

## BAB III

### METODOLOGI PENELITIAN

#### 3.1 Peralatan Pendukung

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

##### 3.1.1 Perangkat Keras

Adapun perangkat keras yang dibutuhkan dalam membangun *website* Royal Flush:

- a. *Processor Intel Dual core Haswell.*
- b. *Memory 2 GB*
- c. *Hardisk 250 GB.*
- d. *Mouse*

##### 3.1.2 Perangkat Lunak

Adapun perangkat lunak yang dibutuhkan dalam membangun *website* Royal Flush:

- a. *Sublime Text Editor 3*
- b. *MySql*
- c. *Xampp*
- d. *Bootstrap*
- e. *Windows 10*
- f. *Browser : Microsoft Edge*

#### 3.2 Bahan Penelitian

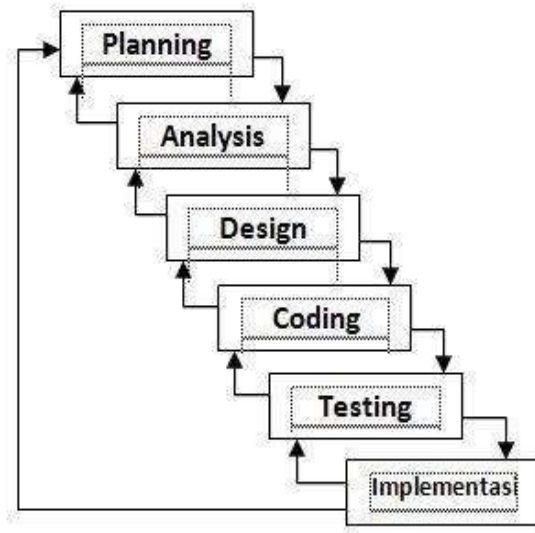
##### 3.2.1 Studi Literatur

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

### 3.2.2 Wawancara

Melakukan tanya jawab dengan pihak yang bersangkutan dengan penelitian secara langsung guna mendapatkan data-data dan informasi yang diinginkan. Wawancara dilakukan dengan pihak Royal Flush.

### 3.3 Langkah Penelitian *Waterfall*



Gambar 3.1 Langkah Metode *Waterfall*

#### 3.3.1 *Requirement*

Langkah pertama menganalisa terhadap kebutuhan sistem yang akan dibangun dan melakukan pengumpulan data secara langsung melalui pengamatan dan wawancara.

#### 3.3.2 *Desain Sistem*

Langkah kedua mulai melakukan perancangan perangkat lunak seperti, struktur data, arsitektur perangkat lunak dan tampilan antar muka.

#### 3.3.3 *Coding dan Testing*

Langkah ketiga melakukan *coding* yang merupakan proses penerjemah *design* dalam bahasa yang dikenal oleh komputer. Setelah *coding* selesai maka akan dilakukan testing terhadap sistem yang telah dibuat.

### **3.4 Requirement Sistem**

#### **3.4.1 Analisis Pemesanan Produk**

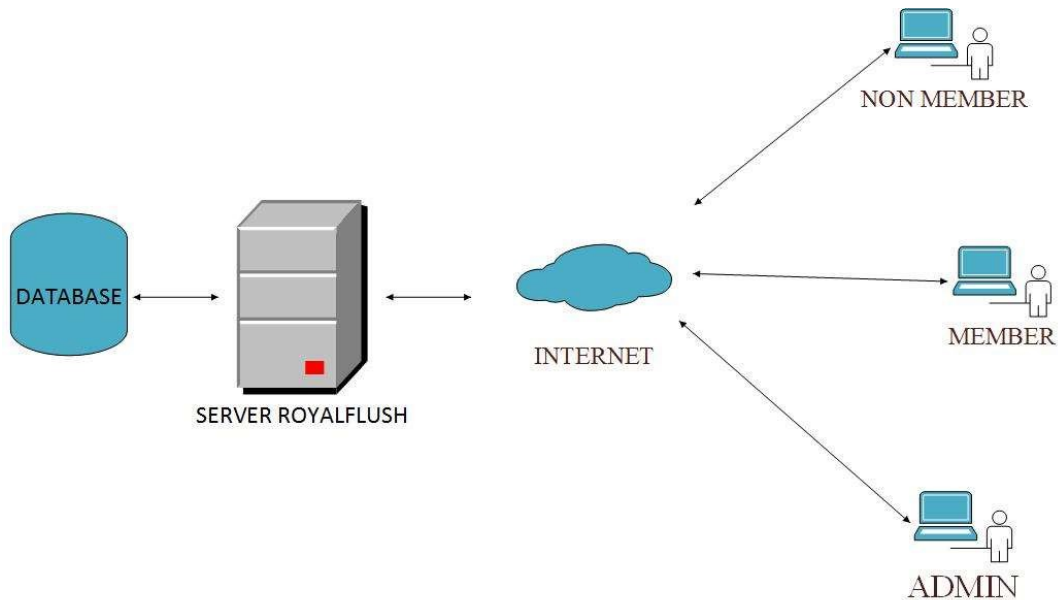
1. Calon *member* harus mendaftar untuk mempunyai akun.
2. Setelah memiliki akun *member* sudah dapat melakukan pembelian produk yang ada di Royal Flush
3. Sebelum melakukan pembayaran *member* mengkonfirmasi jumlah nominal pembayaran, identitas asli *member*, informasi rekening bank yang digunakan *member* dan alamat pengiriman produk.
4. Pembayaran dapat dilakukan *via transfer bank*
5. Setelah pembayaran *admin* akan mengkonfirmasi pembayaran sesuai dengan informasi yang diberikan *member* sebelumnya
6. Setelah di konfirmasi, *admin* memberikan notifikasi bahwa pembayaran telah dikonfirmasi dan produk siap dikirim ke alamat *member*.
7. Website dapat menampilkan produk sesuai dengan kategori.

#### **3.4.2 Analisis Proses Laporan Penjualan Produk**

*Admin* melakukan pengecekan terhadap laporan penjualan dan *update* produk terbaru

### **3.5 Perancangan Sistem**



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

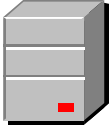


**Gambar 3.2** Arsitektur Sistem yang Diusulkan

*Database server* yang digunakan 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 antarmuka dan melakukan pengambilan data yang diperlukan dari *database server*. Melalui antarmuka yang dimuat *web server* pengguna bisa menyimpan data ke *database server*.

**Tabel 3.1** Keterangan Komponen Arsitektur

	<p><i>Non Member</i></p> <p><i>Member</i></p> <p><i>Admin</i></p>
	<p>Berisi data-data sebagai berikut:</p> <ol style="list-style-type: none"> <li>1. Kategori Produk</li> <li>2. Informasi Perusahaan</li> <li>3. Data Produk</li> <li>4. Data <i>Member</i></li> <li>5. Data <i>Admin</i></li> </ol>

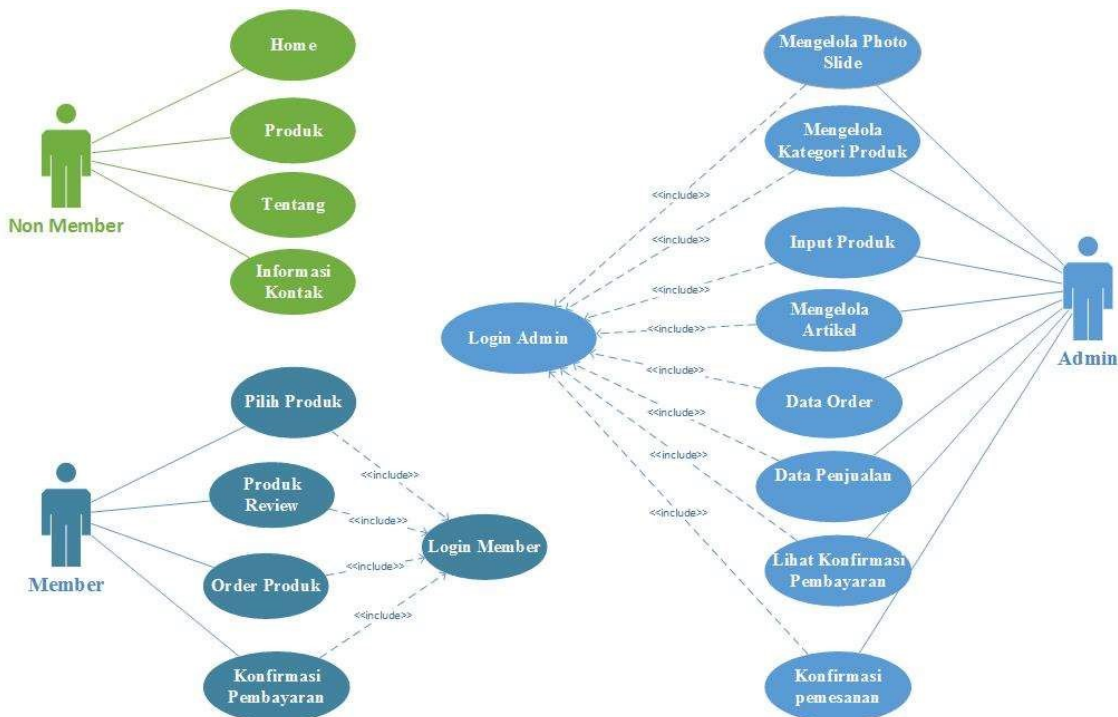
	6. Data Penjualan
	<i>Server</i> digunakan sebagai tempat untuk penyimpanan data-data tersebut.

### 3.6 Perancangan Proses

Dalam pembuatan aplikasi dilakukan perancangan *database* menggunakan bantuan *Diagram ER*. Metode perancangan lain yang digunakan dalam pembuatan aplikasi ini adalah *Unified Markup Language (UML)* yang digunakan dalam pengembangan aplikasi yaitu model *Use Case Diagram*, *Activity Diagram*, dan *Class Diagram*.

#### 3.6.1 Use Case Diagram

Gambar *Use Case Diagram* yang digunakan dalam aplikasi dapat dilihat pada Gambar 3.3



Gambar 3.3 *Use Case Diagram* Aplikasi

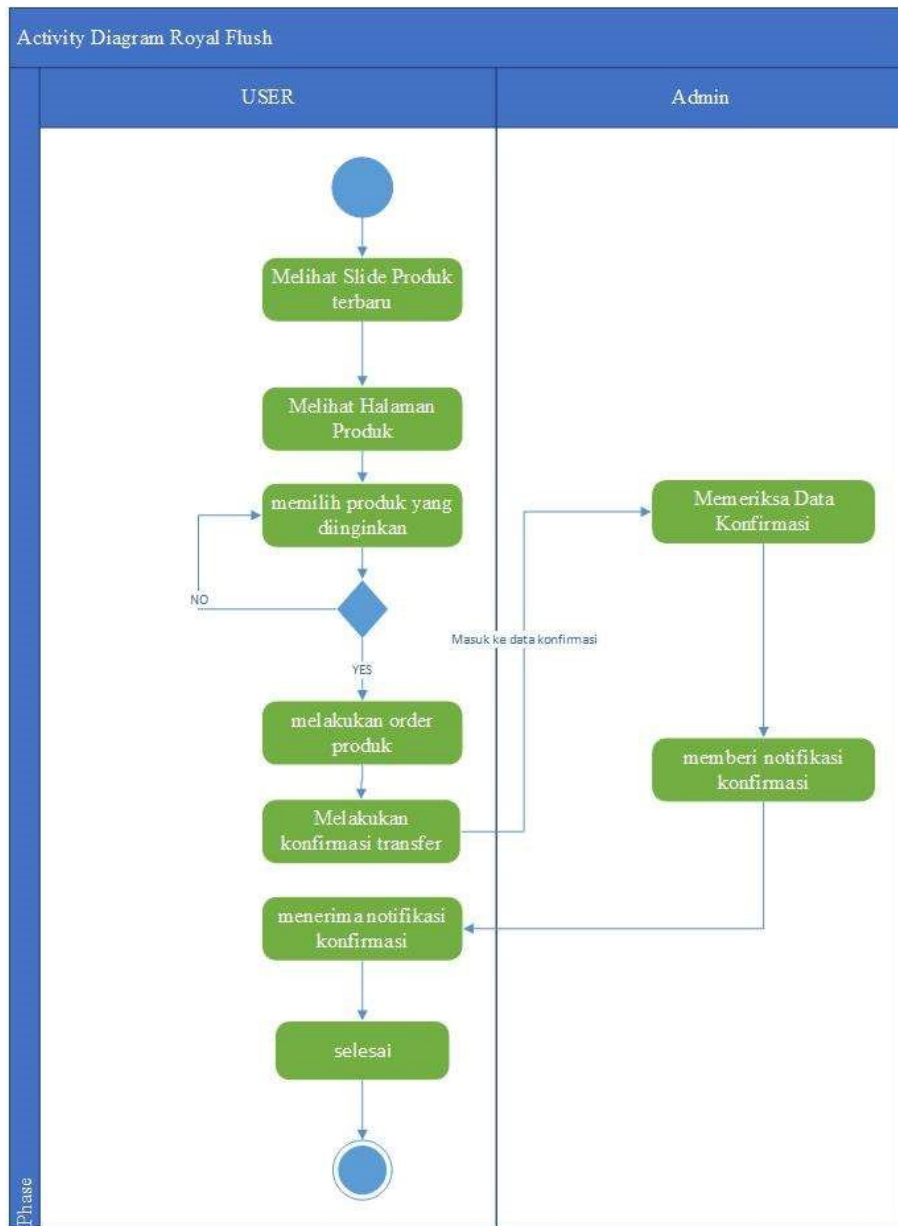
Berikut penjelasan tentang Gambar 3.3:

Terdapat 3 aktor pada *use case diagram* aplikasi yakni *admin*, *non member*, dan *member*.

