

## **BAB 3**

### **METODOLOGI PENELITIAN**

#### **3.1 Metode Pengumpulan Data**

metode pengumpulan data ini, ekstraksi informasi dari penelitian yang dilakukan memegang peranan penting. Data topik yang relevan merupakan indikator keberhasilan penyelidikan. Pengumpulan data harus sangat tepat.

Metode pengumpulan data yang digunakan dalam penelitian ini yaitu sebagai berikut:

##### **3.1.1 Wawancara**

Wawancara merupakan suatu metode pengumpulan data dimana data di dapatkan dari narasumber yang terkait dengan topik bahasan. Pada penelitian ini peneliti mengajukan beberapa pertanyaan kepada pihak penjual dan pembeli di kantin Darmajaya Society Centre (DSC)

##### **3.1.2 Kuisisioner**

Kuisisioner adalah suatu metode pengumpulan data dimana data di dapatkan dari membagikan form pertanyaan yang akan di isi oleh orang yang memiliki hubungan dengan topik permasalahan. Dalam penelitian ini kuisisioner dibagikan dengan tujuan untuk mengetahui respon pembeli dan penjual di kantin Darmajaya Society Centre (DSC) terhadap aplikasi yang di bangun.

##### **3.1.3 Studi Literatur**

Studi literatur di lakukan dengan mengumpulkan literatur yang bersumber dari jurnal, buku atau hasil karya orang lain yang berkaitan dengan penelitian ini.

## **3.2 Pengembangan Perangkat Lunak**

### **3.2.1 Menentukan Kebutuhan Proyek**

Di tahap ini dilakukan proses pengumpulan kebutuhan terkait data yang nantinya akan digunakan dalam membangun aplikasi berbasis android ini.

### **3.2.2 Membuat Prototype Desain Sistem**

Tahap ini merupakan tahap yang bertujuan pada desain tampilan aplikasi yang nantinya akan di implementasikan kedalam bentuk code program

### **3.2.3 Implementasi dan pengkodean**

Tahap ini adalah tahap mengubah desain prototype sistem menjadi sebuah sistem informasi berbasis mobile dengan menggunakan Bahasa pemrograman yang telah ditentukan sebelumnya.

### **3.2.4 Testing atau Pengujian**

Setelah melakukan pengkodean program dan aplikasi sudah dirancang, sebelum menyerahkan aplikasi ini kepada user, maka harus di uji terlebih dahulu kinerja dari aplikasi ini apakah sudah sesuai dengan yang di harapkan atau tidak. Tahap pengujian ini akan menggunakan metode black box testing. Dimana blackbox testing adalah testing yang dilakukan di akhir proses yang meliputi pengujian kinerja loading, pengujian interface, pengujian database dan pengujian

### **3.2.5 Penerapan Program dan Pemeliharaan**

Setelah melakukan pengujian dan aplikasi berjalan sesuai dengan tujuan yang diharapkan, tahap selanjutnya adalah memberikan program tersebut ke pengguna yang akan menggunakan aplikasi, serta di lakukannya pemeliharaan aplikasi dengan tujuan untuk menjaga agar aplikasi tetap berjalan sesuai dengan fitur yang telah diharapkan.

### 3.3 Implementasi Metode FIFO dalam Memproses orderan di dalam aplikasi ini

Pada kantin (DSC) terdapat 9 stan yang menjual makanan dan minuman dan terdapat dua kantin yaitu kisheal dan TITIS. Terdapat beberapa menu yang sudah mereka inputkan beserta harganya. Sekelompok mahasiswa Teknik Informatika Angkatan 2018 yang sudah meregister kedalam aplikasi ingin menggunakan aplikasi ini dan ingin memesan menu yang ada di Kisheal dan TITIS. Berikut adalah contoh tabel implementasi dari dua kantin tersebut:

Urutan Order ke-	Nama Customer	Jam Order	Nama Kantin
1	Jono	09.45	Kisheal
2	Bella	09.49	
3	Endang	10.02	
4	Martin	10.03	
1	Khodijah	11.45	TITIS
2	Bayu	11.46	
3	Julian	11.49	
4	Mukhlas	12.10	

Berikut adalah cara metode fifo dalam memproses urutan dalam pemesanan tersebut

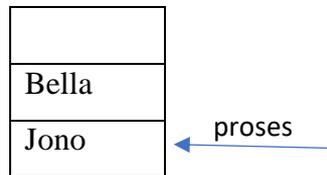
(Kantin Kisheal)

1. Antrean Kosong

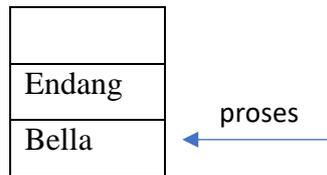

2. Urutan Pertama masuk dan di proses

Jono

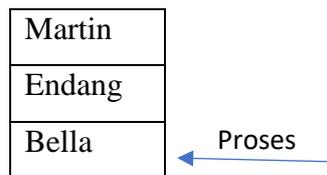
3. Urutan pertama diproses dan urutan kedua masuk



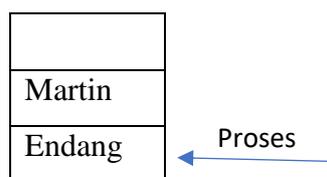
4. Urutan pertama selesai, Urutan kedua di proses dan urutan ketiga masuk



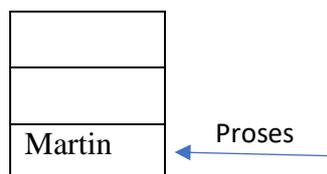
5. Urutan kedua di proses, urutan ketiga sudah dalam antrean, urutan keempat masuk



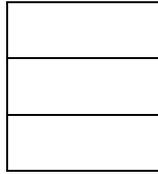
6. Urutan kedua selesai, urutan ketiga di proses, urutan keempat sudah di dalam antrean



