BAB 3
ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem

Analisis bertujuan untuk mencari tahu permasalahan-permasalahan yang ada pada sistem serta menentukan kebutuhan apa saja yang dibutuhkan oleh sistem yang akan dibangun. Analisis yang akan dibahas adalah analisis masalah, analisis kebutuhan perangkat lunak (fungsional) dan analisis non fungsional.

Untuk membangun sebuah aplikasi yang nantinya akan berfungsi sesuai dengan kebutuhannya sangatlah diperlukan analisis terhadap sistem yang sedang berjalan pada saat ini atau yang berjalan secara umum. Analisis ini bertujuan untuk membuat sistem yang akan dibangun tidak keluar dari sistem yang sedang berjalan saat ini.

3.1.1 Analisis Masalah

Analisis masalah bertujuan untuk mengidentifikasi masalah-masalah yang akan dipecahkan. Analisis masalah juga merupakan langkah pertama dalam tahap analisis sistem. Masalah akan dapat teridentifikasi bisa dari suatu pertanyaan. Masalah ini yang menyebabkan sasaran dari sistem tidak tercapai. Maka dari itu, langkah pertama yang harus dilakukan pada tahap analisis masalah adalah mengidentifikasi terlebih dahulu masalah-masalah yang terjadi.

Berdasarkan hasil kuisioner yang telah dilakukan ditemukan beberapa masalah, yaitu :

1. Masyarakat membutuhkan sebuah aplikasi yang dapat mencari tempat kuliner asia di kota Bandung sesuai jalur yang diinginkan.
2. Masyarakat membutuhkan sebuah aplikasi yang dapat mencari tempat kuliner asia di kota Bandung sesuai dengan nama makanan.
3. Masyarakat membutuhkan sebuah aplikasi yang dapat memberikan info kuliner asia apa yang sedang trending di Instagram.

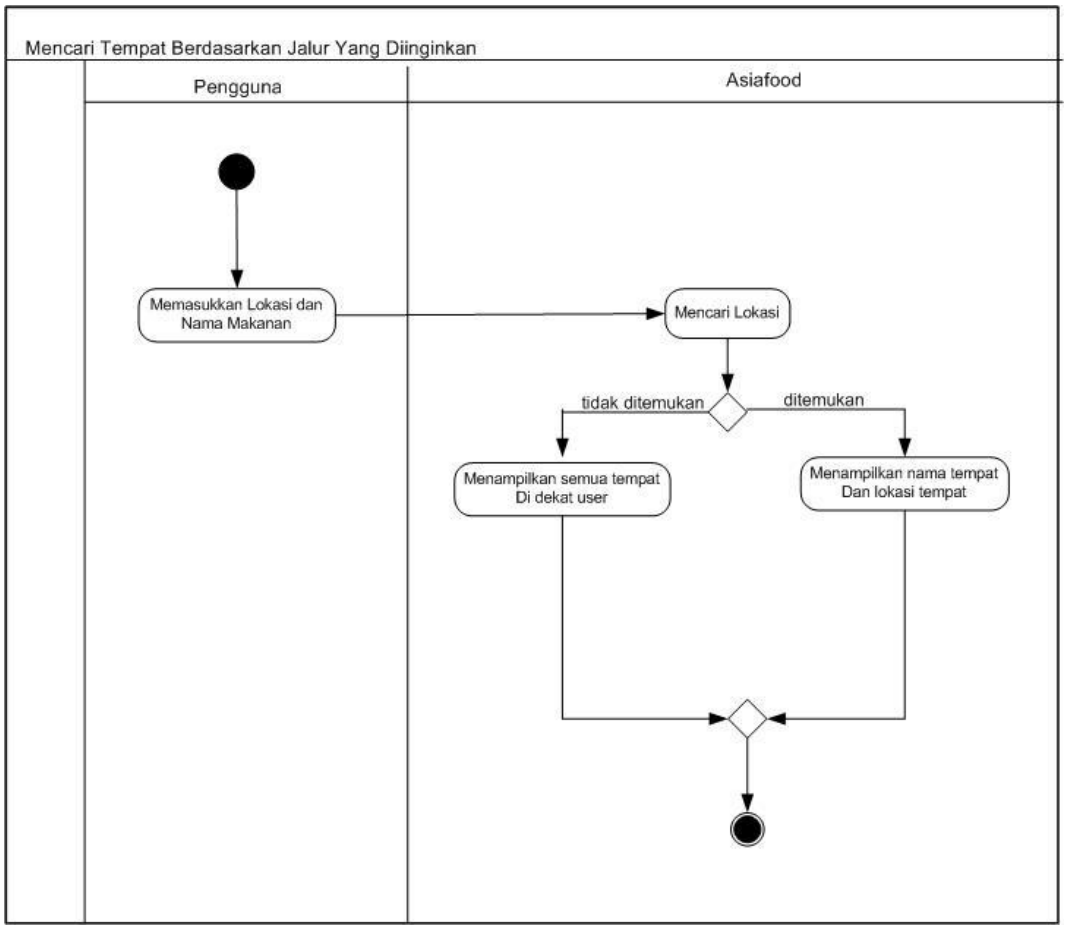
3.1.2 Analisis Aturan Bisnis

Analisis aturan bisnis dibuat bertujuan untuk mengetahui aturan bisnis perangkat lunak yang akan dibangun.

3.1.2.1 Analisis Sistem Yang Akan Dibangun

Analisis sistem yang akan dijelaskan disini adalah analisis sistem yang akan dibangun seperti gambaran bagaimana aturan dan proses dalam perangkat lunak yang akan di bangun ini, berikut adalah analisis yang di lakukan :

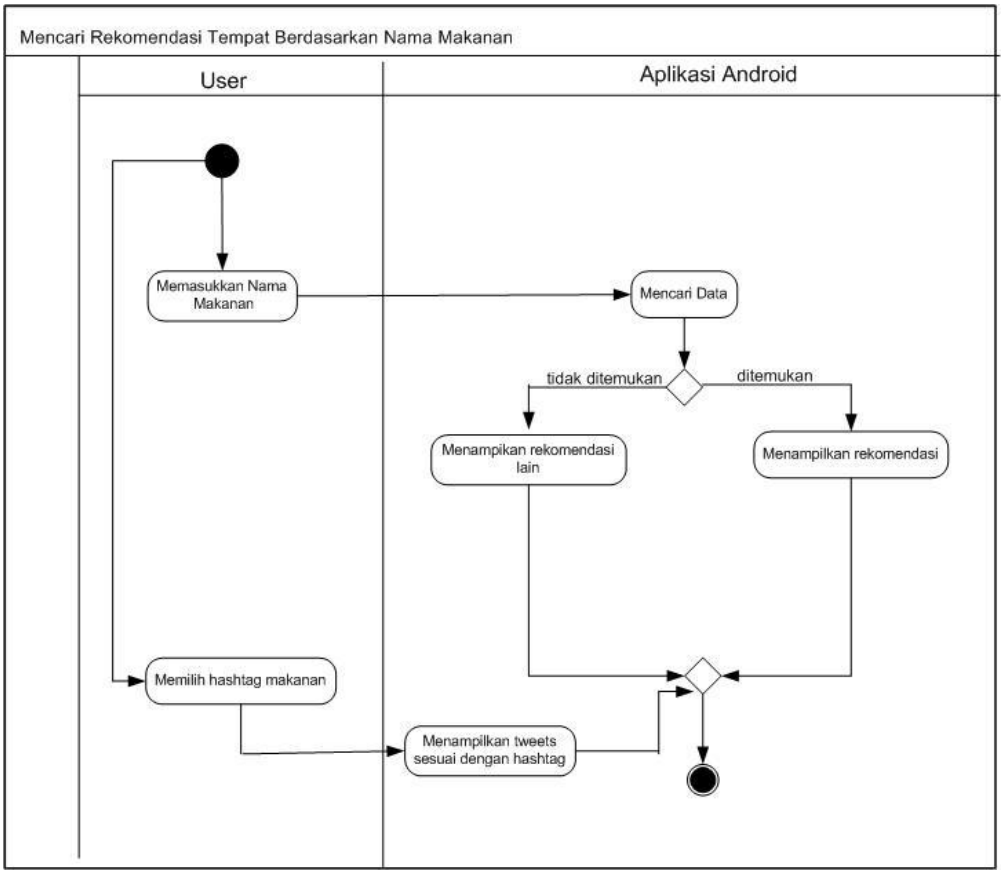
1. Mencari Tempat Berdasarkan Jalur Yang Diinginkan