1. *Admin* dapat mengelola semua data-data pada *website* Royal Flush Cloth.
2. *Non member* hanya dapat masuk kedalam halama utama, melihat Produk dan Informasi – informasi perusahaan.
3. *Member* dapat melakukan *order* produk yang diinginkan, setelah melakukan *order* produk maka *member* harus melakukan pembayaran *via transfer* dan melakukan konfirmasi pembayaran.

### **3.6.2 Activity Diagram Aplikasi**

Gambaran *Activity Diagram* yang digunakan dalam aplikasi dapat dilihat pada Gambar 3.4.



**Gambar 3.4 Activity Diagram**

Berikut penjelasan tentang Gambar 3.4:

1. Terdapat alur dari kegiatan transaksi, yaitu di mulai dari *member* mengecek slide - slide produk terbaru.
2. *Member* juga dapat melihat produk Royal Flush lainnya pada halaman Produk.

3. *Member* dapat memilih produk produk yang diinginkan dan bisa melakukan *order*.
4. *Member* melakukan *order* barang, dalam pembayaran dilakukan *via transfer*. Selanjutnya *member* akan menunggu konfirmasi dari *admin*.
5. *Admin* selanjutnya melakukan pengecekan informasi pembayaran *member*, apabila sudah lengkap dengan ketentuan pembayarannya maka *admin* melakukan konfirmasi *order* yang di pesan oleh ember dan *memberikan* notifikasi.
6. Pesanan *member* terkonfirmasi.

### **3.6.3 ER Diagram**

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





Gambar 3.5 Menunjukkan *Entity Relationship Diagram* dan memiliki 12 buah *entitas* yaitu:

1. *Member*
2. Produk
3. Foto
4. Kategori
5. Penjualan\_detail
6. Penjualan
7. Konfirmasi
8. Bank
9. Kota
10. Petugas
11. Artikel
12. Setting

Berikut penjelasan tentang Gambar 3.5:

- a. *Entitas member* memiliki relasi dengan *entitas* produk *one to many* yang artinya 1 *member* dapat melihat banyak produk yang tersedia di *website*.
- b. *Entitas* Kategori memiliki relasi *one to many* dengan produk yang artinya satu kategori memiliki lebih dari satu produk.
- c. *Entitas* Foto memiliki relasi *one to one* dengan produk, yang artinya satu produk hanya memiliki 1 foto
- d. *Entitas* produk memiliki relasi dengan penjualan\_detail *many to one* yang berarti *member* dapat bisa memilih banyak produk dalam satu kali pembelian.
- e. *Entitas* penjualan\_detail memiliki relasi *one to one* dengan penjualan yang artinya setiap 1 penjualan\_detail akan tersimpan pada penjualan
- f. *Entitas* penjualan\_detail memiliki relasi *one to one* dengan konfirmasi yang berarti setiap 1 penjualan\_detail akan melakukan 1 kali konfirmasi.
- g. *Entitas* konfirmasi memiliki relasi *one to one* dengan bank dan kota yang berarti pada saat akan melakukan konfirmasi, 1 konfirmasi hanya boleh memilih 1 bank dan 1 kota tujuan pengiriman.
- h. *Entitas* petugas memiliki relasi *one to many* dengan produk artinya 1 orang petugas dapat menambahkan banyak produk baru ke dalam *website*.

- i. *Entitas* petugas memiliki relasi *one to many* dengan konfirmasi dan penjualan, artinya 1 orang petugas dapat mengelola banyak penjualan dan konfirmasi.
- j. *Entitas* petugas memiliki relasi dengan *setting* artinya petugas dapat melakukan *setting* atau pengaturan pada *website*.
- k. *Entitas* petugas memiliki relasi *one to one* dengan artikel yang artinya satu orang petugas hanya bisa menambahkan 1 buah artikel pada *website*.

1. Tabel Petugas

Table petugas merupakan tabel yang berfungsi untuk menyimpan data - data petugas dari *website* Royal Flush. Pada tabel ini akan disimpan data seperti id petugas, nama, alamat, telepon, user petugas dan passwordnya.

**Tabel 3.2** Struktur tabel petugas

No	Field Name	Data Type	Key Type	Length
1	Ptg_id	Int	PK	
2	Ptg_nama	Varchar		50
3	Ptg_telp	Varchar		15
4	Ptg_alamat	Varchar		255
5	Ptg_user	Varchar		16
6	Ptg_pass	Varchar		12
7	Ptg_create	Datetime		

## 2. Tabel kategori

Tabel kategori merupakan tabel yang berfungsi untuk menyimpan data kategori produk. Pada tabel ini akan disimpan data berupa id kategori, dan nama kategori.

**Tabel 3.3** Struktur tabel kategori

No	Field Name	Data Type	Key Type	Length
1	kat_id	Int	PK	
2	kat_nama	Varchar		50
3	Kat_create	Datetime		

## 3. Tabel kota

Tabel kota ini berfungsi untuk menyimpan informasi tentang kota yang akan menjadi tujuan pengiriman produk. Pada tabel ini akan menyimpan data berupa id kota, nama kota, serta ongkos kirim dari setiap kota.

**Tabel 3.4** Struktur tabel kota

No	Field Name	Data Type	Key Type	Length
1	Kt_id	Int	PK	
2	Kt_nama	Varchar		50
3	Kat_ongkir	Varchar		15
4	Kt_create	Datetime		

## 4. Tabel Member

Tabel *member* merupakan tabel yang berfungsi untuk menyimpan data *member*. Pada tabel ini akan berisi data *id member*, nama *member*, telepon

*member*, alamat *member*, user *member*, *member register*, user *member*, password *member*.

**Tabel 3.5** Struktur tabel *member*

<b>No</b>	<b>Field Name</b>	<b>Data Type</b>	<b>Key Type</b>	<b>Length</b>
1	Mem_id	Int	PK	
2	Mem_nama	Varchar		50
3	Mem_tlp	Varchar		15
4	Mem_alamat	Varchar		255
5	Mem_register	Varchar		25
6	mem_user	Varchar		25
7	mem_password	Vaarchar		16
8	Mem_create	Datetime		

#### 5. Tabel Foto

Tabel Foto merupakan tabel yang berfungsi untuk menyimpan data - data foto dari produk. Pada tabel ini akan menyimpan data *id* foto, nama foto, dan url dari foto tersebut.

**Tabel 3.6** Struktur tabel foto

<b>No</b>	<b>Field Name</b>	<b>Data Type</b>	<b>Key Type</b>	<b>Length</b>
1	ft_id	Int	PK	
2	pr_id	Varchar		10
3	ft_nama	Varchar		50
4	ft_url	Varchar		20
5	ft_create	Datetime		
6	ft_slide	Varchar		5
7	ft_judul	Varchar		25
8	ft_note	Varchar		25

6. Tabel Penjualan\_detail

Tabel penjualan\_detail merupakan tabel yang berfungsi untuk menyimpan data data proses penjualan yang telah dilakukan. Tabel ini menyimpan data seperti id produk yang dibeli, jumlah penjualan, id penjualan detail.

**Tabel 3.7** Struktur tabel penjualan\_detail

<b>No</b>	<b>Field Name</b>	<b>Data Type</b>	<b>Key Type</b>	<b>Length</b>
1	pd_id	Int	PK	
2	pd_jml	Int		
3	pnj_id	Int		
4	Pr_id	Int		

## 7. Tabel Konfirmasi

Tabel konfirmasi merupakan tabel yang berfungsi untuk menyimpan data konfirmasi. Tabel ini menyimpan data berupa konfirmasimasi id, penerima, alamat kirim, atas nama bank, nama bank, total belanja, dan total transfer yang nanti akan dilakukan.

**Tabel 3.8** Struktur tabel konfirmasi

<b>No</b>	<b>Field Name</b>	<b>Data Type</b>	<b>Key Type</b>	<b>Length</b>
1	kon_id	Int	PK	
2	kon_penerima	Varchar		50
3	kon_telp	Varchar		15
4	kon_alamat	Text		
5	kon_an	Varchar		50
6	Kon_an_bank	Varchar		25
7	Kon_total	Varchar		16
8	Kon_tr_jum	Varchar		16
9	Pnj_id	Int		
10	B_id	Int		
11	Kt_id	Int		
12	Kon_create	Timestamp		
13	Kon_status	Int		

## 8. Tabel Artikel

Tabel artikel ini merupakan tabel yang berfungsi untuk menyimpan data data artikel untuk *website*. Artikel ini bisa berupa informasi perusahaan. Pada tabel ini akan menyimpan data berupa id artikel, judul dan isi dari artikel tersebut.

**Tabel 3.9** Struktur tabel artikel

<b>No</b>	<b>Field Name</b>	<b>Data Type</b>	<b>Key Type</b>	<b>Length</b>
1	art_id	Int	PK	
2	art_judul	Varchar		20
3	art_isi	Text		
4	art_create	Datetime		
5	Ptg_id	Int		

## 9. Tabel *Setting*

Tabel *Setting* merupakan tabel yang berfungsi untuk menyimpan data data *setting* pada *website*. Tabel ini berisi data data seperti *setting id*, judul web, logo.