7. Urutan ketiga selesai, urutan keempat di proses

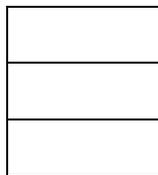


## 8. Urutan empat selesai dan antrean kosong

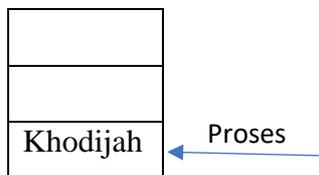


(Kantin TITIS)

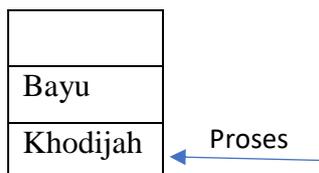
## 1. Antrean Kosong



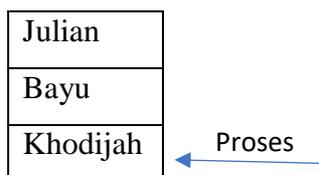
## 2. Urutan pertama masuk dan di proses



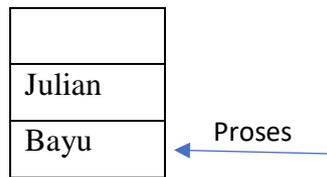
## 3. Urutan pertama di proses dan urutan kedua masuk



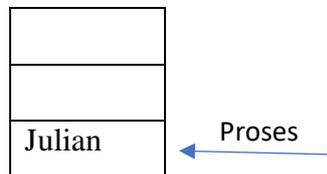
## 4. Urutan pertama di proses, urutan kedua sudah dalam antrean, urutan ketiga masuk



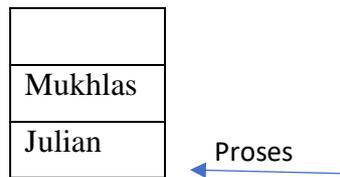
5. Urutan pertama selesai, urutan kedua di proses dan urutan ketiga sudah di dalam antrean



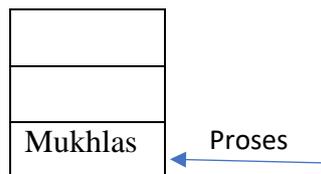
6. Urutan kedua selesai, urutan ketiga di proses



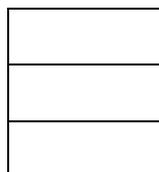
7. Urutan ketiga di proses, urutan ke empat masuk dalam antrean



8. Urutan ketiga selesai, urutan keempat di proses

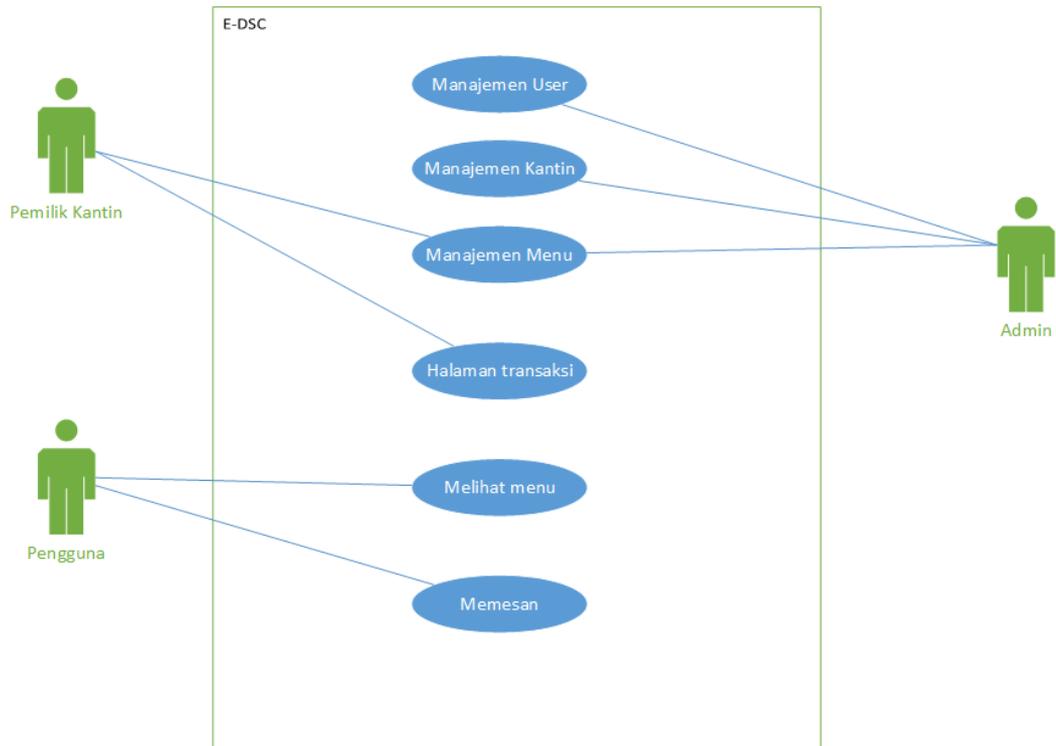


9. Urutan keempat selesai dan antrean kosong



Dan jika antrean kosong maka masuk antrean selanjutnya akan langsung diproses

### 3.4 Usecase Diagram



Gambar 3.1 Usecase Aplikasi E-DSC

#### USECASE SCENARIO

##### 1. Usecase Scenario Register

**Aktor:** Pengguna, Pemilik Kantin

**Deskripsi:** Proses untuk mendaftarkan diri kedalam aplikasi

**Kondisi Awal:** User belum memiliki akun untuk login kedalam aplikasi

**Skenario Utama:**

1. User memasukkan seluruh data diri / data kantinnya sesuai dengan field yang ada di halaman register
2. Mengklik tombol daftar

