# BAB 3. ANALISIS DAN DESAIN SISTEM

Pada bab ini akan dibahas mengenai desain dan analisis sistem mengenai Sistem Penunjang Belanja Pedagang Keliling di Lokasi Sekitar Menggunakan Haversine Berbasis Android.Pada bab ini pula akan dijelaskan analisis data yang akan membahas mengenai *usecase diagram, activity diagram* dan *user interface* yang dipakai.

## 3.1.1 Analisis Data

Selama pembuatan skripsi, dilakukan analisis data terhadap gomart dan grabmart sebagai studi pembanding. Data yang digunakan dalam pembuatan sistem ini pertama adalah lokasi atau koordinat *latitude* dan *longitude* dari pengguna pedagang keliling dan pengguna. Data kedua yang diberikan adalah file gambar berupa *file* jpg/jpeg untuk produk dari pedagang keliling.Yang digunakan untuk mengupload gambar produk ke server. Data yang selanjuntya yang dibutuhkan adalah data pribadi dari pedagang keliling yang berupa region/daerah yang dibutuhkan untuk menghasilkan query bagi pelanggan. Data selanjutnya real time location pelanggan yang digunakan untuk mendapat list dari pedagang keliling terdekat diareanya.

## 3.1.2 Analisis Sistem

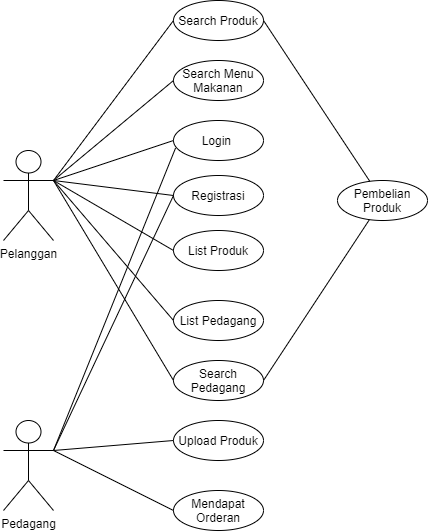
Pada penulisan skripsi ini dilakukan analisis terhadap aplikasi sistem aplikasi gojek dan gomart untuk melihat – lihat fitur yang telah ada. Dan mencari tau fitur yang dapat ditambah dan digunakan dalam pembuatan sistem penunjang belanja pedagang keliling di lokasi sekitar.

## 3.1.3 Analisis Permasalahan

Kesusahannya pedagang keliling untuk menjual barangnya secara manual dan kesusahan pada keliling untuk mengetaui siapa saja yang ingin membeli barang daganganya sehingga pedagang keliling tidak usah untuk putar –putar untuk mendapatkan pelanggannya. Diharapkan juga membantu para pelanggan pedagang keliling untuk mengetahui lokasi pedagang keliling atau penjual yang dibutuhkan. Membantu pelanggan dalam pembelian hari – H dan memberikan catatan untuk pembelian di esok paginya.

## 3.2 Use Case Diagram

*Use case diagram* merupakan *diagram* yang mengambarkan sebuah aktor dengan sistem. Use case diagram dapat mendeskripsikan satu actor atau lebih dari sebuah sistem. Didalam use case diagram dalam penelitian ini terdapat dua aktor. Pelanggan yang memiliki hak kegunaan *login*, registrasi, *search* produk, *search* pedagang keliling, melihat list pedagang keliling dan melakukan order produk. Pedagan keliling memilik hak akses login, registrasi , upload produk ,mendapat orderan.

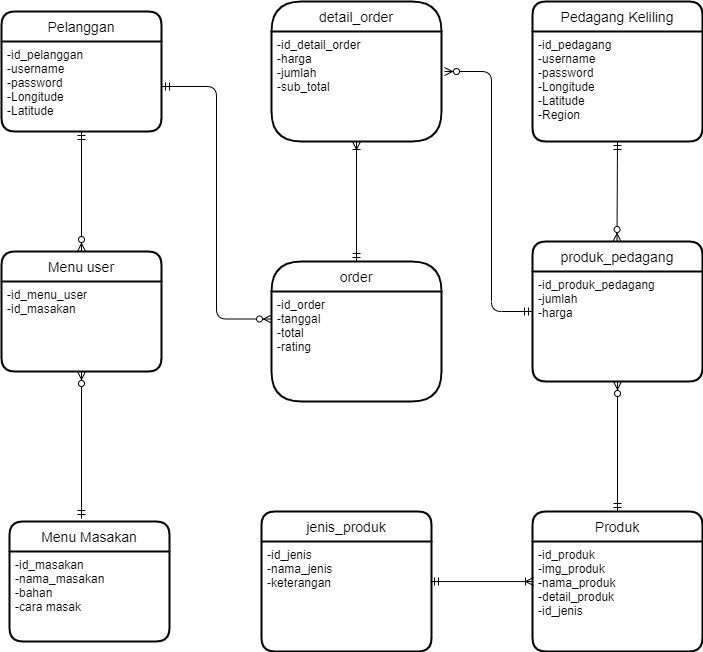
****

Gambar 3.1 Use case diagram

## 3.4.1 Entity Relation Diagram

Pada penelitian ini memiliki 5 entity. Entity Pelanggan yang memiliki *attribute* id\_pelanggan, username, password, longitude dan latitude. Entity Pedagang Keliling memiliki attribute id\_pedangang, username, password, longitude, latitude, region. Entity Produk id\_produk, foreign key id\_pedagang, img\_produk, nama\_produk, jumlah\_produk dan detail\_produk.