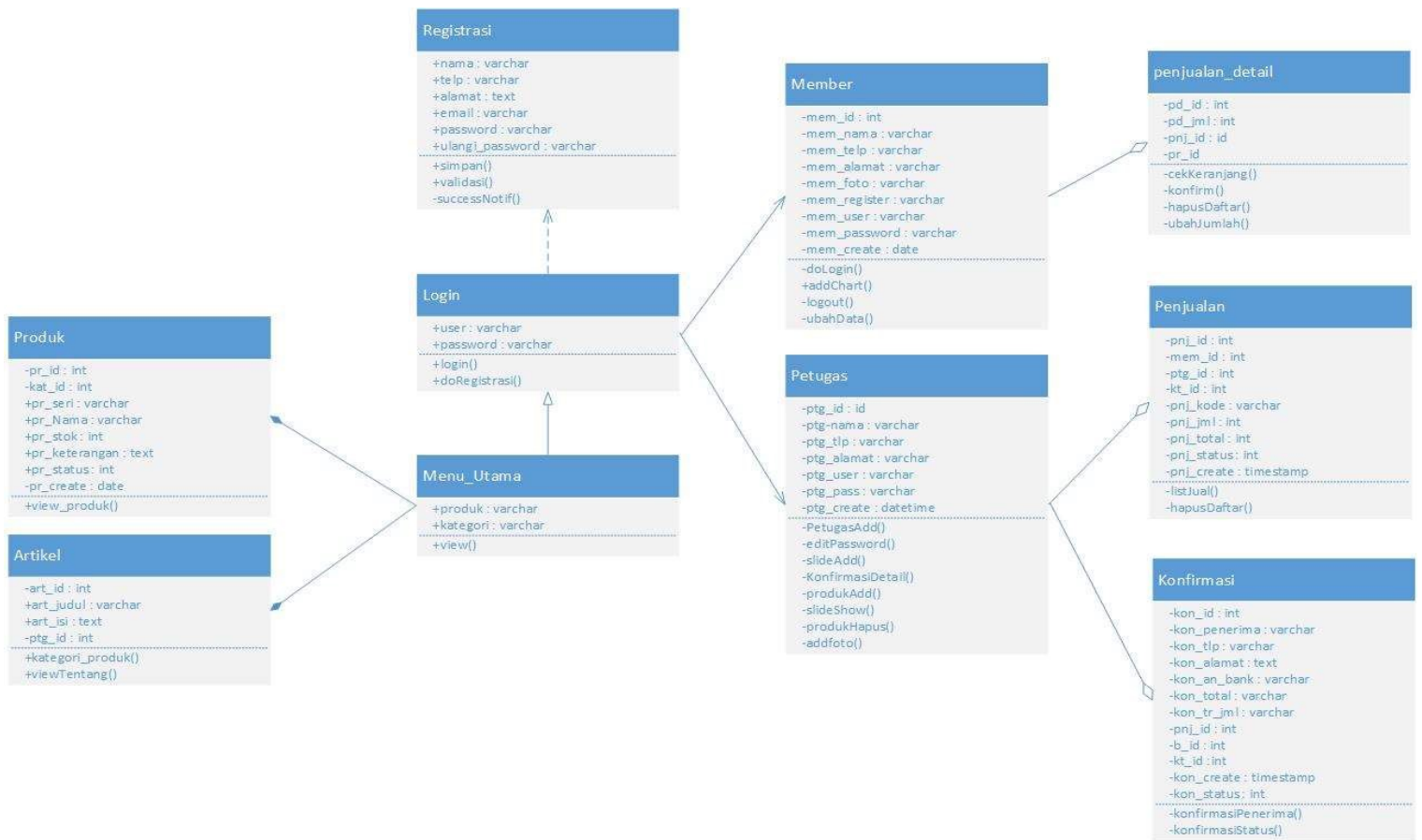


**Tabel 3.10** Struktur tabel *setting*

<b>No</b>	<b>Field Name</b>	<b>Data Type</b>	<b>Key Type</b>	<b>Length</b>
1	Set_id	Int	PK	
2	Set_judulweb	Varchar		15
3	Set_kodepembelian	Varchar		20
4	Set_kodepenjualan	Varchar		20
5	Set_tentang	Text		
6	Set_informasi	Text		
7	Set_logo	Varchar		15

#### **3.6.4 Class Diagram**

*Class Diagram* yang digunakan dalam aplikasi dapat dilihat pada Gambar 3.6.



Gambar 3.6 Class Diagram Aplikasi

Berikut adalah penjelasan class diagram pada gambar 3.6:

Pada *class login* adalah *class* yang digunakan untuk *memberi* akses untuk *member* dan *admin* agar bisa mengakses *website*. Pada *class* ini terdapat beberapa *method* yaitu sebagai berikut:

1. *Method login()*, berfungsi sebagai validasi login *member* dan *admin*. data *member* dan *admin* akan diperiksa di *database* apakah tersedia atau tidak.
2. *Method doRegistrasi()*, berfungsi untuk calon *member* yang ingin mendaftar untuk menjadi *member*.

Pada *class Menu\_utama* digunakan untuk menampilkan halaman utama. Pada *class* ini terdapat beberapa *method* seperti berikut:

1. *Method view()*, berfungsi untuk menampilkan konten – konten halaman utama.

Pada *class produk* digunakan untuk menampilkan produk produk yang ada. Pada *class* ini terdapat beberapa *method* sebagai berikut:

1. *Method viewProduk()*, berfungsi untuk melihat produk yang ada pada web.

Pada *class Artikel* digunakan oleh *member* untuk melihat artikel. Pada *class* ini terdapat beberapa *method* seperti:

1. *Method KategoriProduk()*, berfungsi untuk melihat kategori produk.
2. *Method viewTentang()*, berfungsi untuk melihat informasi-informasi tentang perusahaan.

Pada *class Registrasi* digunakan untuk mendaftar *member* baru. Pada *class* ini memiliki beberapa *method* seperti berikut:

1. *Method simpan()*, berfungsi untuk menyimpan informasi – informasi *member* baru ke dalam *database*.
2. *Method validasi()*, berfungsi untuk memvalidasi informasi yang dimasukkan oleh calon *member* seperti *password* dan *captcha*.

3. *Method succesnotif()*, berfungsi untuk menampilkan notifikasi jika proses registrasi telah sukses.

Pada *class Member* digunakan oleh *member* untuk berbelanja produk yang ada pada web. Pada *class* ini memiliki beberapa method seperti berikut:

1. *Method doLogin()*, berfungsi untuk validasi *login member*.
2. *Method addchart()*, berfungsi untuk menambahkan item ke dalam keranjang belanja.
3. *Method logout()*, berfungsi untuk melakukan *log out*.
4. *Method ubahData()*, berfungsi untuk mengubah data *member*.

Pada *class* petugas digunakan oleh *admin*. pada *class* ini memiliki beberapa method seperti berikut:

1. *Method petugasAdd()*, berfungsi untuk menambahkan petugas baru.
2. *Method editpass()*, berfungsi untuk melakukan perubahan password.
3. *Method slideAdd()*, berfungsi untuk menambahkan gambar yang akan ditampilkan pada *slide*.
4. *Method konfirmasiDetail()*, berfungsi untuk melihat daftar konfirmasi pembayaran.
5. *Method productAdd()*, berfungsi untuk menambahkan produk baru.
6. *Method slideshow()*, berfungsi untuk menampilkan gambar mana yang akan digunakan pada *slide*.
7. *Method produkHapus()*, berfungsi untuk menghapus produk.
8. *Method addFoto()*, berfungsi untuk menambahkan foto untuk produk.

Pada *class* penjualan\_detail digunakan oleh *member* untuk menyimpan *list* produk yang telah dibeli *member* sebelumnya. Pada *class* ini memiliki beberapa *method* seperti berikut:

1. *Method cekKeranjang()*, berfungsi untuk melihat isi keranjang belanja *member*.
2. *Method konfirmasi()*, berfungsi untuk melakukan konfirmasi pembayaran.

3. *Method hapusDaftar()*, berfungsi untuk menghapus daftar item dari keranjang.
4. *Method ubahJumlah()*, berfungsi untuk mengubah jumlah item yang ada di keranjang sebelum dilakukan konfirmasi.

Pada *class* Penjualan digunakan oleh *admin* untuk melihat daftar data penjualan yang ada. Pada *class* ini terdapat beberapa *method* seperti berikut:

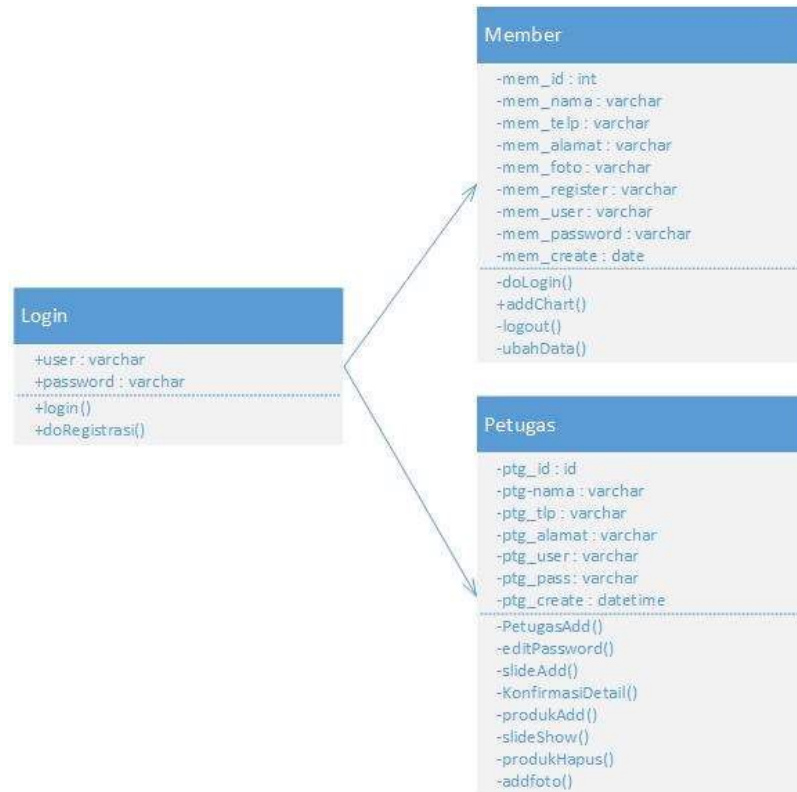
1. *Method listJual()*, berfungsi untuk menampilkan daftar data penjualan.
2. *Method hapusDaftar()*, berfungsi untuk menghapus daftar yang diinginkan *admin*.

Pada *class* Konfirmasi digunakan oleh *admin* untuk melihat serta memvalidasi konfirmasi pembayaran dari *member*. Pada *class* ini terdapat beberapa *method* seperti berikut:

1. *Method konfirmasiPenerima()*, berfungsi untuk menampilkan daftar konfirmasi pembayaran dari *member*.
2. *Method konfirmasiStats()*, berfungsi untuk memvalidasi konfirmasi tersebut serta memberikan notif kepada *member*.

Pada gambar 3.6 menjelaskan bahwa aplikasi berbasis *web* yang akan dibuat dibagi atas beberapa *class* yang saling berhubungan. Kelas-kelas yang saling berhubungan memiliki arti antara hubungan yang satu dengan yang lain. Kelas-kelas yang berelasi antara lain:

a. Relasi antara *class login*, *class member* dan *class petugas*.



**Gambar 3.7** Relasi antara *class login*, *class member* dan *class petugas*

Relasi antara *class login*, *class member* dan *class petugas* dihubungkan dengan relasi *Directed Association* yaitu relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain. *Class login* digunakan untuk login oleh *member*, dan *admin* yang dihubungkan melalui *method login()*

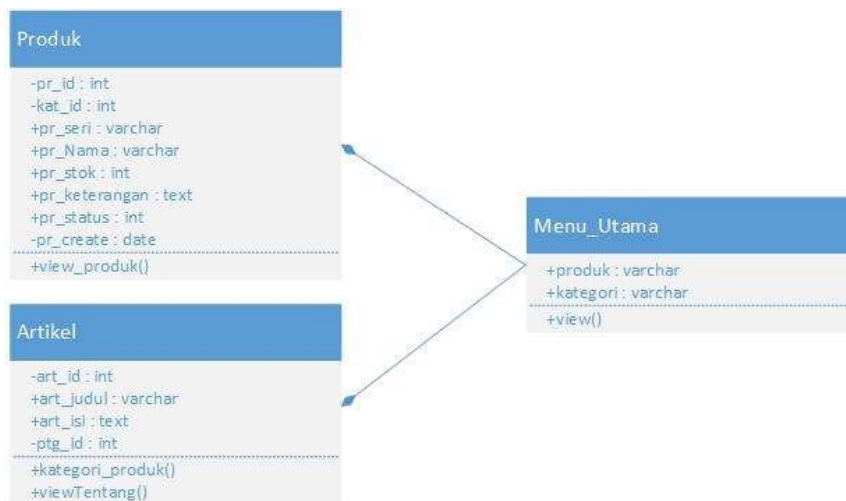
b. Relasi antara class *Menu\_Utama* dengan class *Login*



**Gambar 3.8** Relasi antara class *Menu\_Utama* dengan class *Login*

Relasi antara class *Menu\_utama* dengan class *login* terhubung dengan relasi generalisasi. Menu utama akan terbuka ketika *member* telah melakukan *login* dan sukses jika posisinya adalah pengguna ingin masuk langsung sebagai *member*.

c. Relasi antara class *Menu\_utama*, class *Produk*, class *Artikel*



**Gambar 3.9** Relasi antara class *Menu\_utama*, class *Produk*, class *Artikel*

Relasi antara class *Menu\_utama*, class *Produk*, class *Artikel* terhubung dengan relasi komposisi. Artinya ada 2 class penyusun class *Menu\_utama*. Class penyusun

tersebut juga tidak dapat digunakan jika tidak difungsikannya *class* menu\_utama, begitu juga sebaliknya.

d. Relasi antara *class login* dengan *class* registrasi



**Gambar 3.10** Relasi antara *class login* dengan *class* registrasi

Relasi antara *class login* dengan *class* registrasi dihubungkan dengan relasi *dependency* atau dalam bahasa indonesianya relasi dependensi. Relasi dependensi merupakan Relasi antar kelas dengan makna ketergantungan antar kelas. Jadi ketika pengguna akan melakukan *login*, maka pengguna tersebut harus terdaftar terlebih dahulu yaitu dengan mendaftarkan dirinya pada *class* registrasi untuk mendapatkan *username* dan *password* untuk *login*.