Gambar 3.1 Proses Bisnis Mencari Tempat Berdasarkan Jalur Yang Diinginkan

a. Untuk mencari tempat nantinya pengguna akan bisa mencari tempat berdasarkan jalur/jalan yang akan dilalui oleh pengguna.
b. Pengguna sebelumnya harus memasukkan jalur yang akan dilewati, pertama aplikasi akan menunjukkan tempat pengguna berada lalu pengguna memasukkan tempat yang akan dituju dan aplikasi yang akan dibangun ini menunjukkan beberapa nama tempat yang disepanjang jalan yang akan pengguna lalui.
c. Pengguna juga selain bisa memasukkan nama jalan, pengguna juga bisa memasukkan nama makanan yang ingin dibeli/dicari. Aplikasi yang akan dibangun ini lalu akan menampilkan tempat yang sesuai dengan nama makanan yang pengguna masukkan. Nama tempat yang tampil merupakkan nama tempat yang berada di jalur yang dilalui oleh pengguna.

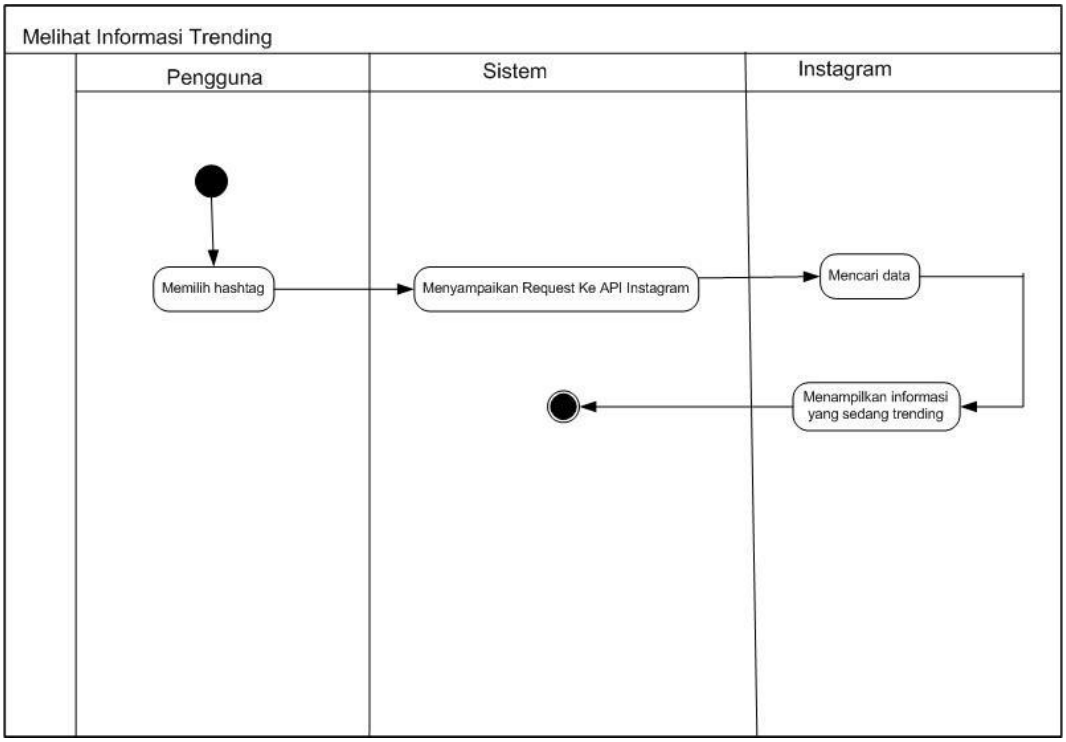
2. Mencari Tempat Berdasarkan Nama Makanan



Gambar 3.2 Proses Bisnis Mencari Tempat Berdasarkan Nama Makanan

a. Untuk menu mencari tempat berdasarkan nama makanan, nanti perangkat lunak yang akan dibangun ini akan akan menampilkan nama tempat sesuai dengan nama makanan yang dimasukkan oleh pengguna.
b. Nama makanan yang dimasukkan adalah nama makanan yang diinginkan oleh pengguna.
c. Nama tempat yang akan tampil adalah hasil cari dari nama makanan yang dimasukkan oleh pengguna, aplikasi yang akan dibangun ini juga menampilkan tempat sesuai dengan rating yang bagus terlebih dahulu. Rating tersebut didapatkan dari Zomato.

3. Melihat Informasi Trending



Gambar 3.3 Proses Bisnis Melihat Informasi Trending

a. Pengguna dapat melihat informasi trending makanan berdasarkan hashtag yang dipilih.
b. Pengguna dapat melihat informasi trending yang diambil dari Instagram.
c. Hashtag yang dipilih nanti merupakan nama-nama negara di asia.

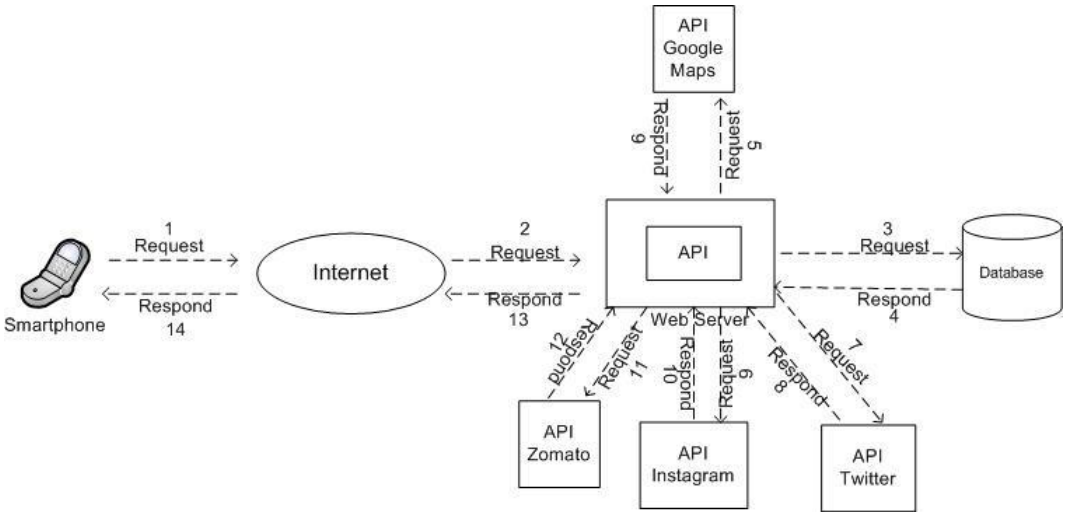
d. Setelah pengguna memilih hashtag tersebut, aplikasi yang akan dibangun ini menampilkan beberapa foto yang sedang trending di Instagram berdasarkan hashtag yang telah dipilih.