Entity kerajang memiliki attribute id\_keranjang, foreign key id\_produk dan foreign key id\_pelanggan. Entity Order memiliki attribute id\_order dan id\_kerajang. Pelanggan memiliki relation 1 to many ke pedagang keliling. Pedagang keliling memiliki relasi 1 to many ke produk. Pelangan memilik relasi 1 to many ke keranjang. Produk memiliki relasi 1 to many ke keranjang. Keranjang memiliki relasi 1 to 1 ke order.



Gambar 3.2 Entity Relation Diagram

## 3.4.2 Penjelasan Struktur Tabel

Berikut ini adalah penjelasan struktur table yang ada di dalam database aplikasi. Bedasarkan ERD ( *Entity Relationship Diagram* )yang sudah didesain sesuai pada Gambar 3.4, dibuatlah database dengan rincian tabel yang digunakan seperti dibawah ini.

Table 3.1 Desain Table Pelanggan

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama** | **Tipe Data** | **Key** | **Deskripsi** |
| id\_pelanggan | Int | *Primary key* | Id Pelanggan auto increment |
| nama\_pelanggan | Varchar(100) | - | nama pelanggan |
| username | Varchar(100) | - | username pelanggan |
| password | Varchar(100) | - | password pelanggan |
| longitude | Varchar(255) | - | Longitude pelanggan |
| latitude | Varchar(255) | - | Latitude pelanggan |
| rating | float | - | rating pelanggan |

Desain table pelanggan pada Table 3.1 digunakan untuk menyimpan data pelanggan yang ada. Data yang disimpan berupa username, password, nama\_pelanggan, longitude dan latitude dari pelanggan.

Table 3.2 Desain Table Menu Masakan

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama** | **Tipe Data** | **Key** | **Deskripsi** |
| id\_menu\_masakan | Int | *Primary key* | Id menu masakan auto increment |
| nama\_masakan | Varchar(100) | - | nama masakan |
| bahan\_masakan | Varchar(100) | - | bahan makanan |
| cara\_memasak | Varchar(100) | - | cara- cara membuat makanan |
| pembuat\_masakan | Varchar(100) | - | pembuat menu masakan |

Desain Table Menu Masakan pada Table 3.2 digunakan untuk menyimpan data masakan yang di *scraping*. Data yang disimpan berupa nama\_masakan, bahan\_masakan, cara\_memasak, dan pembuat masakan.

Table 3.3 Desain Table Menu Makanan Pelanggan

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama** | **Tipe Data** | **Key** | **Deskripsi** |
| id\_menu\_makanana\_user | Int | *Primary key* | Id menu masakan auto increment |
| id\_pelanggaan | int | *Foreign key* | id\_pelanggan |
| id\_menu\_masakan | int | *Foreign key* | id\_menu masakan dari table menu masakan |
|  |  |  |  |

Desain Table Menu Masakan pada Table 3.3 digunakan untuk menyimpan data menu\_makanan user. Data yang disimpan berupa id\_pelanggan dan id\_menu\_masakan. Yang keduanya merupakan foreign key.

Table 3.4 Desain Pedagang Keliling

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama** | **Tipe Data** | **Key** | **Deskripsi** |
| id\_pedagang \_keliling | Int | *Primary key* | Id pedagang keliling auto increment |
| nama\_pedagang\_keliling | Varchar(100) | - | nama pedagang keliling |
| username | Varchar(100) | - | username pedagang keliling |
| password | Varchar(100) | - | password pedagang keliling |
| longitude | Varchar(255) |  | Longitude pedagang keliling |
| latitude | Varchar(255) |  | Latitude pedagang keliling |
| region\_latitude | Varchar(255) |  | Region latitude pedagang keliling |
| region\_longitude | Varchar(255) |  | Region longitude pedagang keliling |

Desain table pelanggan pada Table 3.4 digunakan untuk menyimpan data pedagang keliling yang ada. Data yang disimpan berupa username, password, nama\_pedagang\_keliling, longitude, latitude, region longitude dan region latitude.

Table 3.5 Desain Table Produk

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama** | **Tipe Data** | **Key** | **Deskripsi** |
| id\_produk | Int | *primary key* | Id produk auto increment |
| img\_produk | Varchar(200) |  | Menyimpan link berupa gambar. |
| nama\_produk | Varchar(100) |  | Nama\_produk |
| detail\_produk | Varchar(255) |  | Menyimpan keterangan produk |
| id\_jenis | int | *foreign key* | foreign key dari table jenis\_produk |

Desain table produk pada Table 3.5 digunakan untuk menyimpan produk – produk yang di data. Data yang disimpan berupa img\_produk , nama\_produk , detail\_produk, id\_jenis.