**Skenario Alternatif:**

1. Jika username akun sudah terdaftar maka sistem akan memunculkan notifikasi username sudah terdaftar

**Kondisi Akhir:** User sudah memiliki akun untuk login kedalam aplikasi

## 2. Usecase Scenario Login

**Aktor:** Pengguna, admin, dan pemilik kantin

**Deskripsi:** Proses untuk seluruh user masuk kedalam sistem ini

**Kondisi Awal:** Seluruh user belum masuk kedalam sistem ini

**Skenario Utama:**

1. User memasukan username dan password yang telah di berikan
2. Mengklik tombol login

**Skenario Alternatif:**

1. Jika akun belum terdaftar maka sistem akan memunculkan notifikasi akun belum terdaftar

**Kondisi Akhir:** user masuk kedalam dashboard akun masing-masing

## 3. Usecase Scenario Akses Manajemen Kantin

**Aktor:** Admin

**Deskripsi:** Admin dapat menambah, mengedit, dan menghapus data kantin di dalam sistem

**Kondisi Awal:** Data kantin belum terupdate

**Skenario Utama:**

1. Admin mengklik halaman manajemen kantin
2. Admin menambah data kantin jika di perlukan
3. Admin mengedit data kantin jika di perlukan
4. Admin menghapus data kantin jika di perlukan

**Skenario Alternatif:**

1. Jika belum adanya data kantin di dalam aplikasi, maka admin tidak dapat mengedit maupun menghapus data kantin

**Kondisi Akhir:** Data kantin sudah terupdate

## 4. Usecase Scenario Akses Manajemen User

**Aktor:** Admin

**Deskripsi:** Admin dapat menambah, mengedit, dan menghapus data user di dalam sistem

**Kondisi Awal:** Data user belum terupdate

**Skenario Utama:**

1. Admin mengklik halaman manajemen user
2. Admin menambah data user jika di perlukan
3. Admin mengedit data user jika di perlukan
4. Admin menghapus data user jika di perlukan

**Skenario Alternatif:**

1. Jika belum adanya data user di dalam aplikasi, maka admin tidak dapat mengedit maupun menghapus data user

**Kondisi Akhir:** Data user sudah terupdate

5. Usecase Scenario Akses Detail Menu Kantin

**Aktor:** Admin

**Deskripsi:** Admin dapat menambah, mengedit, dan menghapus data menu kantin di dalam sistem

**Kondisi Awal:** Data menu kantin belum terupdate

**Skenario Utama:**

1. Admin mengklik halaman detail menu kantin
2. Admin menambah data menu kantin jika di perlukan
3. Admin mengedit data menu kantin jika di perlukan
4. Admin menghapus data menu kantin jika di perlukan

**Skenario Alternatif:**

1. Jika belum adanya data menu kantin di dalam aplikasi, maka admin tidak dapat mengedit maupun menghapus data menu kantin

**Kondisi Akhir:** Data menu kantin sudah terupdate

6. Usecase Scenario Akses Halaman Transaksi

**Aktor:** Admin

**Deskripsi:** Admin dapat melihat dan menghapus transaksi yang ada di dalam sistem

**Kondisi Awal:** admin belum mengetahui detail transaksi yang sudah berlangsung

**Skenario Utama:**

1. Admin mengakses halaman transaksi
2. Admin dapat melihat dan menghapus transaksi yang telah berlangsung di dalam aplikasi ini.

**Skenario Alternatif:**

1. Jika belum adanya transaksi yang telah berlangsung, admin tidak dapat menghapus transaksi apapun.

**Kondisi Akhir:** Admin sudah mengetahui detail transaksi yang sudah berlangsung

## 7. Usecase Scenario Setting Akun

**Aktor:** Pengguna, Admin, Pemilik Kantin

**Deskripsi:** User dapat mengedit username dan password guna mempermudah proses login nantinya

**Kondisi Awal:** Username dan password belum terupdate

**Skenario Utama:**

1. User mengakses menu setting akun
2. User memasukan username baru jika di perlukan
3. User memasukan username lama jika di perlukan
4. Mengklik tombol save

**Skenario Alternatif:**

1. Jika adanya username yang sama, maka sistem akan memunculkan notifikasi username telah ada

**Kondisi Akhir:** Username dan password sudah terupdate

## 8. Usecase Scenario Akses Halaman Menu

**Aktor:** Pemilik Kantin

**Deskripsi:** Pemilik kantin dapat menambah, mengedit, menghapus detail menu yang ada di dalam sistem

**Kondisi Awal:** Data menu belum terupdate

**Skenario Utama:**

1. Pemilik kantin mengakses halaman menu

2. Pemilik kantin dapat menambah menu jika di perlukan
3. Pemilik kantin dapat mengedit menu jika di perlukan
4. Pemilik kantin dapat menghapus menu jika di perlukan

**Skenario Alternatif:**

1. Jika pemilik kantin sama sekali belum pernah menambah menu di dalam aplikasi, maka pemilik kantin tidak dapat mengakses fitur hapus dan edit menu

**Kondisi Akhir:** Data menu sudah terupdate.

9. Usecase Scenario Akses Detail Menu

**Aktor:** Pemilik Kantin

**Deskripsi:** Pemilik kantin dapat melihat detail menu yang pernah di inputkan kedalam aplikasi

**Kondisi Awal:** Pemilik kantin belum melihat detail menu apa saja yang pernah di inputkan kedalam aplikasi

**Skenario Utama:**

1. Pemilik kantin mengakses halaman detail menu
2. Pemilik kantin dapat melihat seluruh detail menu yang sudah pernah di inputkan kedalam aplikasi

**Skenario Alternatif:**

1. Jika belum adanya menu yang di inputkan, maka sistem akan memunculkan notifikasi tidak ada menu yang telah di record

**Kondisi Akhir:** Pemilik kantin sudah melihat detail menu apa saja yang pernah di inputkan kedalam aplikasi.

10. Usecase Scenario Akses Halaman Transaksi

**Aktor:** Pemilik Kantin

**Deskripsi:** Pemilik kantin dapat melihat seluruh transaksi dan proses transaksi yang sedang berlangsung

**Kondisi Awal:** Pemilik kantin belum melihat ataupun memproses transaksi berlangsung

**Skenario Utama:**

1. Pemilik kantin mengakses menu transaksi
2. Pemilik kantin dapat melihat transaksi yang telah selesai
3. Pemilik kantin dapat memproses transaksi yang sedang berlangsung

**Skenario Alternatif:**

1. Jika belum adanya transaksi yang telah selesai, maka sistem akan memunculkan notifikasi tidak ada transaksi yang telah di record.