3.1.3 Analisis Arsitektur Sistem

Analisis arsitektur sistem tujuannya untuk menidentifikasi arsitektur sistem yang akan dibangun. Berikut adalah Arsitektur Sistem yang digunakan dalam pembangunan aplikasi ini.

1. Platform *Mobile*

Platform mobile android merupakan salah satu sistem yang telah dipilih untuk membangun perangkat lunak ini. Disini akan dijelaskan arsitektur perangkat lunak dari platform mobile android. Arsitektur perangkat lunak akan menggambarkan bagaimana interaksi sistem berinteraksi satu sama lain. Ilustrasi tersebut dapa dilihan pada Gambar 3.4 Arsitektur Perangkat Lunak *Platform Mobile*.



Gambar 3. 4 Arsitektur Perangkat Lunak *Platform Mobile*

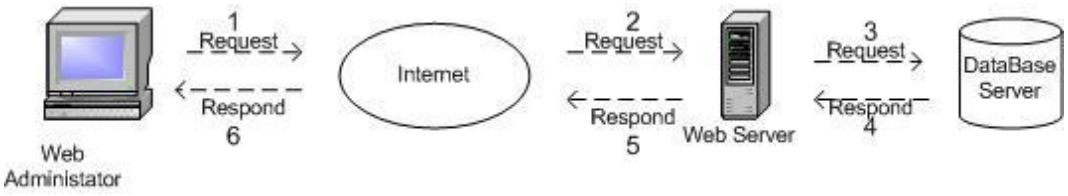
1. Pengguna yang mengakses aplikasi pada *smartphone* me*request* sebuah perintah yang harus melewati internet terlebih dahulu.

2. *Web server* menerima *request* dari *smartphone* dan menentukan jenis *request*. Perangkat *smartphone* pengguna melakukan request data *ke web server* melalui *API*.
3. Setelah itu jika *web server* telah menerima *request* dari *smartphone* pengguna maka akan di check kembali jika membutuhkan data di *database* maka *web server* akan mengambil data yang ada di *database*.
4. Setelah *web server* menerima data yang diminta, maka data akan dikembalikan ke API sebagai sebuah API.
5. Jika ternyata *web server* membutuhkan data lokasi maka *server* memanggil API Google Maps untuk mendapatkan *koordinat*.
6. Lalu dikembalikan pada *web server* lagi dengan mengirimkan *longitude* dan *latitude* yang nantinya akan dikembalikkan pada *smartphone* melalui *internet* terlebih dahulu.
7. Jika ternyata data yang diminta dari *web server* adalah data yang harus di ambil dari twitter maka *web server* akan melakukan proses pengambilan data/*request* data ke twitter melalui API twitter. Selain melakukan *request* terhadap API Twitter *web server* akan melakukan *request* terhadap *database* untuk *merequest hashtag* sesuai dengan *request* pengguna.
8. *Request* akan dikembalikan sebagai respond ke *web server*.
9. Jika ternyata data yang diminta dari *web server* adalah data yang harus di ambil yang merupakan data *trending* dari Instagram, *web server* akan mengirimkan *request* data pada Instagram melalui API Instagram. Selain melakukan melakukan request terhadap API Instagram web server juga akan melakukan request ke database untuk me*request* hashtag sesuai dengan *request* pengguna.
10. *Request* akan dikembalikkan sebagai *respond* ke *web server*.
11. Jika ternyata data yang diminta dari *web server* adalah data yang harus di ambil yang merupakan data tempat makan, maka *web service* akan mengirimkan *request* ke API Zomato untuk mendapatkan nama tempat makan dan juga lokasinya yang nanti akan ditembakkan longitude dan latitudenya ke API Google Maps untuk mendapatkan koordinat tempatnya.

12. Request akan dikembalikkan sebagai *respond* ke *web server*.
13. Jika data *request* sudah didapatkan maka *web server* akan mengembalikkan dalam bentuk JSON untuk di proses di *smartphone* pengguna yang sebelumnya harus melalui *internet* terlebih dahulu.
14. Hasil akhir dikirimkan berupa *respond* sesuai dengan request yang diminta oleh pengguna *smartphone* kepada sistem.

2. *Platform Web*

Platform web adalah merupakan salah satu sistem yang dipilih untuk membangun perangkat lunak ini. Nantinya yang akan menggunakan *platform web* adalah admin. Admin disini tugasnya untuk mengontrol data yang dibutuhkan oleh *platform mobile* seperti data *hashtag*, data tempat rekomendasi, dan juga data dari pengguna. Ilustrasi dari sistem yang berkerja pada platform web dapat dilihat pada Gambar 3.5 Arsitektur Perangkat Lunak *Platform Web*



Gambar 3. 5 Arsitektur Sistem *Platform Web*

1. *Web administrator* melakukan request data kepada *web server* yang harus melawati dulu koneksi internet.
2. Lalu *web server* menerima *request* dari *web administrator*. *Web server* lalu menentukan jenis *reques*t apa yang telah diminta oleh *web administrator*.
3. Bila *web server* menerima *request* makan *reques*t akan diteruskan pada *database server* untuk mengambil data yang berada pada *database server*.
4. *Request* lalu dikembalikan menjadi berupa *respond* yang berupa data yang sudah diambil dari *database server*.

5. *Web server* mengiriminkan data yang sudah didapatkan, data harus melewati internet terlebih dahulu.
6. Pada step terakhir komputer *web administrator* akan menerima data yang diminta, data itu didapatkan setelah *request* dikembalikkan melalui internet.

3.1.4 Analisis Teknologi Yang Digunakan

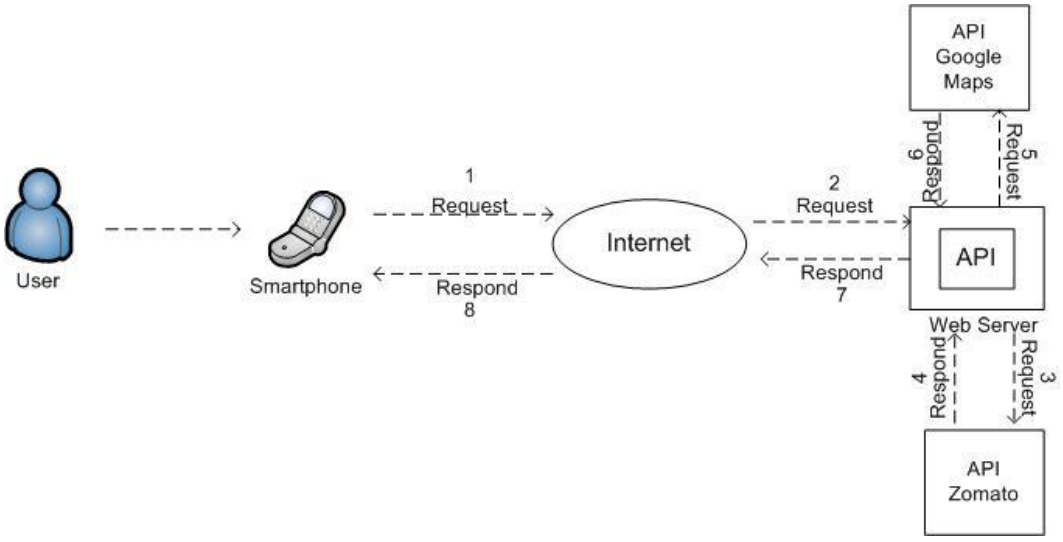
Analisis teknologi dilakukan bermaksud untuk mendeskripsikan teknologi apa saja yang digunakan dalam pembangunan perangkat lunak ini.

