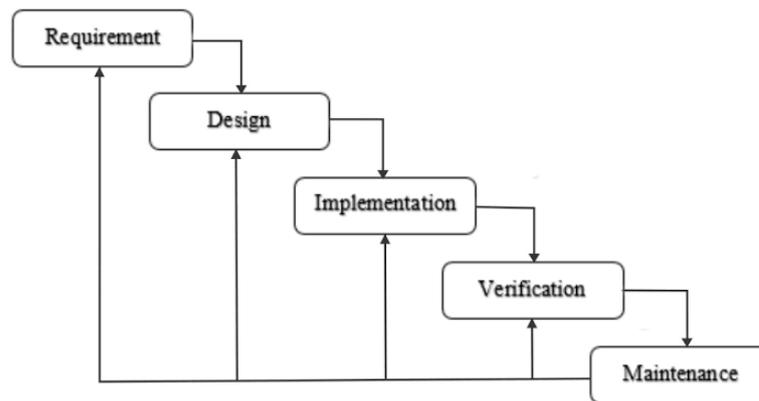


## BAB III METODOLOGI PENELITIAN

### 3.1 Metode Penelitian

Adapun metode penelitian yang digunakan adalah metode *waterfall*. metode *waterfall* merupakan suatu proses pengembangan perangkat lunak berurutan, dimana kemajuan dipandang terus mengalir ke bawah ( seperti air terjun ) melewati fase-fase perencanaan, pemodelan, implementasi dan pengujian. secara umum, penelitian ini terdiri dari 4 tahap yang dapat dilihat pada gambar 3.1.



**Gambar 3.1** *Waterfall*

#### 1. *Requirement*

Tahap perencanaan menyangkut studi tentang kebutuhan pengguna dan kelayakan sistem baik secara teknik maupun secara teknologi. Pada tahap ini dilakukan perencanaan tentang sistem yang akan dibangun. Dalam hal ini adalah *website* Pastel Ondrowino di Legend Coffee.

#### 2. *Design*

Proses *design* digunakan untuk mengubah kebutuhan-kebutuhan di atas menjadi representasi ke dalam bentuk *blueprint software* sebelum *coding* dimulai. Tahap *design* menyangkut perancangan sistem dimana kita akan memberikan solusi dari masalah yang muncul pada tahap analisis.

#### 3. *Implementation*

Tahap implementasi adalah perencanaan sistem diimplementasikan ke situasi nyata dengan pemilihan perangkat keras dan penyusunan desain (*coding*). Untuk implementasi sistem yaitu dengan memasukkan desain yang sudah dibuat ke dalam perangkat yang disediakan.

#### 4. *Verification*

Sesuatu yang dibuat haruslah diuji. Demikian juga *software*, semua fungsi-fungsi *software* harus diuji secara teliti agar *software* bebas dari *error*, dan hasilnya harus benar-benar sesuai dengan kebutuhan yang sudah didefinisikan pada tahap sebelumnya. Tahap pengujian juga merupakan tahap yang menentukan apakah desain yang sudah dibuat sudah sesuai kebutuhan pengguna atau belum. Tujuan dari pengujian ini adalah untuk meminimalisir cacat desain *website* sehingga sistem yang dikembangkan benar-benar dapat berjalan dengan sebaik mungkin. Pengujian ini akan dilakukan dengan mewawancarai beberapa orang sebagai penguji untuk menilai apakah sistem yang dibuat sesuai dengan kebutuhan Pastel Ondrowino di Legend Coffee.

## 5. Maintenance

Tahap pemeliharaan adalah tahap dilakukannya perawatan dan pemeliharaan *website*. Jika perlu akan dilakukan perbaikan kecil dikemudian jika periode sistem sudah habis. Maka akan kembali lagi pada tahap perencanaan.

### 3.2 Analisis Kelemahan Sistem

Dalam sistem yang berjalan saat ini pada legend coffee untuk pemesanan pastel ondrowino sendiri dianggap kurang efisien dalam proses pemesanan khususnya bagi konsumen yang ingin memesan secara *online*, dikarenakan dalam web legend coffee belum memiliki sistem pemesanan secara *online* untuk pastel ondrowino. Dengan sistem yang berjalan saat ini dianggap kurang efektif karena promosi untuk penjualan pastel ondrowino dengan sistem saat ini akan memakan waktu lebih lama dibandingkan dengan menggunakan *system online*, selain itu konsumen memerlukan biaya tambahan yaitu biaya telepon jika ingin menghubungi pihak cafe, dan konsumen yang memiliki jarak yang cukup jauh tetapi menginginkan untuk memesan pastel ondrowino memerlukan waktu perjalanan agar sampai di cafe legend coffee, dimana hal itu akan lebih mudah dilakukan dengan pemesanan *online*. Karena dengan pemesanan melalui *online* atau *web* konsumen dapat menyingkat waktu dan lebih praktis dalam proses pemesanan pastel ondrowino yang sebelumnya masih menggunakan cara manual.

### 3.3 Analisis Kebutuhan Sistem

#### 3.3.1 Analisis Kebutuhan Sistem

Dalam perancangan sistem informasi yang sangat diperlukan adalah seberapa jauh pihak *user* yang dihasilkan dari aplikasi tersebut. dalam hal ini *output* yang diperlukan adalah laporan penjualan dan laporan stok makanan. Untuk itu informasi dibutuhkan dalam perancangan sistem ini adalah bagaiman

seorang admin atau karyawan menginput data makanan ke sistem dan bagaimana proses pemesanan yang dilakukan oleh pelanggan. berikut analisis kebutuhan sistem fungsional :

1. *Member*

- a. Sistem harus dapat *update profil member*.
- b. Sistem harus dapat melakukan transaksi pemesanan.
- c. Sistem harus dapat melakukan konfirmasi pembayaran.
- d. Sistem harus dapat menampilkan produk makanan.

2. Publik

Publik hanya akan dapat melakukan *view* produk makanan saja. jika publik ingin melakukan pemesanan makanan maka publik harus mendaftarkan diri sebagai *member*.

3. Admin

- a. Sistem harus mengubah dan dapat menampilkan data *member*.
- b. Sistem harus dapat menghapus data *member*.
- c. Sistem harus dapat mengolah data pemesanan.

4. Super Admin

- a. Sistem harus dapat *entry* data makanan.
- b. Sistem harus dapat olah data admin.
- c. Sistem harus dapat olah d
- d. Sistem harus dapat olah d
- e. Sistem harus dapat olah d
- f. Sistem harus dapat olah data pengaturan.
- g. Sistem harus dapat menampilkan laporan penjualan.