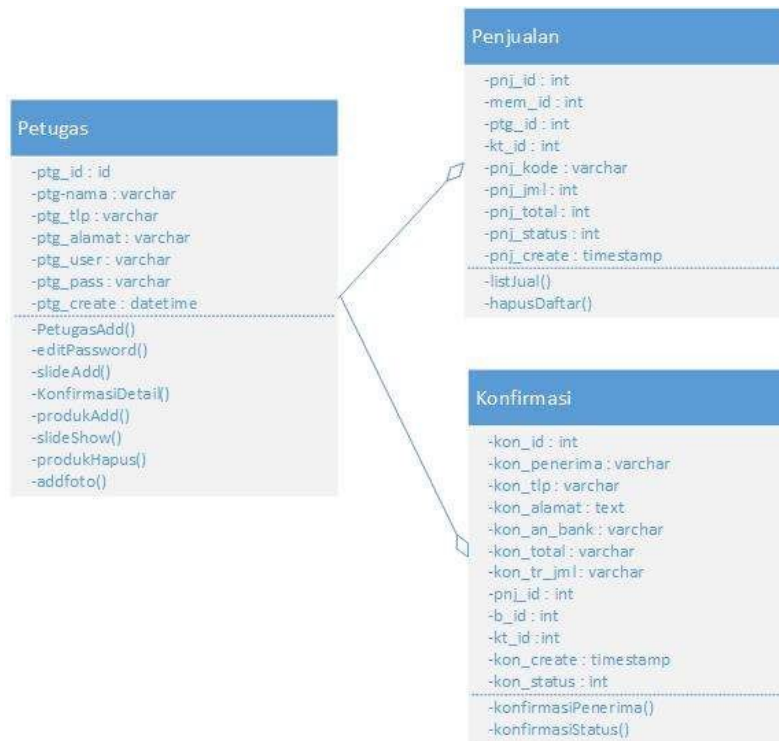
e. Relasi antara *class Member* dengan *class penjualan\_detail*



**Gambar 3.11** Relasi antara *class Member* dengan *class penjualan\_detail*

Relasi antara *class Member* dengan *class penjualan\_detail* dihubungkan dengan relasi agregasi. *Class member* merupakan *main class* dari *subclass penjualan\_detail*

f. Relasi antara *class* Petugas, *class* Penjualan, dan *class* Konfirmasi



**Gambar 3.12** Relasi antara *class* Petugas, *class* Penjualan, dan *class* Konfirmasi

Relasi antara *class* Petugas, *class* Penjualan, dan *class* Konfirmasi dihubungkan dengan relasi agregasi. Artinya *class* Petugas merupakan *main class* dari *subclass* Penjualan dan Konfirmasi.

### 3.7 Perancangan *Interface*

Antarmuka sangatlah penting dalam suatu aplikasi karena merupakan bagian dari perangkat lunak yang menjadi sarana komunikasi antara pengguna dengan sistem serta dapat *memberikan* kemudahan bagi pengguna dalam melakukan aktifitasnya.

#### 3.7.1 Rancangan Antarmuka Halaman Utama

Rancangan antarmuka halaman utama merupakan tampilan utama dari aplikasi yang dapat dilihat oleh *member*, dan *admin*. Gambar rancangan antarmuka halaman utama aplikasi dapat dilihat pada Gambar 3.13.

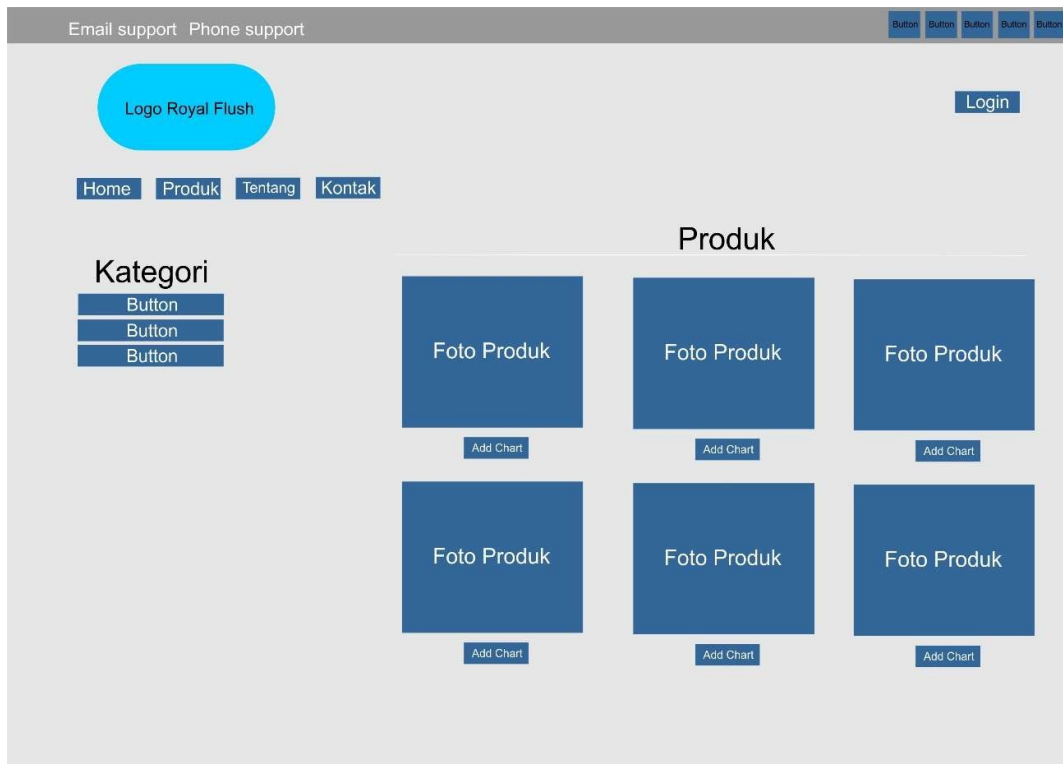


**Gambar 3.13** Rancangan Halaman Utama

Terdapat *menu-menu* yang dapat dibuka sesuai dengan kebutuhan pengguna. Selain *menu-menu* terdapat juga foto produk terbaru dari Royal Flush. Jika pengguna sudah masuk pada halaman utama, semua menu akan aktif sesuai dengan fungsinya.

### 3.7.2 Rancangan Antarmuka Halaman Produk

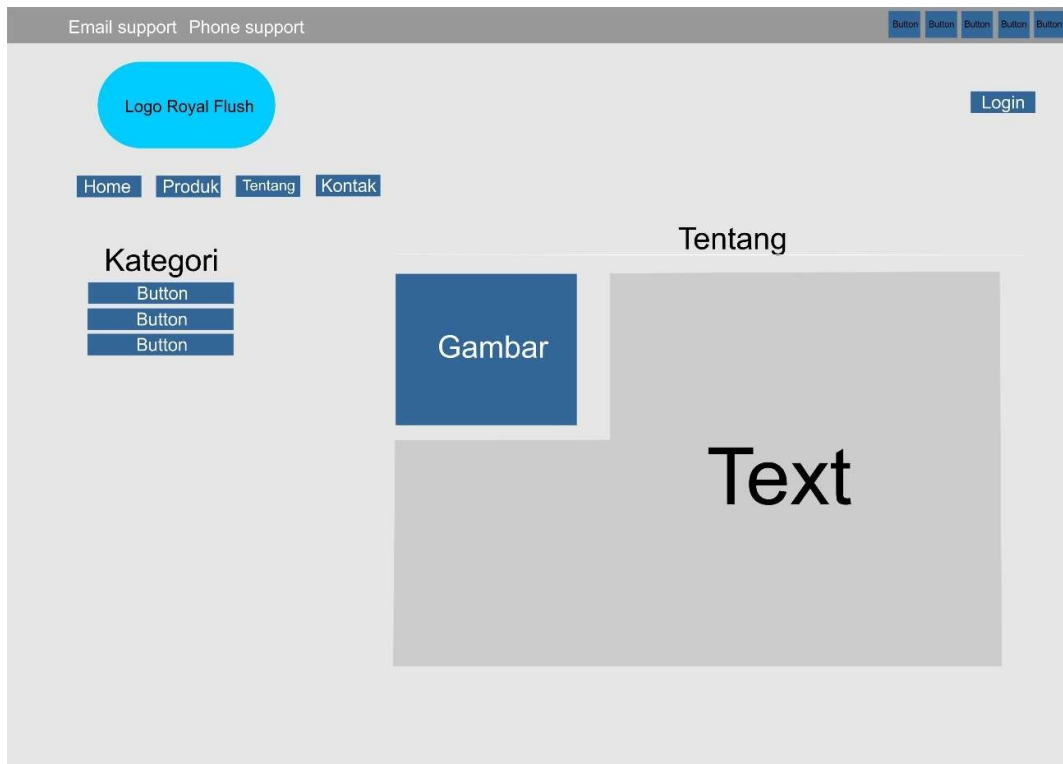
Rancangan antarmuka halaman *Produk* berisi semua daftar produk-produk Royal Flush dari yang terbaru hingga yang terlama yang tentunya masih tersedia bagi calon *member*. Gambar rancangan halaman antar muka produk dapat dilihat pada Gambar 3.14.



**Gambar 3.14** Rancangan Antarmuka Halaman *Produk*

### 3.7.3 Rancangan Antarmuka Halaman Tentang

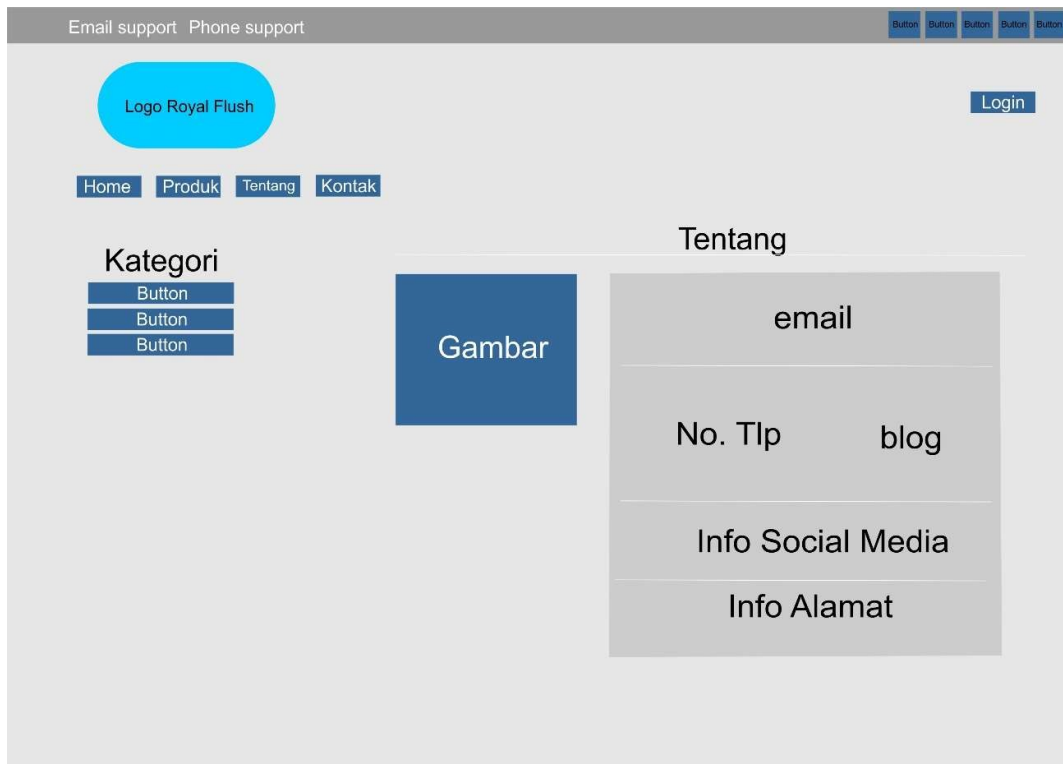
Rancangan antarmuka halaman Tentang ini berisi beberapa informasi sejarah singkat tentang Royal Flush. Desain antarmuka halaman tentang dapat dilihat pada gambar 3.15.