Table 3.6 Desain Table Produk Pedagang

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama** | **Tipe Data** | **Key** | **Deskripsi** |
| id\_produk\_pedagang | *Int* | *primary key* | Id produk pedagang auto increment |
| id\_produk | *int* | *foreign key* | merupakan foreign key dari produk |
| id\_pedagang\_keliling | *int* | *foreign key* | foreign key dari pedagang keliling |
| jumlah | *int* |  | jumlah produk yang tersedia |
| harga | *float* |  | harga dari produk |

Desain table produk pedagang pada Table 3.6 digunakan untuk menyimpan barang dagangan yang dimiliki oleh pedagang keliling. Data yang disimpan berupa id\_produk, id\_pedagang\_keliling, jumlah dan harga.

Table 3.7 Desain Table Detail Order

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama** | **Tipe Data** | **Key** | **Deskripsi** |
| id\_detail\_order | Int | *primary key* | Id detail order auto increment |
| id\_produk\_pedagang | Int | *foreign key* | foreign key dari table produk\_pedagang |
| id\_order | int | *foreign key* | id\_order |
| jumlah\_detail\_order | int |  | jumlah barang yang diorder |
| sub\_total | float |  | total harga dari barang yang diorder |

Desain table detail order merupakan table yang menyimpan keranjang barang yang ingin dipesan. Digunakan untuk menentukan barang apa saja yang ingin dibeli. Data yang disimpan berupa id\_detail\_order, id\_produk\_pedagang, jumlah\_detail\_order, dan subtotal.

Table 3.8 Desain Table Order

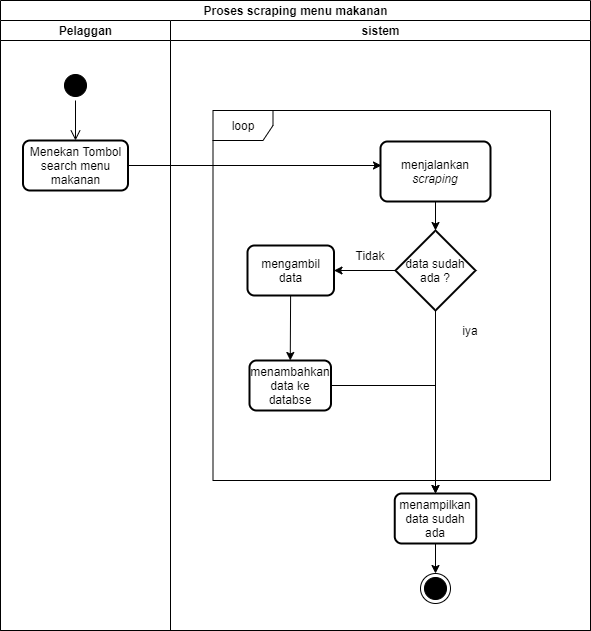
|  |  |  |  |
| --- | --- | --- | --- |
| **Nama** | **Tipe Data** | **Key** | **Deskripsi** |
| id\_order | Int | primary key | id order auto increment |
| id\_detail\_order | Int | foreign key | foreign key dari table produk\_pedagang |
| tanggal | date |  | menyimpan tanggal berapa barang dipesan |
| total | float |  | total harga dari produk yang diorder |
| rating | float |  | menyimpan rating produk dan pelanggan |
| id\_pelanggan | int | foreign key | foreign key dari table pelanggan |

Desain table order digunakan untuk menyimpan data pembelian pelanggan dan juga untuk menampilkan history pembelian. Data ini menyimpan id\_detail\_order, tanggal , total dan id \_pelanggan

# 3.5 Proses Scraping

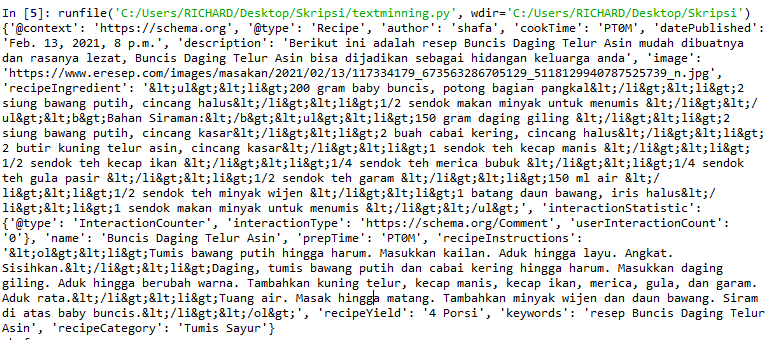
Proses scraping merupakan salah satu proses dalam skripsi ini, data yang didapatkan dari proses scraping ini merupakan menu makanan, bahan makanan dan cara memasak. Perlunya memahami struktur *HTML* dari situs agar data yang diperlukan bisa diambil dengan lancar dan tanpa ada kesalahan.

Acitivity diagram digunakan untuk melihat proses *scraping*, baik di situs eresep atau situs resep lainya. Pengguna akan menuliskan menu makan yang akan dipilih untuk melakukan *scraping* resep masakan dari situs yang dituju, sehingg sistem akan menjalankan proses scraping. Untuk setiap resep masakan yang terdapat akan diperiksa terlebih dahulu apakah data yang akan diambil sudah terdapat di database. Hal itu dilakukan dengan cara melakukan *query* dengan judul masakan dan author resep masakan. Apabila data belum ada didalam database, maka proses scrapping untuk data tersebut diselesaikan dan proses dimasukan ke dalam database.