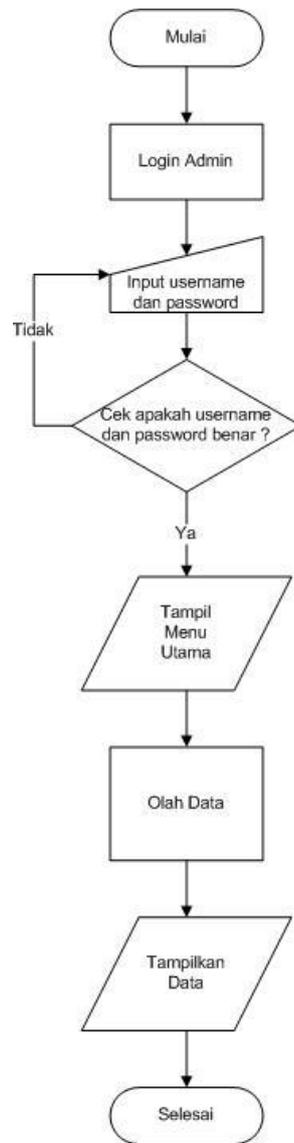
### **3.3.2 Analisis Kebutuhan Sistem Non Fungsional**

1. Operasional

a. Perangkat Keras

- i. Menggunakan sistem operasi windows xp (bisa juga dibuka dengan menggunakan sistem operasi windows 7, *machintos*, *linux*).
- ii. *Proccesor intel pentium iv* atau lebih.

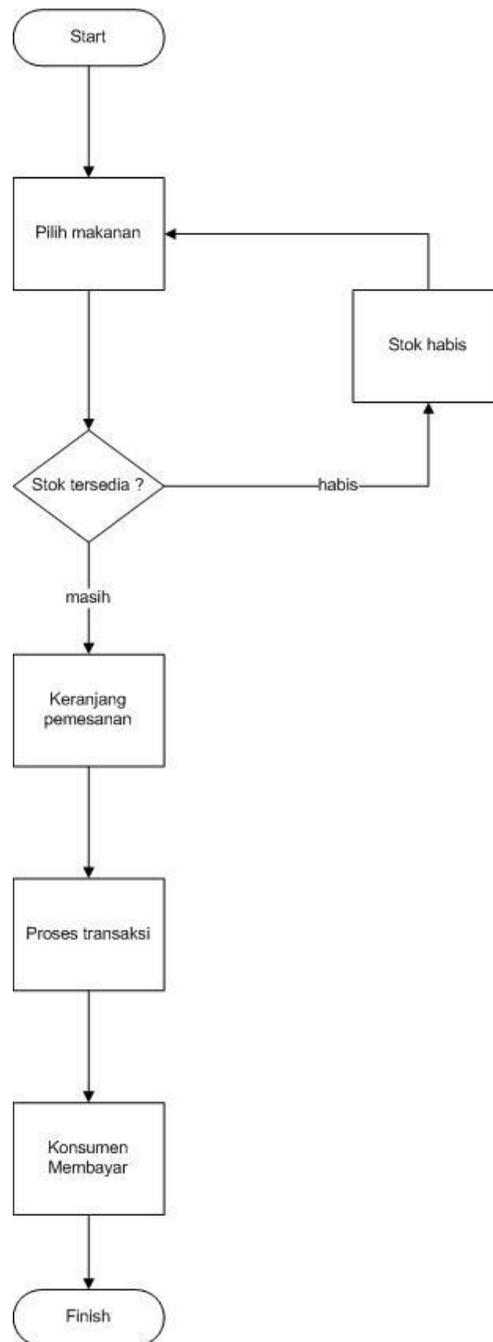




**Gambar 3.2** *Flowchart Set Up Admin*

Setelah admin melakukan *update* data makanan maka data makanan akan tertampil di halaman *web* publik. Pelanggan akan memilih makanan mana yang akan dibeli. jika makanan masih tersedia maka bisa dipesan untuk kemudian melakukan proses transaksi pemesanan. setelah selesai melakukan pemesanan maka pelanggan memperoleh nomor order untuk melakukan pembayaran seperti terlihat pada gambar 3.3 *flowchart* pemesanan di bawah ini

:



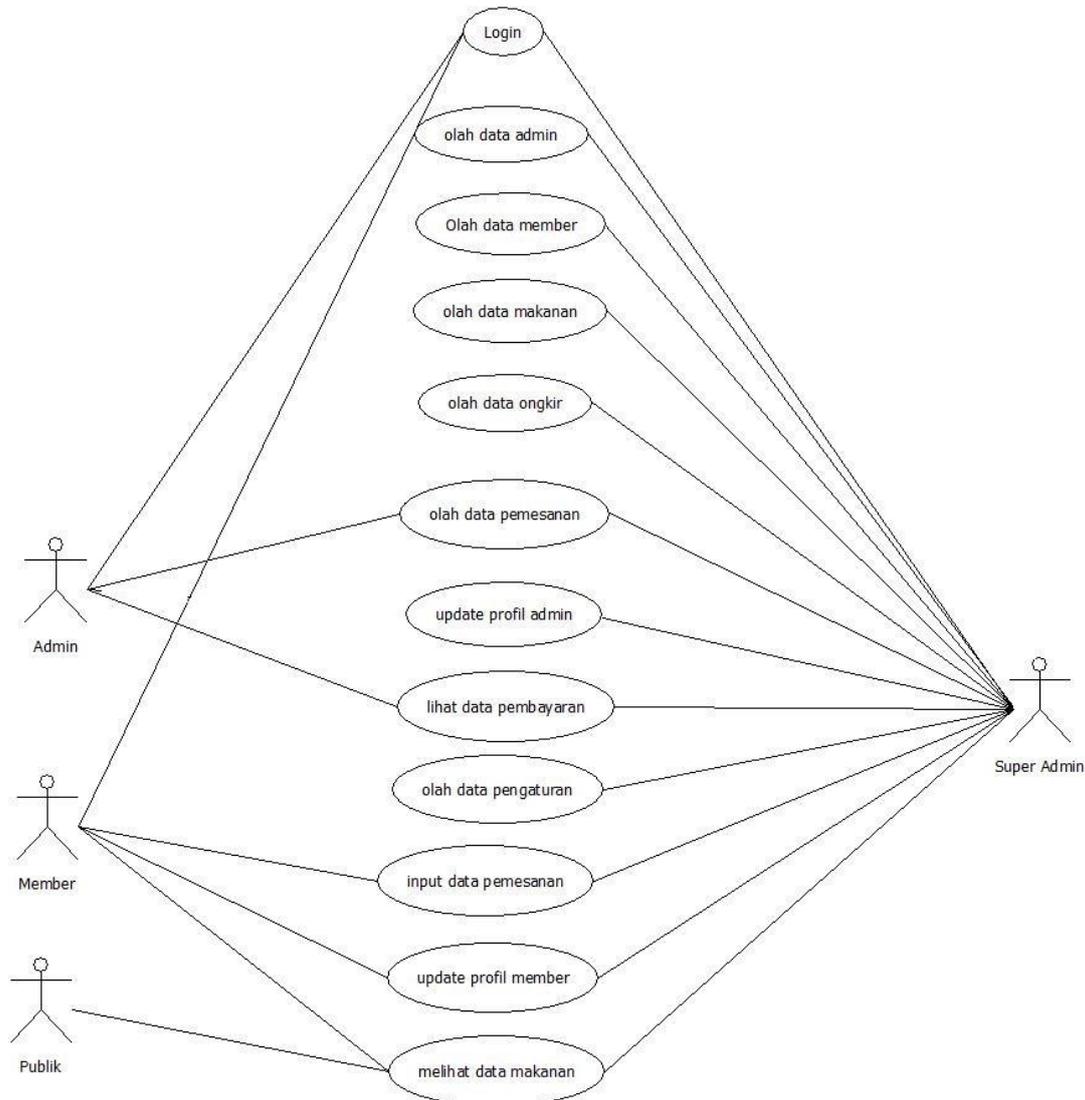
**Gambar 3.3** *Flowchart* Pemesanan

### 3.4.2 Use Case Diagram

*Use Case Diagram* melibatkan 4 aktor. yaitu super admin, admin, *member* dan publik yang memiliki peran berbeda. Super admin memiliki hak akses yang lebih banyak dan spesifikasi. Super admin dapat menambah data admin, d

elola aplikasi dan mempunyai Pada sistem ini seorang super admin, data ongkos kirim, olah

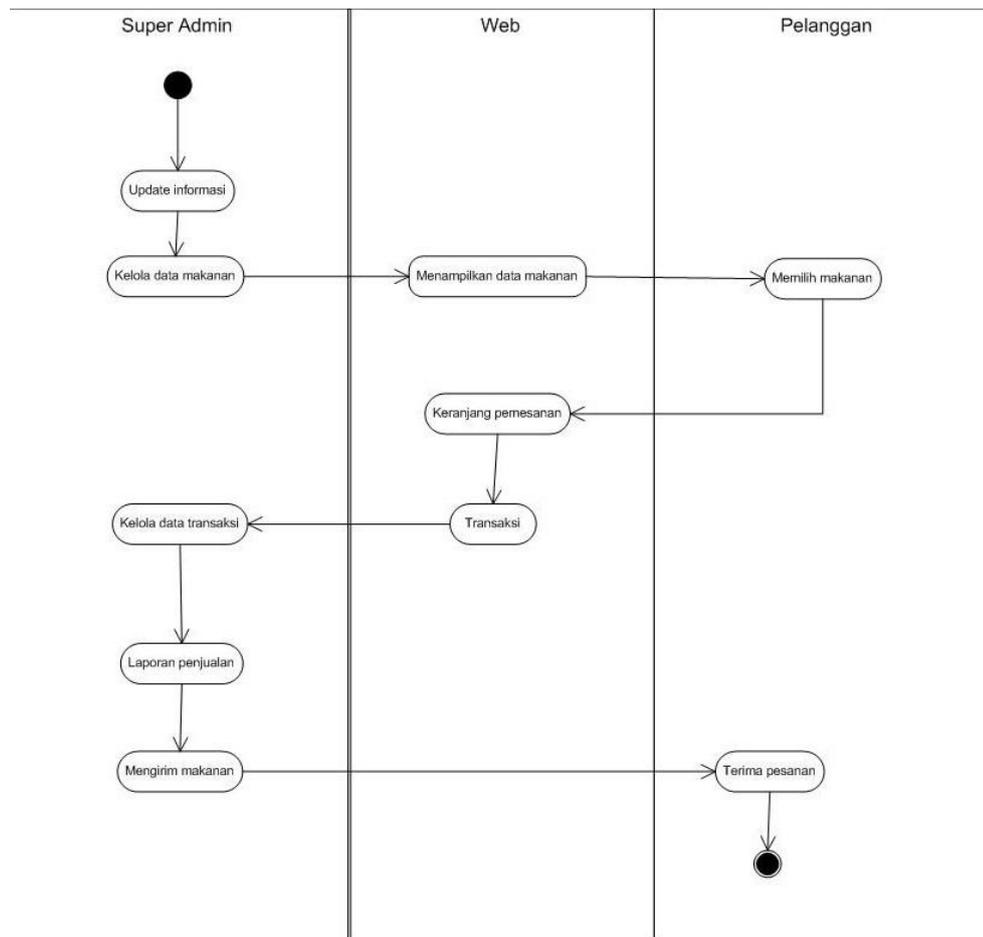
data jadwal dan olah data pemesanan. seorang admin dapat mengolah data pemesanan dan *update* profil admin. Sedangkan seorang member dapat melakukan pemesanan makanan. Sebelum melakukan salah satu dari 5 proses tersebut, admin diharuskan melakukan *login* terlebih dahulu, kemudian akan dilakukan verifikasi (pengecekan) kebenaran data *login* (*user name* dan *password*) dengan cara dibandingkan dengan data dalam *database server*. Apabila *verifikasi login* tersebut gagal, maka admin dapat mengulang kembali proses *login*. perancangan *use case* diagram ditampilkan pada gambar 3.4 :



**Gambar 3.4** Use Case Diagram

### 3.4.3 Activity Diagram

Berdasarkan *use case* sebelumnya maka didapatkan aktor yang terlibat dalam *use case diagram*. *Activity diagram* ini menggambarkan berbagai alir aktifitas dalam sistem yang sedang dirancang. *Activity diagram* dalam aplikasi ini terdiri dari beberapa proses. perancangan *activity diagram* sistem pemesanan makanan dapat dilihat pada gambar 3.5 :