**Kondisi Akhir:** Pemilik kantin sudah melihat ataupun memproses transaksi berlangsung

11. Usecase Scenario Akses Lupa Password

**Aktor:** Pengguna

**Deskripsi:** Pengguna dapat menggunakan fitur ini jika lupa password

**Kondisi Awal:** Pengguna lupa password lama

**Skenario Utama:**

1. Pengguna mengakses halaman lupa password
2. Pengguna memasukan emailnya di field yang tersedia
3. Link ubah password akan di kirimkan ke email yang di inputkan oleh pengguna
4. Pengguna menginputkan password baru
5. Klik tombol ubah password

**Skenario Alternatif: -**

**Kondisi Akhir:** Password lama telah terganti dengan password baru

12. Usecase Scenario Akses Halaman Dashboard

**Aktor:** Pengguna dan pemilik kantin

**Deskripsi:** user Langsung di arahkan ke halaman dashboard ketika sukses melakukan login

**Kondisi Awal:** User masih dalam proses login

**Skenario Utama:**

1. User melakukan login di halaman login
2. Ketika berhasil user Langsung di arahkan ke halaman dashboard

**Skenario Alternatif:**

1. Jika akun tidak terdaftar atau adanya inputan yang salah maka user tidak dapat melakukan proses login ke dalam sistem ini

**Kondisi Akhir:** User sudah berada di halaman dashboard akun masing-masing

## 13. Usecase Scenario Akses Halaman Detail Kantin

**Aktor:** Pengguna

**Deskripsi:** pengguna dapat melihat detail kantin sebelum mereka melakukan pembelian makanan/minuman

**Kondisi Awal:** Pengguna belum menemukan kantin yang mereka inginkan

**Skenario Utama:**

1. Pengguna mengakses halaman detail kantin
2. Masing-masing kantin telah memasukan data detail kantin sehingga pengguna dapat Langsung melihat detail kantin

**Skenario Alternatif:**

1. Jika adanya kantin yang belum memasukan data detail kantin secara lengkap, pengguna tidak dapat melihat detail kantin tersebut

**Kondisi Akhir:** Pengguna sudah ingin mencari menu di kantin yang mereka inginkan

## 14. Usecase Scenario Akses Halaman Detail Menu

**Aktor:** Pengguna

**Deskripsi:** Pengguna dapat melihat detail menu yang ada di kantin yang di inginkan terkait deskripsi, harga dan sebagainya

**Kondisi Awal:** pengguna baru ingin mencari menu yang ada di kantin yang di inginkan

**Skenario Utama:**

1. Pengguna memilih menu yang di inginkan
2. Pengguna dapat melihat detail menu tersebut seperti deskripsi, harga dan lainnya

**Skenario Alternatif:**

1. Jika ada kantin yang tidak memasukan detail menu, maka pengguna tidak dapat melihat detail menu yang ada di kantin tersebut

**Kondisi Akhir:** Pengguna sudah di arahkan ke form pesan menu.

## 15. Usecase Scenario Akses Form Pesan Menu

**Aktor:** Pengguna

**Deskripsi:** Pengguna mengisi form pemesanan menu di salah satu kantin

**Kondisi Awal:** Pengguna ingin memesan suatu menu yang ada di salah satu kantin

**Skenario Utama:**

1. Pengguna memesan menu di salah satu kantin dan di arahkan ke form pemesanan menu
2. Di form ini pengguna melakukan input terkait lokasi pengantaran, dan lainnya
3. Form berhasil terisi dan transaksi Langsung di arahkan ke keranjang belanja

**Skenario Alternatif:**

1. Jika pengguna tidak menginputkan data yang di minta secara detail, maka transaksi tidak dapat di lakukan

**Kondisi Akhir:** Pengguna di arahkan ke halaman keranjang belanja

## 16. Usecase Scenario Akses Keranjang Belanja

**Aktor:** Pengguna

**Deskripsi:** Pengguna dapat mengecek transaksi yang sedang berlangsung di halaman keranjang belanja

**Kondisi Awal:** transaksi baru di lakukan dan terdapat detail transaksi di halaman keranjang belanja

**Skenario Utama:**

1. Ketika selesai melakukan penginputan, maka pengguna dapat mengecek status pemesanan di detail transaksi yang ada di halaman keranjang belanja
2. Pihak kantin mengantarkan makanan, pengguna membayar dan pihak kantin menyelesaikan transaksi

**Skenario Alternatif:**

1. Jika pihak kantin menolak dalam merequest pesanan, maka pesanan pengguna akan masuk ke halaman riwayat transaksi dengan label di tolak / transaksi gagal

**Kondisi Akhir:** transaksi selesai dan transaksi yang telah berhasil tersebut akan di masukan ke daftar riwayat pesanan

17. Usecase Scenario Akses Riwayat Pesanan

**Aktor:** Pengguna

**Deskripsi:** Pengguna dapat melihat riwayat pesanan, baik pesanan yang berhasil ataupun transaksi gagal di dalam menu riwayat pesanan

**Kondisi Awal:** transaksi masih ada di dalam keranjang belanja

**Skenario Utama:**

1. Pihak kantin akan menyelesaikan transaksi di dalam sistem
2. Transaksi yang ada di dalam akun pengguna akan otomatis pindah ke dalam halaman riwayat pesanan

