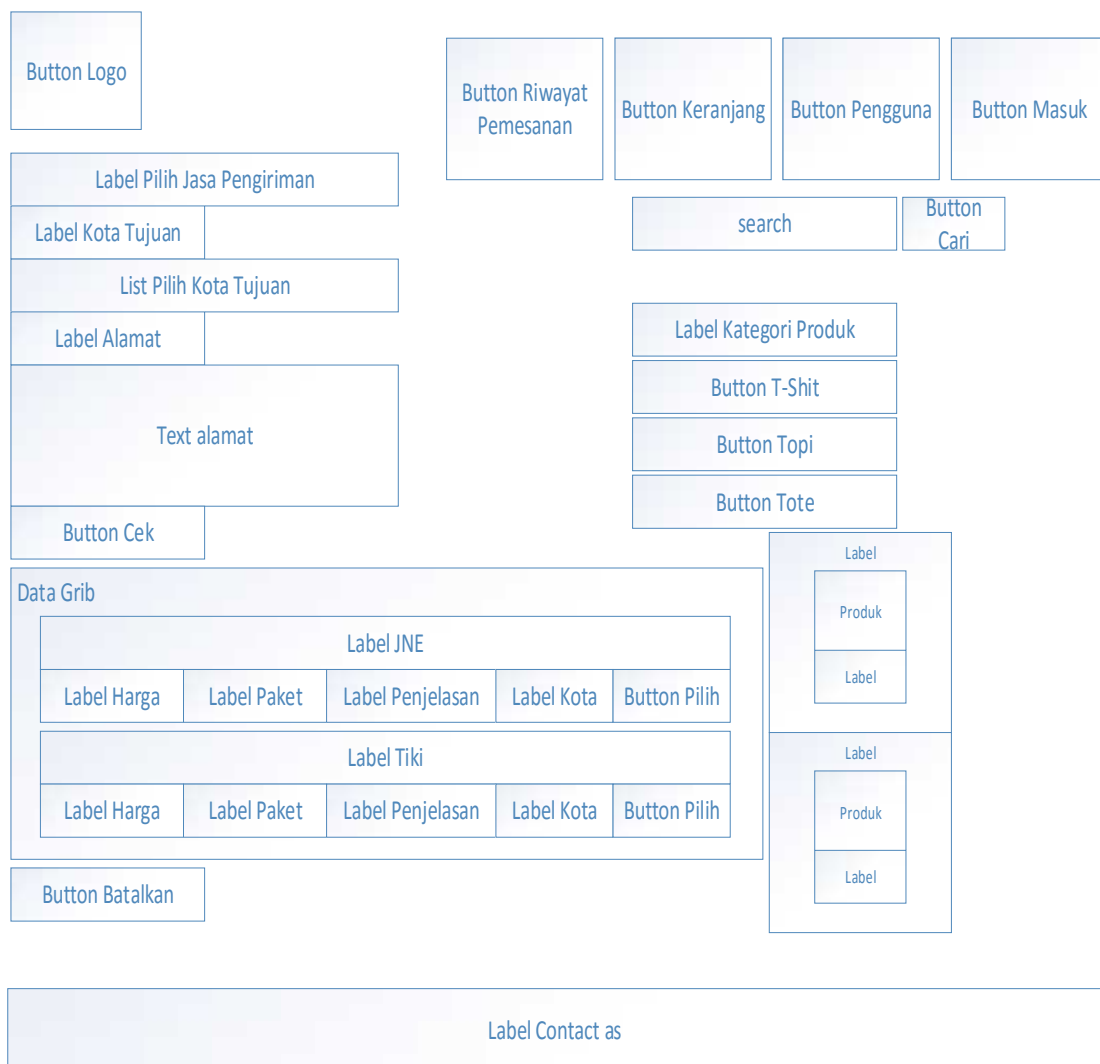
*button* lanjutkan, *button* batalkan pesanan merupakan pembatalan pemesanan yang sudah dibeli oleh *member*, sedangkan *button* lanjutkan merupakan pesanan barang yang sudah dibeli oleh *member*. Gambaran rancangan atarmuka halaman keranjang belanja terdapat pada Gambar 3.14.