**Gambar 3.15** Rancangan Antarmuka Halaman Tentang

### 3.7.4 Rancangan Antarmuka Halaman Kontak

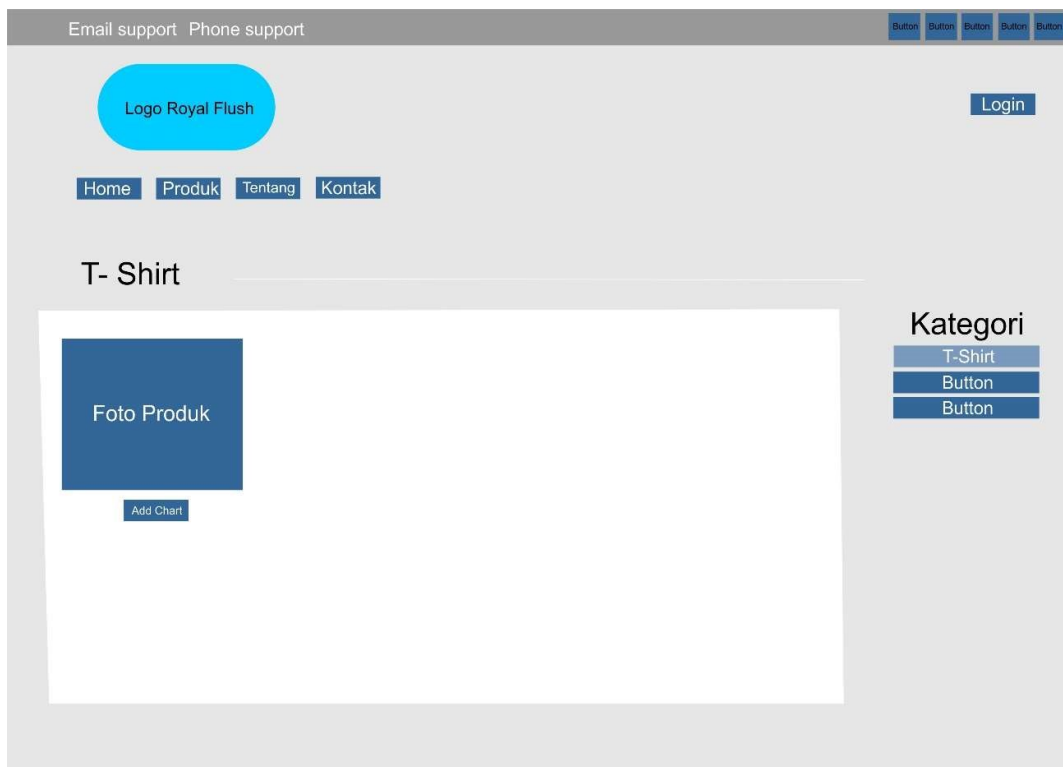
Rancangan antarmuka halaman Kontak ini berisi tentang informasi kontak dari Royal Flush berupa alamat *e-mail*, aplikasi sosial media dan informasi kontak lainnya. Desain antarmuka halaman tentang dapat dilihat pada gambar 3.16.



**Gambar 3.16** Rancangan Antarmuka Halaman Kontak

### 3.7.5 Rancangan Antarmuka Halaman Kategori

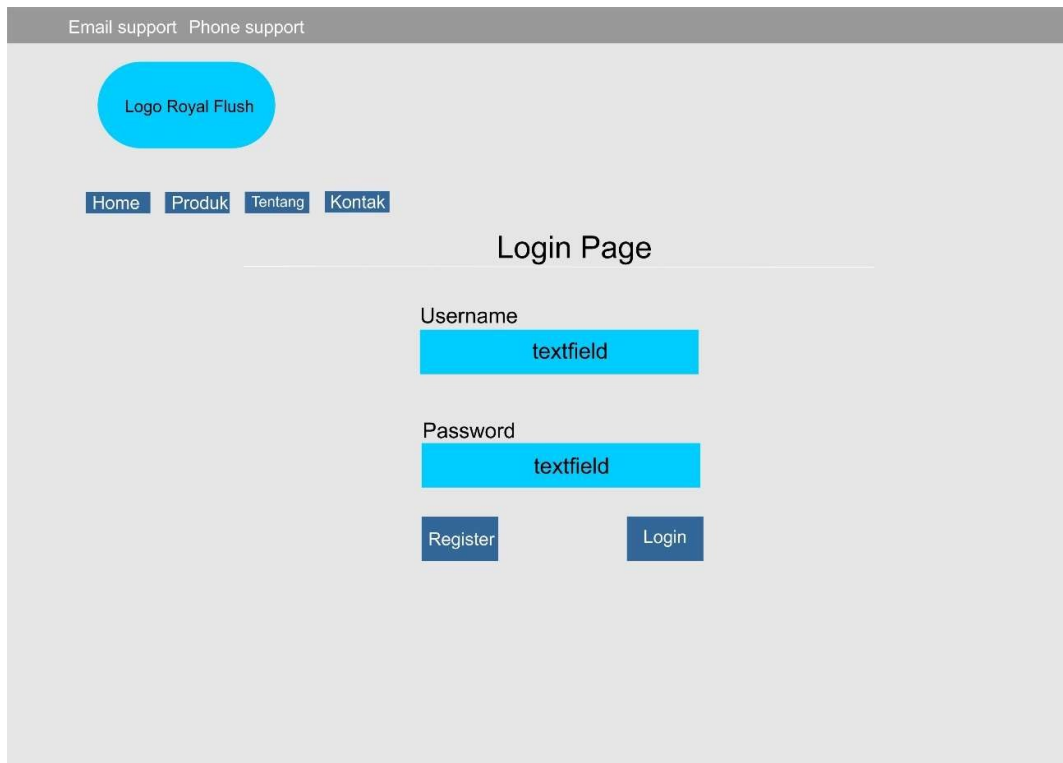
Rancangan antarmuka halaman kategori yang berisi tentang daftar produk sesuai dengan kategori yang pada Royal Flush, rancangan antar muka dapat dilihat pada gambar 3.17.



**Gambar 3.17** Rancangan Antarmuka Halaman Kategori Produk

### 3.7.6 Rancangan Antarmuka Halaman *Login*

Rancangan untuk halaman *login* terdapat dua *label*, dua *text field* dan 2 *button* masuk. Gambar rancangan antarmuka halaman *login* dapat dilihat pada Gambar 3.18.

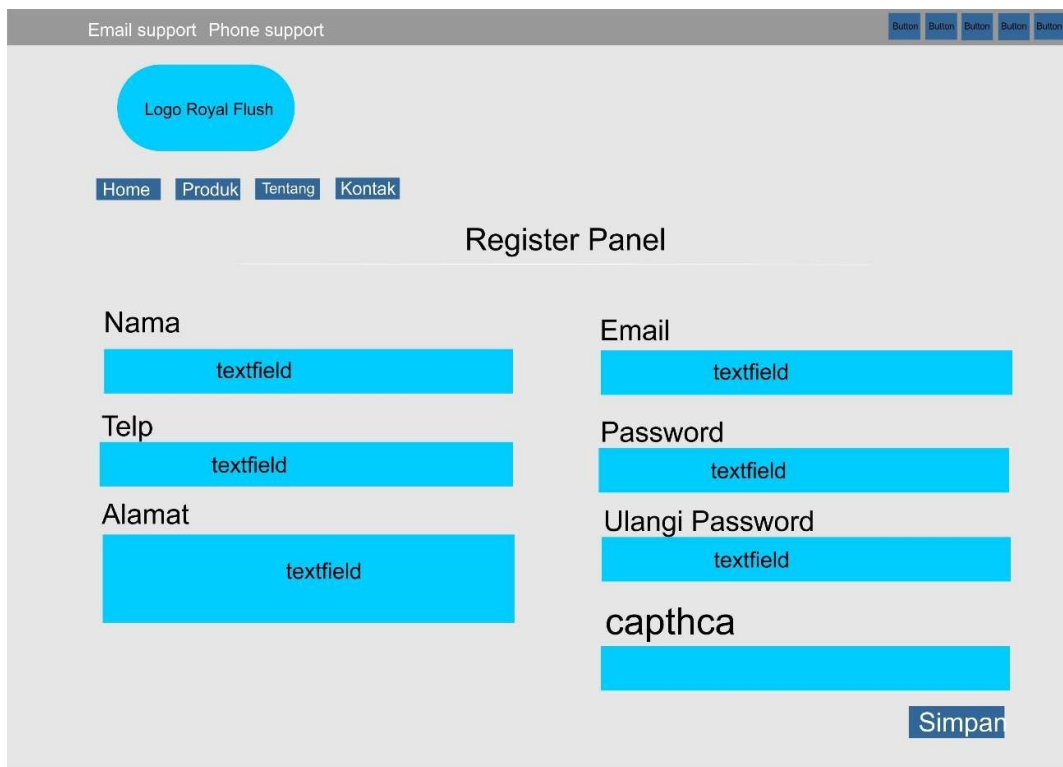


**Gambar 3.18** Rancangan Antarmuka Halaman *Login*



### 3.7.7 Rancangan Antarmuka Halaman Register

Rancangan antarmuka halaman *register*, berisi hal-hal yang berkaitan dengan ketentuan pendaftaran yang dapat dilihat oleh calon *member*. Gambar rancangan antarmuka halaman *register* dapat dilihat pada gambar Gambar 3.19.



The image shows a wireframe of a registration page. At the top, there is a dark grey header with 'Email support' and 'Phone support' on the left, and five blue buttons labeled 'Button' on the right. Below the header is a light grey main area. On the left, there is a blue rounded rectangle labeled 'Logo Royal Flush'. Below it are four blue buttons labeled 'Home', 'Produk', 'Tentang', and 'Kontak'. In the center, the title 'Register Panel' is displayed. The form consists of two columns of input fields. The left column has three fields: 'Nama' (textfield), 'Telp' (textfield), and 'Alamat' (textfield). The right column has three fields: 'Email' (textfield), 'Password' (textfield), and 'Ulangi Password' (textfield). Below these is a 'caphca' field (textfield). At the bottom right of the form is a blue button labeled 'Simpan'.

**Gambar 3.19** Rancangan Antarmuka Halaman *Register Member*

### 3.7.8 Rancangan Antarmuka Halaman *Profile Member*

Rancangan untuk halaman *Profile member* terdapat data informasi tentang *member*, disini *member* dapat melakukan perubahan nama, alamat dan nomer telepon. *Member* juga bisa melihat *list* produk yang telah di pesan oleh *member*. Gambaran rancangan halaman akun *member* dapat dilihat pada Gambar 3.20.