3.1.4.1 API *Zomato*

API *Zomato* memberikan akses informasi paling baru dan paling lengkap untuk lebih dari 1.5 juta restoran di lebih dari 10.000 kota di dunia.

Dalam penelitan ini pemanfaatan API Zomato ini tersendiri ditujukan untuk mendapatkan nama tempat makan dan lokasi tempat makan.

1. Menampilkan Tempat Sesuai Dengan Jalur Yang Diinginkan



Gambar 3.6 Analisis Teknologi API Zomato Menampilkan Tempat Sesuai Dengan Jalur Yang Diinginkan

1. Sistem melakukan *request*, request tersebut berupa tempat yang terdekat dari jalur yang akan dilalui oleh pengguna. Dengan *parameter* yang di *request* adalah :
   - [lat][varchar]
   - [lon][varchar]
2. *Request* yang dipinta oleh sistem melewati internet terlebih dahulu. *Request* ini meminta *request* pertama terhadap *web server* dengan *request http://angpao.web.id/asifood_api/resto_api_zomato.php*
3. *Web server* akan meneruskan *request* terhadap API Zomato, request yang dikirim adalah *longitude* dan *latitude* dari tempat pengguna pengguna berada untuk mendapatkan tempat apa saja yang berada di jalur yang akan dituju oleh pengguna.
4. Sistem akan mengembalikan hasil pengguna menjadi hasil *respond*.
5. Sistem juga akan melakukan *request latitude* dan *longitude* pada API *Google Maps* untuk menampilkan titik dimana saja tempat itu berada.
6. API *Google Maps* mengirimkan kembali *request* menjadi *respond* yang berupa format JSON sebagai hasil akhir.

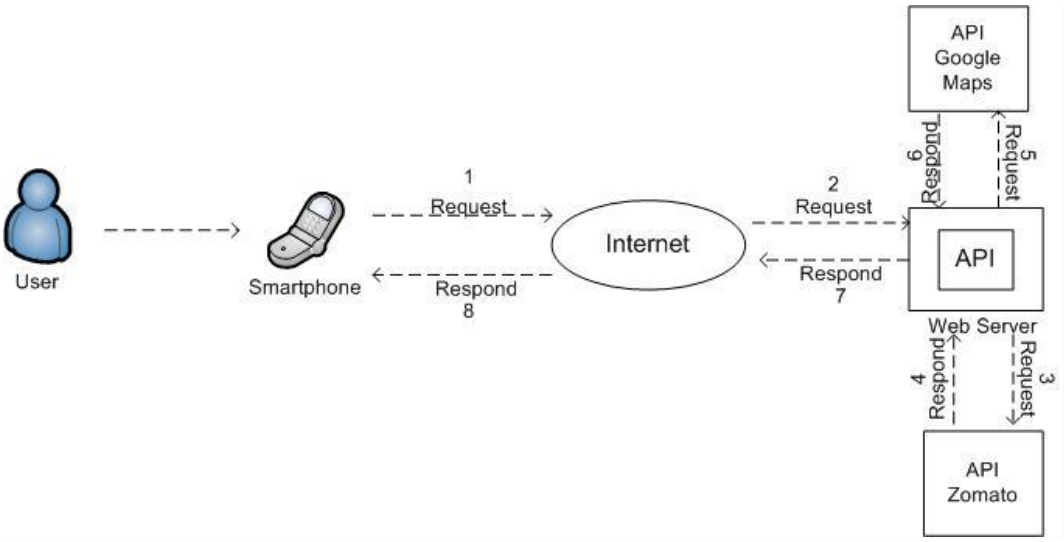
Contoh Respond JSON yang berhasil :


```
{
            "results_found": 2587,
            "results_start": 0,
            "results_shown": 20,
            "restaurants": [
{
                "restaurant": {
                    "R": {
                        "res_id": 18407025
                    },
                    "apikey":
        "c6bbbe623b800201f4aa191bd48ad74c",
```

```
                    "id": "18407025",
                    "name": "Kalpa Tree ",
                    "url":
        "https://www.zomato.com/bandung/kalpa-tree-
        ciumbuleuit?utm_source=api_basic_user&utm_medium=api&utm
        _campaign=v2.1",
                    "location": {
                        "address": "Jl. Kiputih No. 37,
        Ciumbuleuit, Bandung",
                        "locality": "Ciumbuleuit",
                        "city": "Bandung",
                        "city_id": 11052,
                        "latitude": "-6.8662014000",
                        "longitude": "107.6082161000",
                        "zipcode": "",
                        "country_id": 94,
                        "locality_verbose":    "Ciumbuleuit,
        Bandung"
                    },
```


7. Sistem mengirimkan hasil akhir dari *request*, sebelumnya harus melewati *internet* terlebih dahulu.
8. *Respond* yang berupa JSON akan dibaca oleh *smartphone* dan pengguna pada akhirnya bisa mendapatkan.

2. Menampilkan Pencarian Berdasarkan Nama Makanan



Gambar 3.7 Analisis Teknologi API Zomato Untuk Menampilkan Pencarian Berdasarkan Nama Makanan

1. Pengguna akan melakukan *request* pencarian tempat kuliner berdasarkan sesuai dengan nama makanan yang akan melalui *internet* terlebih dahulu.
2. *Request* yang telah melalui internet akan dikirimkan pada web server. *Request* yang dikirimkan pada *web server* adalah *http://angpao.web.id /asifood_api/cari_jenismakanan.php*.
3. *Web Server* melakukan request pada *API* Zomato, melakukan request *longitude* dan *latitude* untuk lokasi dan *request* nama makanan yang telah dimasukkan oleh pengguna dengan menggunakan *parameter* "q".
4. Lalu *API Zomato* mengembalikkan *request* menjadi sebuah *respond*.
5. Sebelum dikirimkan kembali, *web server* mengirimkan *latitude* dan *longitude* dari tempat makan yang telah didapatkan ke *API Google Maps* agar dapat ditampilkan titik lokasi dari tempat tersebut.
6. *API Google Maps* mengirimkan kembali *request* menjadi *respond* yang berupa format JSON sebagai hasil akhir.