**Gambar 3.14.** Rancangan Antarmuka Halaman Keranjang pembelian

### 3.11.9 Rancangan Antarmuka Halaman Jasa Pengiriman

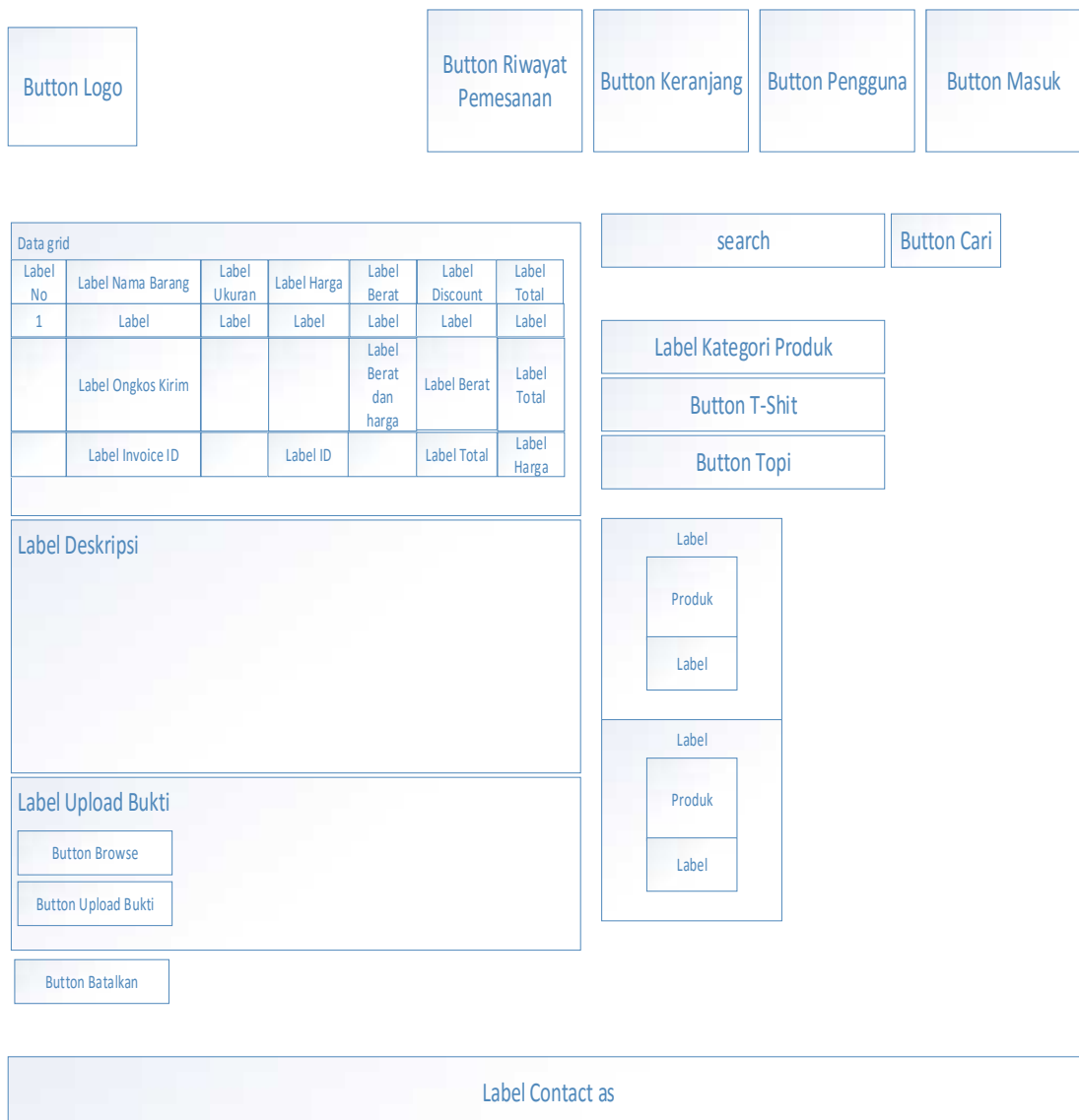
Rancangan antarmuka halaman jasa pengiriman terdapat pengisian data kota dan alamat *member*, terdapat dua jasa pengiriman yaitu melalui jasa JNE dan TIKI. Gambaran rancangan antarmuka halaman jasa pengiriman terdapat pada Gambar 3.15.



**Gambar 3.15.** Rancangan Antarmuka Halaman Jasa Pengiriman

### 3.11.10 Rancangan Antarmuka Halaman *Detail Pemesanan*

Rancangan antarmuka halaman *detail* pemesanan terdapat data pemesanan dan *upload* bukti pembayaran. pembelian *member*. Gambaran rancangan antarmuka halaman *detail* pesanan terdapat pada Gambar 3.16.