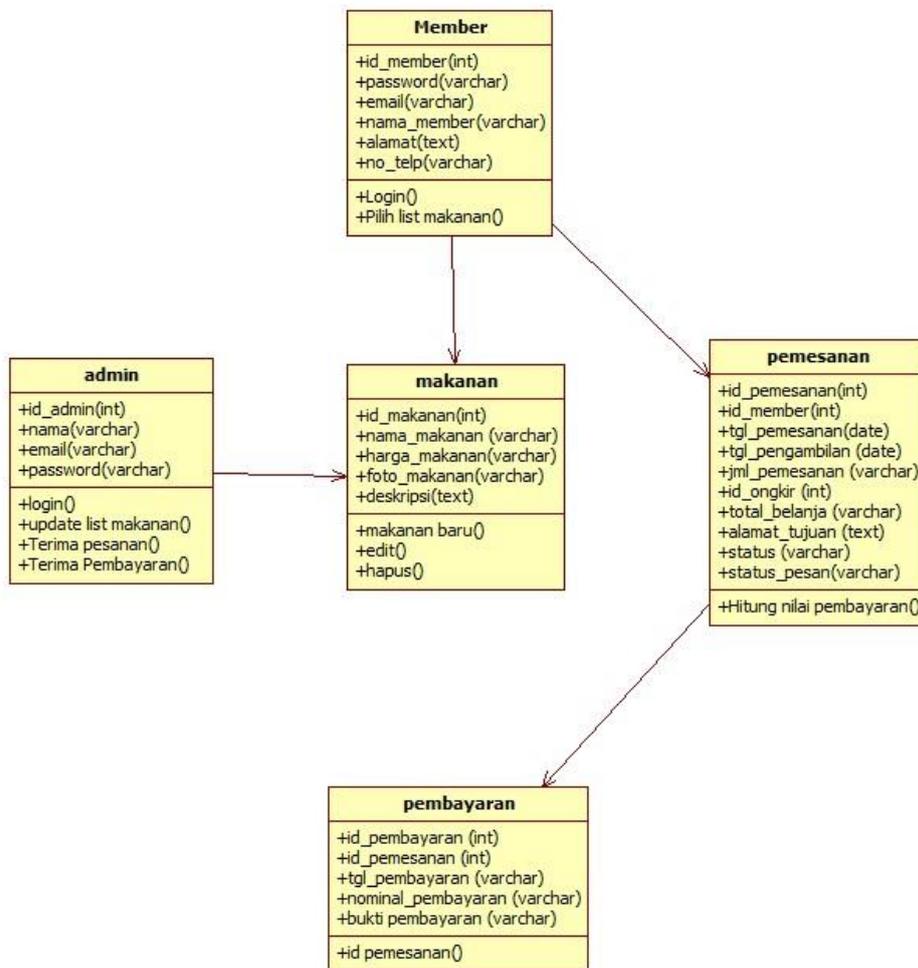
**Gambar 3.5** Activity Diagram Sistem Pemesanan Makanan

Penjelasan tentang gambar 3.5 adalah sebagai berikut terdapat alur dari kegiatan input data oleh admin yaitu dimulai dari melakukan login. setelah admin berhasil *login* maka melakukan *input* atau *update* data maka data akan ditampilkan di halaman web publik. data yang tertampil di halaman publik bisa dilihat oleh pelanggan untuk kemudian pelanggan memilih menu produk yang akan dibeli. setelah memilih makanan yang akan dibeli maka pelanggan mengolah data pemesanan pada keranjang belanja (*cart*). Pelanggan melakukan transaksi *check out* untuk mendapatkan nomor pembelian untuk kemudian melakukan pembayaran setelah selesai melakukan olah data pesanan di menu keranjang belanja. Admin mendapatkan laporan pemesanan dan kemudian mengirim makanan berdasarkan data alamat yang diinputkan oleh pelanggan.

### 3.4.4 Class Diagram

*Class diagram* dibuat untuk menjelaskan hubungan antara kelas yang satu dengan kelas yang lainnya. didalam beberapa kelas, terbut yang menjelaskan komponen apa saja

yang terdapat didalam kelas tersebut dan terdapat *operation* yang berfungsi sebagai penjelas kegiatan apa saja yang dapat dilakukan oleh suatu aktor ketika berada didalam kelas tersebut. berikut perancangan *class* diagram sistem pemesanan makanan pada gambar 3.6:



**Gambar 3.6** Perancangan *Class Diagram* Sistem Pemesanan Makanan

Keterangan Diagram :

1. *Class* admin bisa berdiri sendiri.
2. *Class* makanan tidak akan ada jika tidak ada *class* admin.
3. *Class* pemesanan tidak akan ada jika tidak ada *class* makanan dan *class* member.
4. *Class* pembayaran tidak akan ada jika tidak ada *class* pemesanan sehingga tidak bisa berdiri sendiri.

### 3.5 Perancangan Basis Data

#### 3.5.1 Perancangan Struktur Tabel

Basis data pada sistem ini digunakan untuk menyimpan informasi data reservasi. berikut perancangan struktur tabel yang akan digunakan dalam *database* db\_pastel.

##### 1. Tabel admin

Tabel admin berisi data-data admin yang mengelola sistem. Pada tabel tersebut terdapat beberapa *field* yaitu *id\_admin*, *password*, *nama\_admin*, *email*, *no\_telp*, *alamat*. rancangan struktur tabel *user* dapat dilihat pada tabel 4.5 berikut:

**Tabel 3. 1** Struktur Tabel User

<i>Field</i>	<i>Type</i>	<i>Size</i>	<i>Index</i>
Id_admin	Int	11	<i>Primary Key</i>
Password	Varchar	25	
nama_admin	Varchar	25	
Email	Varchar	25	
no_telp	Varchar	25	
Alamat	Text		

##### 2. Tabel Super admin

Tabel super admin digunakan untuk menyimpan data super admin atau pemilik cafe.

**Tabel 3. 2** Struktur Tabel Super Admin

	<b>Field</b>	<b>Type</b>	<b>Length</b>	<b>Extra</b>
*	id_super	Int	11	Primary key
	<b>Field</b>	<b>Type</b>	<b>Length</b>	<b>Extra</b>
	Nama_super		25	
	Email		25	
	Password		25	



3	Email	Varchar	25	
4	Password	Varchar	25	
5	No_telp	Varchar	15	
6	Alamat	Text		
7	Instansi	Varchar	25	

#### 6. Tabel Ongkir

Tabel ongkir digunakan untuk menyimpan data ongkos kirim setiap kota.

**Tabel 3. 6** Struktur Tabel Ongkir

	Field	Type	Length	Extra
1	id_ongkir	Int	5	<i>Primary Key</i>
2	Provinsi	varchar	25	
3	Harga	Int	10	

#### 7. Tabel Pemesanan

Tabel pemesanan digunakan untuk menyimpan data pemesanan yang diinputkan oleh pelanggan.