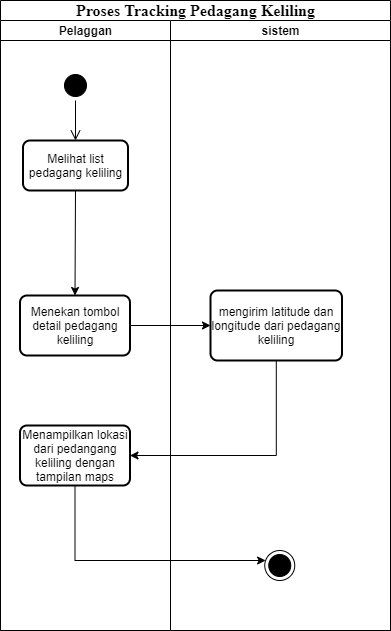
Gambar 3.3 Activity Diagram Proses *Scraping* Menu Makanan.

Menggunakan *library python* *BeutifulSoup* untuk melakukan pengambilan data dari situ eresep.com. *Beautifulsoup* digunakan untuk mengambil data yang ada disitus tersebut. BeutifulSoup menggunakan link yang dipasing kedalam *parameter python* yang kemudian digunakan untuk mengakses halaman situs tersebut. Data yang diambil berupa file JSON yang didapat didalam situs tersebut.

**Gambar 3.4 *Scraping JSON* di eresep.com

**3.6.1 Proses Tracking Pedagang Keliling**

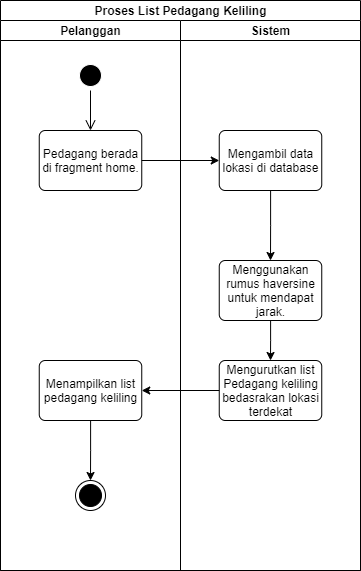
Pada Gambar 3.5 merupakan activity diagram untuk melakukan pencarian barang atau pedagang keliling. Apabila pengguna masuk ke halaman aplikasi untuk melihat list pedagang keliling. Setelah melihat list pedagang keliling maka akan menekan tombol lihat lokasi. Akan masuk kedalam aktivitas detail lokasi pedagang keliling. Di detail lokasi pedagang akan terlihat lokasi dari pedagang keliling yang ditampilkan di *google maps*.

****

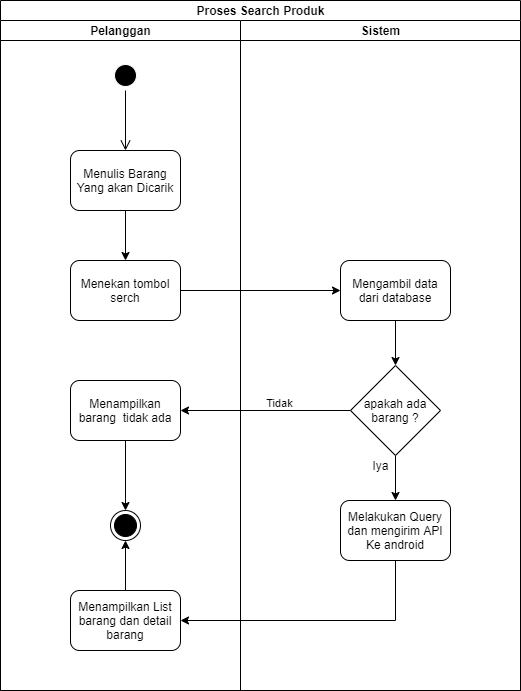
Ganbar 3.5 Activity Diagram Tracking Pedagang Keliling

**3.6.2 Proses List Pedang Keliling dan Search Produk**

Menampilakan list pedagang keliling bedasarkan jarak terdekat dengan lokasi yang dimiliki saat ini. Melakukan proses search dan filter sesuai dengan attribute. Pada Gambar 3.6 merupakan activity diagram untuk menampilkan list pedagang keliling bedasarkan lokasi terdekat. Pada Gambar 3.7 merupakan activity diagram untuk melakukan search produk bedasarkan keyword yang dicari.

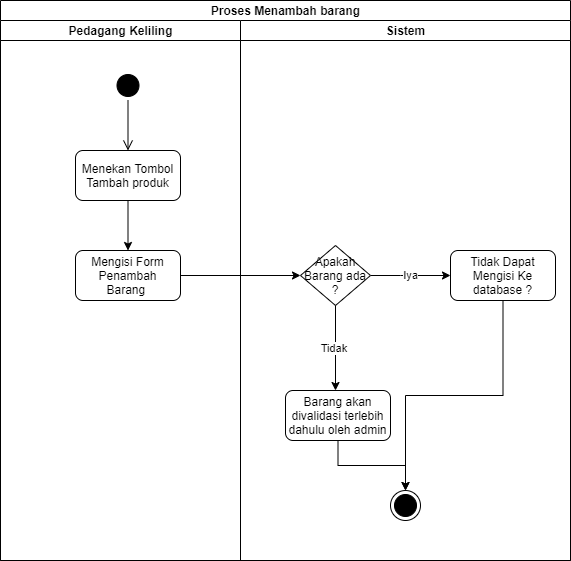


Gambar 3.6 List Pedagang Keliling.