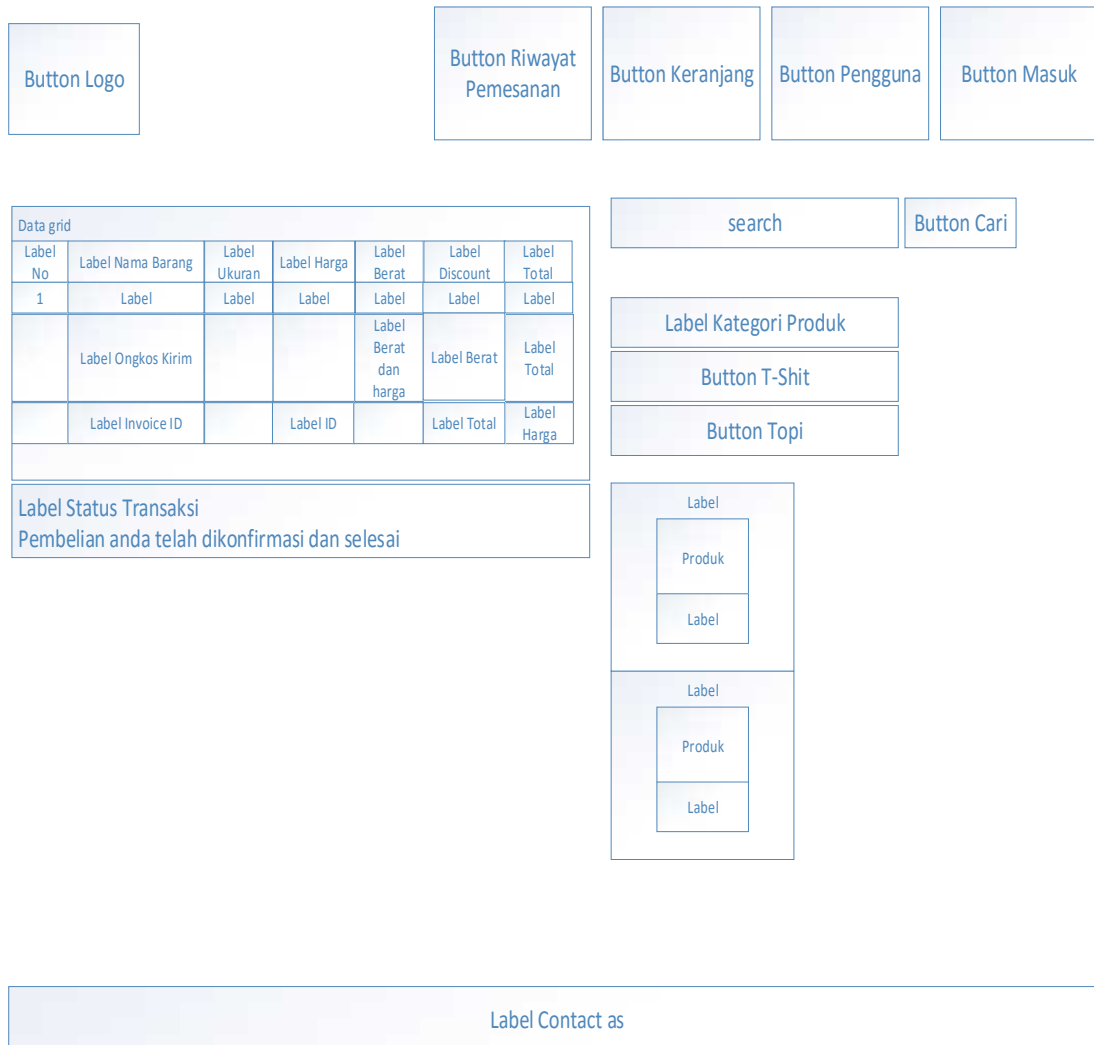


**Gambar 3.16.** Rancangan Antarmuka Halaman *Detail* Pesanan



### 3.11.11 Rancangan Antarmuka Halaman *Detail* Pesanan Berhasil

Rancangan antarmuka Halaman *detail* pesanan berhasil terdapat tampilan status transaksi yang berhasil di *upload* oleh *member*. Gambaran antarmuka halaman *detail* pesanan berhasil terdapat pada Gambar 3.17.