The screenshot shows a web interface for a member's profile. At the top, there are links for 'Email support' and 'Phone support', and a row of five 'Button' labels. Below this is the 'Logo Royal Flush' and a navigation menu with 'Home', 'Produk', 'Tentang', and 'Kontak'. The main content is divided into two sections: 'Akun' (Account) and 'Log Pembelian' (Purchase Log).

**Akun**

Nama: textfield

Telp: textfield

Alamat: textfield

Email: Email@email.com

Tgl Daftar: 10-maret-2012

Save

**Log Pembelian**

25 ▼ textfield

No.	Kode	Tgl	
1	Xp02	10/23/2015 14:20:25	Cek Keranjang

**Gambar 3.20** Rancangan Antarmuka Halaman *Profile Member*

### 3.7.9 Rancangan Antarmuka Halaman Keranjang *Member*

Pada halaman ini *member* dapat melihat isi keranjang belanja *member* yang nantinya akan dikonfirmasi oleh *member*. Rancangan halaman keranjang dapat dilihat pada gambar 3.21.

Email support Phone support

Button Button Button Button Button

Logo Royal Flush

Profile Keranjang Log-out

Home Produk Tentang Kontak

Akun

Nama  
textfield

Telp  
textfield

Alamat  
textfield

Email  
Email@email.com

Tgl Daftar  
10-maret-2012

Save

Daftar Pembelian

25

No.	Foto	Nama	Unit	Jumlah
1	Foto	Produk Name Produk Note Rp 10.000	1	Rp 10.000,-

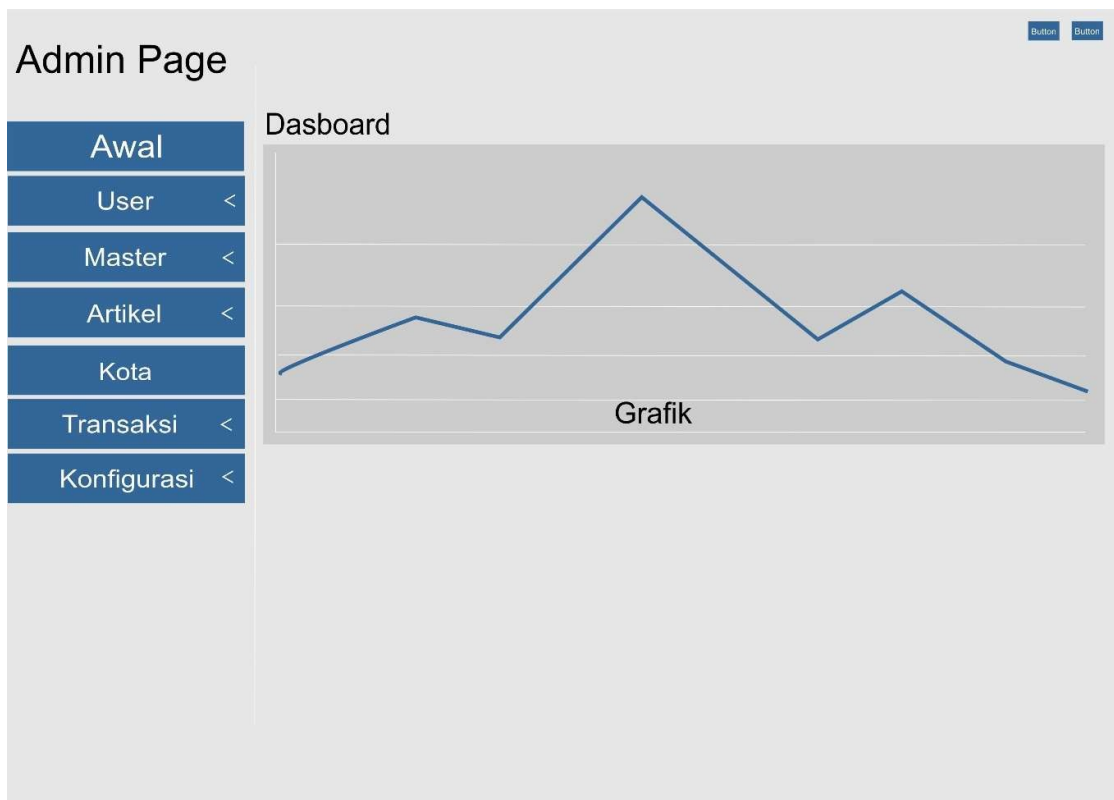
Rp 10.000,-

Konfirmasi

**Gambar 3.21** Rancangan Antarmuka Keranjang *Member*

### 3.7.10 Rancangan Antarmuka Halaman *Admin*

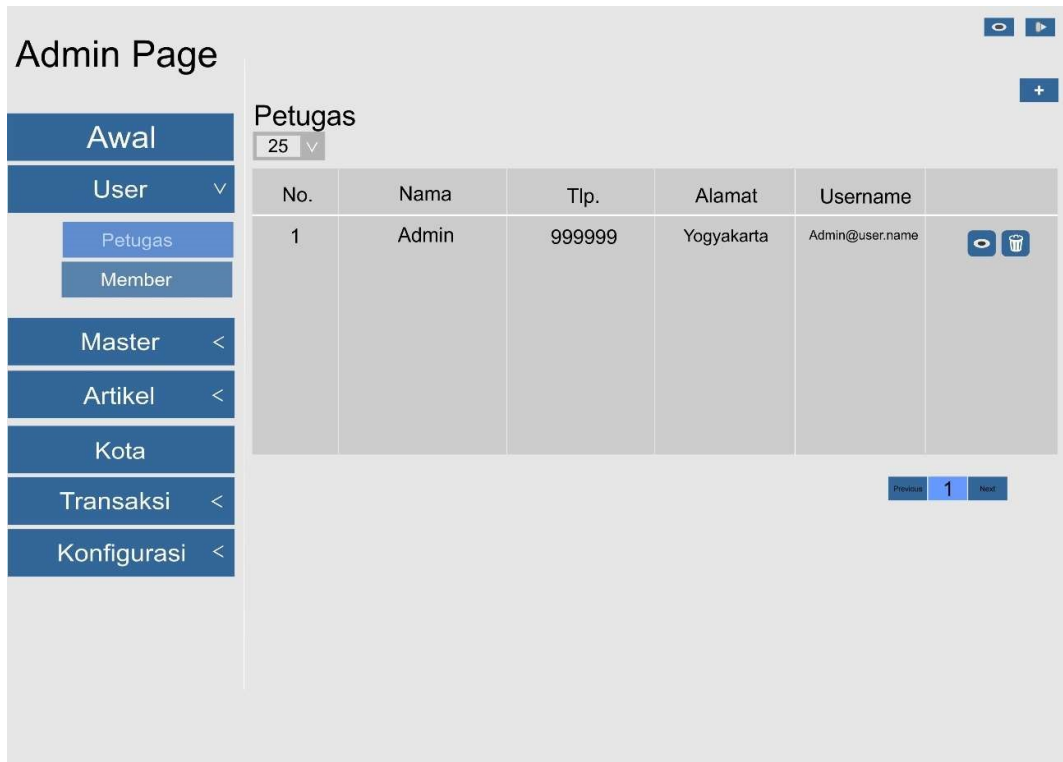
Rancangan antarmuka halaman utama *admin* merupakan tampilan utama dari aplikasi yang dapat dilihat oleh *admin*. Gambar rancangan antarmuka halaman utama *admin* dapat dilihat pada Gambar 3.22.



**Gambar 3.22** Rancangan Antarmuka Halaman Utama *Admin*

### 3.7.11 Rancangan Antarmuka Halaman Petugas

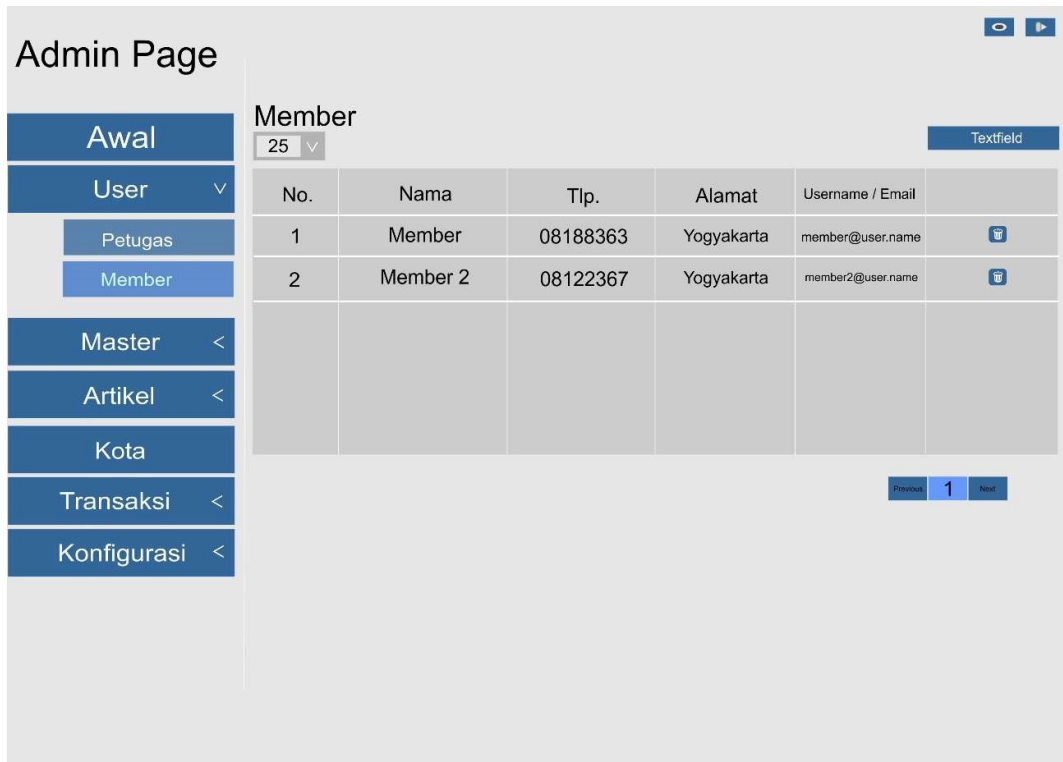
Rancangan antarmuka halaman petugas merupakan halaman yang digunakan untuk *admin* menambah atau menghapus dan mencari petugas. Gambaran rancangan antarmuka halaman petugas dapat dilihat pada Gambar 3.23.



**Gambar 3.23** Rancangan Antarmuka Halaman Petugas

### 3.7.12 Rancangan Antarmuka Halaman *Member*

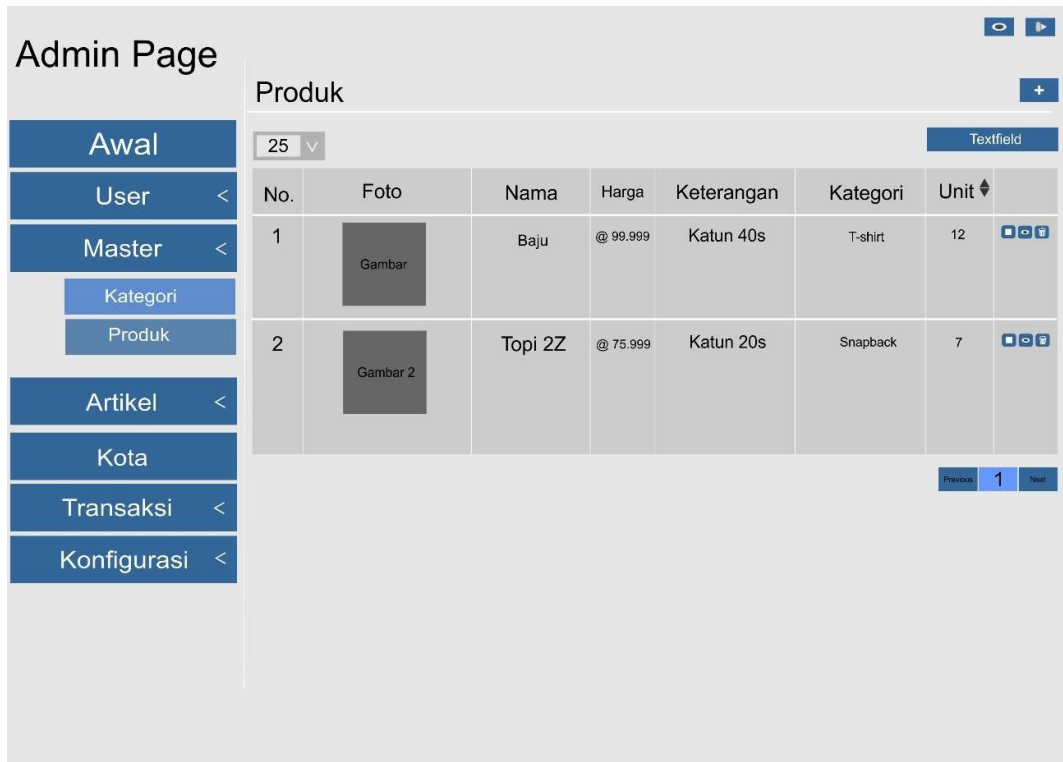
Rancangan antarmuka halaman *member* merupakan halaman yang digunakan untuk *admin* menghapus dan mencari *member*. Gambaran rancangan antarmuka halaman *member* dapat dilihat pada Gambar 3.24.