**Tabel 3. 7** Struktur Tabel Pemesanan

	Field	Type	Length	Extra
1	id_pemesanan	Int	11	<i>Primary key</i>
2	Catatan	Text		
3	Id_member	Int	11	
4	Tgl_pengambilan	Date		
	Field	Type	Length	Extra
5	Jml_pemesanan	Va		
6	Id_ongkir	Int		
7	Total_belanja	Va		
8	Alamat_tujuan	Text		
9	Keterangan	Varchar	25	
10	Ongkir	Int	11	

#### 8. Tabel Pengaturan

Tabel pengaturan digunakan untuk menyimpan data pengaturan nomor rekening pembayaran.

**Tabel 3. 8** Struktur Tabel Pengaturan

	<b>Field</b>	<b>Type</b>	<b>Length</b>	<b>Extra</b>
1	id_pengaturan	Int	11	<i>Primary key</i>
2	Kolom	Varchar	25	
3	Isi	Text		

9. Tabel Detail Pemesanan

Tabel detail pemesanan digunakan untuk menyimpan data makanan yang dipesan oleh pelanggan.

**Tabel 3. 9** Struktur Tabel Detail Pemesanan

	<b>Field</b>	<b>Type</b>	<b>Length</b>	<b>Extra</b>
1	id_detail_pemesanan	Int	11	<i>auto_increment</i>
2	Id_pemesanan	Int	11	
3	Id_makanan	Int	11	
4	Nama_makanan	varchar	25	
5	Harga_makanan	varchar	25	
6	Jml_pesanan	varchar	25	
7	Sub_total_harga	varchar	25	

10. Tabel Pembayaran

Tabel pembayaran digunakan untuk n pemesanan yang sudah melakukan pembayaran.

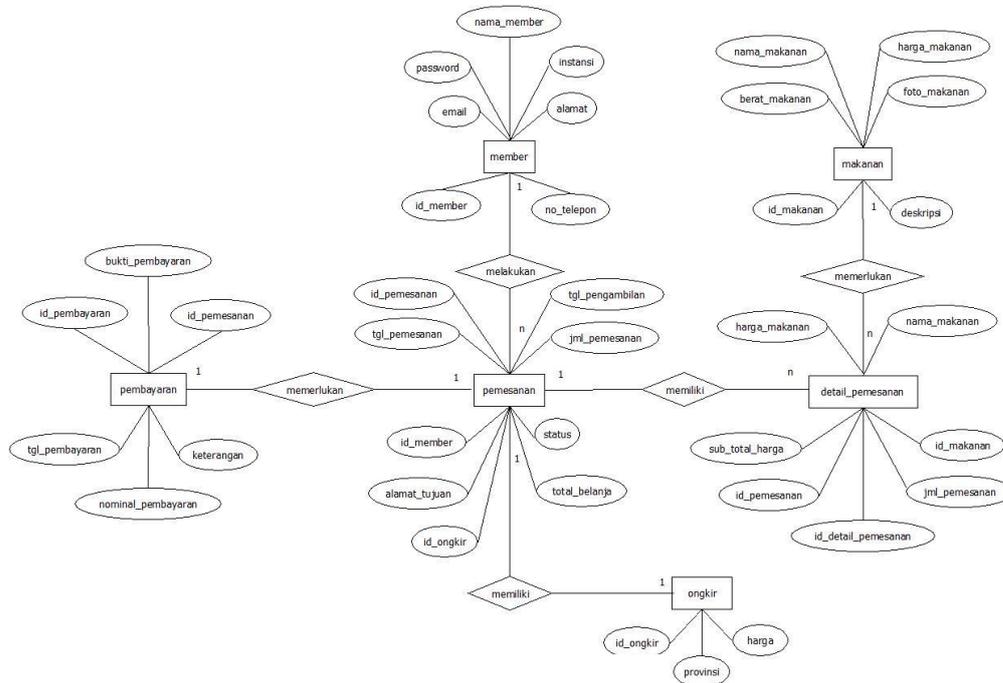
**Tabel 3.9** Struktur Tabel Pembayaran

	<b>Field</b>	<b>Type</b>	<b>Length</b>	<b>Extra</b>
1	id_pembayaran	Int	11	<i>auto_increment</i>
	<b>Field</b>	<b>Type</b>	<b>Length</b>	<b>Extra</b>
2	Id_pemesanan	Int	11	
3	Tgl_pembayaran	Date		
4	Nominal_pembayaran	varchar	25	
5	Bukti_pembayaran	varchar	25	

6	Keterangan	Text		
---	------------	------	--	--

### 3.3.1 Perancangan ERD

*Entity Relationship Diagram (ERD)* merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan relasi. Adapun ERD yang terdapat dalam sistem penjualan berbasis web dapat dilihat pada Gambar 3.7.



Gambar 3.7 Perancangan ERD Sistem Pemesanan Makanan

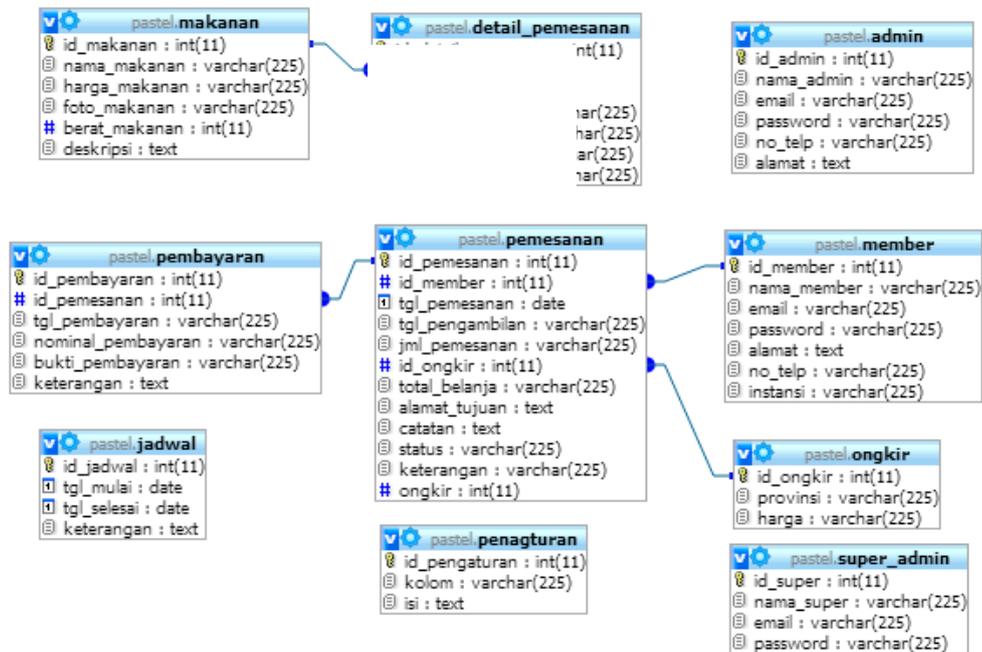
### 3.3.2 Relasi Tabel

Relasi tabel menggambarkan hubungan antara satu tabel dengan tabel yang lainnya yang berfungsi untuk mengatur operasi suatu *database*. Dalam sebuah relasi *database* perlu diketahui istilah sebagai berikut :