**Gambar 3.17.** Rancangan Antarmuka Halaman *Detail* Pesanan Berhasil

### 3.11.12 Rancangan Antarmuka Halaman Riwayat Pemesanan

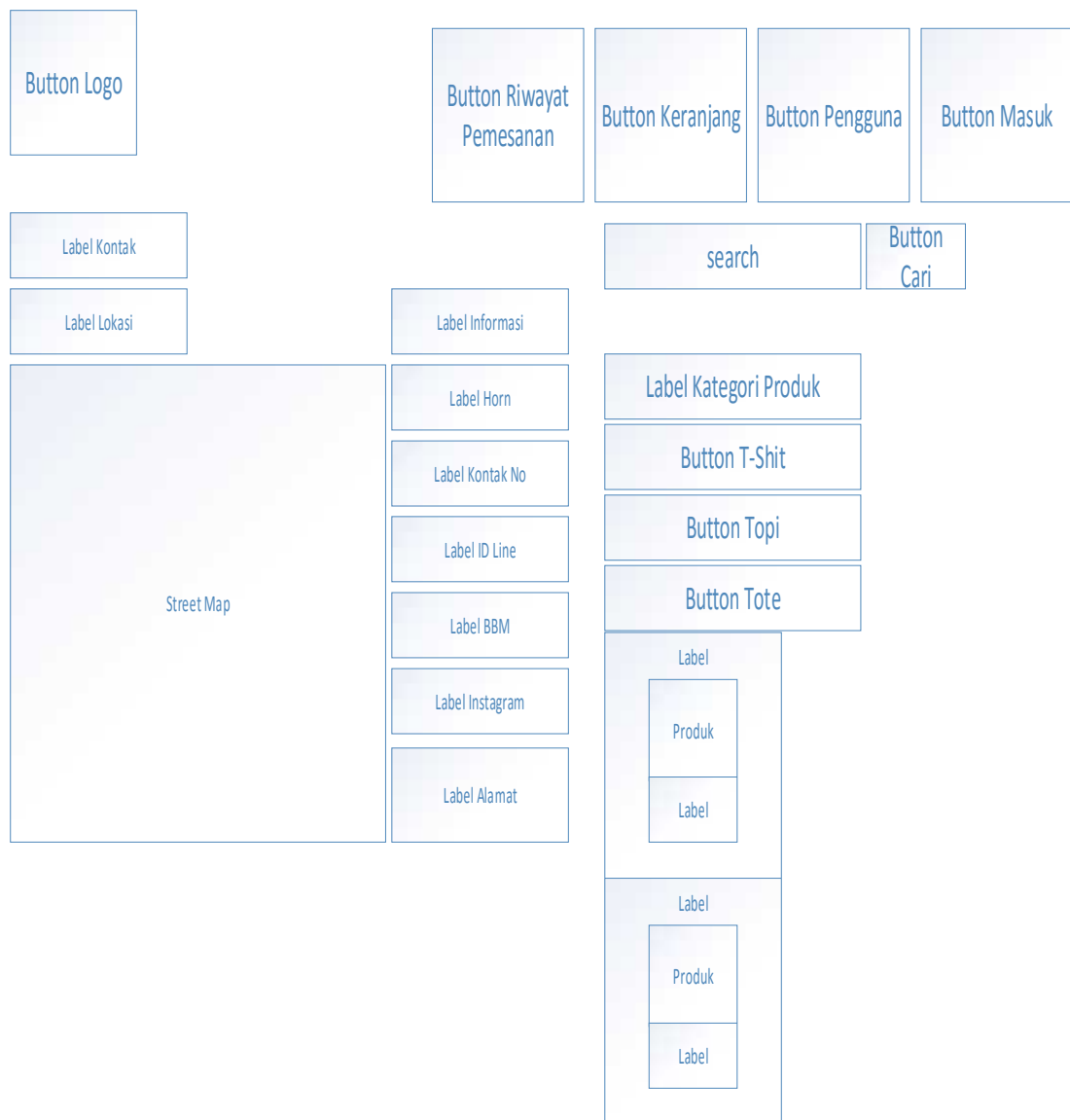
Rancangan antarmuka halaman riwayat pemesanan terdapat informasi data pembelian *member* beserta tanggal pemesanan. Gambaran rancangan antarmuka halaman riwayat pemesanan terdapat pada Gambar 3.18.



**Gambar 3.18.** Rancangan Antarmuka Halaman Riwayat Pemesanan

### 3.11.13 Rancangan Antarmuka Halaman Kontak

Rancangan antarmuka halaman kontak terdapat informasi kontak dan lokasi Horn Music Merch Distro. Gambaran rancangan antarmuka halaman kontak terdapat pada Gambar 3.19.