**Gambar 3.24** Rancangan Antarmuka Halaman *Member*

### 3.7.13 Rancangan Antarmuka Halaman Data Produk

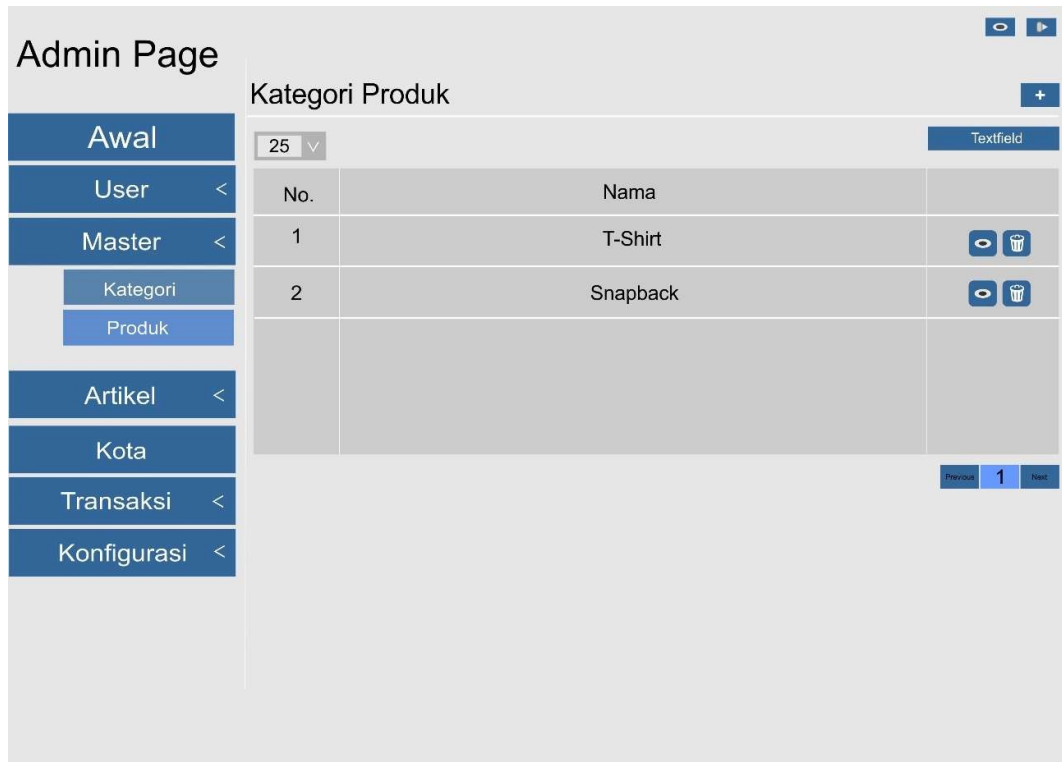
Rancangan antarmuka halaman data produk *admin* dapat melakukan penambahan produk, mungubah harga dan informasi lainnya mengenai produk tersebut. Gambar rancangan antarmuka halaman data produk dapat dilihat pada Gambar 3.25.



**Gambar 3.25** Rancangan Antarmuka Halaman Data Produk

### 3.7.14 Rancangan Antarmuka Halaman Data Kategori

Rancangan antarmuka halaman data kategori *admin* dapat melakukan penambahan dan pengurangan kategori jika di perlukan di dalam *website* tersebut. Gambar rancangan antarmuka halaman data kategori dapat dilihat pada Gambar 3.26.

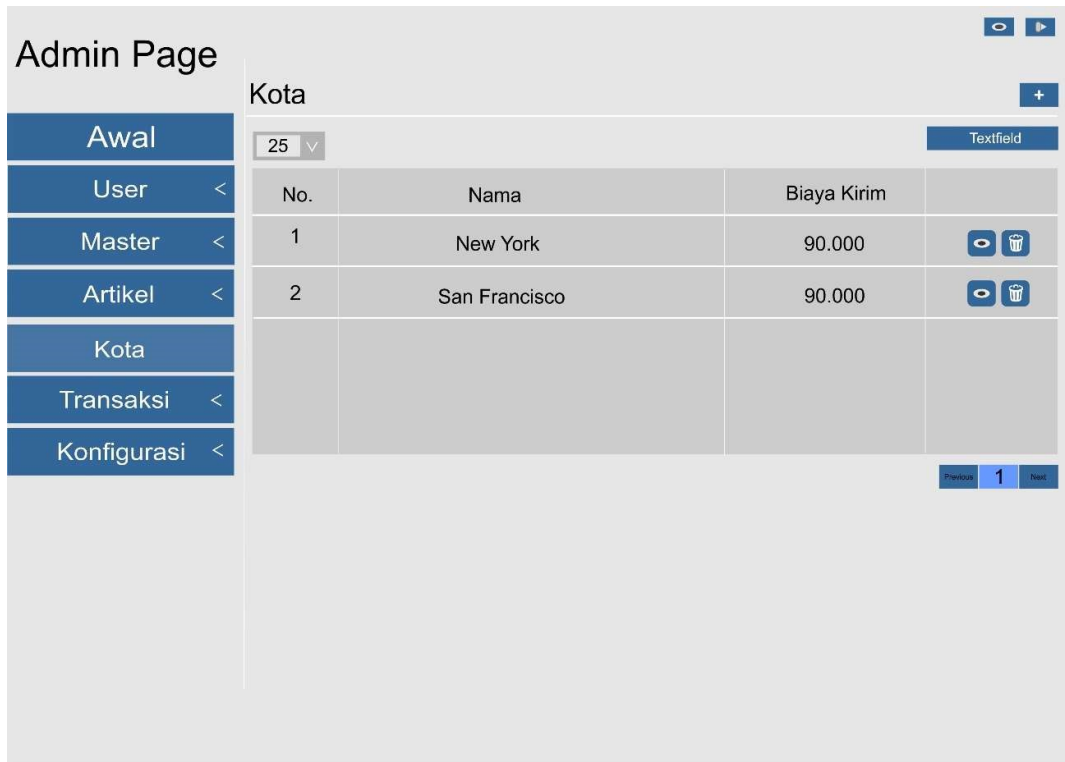


**Gambar 3.26** Rancangan Antarmuka Halaman Data Kategori



### 3.7.15 Rancangan Antarmuka Halaman Data Kota

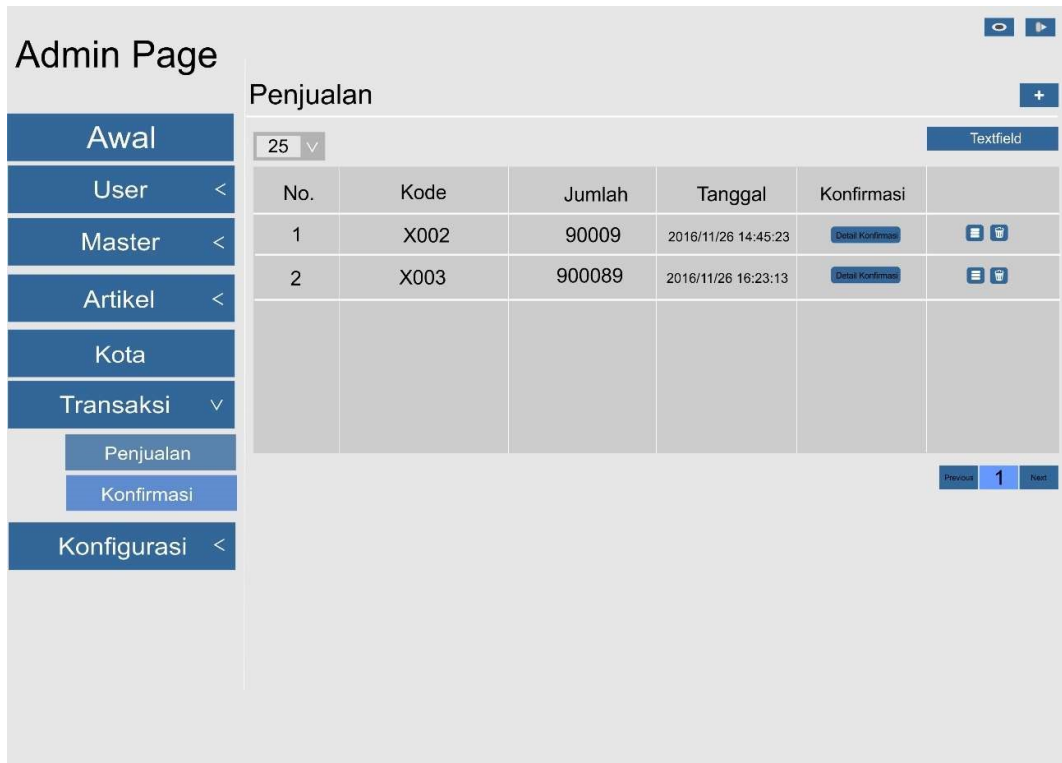
Rancangan antarmuka halaman data kota *admin* dapat melakukan penambahan dan pengurangan daftar kota serta mengubah informasi lainnya pada kota tersebut. Gambar rancangan antarmuka halaman data kota dapat dilihat pada Gambar 3.27.



**Gambar 3.27** Rancangan Antarmuka Halaman Data Kota

### 3.7.16 Rancangan Antarmuka Halaman Data Penjualan

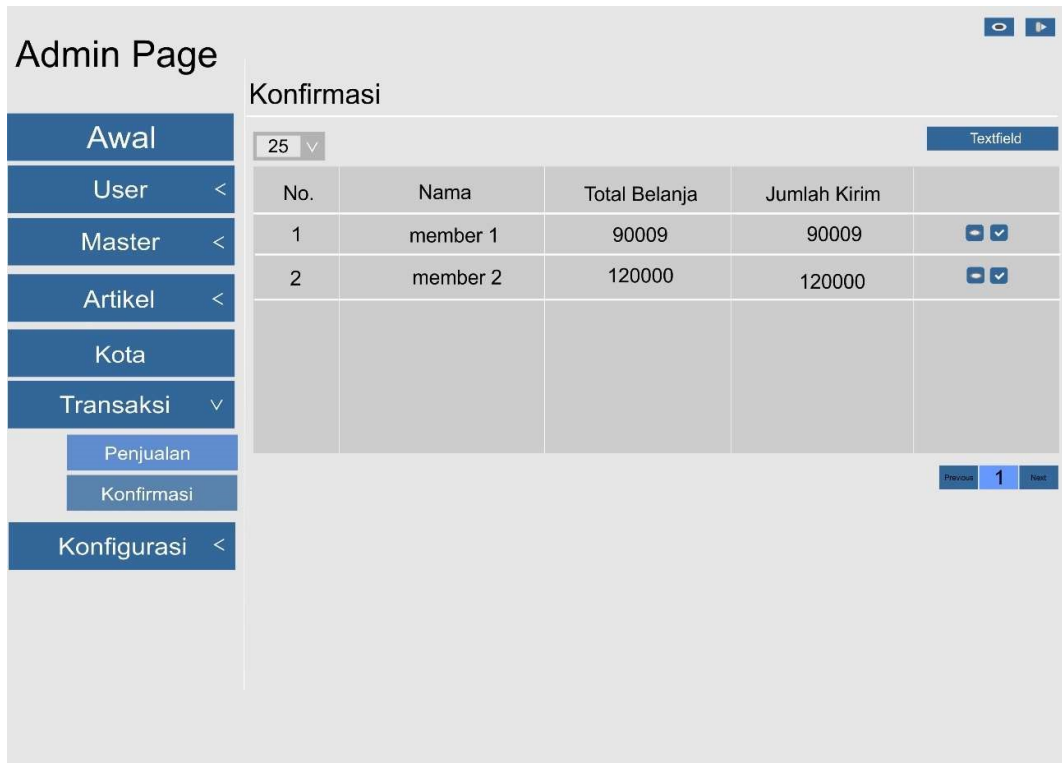
Rancangan antarmuka halaman data penjualan *admin*. Pada halaman ini *admin* dapat melihat daftar penjualan yang otomatis tercatat ketika *member* melakukan proses belanja. Gambar rancangan antarmuka halaman data penjualan dapat dilihat pada Gambar 3.28.