   Contoh Respond JSON yang berhasil :

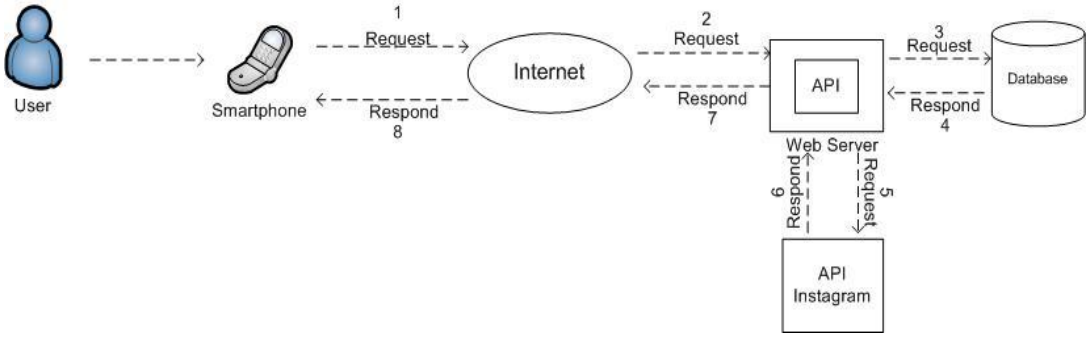
```
{
        "results_found": 24,
        "results_start": 0,
        "results_shown": 20,
        "restaurants": [
          {
                    "restaurant": {
                        "R": {
                            "res_id": 18389082
                        },
                        "apikey":
"c6bbbe623b800201f4aa191bd48ad74c",
                        "id": "18389082",
                        "name": "Bakso Cuangki dan Batagor
Serayu",
                        "url":
"https://www.zomato.com/bandung/bakso-cuangki-dan-
batagor-serayu-
riau?utm_source=api_basic_user&utm_medium=api&utm_campai
gn=v2.1",
                        "location": {
                            "address": "Jl. Serayu No. 2, Riau,
Bandung",
                            "locality": "Riau",
                            "city": "Bandung",
                            "city_id": 11052,
                            "latitude": "-6.9092116072",
                            "longitude": "107.6254244149",
                            "zipcode": "",
                            "country_id": 94,
                            "locality_verbose": "Riau, Bandung"
                        },
```


7. Sistem mengirimkan hasil akhir dari *request*, sebelumnya harus melewati *internet* terlebih dahulu.
8. Respond yang berupa JSON akan dibaca oleh *smartphone* dan pengguna pada akhirnya bisa mendapatkan jawaban.

3.1.4.2 API *Instagram*

Platform API Instagram dapat digunakan untuk membuat non-otomatis, otentik, aplikas, dan service kualitas tinggi yang : membantu individu membagikan konten mereka sendiri dengan aplikasi pihak ke-3. Membantu mereka dan pengiklan memakhami, mengelola pelanggan dan hak media mereka. Membantu penyiar dan penerbit menemukan konten, mendapatkan hak digital terhadap media, dan memabagikan media dengan atribusi yang tepat.

Pemanfaatan Instagram untuk di pembangunan perangkat lunak ini untuk melihat informasi makanan yang sedang trending berdasarkan *hashtag* pada *Instagram*.



Gambar 3.8 Analisis Teknologi API Instagram

1. Pengguna melakukan *request* suatu perintah untuk melihat informasi yang sedang *trending* yang harus melewati internet terlebih dahulu. Dengan parameter yang direquest adalah : - [tag][varchar]
2. *Request* yang telah melewati internet akan dikirimkan pada *web server*. *Request* yang akan diminta akan langsung ditujukan pada *database*.
3. *Web Server* akan mengirimkan *request* ke *database* untuk meminta data *"hashtag"* yang dimana *hashtag* akan mempunyai "id" dan "tag". "id" berupa angka mulai dari angka 1 sampai dengan 7, sedangkan untuk "tag" akan mengandung nama negaranya mulai dari thailand, jepang, korea, china, india, malaysia, indonesia secara berurutan.
4. Lalu *database* akan mengirimkan *respond* pada *web server* yang akan langsung diteruskan kepada *API Instagram*.
5. Setelah didapatkan *hashtag* yang sesuai. *Request* yang akan dikirimkan adalah *https://www.instagram.com/explore/tags/makanan"+tag"/?__a=1*

6. *API Instagram* lalu mengirimkan hasil akhir yang berupa format JSON untuk menampilkan informasi yang sedang trending di *Instagram* sesuai dengan *hashtag.*

   Contoh *JSON* yang berhasil :


```
{
          "tag": {
              "name": "makananthailand",
              "content_advisory": null,
              "media": {
                  "nodes": [
                      {
                          "comments_disabled": false,
                          "id": "1652694250277366079",
                          "dimensions": {
                              "height": 1080,
                              "width": 1080
                          },
                          "owner": {
                              "id": "270659974"
                          },
                          "thumbnail_src":
      "https://scontent-sin6-2.cdninstagram.com/t51.2885-
      15/s640x640/sh0.08/e35/23668065_147623442629611_363467
      3497952747520_n.jpg",
                          "thumbnail_resources": [
                              {
                                  "src":  "https://scontent-
      sin6-2.cdninstagram.com/t51.2885-
      15/s150x150/e35/23668065_147623442629611_3634673497952
      747520_n.jpg",
                                  "config_width": 150,
                                  "config_height": 150
                              },
                              {
                                  "src":  "https://scontent-
      sin6-2.cdninstagram.com/t51.2885-
      15/s240x240/e35/23668065_147623442629611_3634673497952
      747520_n.jpg",
                                  "config_width": 240,
                                  "config_height": 240
                              },
                              {
                                  "src":  "https://scontent-
      sin6-2.cdninstagram.com/t51.2885-
      15/s320x320/e35/23668065_147623442629611_3634673497952
      747520_n.jpg",
```

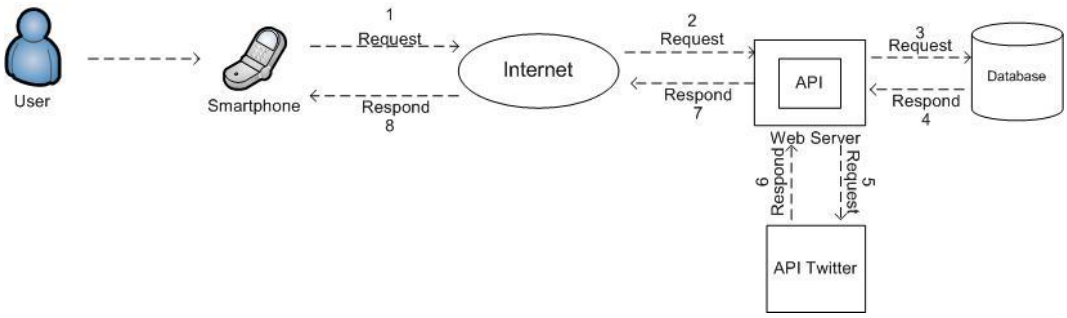
```
                                "config_width": 320,
                                "config_height": 320
                            },
                            {
                                "src":  "https://scontent-
sin6-2.cdninstagram.com/t51.2885-
15/s480x480/e35/23668065_147623442629611_3634673497952
747520_n.jpg",
                                "config_width": 480,
                                "config_height": 480
                            },
                            {
                                "src":  "https://scontent-
sin6-2.cdninstagram.com/t51.2885-
15/s640x640/sh0.08/e35/23668065_147623442629611_363467
3497952747520_n.jpg",
                                "config_width": 640,
                                "config_height": 640
                            }
                        ],
```


7. Hasil akhir yang berupa JSON tersebut dikembalikan kepada *smartphone* tetapi sebelumnya harus melewati internet terlebih dahulu.
8. *Respond* yang berupa JSON akan dibaca oleh *smartphone* dan pada akhirnya pengguna akan mendapatkan jawaban.