1. *Primary Key*. Sebuah *field* atau atribut yang ditandai sebagai kunci utama dari sebuah table dengan tujuan untuk membuat nilai dari *field primary key* sebagai nilai yang unik. Pada table *database* db\_standar yang berperan sebagai *primary key* adalah *field* id\_member, id\_admin, id\_makanan, id\_pemesanan, id\_pengaturan, id\_ongkir, id\_hubungi dan id\_pembayaran.
2. *Foreign Key*. Sebuah Field dalam sebuah table dimana *field* tersebut berasal dari *primary key* dari sebuah table lain dengan tujuan untuk membuat relasi dengan table

tersebut. Pada table *database* db\_standar yang berperan sebagai *foreign key* adalah field *id\_member*, *id\_ongkir*, *id\_pemesanan* dan *id\_makanan*.

Dibawah ini merupakan relasi tabel *database* db\_pastel.

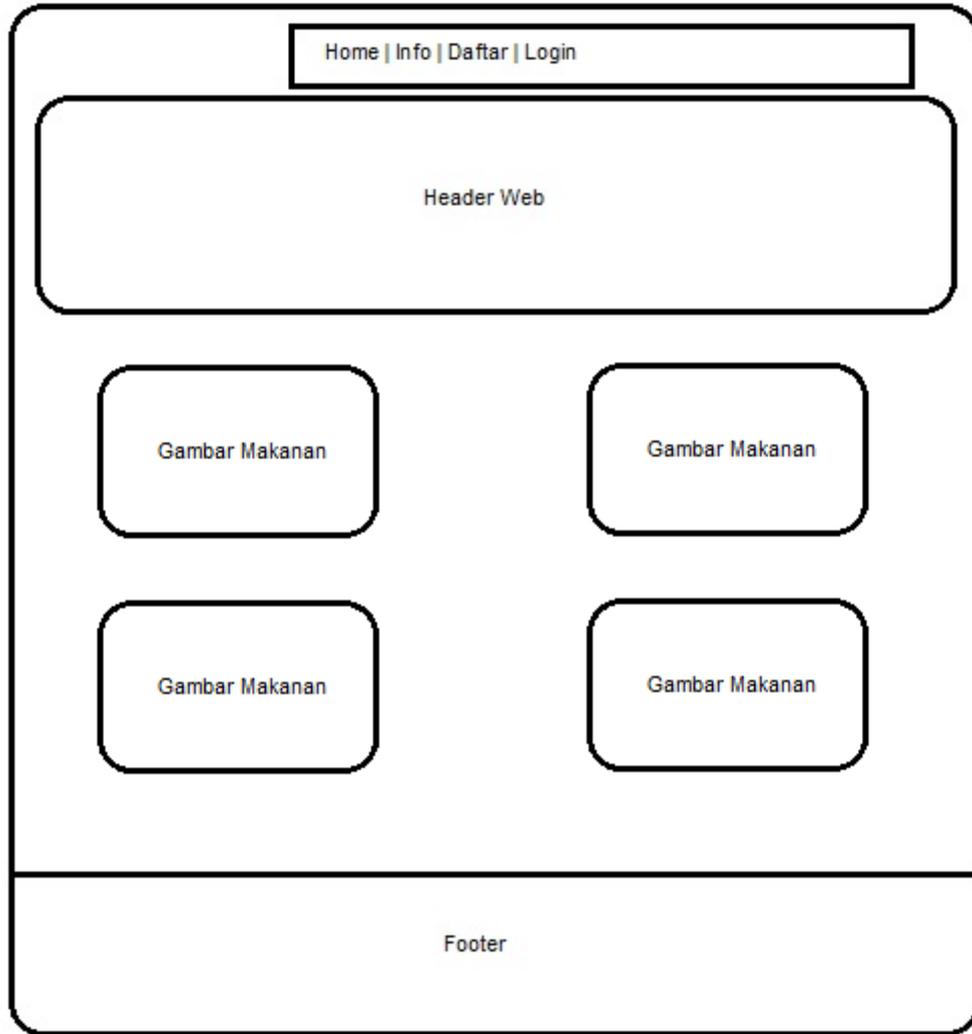


**Gambar 3.8** Relasi Tabel Basis Data db\_pastel

### 3.4 Perancangan Antarmuka

#### 3.4.1 Halaman Utama Publik

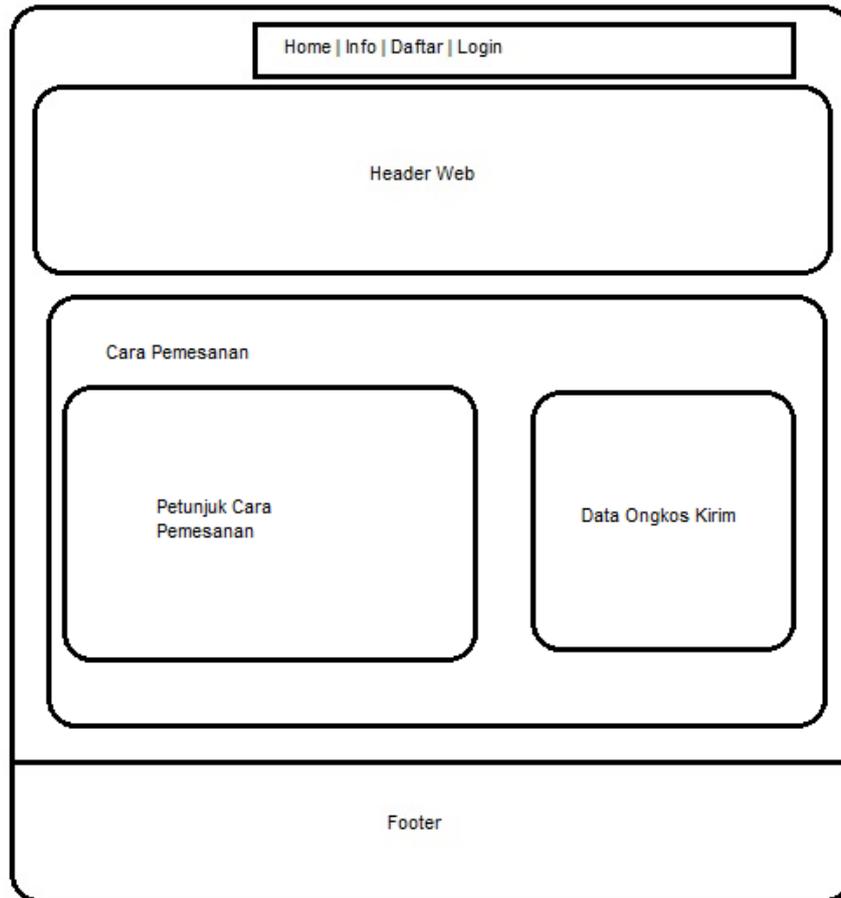
Pada bagian ini menggambarkan rancangan tampilan *home* pada halaman publik. Pada halaman ini menampilkan 6 macam produk terbaru. Seorang Publik hanya mempunyai akses untuk melihat data produk saja. Untuk dapat melakukan pemesanan, publik harus mendaftar sebagai member terlebih dahulu. Berikut rancangan halaman *index home* publik :