**Gambar 3.28** Rancangan Antarmuka Halaman Data Penjualan

### 3.7.17 Rancangan Antarmuka Halaman Data Konfirmasi

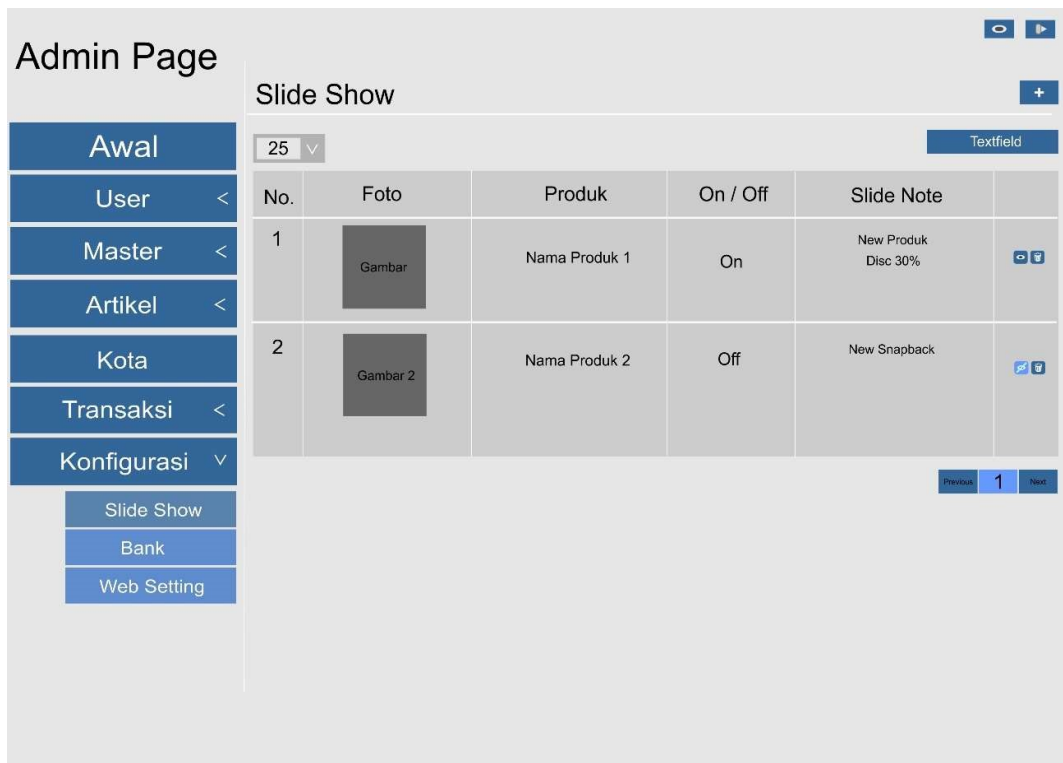
Rancangan antarmuka halaman data konfirmasi *admin*. Pada halaman ini *admin* dapat melihat detail konfirmasi dan melakukan konfirmasi dari daftar proses penjualan yang telah *member* lakukan. Gambar rancangan antarmuka halaman data konfirmasi dapat dilihat pada Gambar 3.29.



**Gambar 3.29** Rancangan Antarmuka Halaman Data Konfirmasi

### 3.7.18 Rancangan Antarmuka Halaman Konfigurasi *Slide Show*

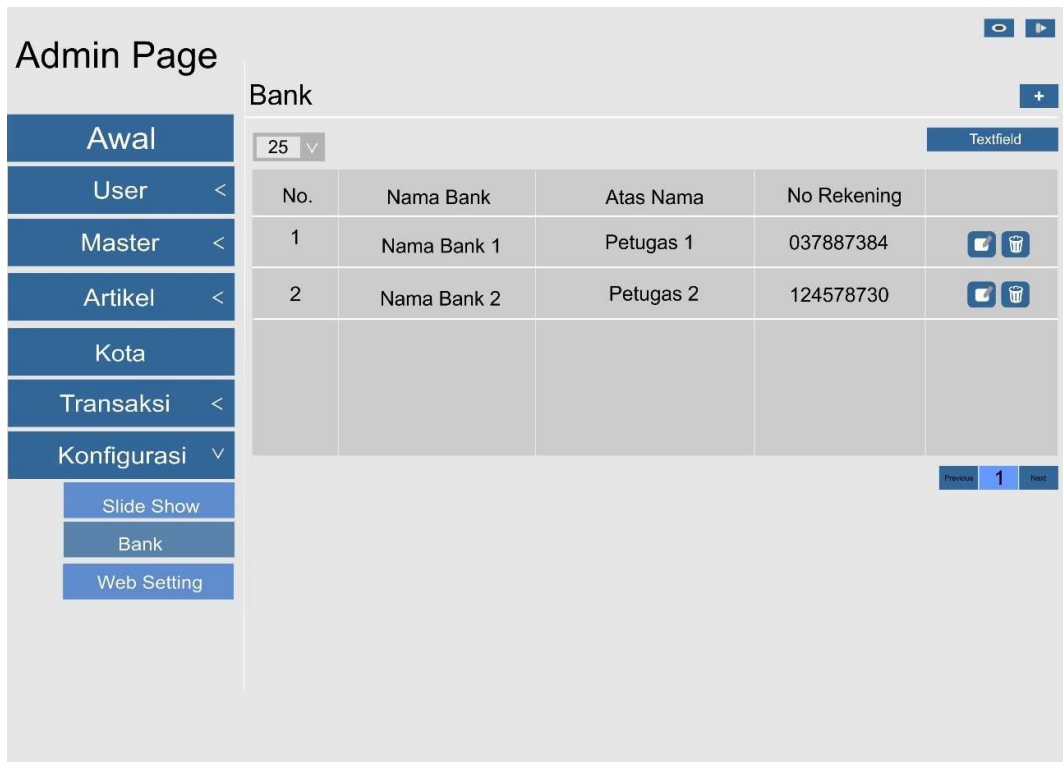
Rancangan antarmuka halaman konfigurasi slide show. Pada halaman ini *admin* dapat mengelola *slide* yang ingin ditampilkan. Gambar dari slide tersebut dapat diambil dari foto produk yang ada atau *admin* juga dapat menambahkan gambar lain untuk *slide* baru. Gambar rancangan antarmuka halaman konfigurasi *slide show* dapat dilihat pada Gambar 3.30.



**Gambar 3.30** Rancangan Antarmuka Halaman Konfigurasi Slide Show

### 3.7.19 Rancangan Antarmuka Halaman Konfigurasi Bank

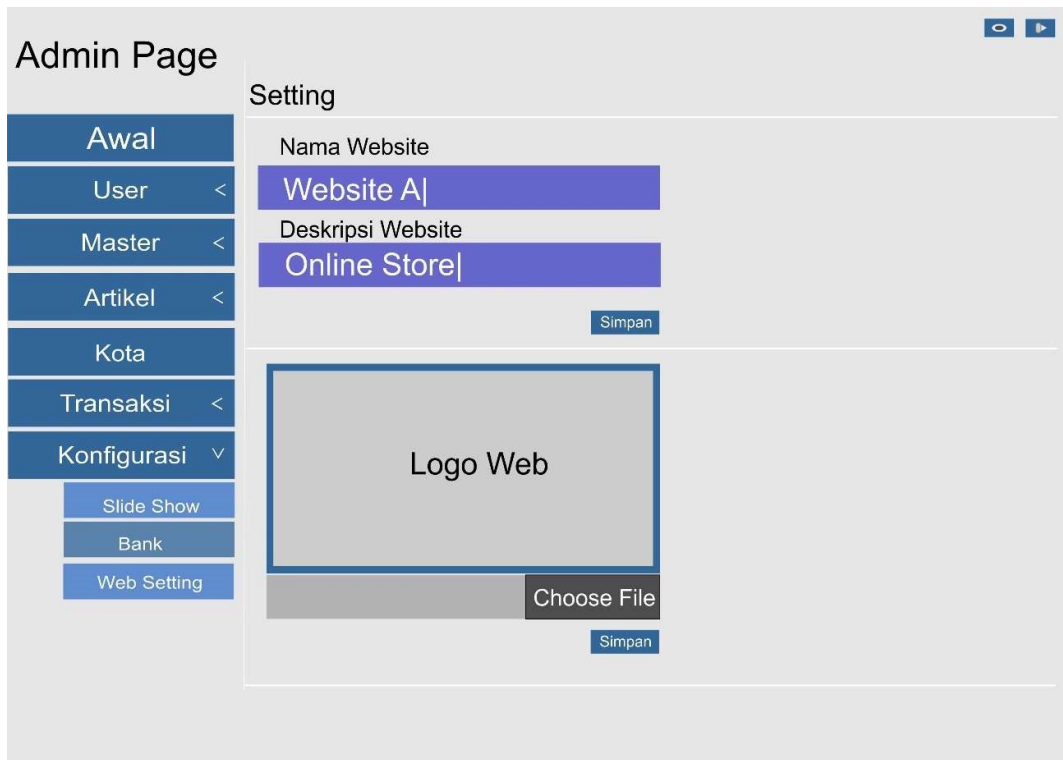
Rancangan antarmuka halaman konfigurasi Bank. Pada halaman ini *admin* dapat menambah dan menghapus bank. Gambar rancangan antarmuka halaman konfigurasi bank dapat dilihat pada Gambar 3.31.



**Gambar 3.31** Rancangan Antarmuka Halaman Konfigurasi Bank

### 3.7.20 Rancangan Antarmuka Halaman *Web Setting*

Rancangan antarmuka halaman *Web setting*. Pada halaman ini *admin* dapat merubah judul *website* dan mengganti logo *website*. Gambar rancangan antarmuka halaman *Web setting* dapat dilihat pada Gambar 3.32.



**Gambar 3.32** Rancangan Antarmuka Halaman *Web Setting*

### 3.8 Pengujian

Pengujian perangkat lunak merupakan suatu kegiatan yang dilakukan untuk memperoleh informasi serta mengevaluasi mengenai kualitas dari produk atau layanan yang sedang diuji. Tujuan pengujian dalam pengembangan aplikasi adalah untuk dapat memenuhi kebutuhan yang diperlukan dengan mendasari pada rancangan dan pengembangan perangkat lunak.

Metode pengujian yang dipakai dalam pengembangan aplikasi adalah *black box testing*. *Black box testing* atau tes fungsional adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak yang dikembangkan.

Hal-hal yang menjadi perhatian dalam pengujian adalah sebagai berikut:

- a. Aplikasi dapat menerima dan menyimpan hasil proses penjualan dari *member*.
- b. Aplikasi dapat menampilkan produk berdasarkan produk terbaru maupun kategori produk.
- c. Aplikasi dapat melakukan proses belanja dan proses konfirmasi dengan mudah.