3.1.4.3 API *Twitter*

API Twitter atau *Application Programming Interface* (API) *Twitter* adalah suatu aplikasi yang dibuat oleh *twitter* untuk mempermudah developer untuk mengakses informasi yang ada di *twitter*.

Pemanfaatan *twitter* pada aplikasi ini adalah untuk mencari rekomendasi melalui *twitter* yang didapatkan melalui *hashtag*.



Gambar 3.9 Analisis Teknologi API Twitter

1. Pengguna melakukan suatu *request* perintah yang dimana *request* tersebut harus melewati *internet* terlebih dahulu untuk aksesnya.
2. *Request* yang berhasil melewati *internet* lalu dikirim ke *web server*. *Request* yang dilakukan adalah untuk mengakses *http://angpao.web.id/asifood_api/twitter.php*
3. *Web Server* mengirimkan *request* ke *database* untuk mengambil data *hashtag* yang sudah ada.
4. *Database* mengirimkan kembali *request* sebagai *respond* ke *web server* untuk selanjutnya dikirimkan kepada *API Twitter hashtag* yang sudah diambil dari *database*
5. *Web server* melakukan *request* sebuah data yang dibutuhkan ke API Twitter. Untuk mendapatkan *hashtag* dan *tweet*, *coding* yang digunakan pada pemograman *java* adalah sebagai berikut ;

```
public void ambilTweet() {
    String url = AppConstans.APITWEET;
    Map<String, String> params = new HashMap<>();
    params.put("hashtag", hashtag);
```

lalu ada pemanggilan *twitteritems* untuk mendapatkan keseluruhan hasil dari *tweet* yang akan diambil. *Coding* ini digunakan untuk mengambil *tweet* secara keseluruhan dan sesuai dengan kebutuhan.

```
text = object.getString("text");
created_at = object.getString("created_at");
user = object.getJSONObject("user");
user_tweet = user.getString("name");
screen_name = user.getString("screen_name");
profile_image_url = user.getString("profile_image_url");
if (object.has("extended_entities")) {
    entities = object.getJSONObject("extended_entities");
    media = entities.getJSONArray("media");
    JSONObject url = media.getJSONObject(0);
    media_url = url.getString("media_url");
```

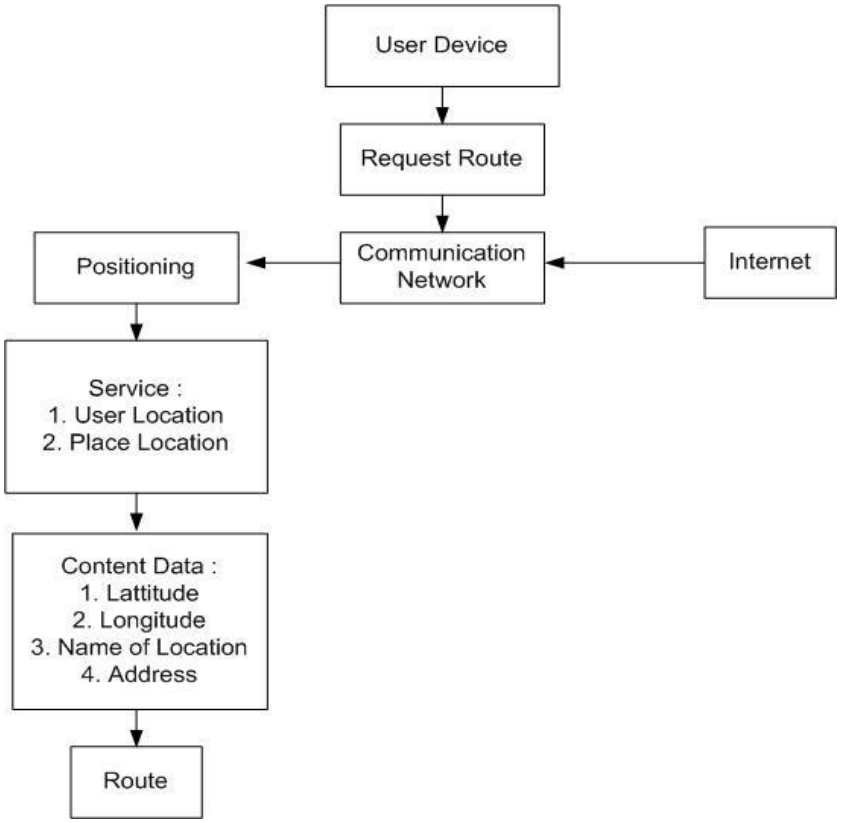
Hasil akhir sebuah *respond* dari request kepada API Twitter yang akan berupa format JSON.

6. Hasil akhir dikirimkan kembali kepada pengguna yang harus melalui *internet* terlebih dahulu.
7. Hasil akhir yang berupa JSON yang siap dibaca oleh *smartphone* yang telah melalui *internet* terlebih dahulu.

3.1.4.4 Metode *Location Based Service* (LBS)

Pada penelitian ini akan menggunakan metode *Location Based Service* (LBS) untuk mendapatkan lokasi tepat dimana lokasi pengguna pemilik *smartphone* berada dan juga untuk mengetahui tempat apa saja yang ada di sekitar tempat pengguna berada untuk rekomendasi.

Berdasarkan hasil penelitian yang ada pada jurnal dengan judul "Pengembangan Aplikasi *Location Based Service* Untuk Informasi dan Pencarian Lokasi Pariwisata di Kota Cimahi Berbasis Android" perangkat lunak yang digunakan pada aplikas informasi dan pencarian rute dititik beratkan pada pengambilan koordinat, perangkat lunak memanfaatkan *GPS*. Kemudia untuk mendapatkan peta dan rute, sistem akan menggunakan *GoogleMaps API* demhan memberikan parameter koordinat bumi. Setelah mengirim parameter tersebut ke *GoogleMap Server* akan berupa peta statik. Didalam pencarian rute, perangkat lunak akan mengirimkan dua koordinat bumi sebagai alamat awal dan alamat tujuan, lalu *GoogleMap Direction Server* akan membalas berupa data dan rute yang akan ditampilkan pada pengguna. **[**20**]**



Gambar 3.10 Diagran Implementasi LBS

3.1.5 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non-fungsional, yaitu merupakan analisis yang diperuntukan untuk menentukan spesifikasi dari kebutuhan sistem yang akan dibangun. Spesifikasi yang dimaksud adalah elemen atau perangkat-perangkat yang dibutuhkan untuk sistem yang akan dibangun ini atau sampai sistem yang dibangun ini sudah dapat diimplementasikan. Analasis kebutuhan non-fungsional juga berfungsi sebagai penentu spesifikasi masukkan apa saja yang diperlukan oleh sistem, juga keluaran (*output*) yang akan dihasilkan. Kebutuhan non-fungsional terbia menjadi beberapa analisis yaitu analisis perangkat keras, analisis perangkat lunak dan analisis pengguna.