**Gambar 3.9** Rancangan Halaman Utama Publik

### 3.4.2 Halaman Info

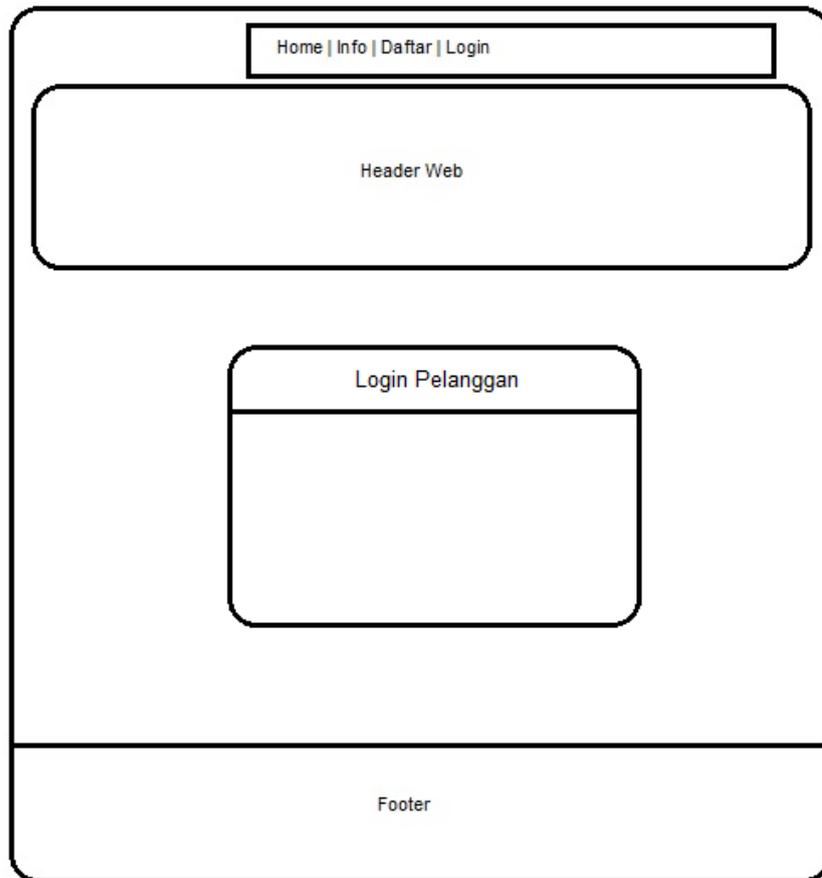
Pada bagian halaman info ini akan menampilkan tata cara pemesanan dan data ongkos kirim di tiap kota yang telah diinputkan oleh admin. Berikut rancangan halaman info :



**Gambar 3.10** Rancangan Halaman Info

### 3.4.3 Halaman *Login Member*

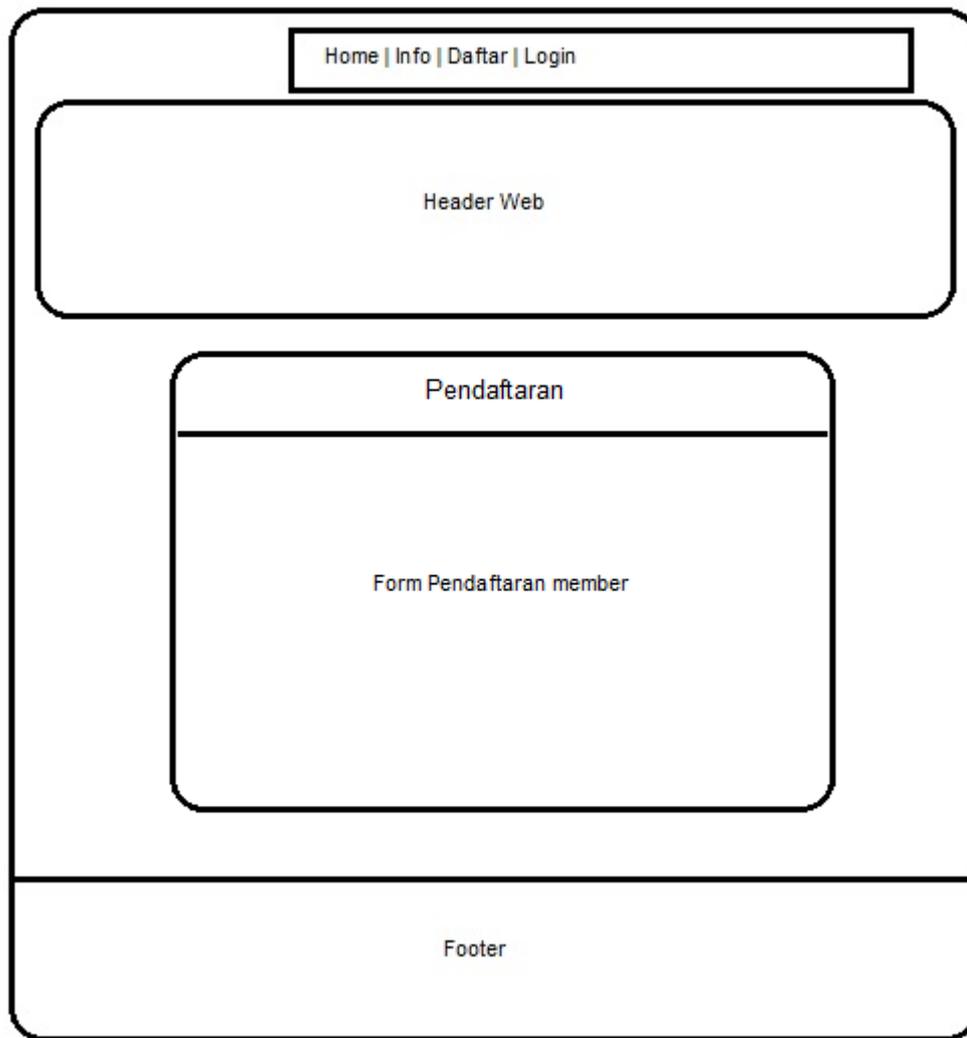
Pada halaman ini menampilkan *form login* untuk member. Pada halaman ini member diminta untuk memasukkan email dan *password* yang sudah pernah didaftarkan sebelumnya. Berikut tampilannya :