Gambar 3.7 Search Produk.

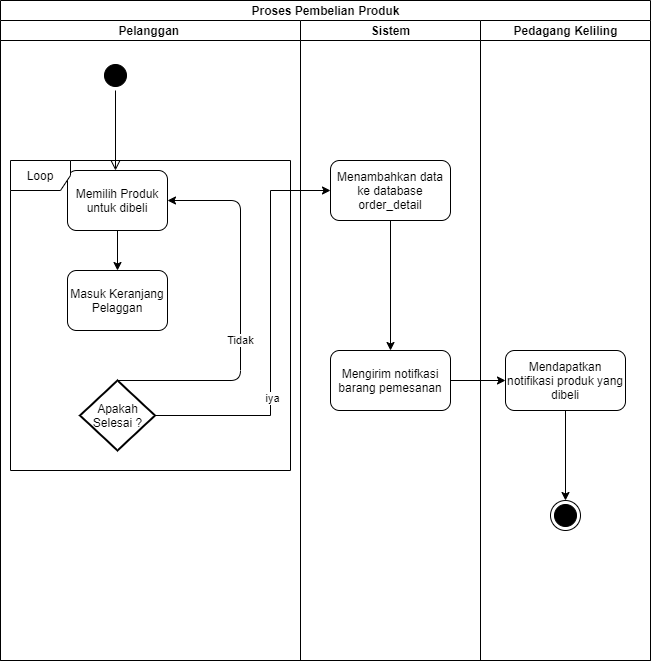
**3.6.3 Proses Menambah Produk Pedagang Keliling**

****

Gambar 3.8 Menambah Barang

Proses menambah barang merupakan salah satu fitur yang dimiliki oleh pedagang keliling. Pada gambar 3.8 merupakan activity diagram untuk proses menambah barang . Menambahkan barang yang akan dibawak atau dijual oleh pedagang keliling .

**3.6.4 Proses Pembelian Barang**

****

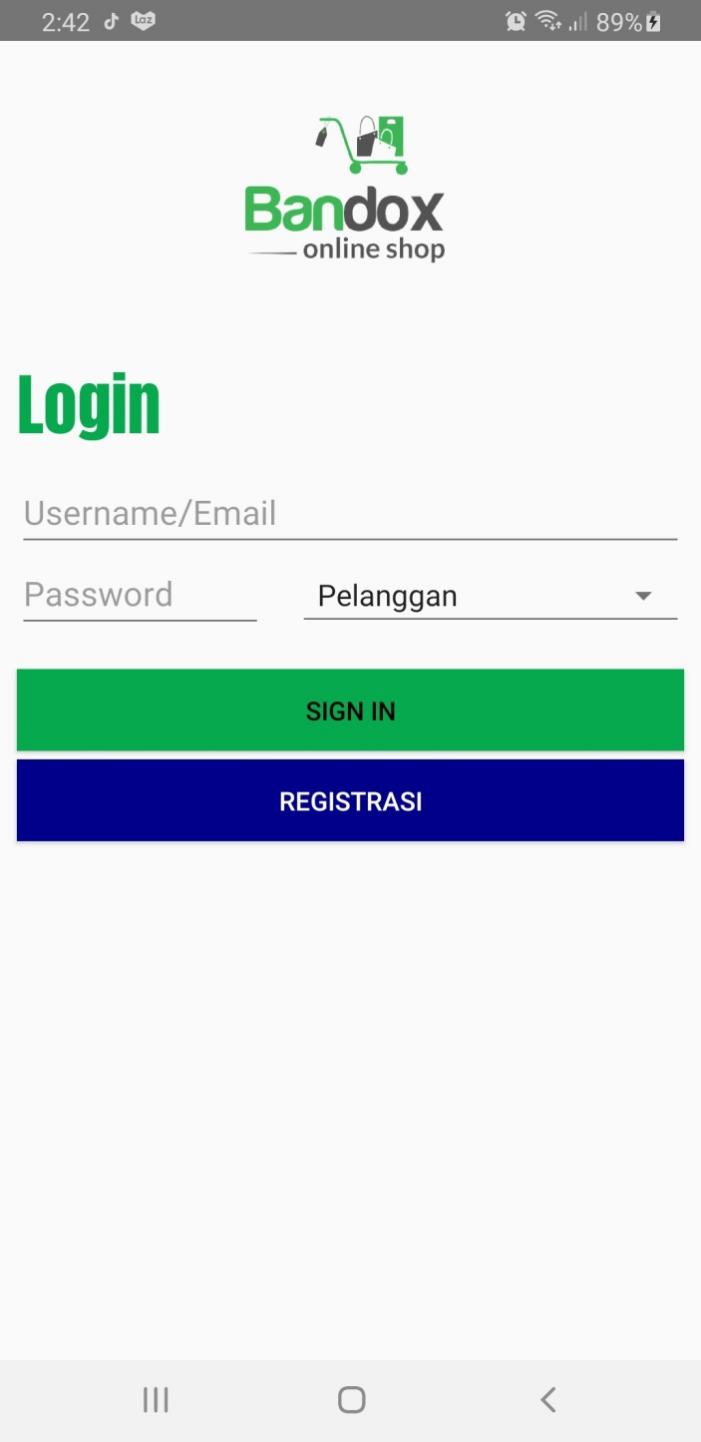
Gambar 3.9 Proses Pembelian Produk

Proses pembelian barang merupakan fitur pelanggan untuk melakukan pembelian barang. Pertama pelanggan akan menginput barang – barang kedalam kerajang belanjaan dan lalu barang akan diantar oleh pedagang keliling.