**Gambar 3.19.** Rancangan Antarmuka Halaman Kontak

### 3.11.14 Rancangan Antarmuka Halaman *Login admin*

Rancangan antarmuka halaman *login admin* terdapat dua *label*, dan dua *text field*, dan satu *button* masuk. Gambaran rancangan antarmuka halaman *login admin* terdapat pada Gambar 3.20.

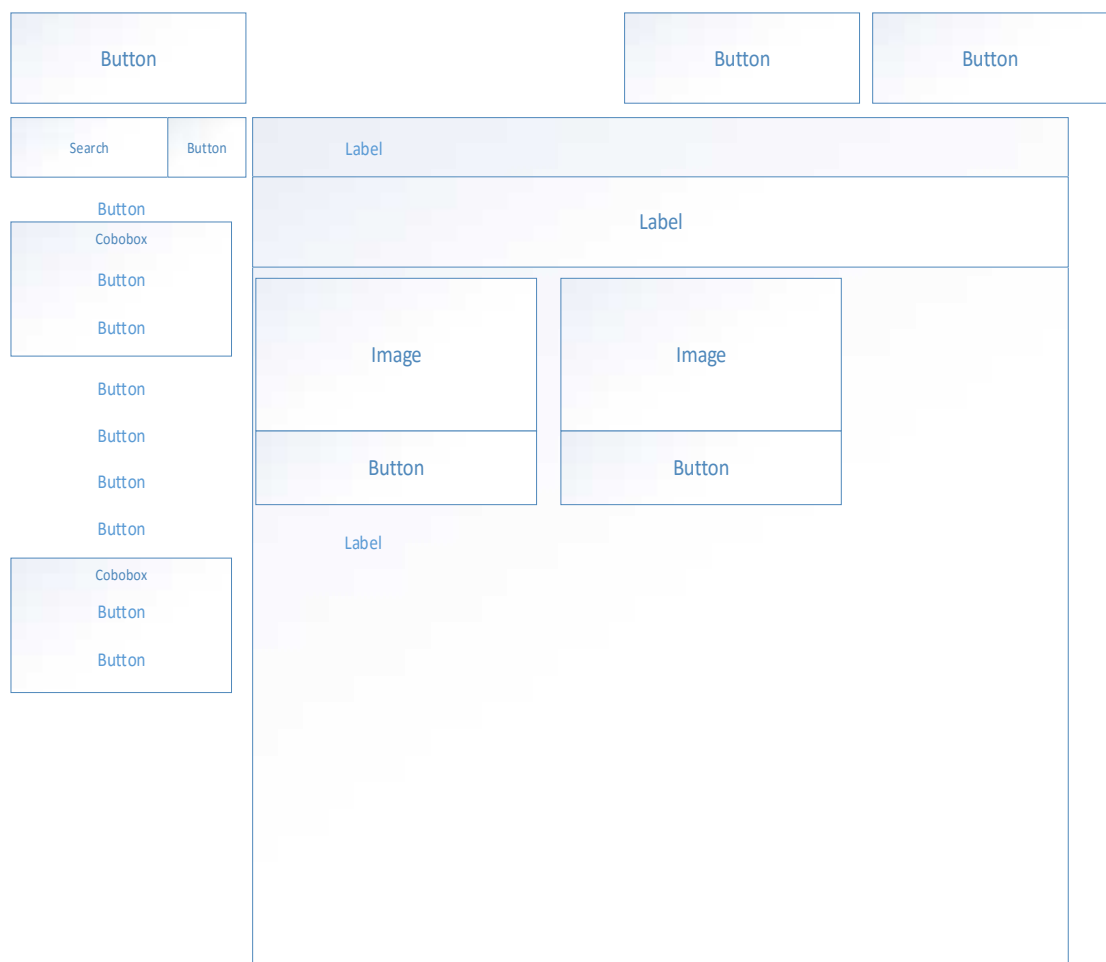
The diagram, titled "Gambar", illustrates the layout of the Admin Login page. It consists of five vertically stacked rectangular components:

- Label UserName
- TextFieldI UserName
- Label Password
- TextFieldI Password
- Button Masuk