**Skenario Alternatif: -**

**Kondisi Akhir:** Transaksi sudah ada di dalam daftar riwayat pesanan

18. Usecase Scenario Logout

**Aktor:** Pengguna, Admin, Pemilik Kantin

**Deskripsi:** User dapat keluar dari sistem ini

**Kondisi Awal:** User masih ada di dalam sistem ini

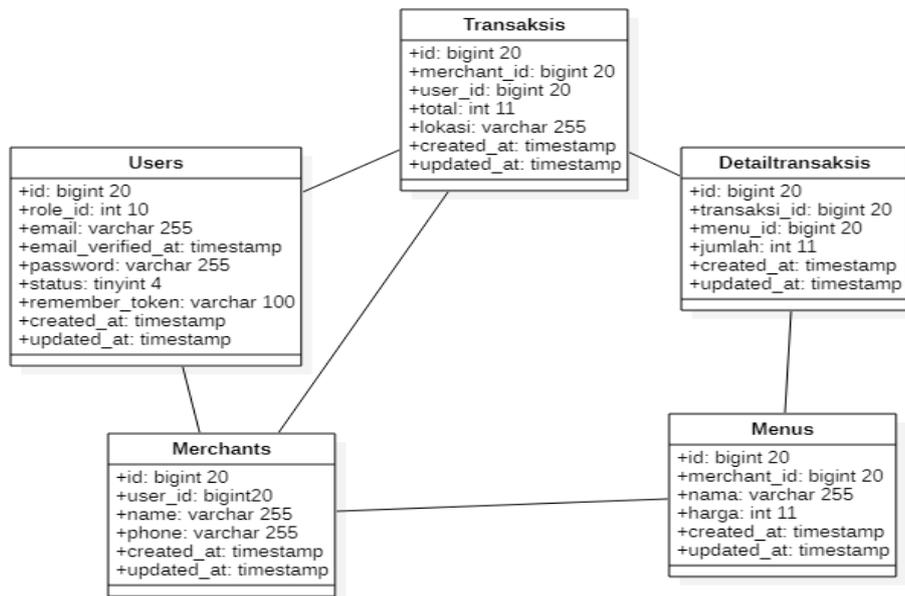
**Skenario Utama:**

1. User mengklik tombol logout
2. User Langsung di arahkan ke halaman login

**Skenario Alternatif: -**

**Kondisi Akhir:** User sudah keluar dari sistem ini

### 3.5 Class Diagram

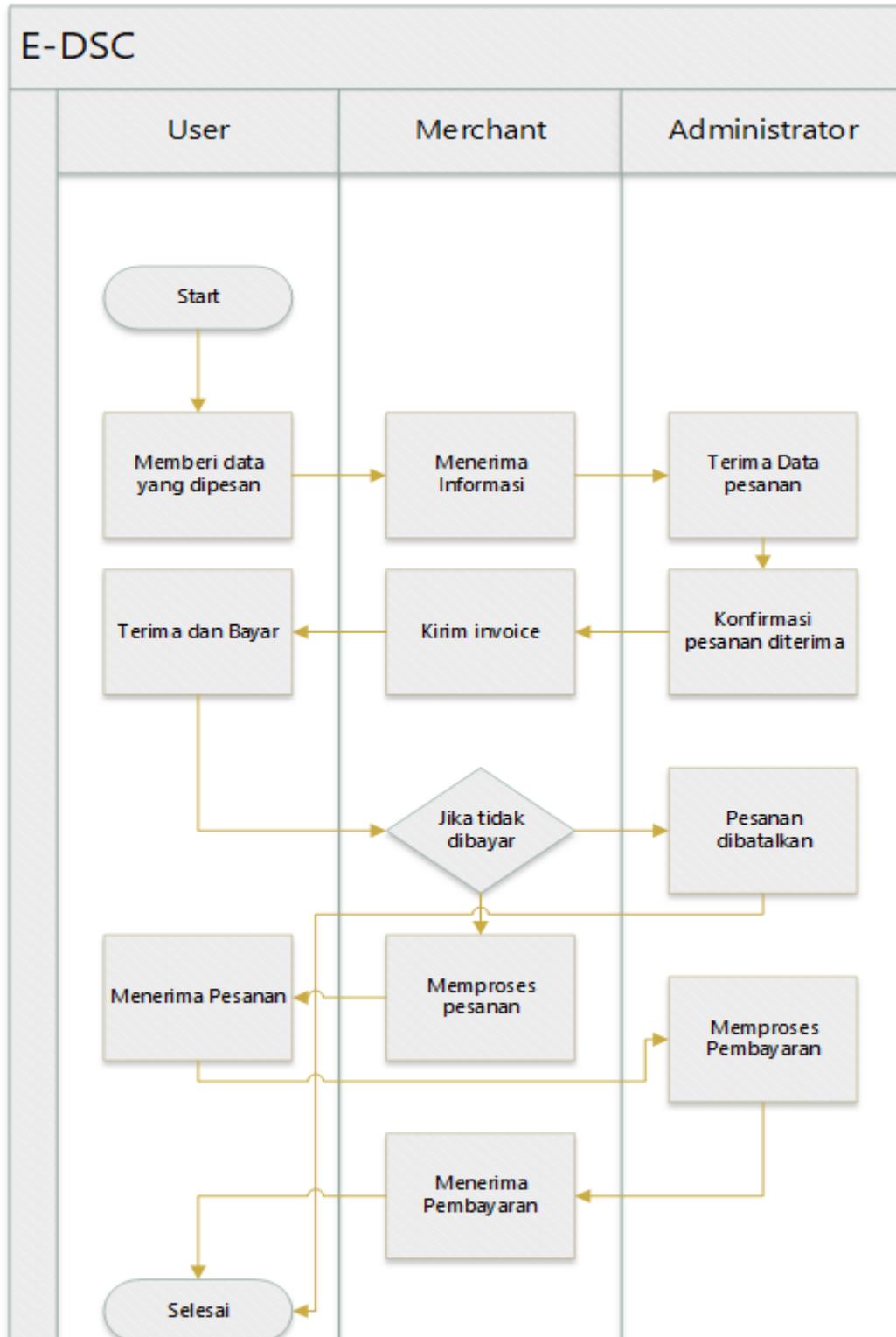


Gambar 3.2 Class Diagram Aplikasi E-DSC

keterangan class diagram di atas:

- Tabel user berelasi dengan tabel transaksi dengan relasi many to many karena banyaknya user yang dapat membuat banyaknya transaksi
- Tabel user berelasi dengan tabel merchants dengan relasi many to many karena banyaknya user yang dapat melihat banyak merchant
- Tabel merchant berelasi dengan tabel transaksi dengan relasi many to many karena banyaknya merchant yang dapat memproses banyaknya transaksi
- Tabel merchant berelasi dengan tabel menu dengan relasi many to many karena banyak merchant dapat menginput banyak meny
- Tabel transaksi berelasi dengan tabel detail transaksi dengan relasi one to one karena satu transaksi hanya memiliki satu detail transaksi
- Tabel detail transaksi berelasi dengan tabel menu dengan relasi one to many karena satu detail transaksi dapat memuat beberapa menu di dalamnya.