3.1.5.1 Analisis Kebutuhan Perangkat Keras (Hardware)

Analisis kebutuhan perangkat keras adalah sebuah uraian atau penjelasan kebutuhan-kebutuhan non fungsional yang berhubungan dengan spesifikasi

perangkat keras (*hardware*) yang berhubungan juga dengan proses pembangunan perangkat lunak.

Spesifikasi dari perangkat keras (*hardware*) minimal yang diperlukan untuk menjalankan perangkat lunak ini pada *platform mobile (android)* adalah seperti yang ditunjukkan pada Tabel 3.1 Spesifikasi Kebutuhan Perangkat Keras *Android* Minimal.

Tabel 3.1 Spesifikasi Perangkat Keras Android

Jenis	Spesifikasi
Processor	Quad Core 850 MHz
RAM	1.0 GB
Memory Internal	16 GB
OS	Android 5.1 Lollipop

Tabel 3.2 Spesifikasi Kebutuhan Perangkat Keras Komputer Minimal

Jenis	Spesifikasi
Processor	Quad Core 850 MHz
RAM	4.0 GB
Harddisk	97.5 GB
Monitor	14 inchi

3.1.5.2 Analisis Kebutuhan Perangkat Lunak (Software)

Spesifikasi kebutuhan perangkat lunak (*software*) untuk membangun aplikasi dan simulasi program adalah seperti yang ditampilkan pada Tabel 3.3 Spesifikasi Kebutuhan Perangkat Lunak.

Tabel 3.3 Spesifikasi Kebutuhan Perangkat Lunak

Jenis	Spesifikasi
Sistem Operasi Android	5.0 Lollipop
Sistem Operasi Komputer	Miscrosoft Windows 7 Ultimate 64 Bit
Integrated Deveploment Environment (IDE)	Android Studio
Mockup Tool	Balsamiq Mockup
UML Tool	Microsoft Visio 2007 dan StarUML

3.1.5.3 Analisis Pengguna

Sebuah aplikasi akan berjalan secara optimal dan semestinya bila aplikasi tersebut dijalankan oleh pengguna yang memiliki kemampuan dalam menjalankan aplikasi tersebut. Untuk karakteristik dari pengguna akan ditunjukan pada Tabel 3. Analisis Pengguna Yang Ada, sebagai berikut :

Tabel 3.4 Analisis Pengguna Yang Ada

User	Karakteristik
Pengguna	- Bisa membaca. - Mampu dan mengerti bagaimana cara mengoperasikan smartphone android. - Mampu mengoperasikan aplikasi.
Administator	- Mengerti bahasa pemograman khususnya pemograman web. - Mampu dan mengerti mengoperasikan database. - Mampu dan mengerti mengoperasikan sebuah website.

3.1.6 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional yang akan dilakukan disini adalah merupakan analisis yang menjelaskan tentang kebutuhan-kebutuhan apa saja yang akan dibutuhkan oleh sistem. Berikut adalah analisis kebutuhan fungsional dan non fungsional pada aplikasi yang akan di bangun ini baik untuk *platform mobile* dan *platform website.*

Tabel 3.5 Spesifikasi Kebutuhan Perangkat Lunak Fungsional Pada *Platform Mobile*

Kode Kebutuhan	Kebutuhan
SKPL-F-01	Sistem menyediakan fasilitas untuk pendaftaran
SKPL-F-02	Sistem menyediakan fasilitas untuk login
SKPL-F-03	Sistem menyediakan fasilitas lupa password
SKPL-F-04	Sistem menyediakan fasilitas untuk mengubah profile pengguna
SKPL-F-05	Sistem menyediakan fasilitas untuk mencari tempat berdasarkan jalur yang diinginkan
SKPL-F-06	Sistem menyediakan fasilitas untuk mencari tempat berdasarkan nama makanan
SKPL-F-07	Sistem menyediakan fasilitas untuk melihat informasi yang sedang trending di Instagram

Tabel 3.6 Spesifikasi Kebutuhan Perangkat Lunak Non-Fungsional Pada *Platform Mobile*

Kode Kebutuhan	Kebutuhan
SKPL-NF-01	Sistem operasi berbasis mobile android
SKPL-NF-02	Sistem akan bekerja lebih maksimal bila smartphone memiliki fitur GPS
SKPL-NF-03	Sistem dapat dijalankan pada android yang memiliki versi minimum Android 5.0 (Lollipop)
SKPL-NF-04	Sistem dapat diakses selama 24 jam
SKPL-NF-05	Sistem dapat diakses selama ada koneksi internet

Tabel 3.7 Spesfikasi Kebutuhan Perangkat Lunak Fungsional Pada *Platform Website*

Kode Kebutuhan	Kebutuhan
SKPL-F-01	Sistem dapat menyediakan fasilitas menambah, mengedit, menghapus data tempat rekomendasi
SKPL-02	Sistem dapat menyediakan fasilitas menambah, mengedit, menghapus data hashtag
SKPL-03	Sistem dapat menyediakan fasilitas menambah, mengedit, menghapus data pengguna

Tabel 3.8 Spesifikasi Kebutuhan Perangkat Lunak Non Fungsional Pada *Platform Website*

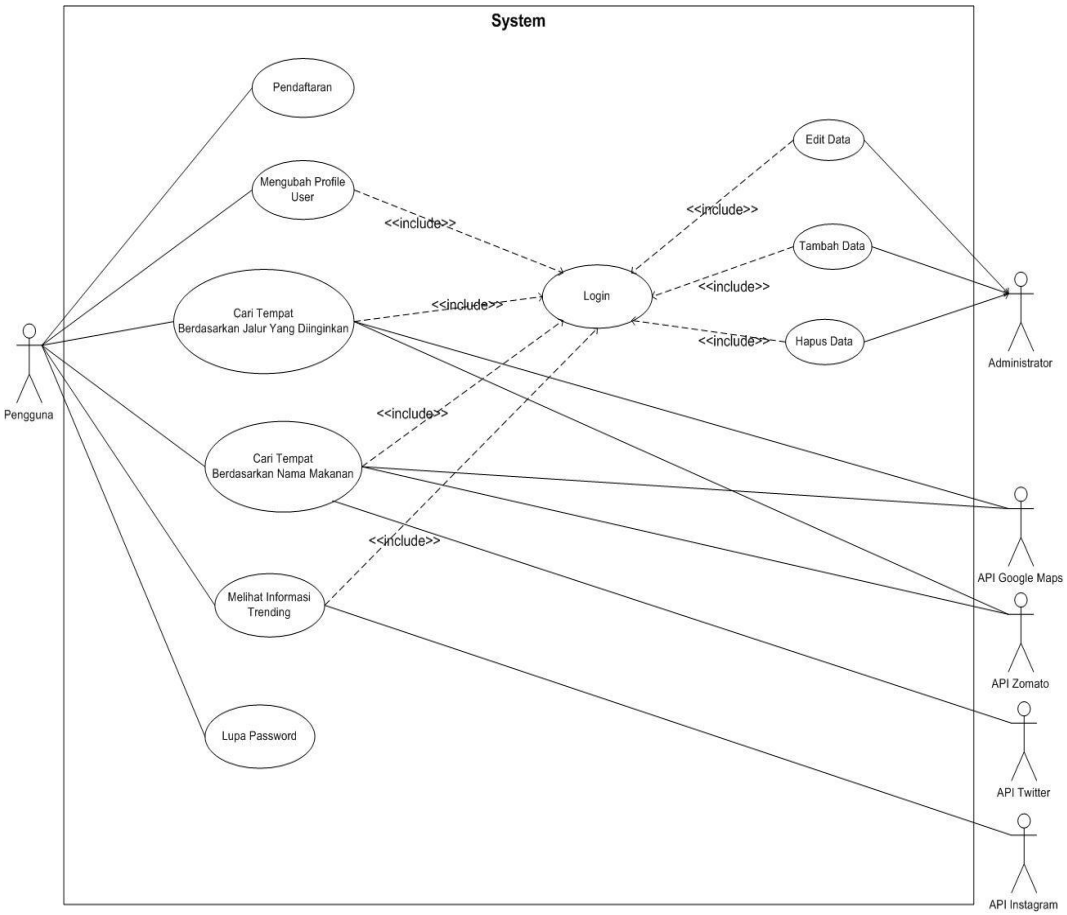
Kode Kebutuhan	Kebutuhan
SKPL-NF-01	Sistem berbasis website
SKPL-NF-02	Sistem tidak dapat diakses selama 24 jam