**3.7 Desain Interface Pelanggan**

Membentuk tampilan yang dibutuhkan oleh kebutuhan *user* pelanggan. Tampilan pertama adalah tampilan Login. Tampilan kedua adalah Registrasi. Tampilan ketiga merupakan tampilan home, aktivitas , menu makanan ,profile dan history user.

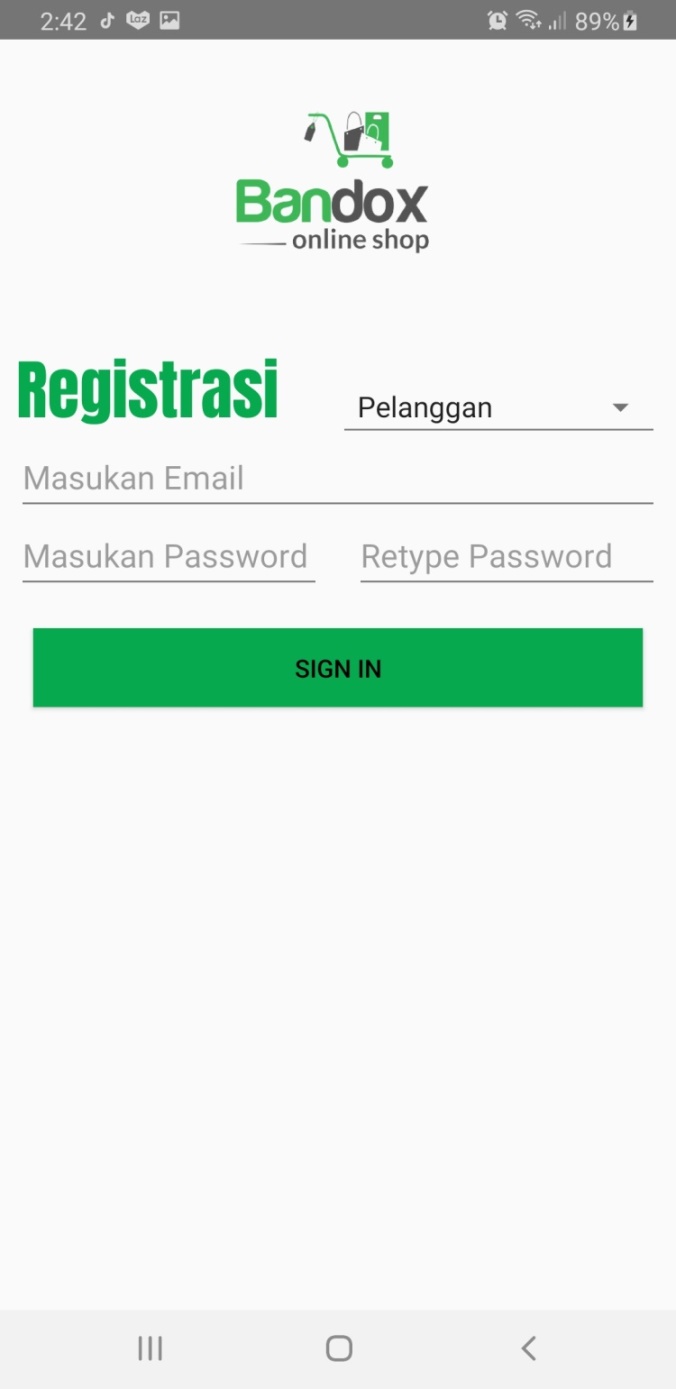