### 3.6 Activity Diagram



Gambar 3.3 Activity Diagram Aplikasi E-DSC

### 3.7 Rancangan Awal Interface aplikasi

#### 3.7.1 Rancangan awal Interface Halaman Detail Order (Merchant)

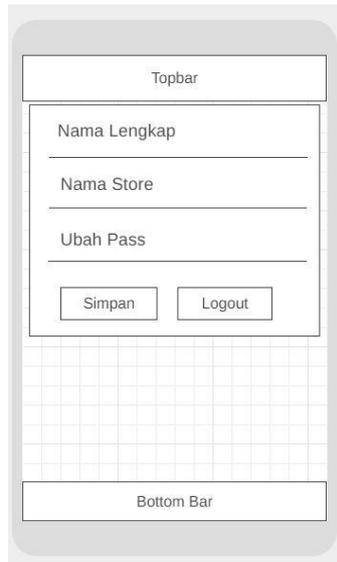
Halaman detail order memiliki topbar yakni berupa nama menu “detail order” dengan background berwarna merah dengan tombol panah yang akan di gunakan untuk back ke menu sebelumnya. Di halaman detail order ini juga berisi detail menu yang di pesan customer beserta nama pemesan dan lokasi pengantaran serta harga total dari orderan tersebut. Di menu ini merchant dapat menerima ataupun menolak orderan yang masuk.



Gambar 3.4 Rancangan Interface Halaman Detail Order (Merchant)

#### 3.7.2 Rancangan awal Interface Halaman Akun (Merchant)

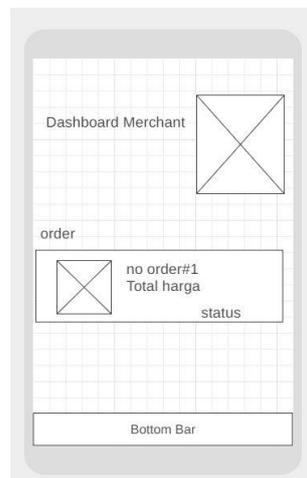
Bagian topbar berisi nama halaman “Akun” dengan background color berwarna merah. Di halaman akun ini berisi 3 field yang harus di isi nantinya yakni nama lengkap, nama store dan ubah password. Di halaman akun juga terdapat dua tombol yakni simpan dan logout untuk mengupdate ataupun membatalkan perubahan. Di halaman ini juga terdapat button bar yang berisi halaman home, menu, dan akun



Gambar 3.5 Rancangan Interface Halaman Akun (Merchant)

### 3.7.3 Rancangan awal Interface Halaman Dashboard (Merchant)

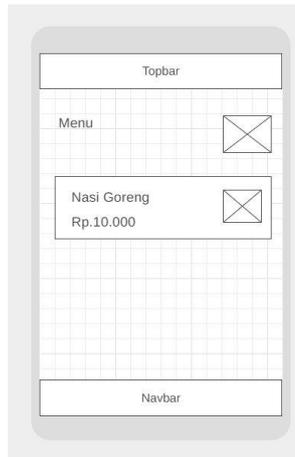
Halaman dashboard berisi Keterangan “Dashboard Merchant” dengan foto profil yang ada di dalamnya. Selain itu di halaman ini juga berisi orderan baru yang masuk ke masing-masing akun merchant. Di halaman ini juga terdapat button bar yang terdiri dari halaman home, menu dan akun.



Gambar 3.6 Rancangan Interface Halaman Dashboard (Merchant)

### 3.7.4 Rancangan awal Interface Halaman Menu (Merchant)

Halaman menu terdapat tombol + yang berguna untuk mengarahkan user ke halaman tambah menu. Selain itu di halaman ini juga terdapat list menu dari menu yang telah di inputkan oleh merchant sebelumnya. Selain itu di halaman ini terdapat navbar yang berisi halaman menu, menu, dan Akun



Gambar 3.7 Rancangan Interface Halaman Menu (Merchant)

### 3.7.5 Rancangan awal Interface Halaman Tambah Menu (Merchant)

Halaman tambah menu terdapat topbar yang berisi nama halaman "Tambah Menu" dan terdapat radio button yang akan user pilih dari menu yang di inputkan berupa makanan atau minuman. Di halaman ini juga terdapat 2 field yang akan di isi, yakni nama menu beserta harga menu. Terdapat 1 tombol simpan yang berguna untuk menyimpan seluruh inputan yang telah di inputkan.



Gambar 3.8 Rancangan Interface Halaman Tambah Menu (Merchant)

### 3.7.6 Rancangan awal Interface Halaman Akun (Merchant)

Bagian topbar berisi nama halaman “Akun” dengan background color berwarna merah. Di halaman akun ini berisi 3 field yang harus di isi nantinya yakni nama lengkap, nama store dan ubah password. Di halaman akun juga terdapat dua tombol yakni simpan dan logout untuk mengupdate ataupun membatalkan perubahan. Di halaman ini juga terdapat button bar yang berisi halaman home, menu, dan akun



Gambar 3.9 Rancangan Interface Halaman Akun (Merchant)

### 3.7.7 Rancangan awal Interface Halaman Menu (Customer)

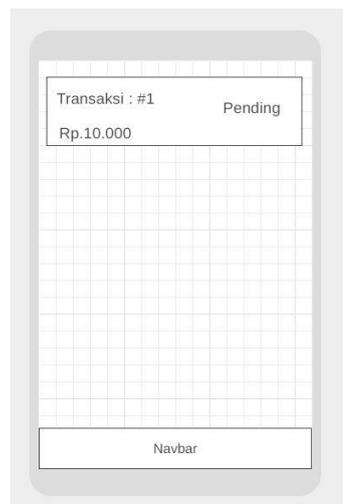
Halaman ini terdapat topbar yang berisi nama halaman “menu” dengan background color berwarna merah. Di halaman ini juga berisi list menu yang dapat user pesan dengan mengklik tombol tambah yang ada di samping masing-masing menu



Gambar 3.10 Rancangan Interface Halaman Menu (Customer)

### 3.7.8 Rancangan awal Interface Halaman Riwayat Transaksi (Customer)

halaman ini berisi seluruh riwayat transaksi yang telah di pesan oleh user. Di halaman ini juga berisi navbar yang berisi halaman home, riwayat dan akun.