**Gambar 3.11** Rancangan Halaman *Form Login Member*

#### **3.4.4 Halaman Daftar**

Pada halaman ini akan menampilkan *form* pendaftaran *user* publik sebagai member. Setelah mendaftar sebagai member maka member tersebut bias melakukan pemesanan makanan dengan login terlebih dahulu. Berikut tampilannya :



**Gambar 3.12** Rancangan Halaman Admin

### 3.4.5 Halaman *Login* Admin atau Super Admin

Pada halaman ini menampilkan *form login* untuk admin. Pada halaman ini user diminta untuk memasukkan email dan *password* yang sudah pernah didaftarkan sebelumnya. Berikut tampilannya :

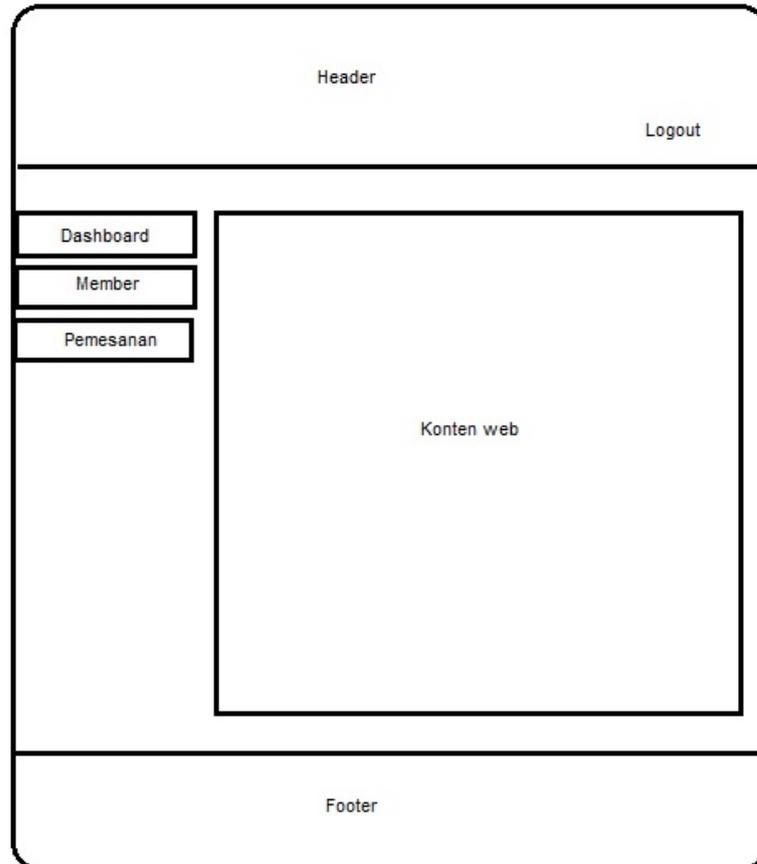
The form is titled 'LOGIN ADMIN' and includes the following fields and a button:

- Username:
- Password:
- Login

**Gambar 3.13** Rancangan Halaman Login Admin atau Super Admin

### 3.4.6 Halaman Admin

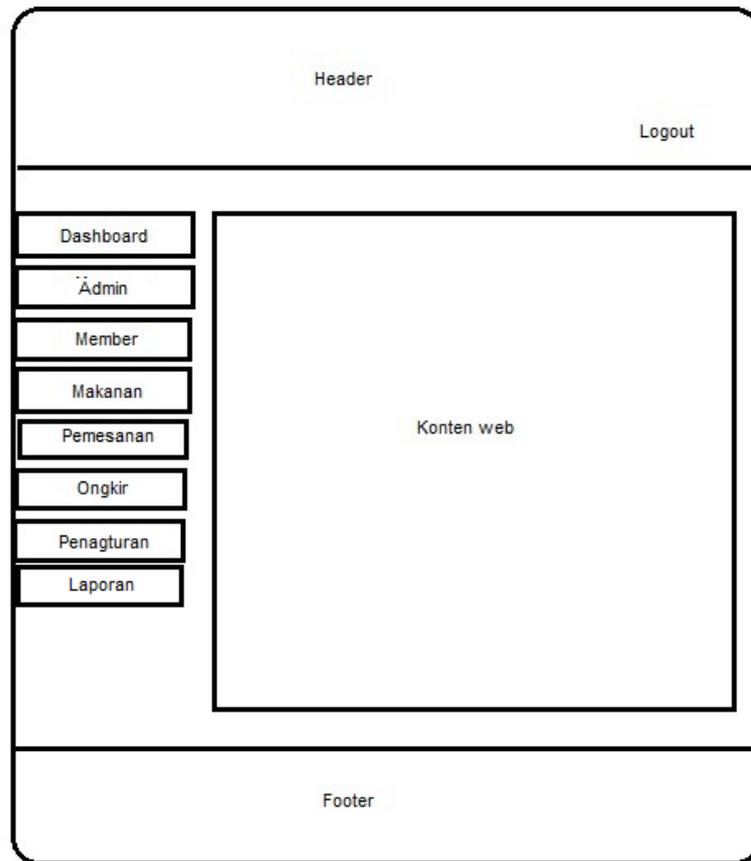
Pada halaman ini menampilkan halaman utama untuk admin. Setelah berhasil login maka admin akan diarahkan ke halaman utama. Seorang admin mempunyai hak akses untuk mengolah data member dan pemesanan saja. Berikut tampilannya :



**Gambar 3.14** Rancangan Halaman Admin

### 3.4.7 Halaman Super Admin

Pada halaman ini menampilkan halaman utama untuk super admin. Setelah berhasil login maka admin akan diarahkan ke halaman utama. Seorang super admin mempunyai hak akses untuk mengolah data member, data pemesanan, data jadwal, data pengaturan, data pembayaran dan pemesanan. Berikut tampilannya :



**Gambar 3.15** Rancangan Halaman Super Admin