**Gambar 3.20.** Rancangan Antarmuka Halaman *Login Admin*

### 3.11.15 Rancangan Antarmuka Halaman Utama *Admin*

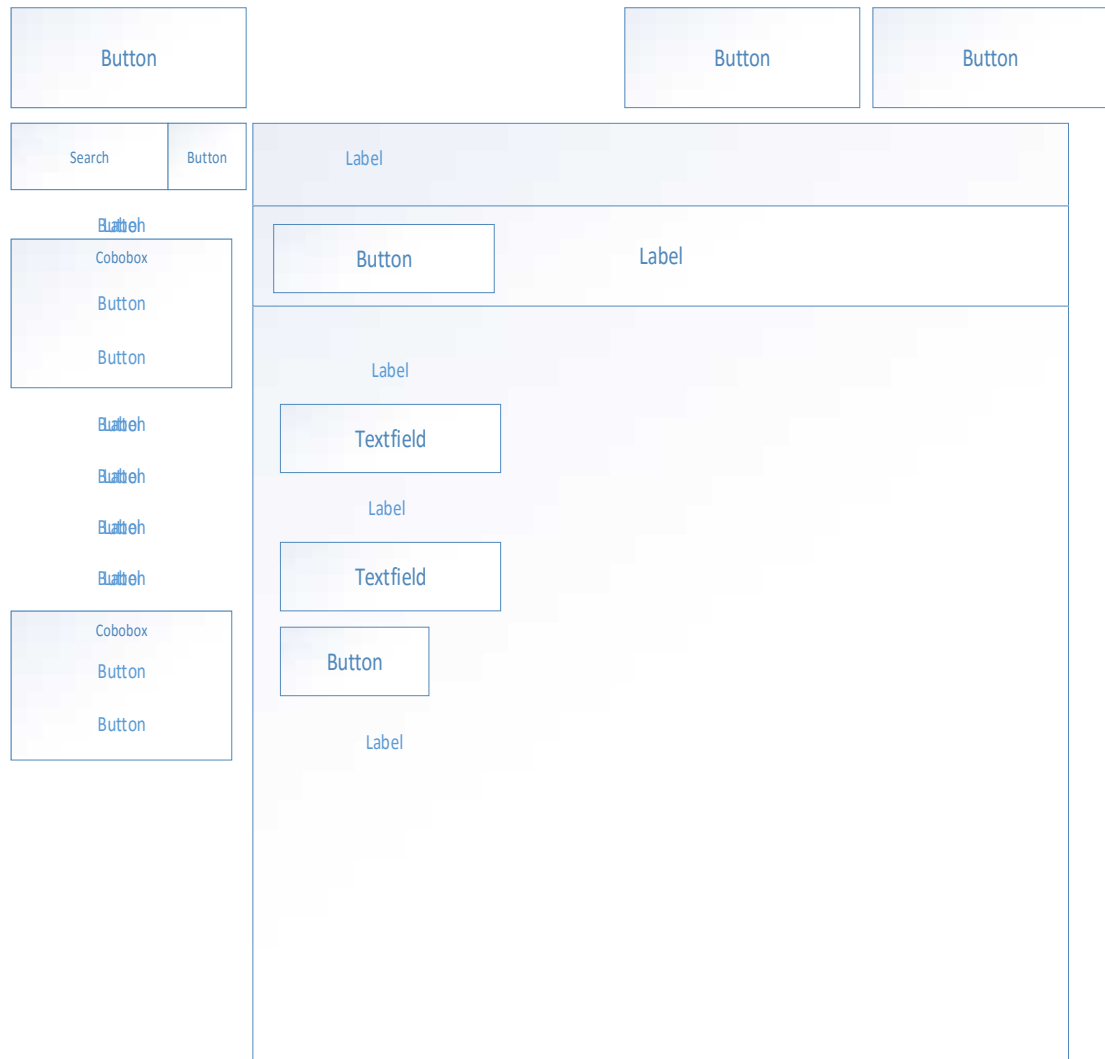
Rancangan antarmuka halaman utama *admin*, terdapat *button*, *label*, dan *image*. *Admin* berfungsi untuk melakukan penambahan produk, menambah stok barang, mengubah *banner*, melihat data pemesanan, melihat data *member*, mengirim bukti transfer, dan melihat laporan penjualan. Gambaran rancangan antarmuka halaman utama *admin* terdapat pada Gambar 3.21.



**Gambar 3.21.** Rancangan Antarmuka Halaman Utama *Admin*

### 3.11.16 Rancangan Antarmuka Halaman Laporan Penjualan

Rancangan antarmuka halaman laporan penjualan *admin* bisa melihat laporan data penjualan. Gambaran rancangan halaman laporan penjualan terdapat pada Gambar 3.22.



**Gambar 3.22.** Rancangan Antarmuka Halaman Laporan Penjualan