Gambar 3.11 Rancangan Interface Halaman Riwayat Transaksi (Customer)

### 3.7.9 Rancangan awal Interface Halaman Detail Order (Customer)

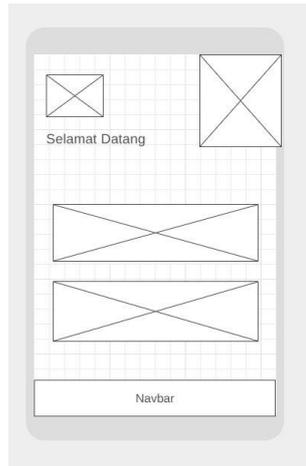
Halaman detail order terdapat topbar yang berisi nama halaman “Detail Order” dengan background color berwarna merah dengan panah yang bertujuan untuk back ke halaman sebelumnya. Di halaman ini juga terdapat seluruh menu yang telah di pesan dan dapat di batalkan dengan mengklik tombol X di masing-masing menu. Di halaman ini juga terdapat total dari orderan yang berlangsung.



Gambar 3.12 Rancangan Interface Halaman Detail Order (Customer)

### 3.7.10 Rancangan awal Interface Halaman Dashboard (Customer)

Halaman dashboard terdapat beberapa gambar yang membuat aplikasi ini semakin user friendly. Selain itu di halaman ini terdapat 2 kategori menu yakni makanan dan minuman yang dapat user pesan di masing-masing merchant. Di halaman ini juga terdapat Navbar yang terdiri dari halaman home, riwayat dan akun.



Gambar 3.13 Rancangan Interface Halaman Dashboard (Customer)

### 3.7.11 Rancangan awal Interface Halaman Keranjang Belanja (Customer)

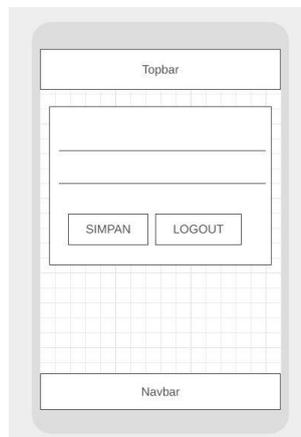
Halaman keranjang belanja terdapat topbar yang berisi nama halaman “Keranjang Belanja” dengan background color warna merah dengan tanda panah yang jika di klik akan mengarahkan user ke halaman sebelumnya. Di halaman ini juga berisi seluruh menu yang ingin di checkout. Jika ingin membatalkan salah satu menu bisa mengklik tombol X di samping menu yang ingin di batalkan. Jika pesanan sudah sesuai klik tombol simpan yang berada di bawah total harga.



Gambar 3.14 Rancangan Interface Halaman Keranjang Belanja (Customer)

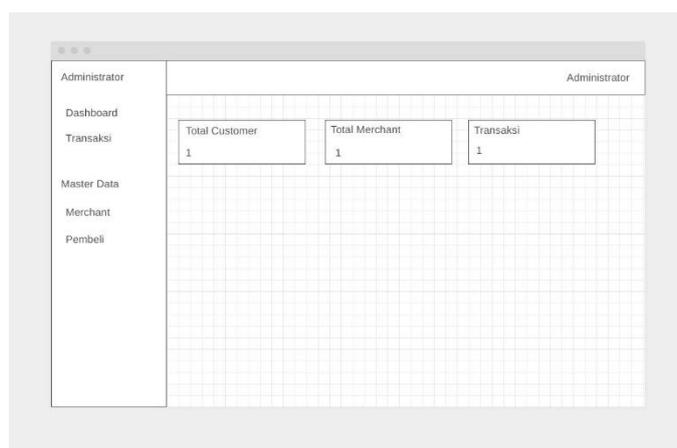
### 3.7.12 Rancangan awal Interface Halaman Akun (Customer)

Bagian topbar berisi nama halaman “Akun” dengan background color berwarna merah. Di halaman akun ini berisi 2 field yang harus di isi nantinya yakni nama lengkap dan ubah password. Di halaman akun juga terdapat dua tombol yakni simpan dan logout untuk mengupdate ataupun membatalkan perubahan. Di halaman ini juga terdapat button bar yang berisi halaman home, riwayat, dan akun



Gambar 3.15 Rancangan Interface Halaman Akun (Customer)

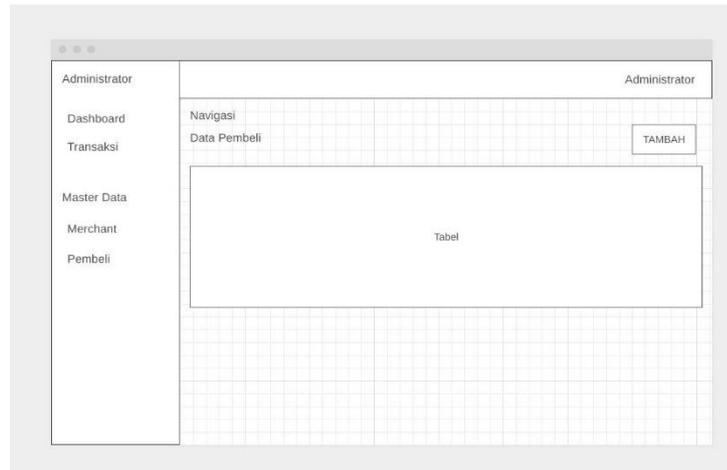
### 3.7.13 Rancangan awal Interface Halaman Dashboard (Administrator)