**3.7.1 Tampilan Login Pelanggan**

****

Gambar 3.9 Tampilan Login

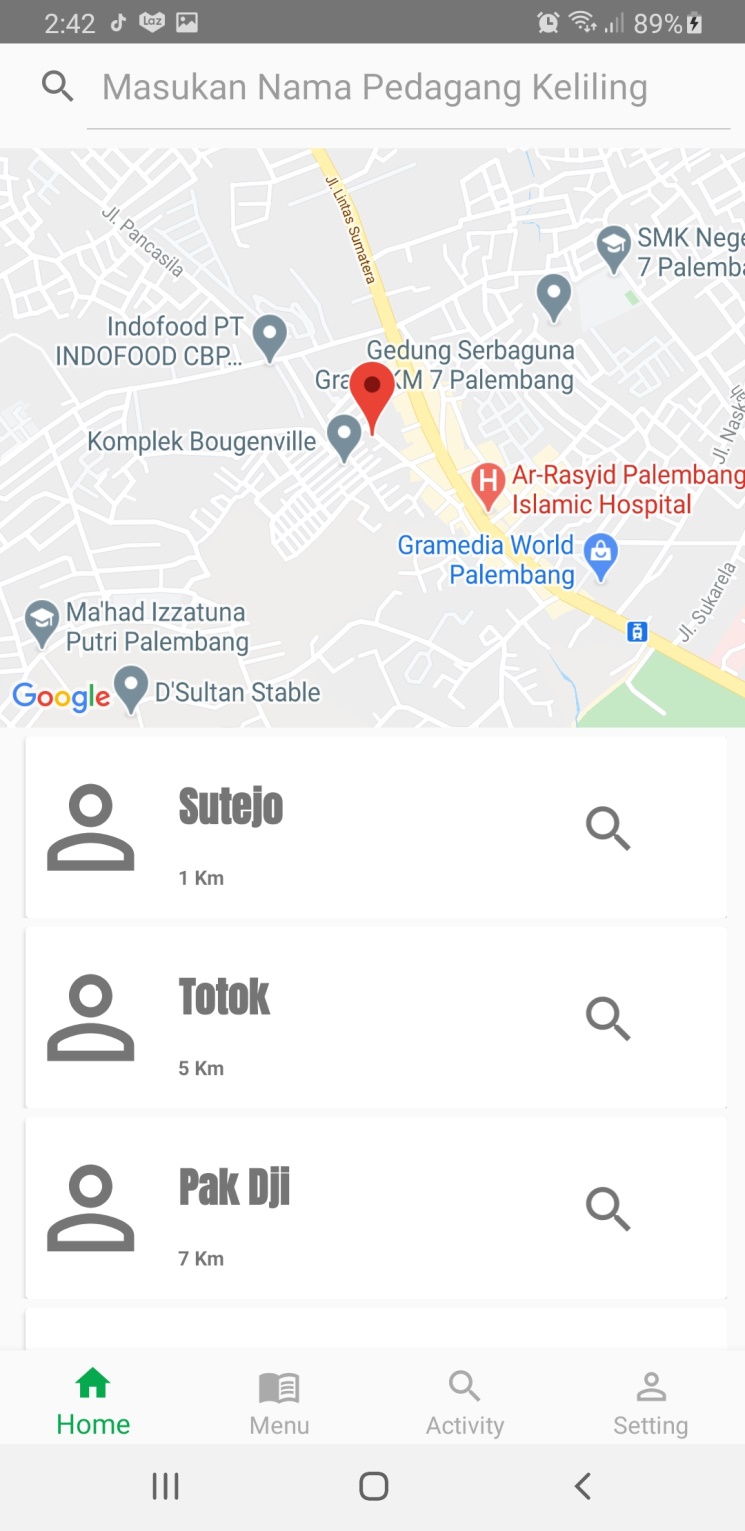
Tampilan desain login dengan tema bewarna hijau. Melakukakn pengecekan *user* pelanggan melalui API.

**3.7.2 Tampilan Registrasi Pelanggan**

****

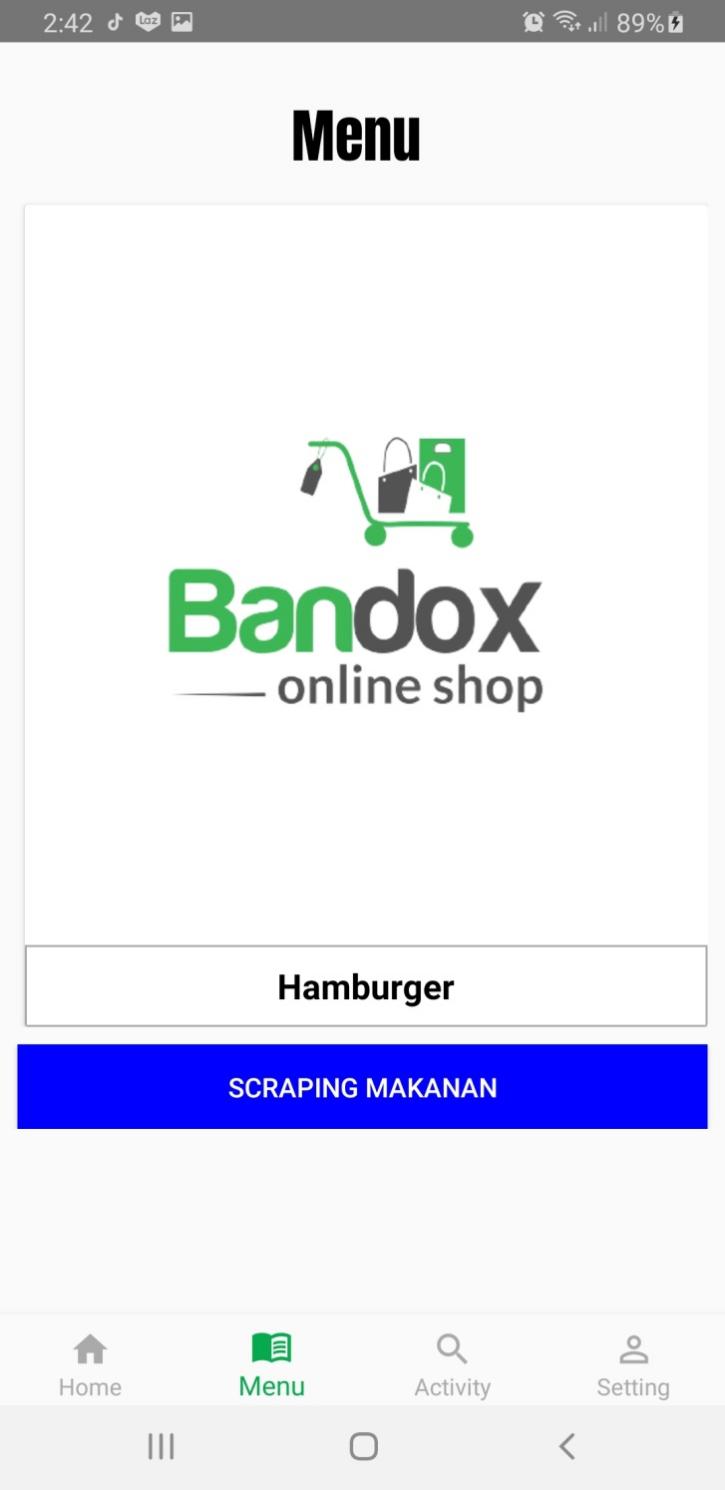
Gambar 3.9 Tampilan Registrasi Pelanggan

**3.7.3 Tampilan *Home* Pelanggan**

****

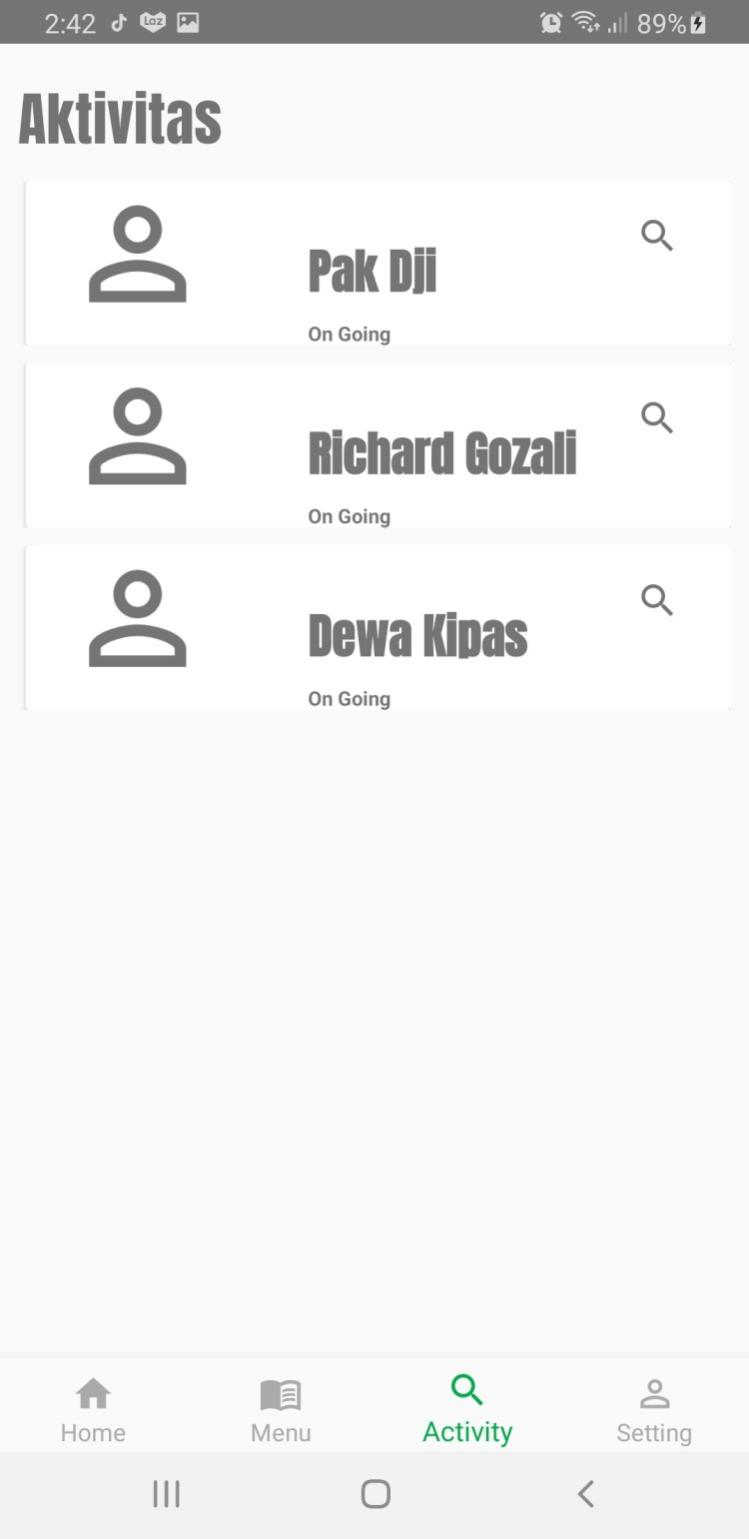
Gambar 4.0 Tampilan *Home* Pelanggan

**3.7.3 Tampilan Menu Makanan Pelanggan**

****

Gambar 4.1 Tampilan Menu Makanan

**3.7.4 Tampilan Aktivitas Pelanggan**

****

Gambar 4.2 Tampilan Menu Aktivitas Pelanggan

**3.8 Desain Interface Pedagang Keliling**

**3.9 Desain Interface Admin**