Gambar 3.16 Rancangan Interface Halaman Dashboard (Admin)

### 3.7.14 Rancangan awal Interface Halaman Master Data Customer (Administrator)

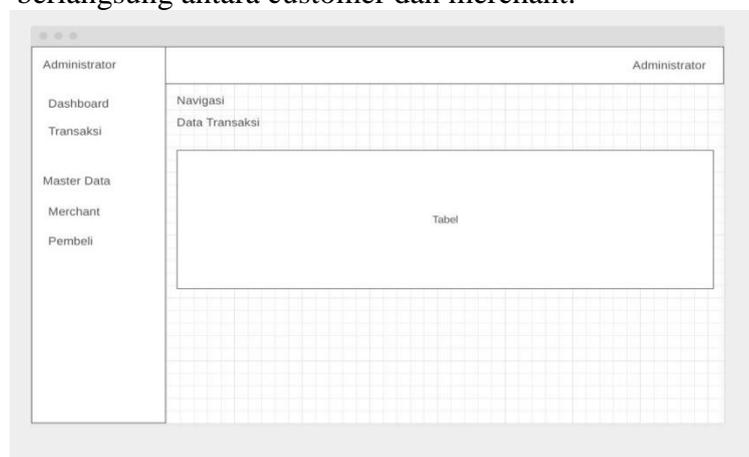
Halaman master data customer terdapat tombol tambah yang berfungsi untuk menambah data di dalam tabel. Di halaman ini juga berisi tabel data customer yang akan menampilkan data yang ada di dalam database



Gambar 3.17 Rancangan Interface Halaman Master Data Customer (Admin)

### 3.7.15 Rancangan awal Interface Halaman Master Data Transaksi (Administrator)

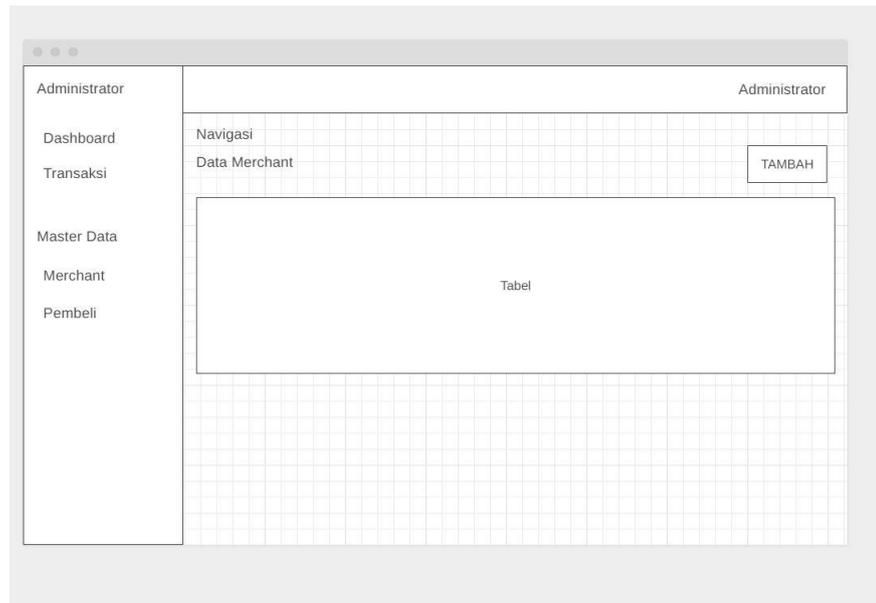
Halaman master data berisi tabel data transaksi yang akan menampilkan data yang ada di dalam database. Data ini di dapatkan dari transaksi yang sudah berlangsung atau sedang berlangsung antara customer dan merchant.



Gambar 3.18 Rancangan Interface Halaman Master Data Transaksi (Admin)

### 3.7.16 Rancangan awal Interface Halaman Master Data Merchant (Administrator)

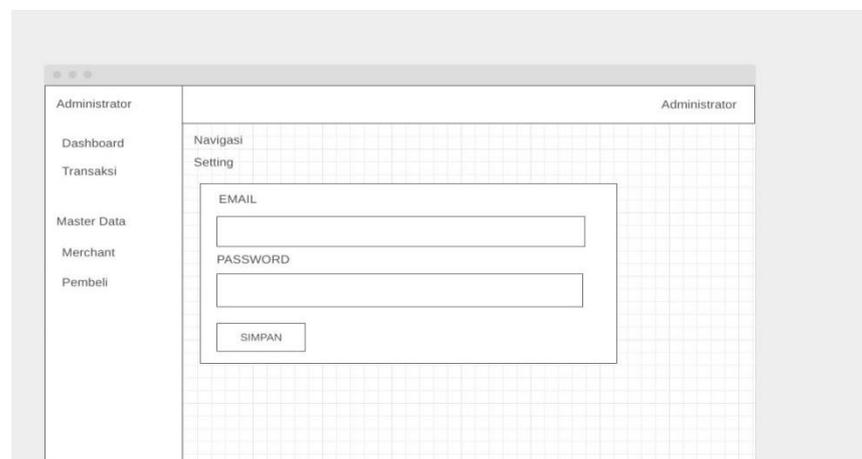
Halaman master data merchant terdapat tombol tambah yang berfungsi untuk menambah data di dalam tabel. Di halaman ini juga berisi tabel data merchant yang akan menampilkan data yang ada di dalam database



Gambar 3.19 Rancangan Interface Halaman Master Data Merchant (Admin)

### 3.7.17 Rancangan awal Interface Halaman Setting (Administrator)

Halaman setting terdapat dua field yang harus di isi yakni field email dan password. Di halaman ini terdapat tombol simpan yang berfungsi untuk mengupdate data yang telah di inputkan oleh administrator.



Gambar 3.20 Rancangan Interface Halaman Setting (Admin)