3.1.6.1 Analisis Kebutuhan Fungsional Pada *Platform Mobile*

Analisis kebutuhan fungsional pada platform mobile akan dilakukan menggunakan tools yaitu UML. Tahap dari analisis dengan menggunakan UML meliputi *Use Case* Diagram, *Sequence* Diagram, *Activity* Diagram, *Class* Diagram. Penjelasan dari analisis kebutuhan fungsional pada platform mobile akan dijelaskan sebagai berikut :

3.1.6.1.1 *Use Case* Diagram

Use Case diagram merupakan sebuah kontruksi yang terjadi di dalam sistem aplikasi ini. Konstuksi ini sendiri akan mendeskripsikan hubungan antar actor dengan aktifitas yang terdapat dalam sistem aplikasi yang akan dibangun ini. Bisa dilihat dibawah ini merupakan *Use Case Diagram* untuk *Platform Android*.



Gambar 3.11 *Use Case* Diagram

Untuk detail dari use case akan digunakan use case scenario untuk menunjukkan kondisi yang sedang terjadi dan langkah-langkah apa saja yang terlibat pada setiap use case. Karena use case diagram tidak memberikan detail yang cukup jelas bagaimana kebutuhan sistem akan terpenuhi. Use case scenario untuk aplikasi yang akan dibangun ini akan dijelaskan sebagai berikut :

1. *Use Case Scenario* Pendaftaran

Scenario use case Pendaftaran akan menjelaskan bagaimana alur proses pada diagram use case tersebut. Berikut adalah tabel penjabaran dari use case Pendaftaran.

Tabel 3.9 *Use Case* Scenario Pendaftaran

Identifikasi	
Nomor	1
Nama	Pendaftaran
Deskripsi	Pengguna melakukan pendaftaran untuk mendapatkan akun dan agar bisa melakukan *login* pada aplikasi.
Aktor	Pengguna
Skenario Utama	
Kondisi Awal	Menampilkan halaman *Login*
Aksi Aktor	**Reaksi Sistem**
1. Membuka/Memulai Aplikasi	
	2. Menuju halaman pendaftaran
3. Menampilkan halaman pendaftaran	
4. Mengisi identitas seperti Nama, Email, Kata Sandi, dan Pengulangan Kata Sandi.	5. Sistem melakukan validasi data yang telah di inputkan.
	6. Sistem memgirimkan email verifikasi akun kepada email yang digunakan oleh pengguna.
7. Pengguna mengecheck email dan melakukan verifikasi	8. Sistem memvalidasi akun yang sudah terverifikasi.
	9. Menampilkan halaman sudah masuk ke halaman utama.
Skenario Alternatif	
Aksi Aktor	**Reaksi Sistem**
1. Kata sandi minimal 8 karakter	2. Menampilkan pesan/alert bahwa kata sandi kurang dari 8 karakter
	3. Menampilkan *Form* Pendaftaran
Kondisi Akhir	Sistem menampilkan halaman utama

2. *Use Case Scenario* Login

Scenario *use case* Login akan menjelaskan bagaimana alur proses pada diagram use case. Berikut adalah tabel penjabaran dari *use case* Login.

Tabel 3.10 *Use Case* Scenario Login

Identifikasi	
Nomor	2
Nama	Login
Deskripsi	Pengguna melakukan proses login untuk masuk ke dalam aplikasi
Aktor	Pengguna
Skenario Utama	
Kondisi Awal	Halaman awal Login
Aksi Aktor	**Reaksi Sistem**

1. Membuka/Memulai Aplikasi	
	2. Menampilkan halaman login
3. Memasukkan alamat email dan kata sandi jika sudah terdaftar	
4. Menekan tombol masuk	5. Memvalidasi alamat email dan kata sandi yang sudah di masukkan
	6. Menampilkan halaman utama
Skenario Alternatif	
Aksi Aktor	**Reaksi Sistem**
1. Memasukkan kombinasi alamat email dan password yang salah	2. Mengeluarkan peringatan/pesan bahwa pengguna gagal login .
	3. Menampilkan form login
Kondisi Akhir	Sistem menampilkan halaman utama

3. *Use Case Scenario* Lupa Password

Skenario *use case* Lupa Password akan menjelaskan bagaimana alur proses pada diagram use case. Berikut adalah tabel penjabaran dari *use case* Lupa Password.

Tabel 3.11 *Use Case* Scenario Lupa Password

Identifikasi	
Nomor	3
Nama	Lupa Password
Deskripsi	Pengguna melakukan verifikasi lupa password
Aktor	Pengguna
Skenario Utama	
Kondisi Awal	Aplikasi menampilkan halaman log in
Aksi Aktor	**Reaksi Sistem**
1. Membuka/ Memulai Aplikasi	
2. Menuju ke halaman lupa password	3. Membuka halaman lupa password
4. Mengisi password baru	5. Memvalidasi perintah
	6. Password berhasil diganti
Kondisi Akhir	Sistem menampilkan bahwa pergantian password yang baru, sudah berhasil

4. *Use Case Scenario* Mengubah *Profile* Pengguna

Skenario *use case* Mengubah *Profile* Pengguna akan menjelaskan bagaimana alur proses pada diagram use case. Berikut adalah tabel penjabaran dari *use case* Mengubah *Profile* Pengguna.

Tabel 3.12 *Use Case* Scenario Mengubah *Profile* Pengguna

Identifikasi	
Nomor	4
Nama	Mengubah Profile Pengguna
Deskripsi	Pengguna melakukan proses login untuk Mengubah Profile mereka.
Aktor	Pengguna
Skenario Utama	
Kondisi Awal	Halaman utama aplikasi setelah berhasil Login
Aksi Aktor	**Reaksi Sistem**
1. Membuka halaman edit info akun	
	2. Menampilkan halaman info akun
3. Melakukan perubahan pada profil	
4. Menekan tombol Ubah Profil	5. Menyimpan profile baru
	6. Profile dirubah
Kondisi Akhir	Profile berhasil diperbaharui

5. *Use Case Scenario* Mencari Tempat Berdasarkan Jalur Yang Diinginkan

Skenario *use case* Mencari Tempat Berdasarkan Jalur Yang Diinginkan akan menjelaskan bagaimana alur proses pada diagram use case. Berikut adalah tabel penjabaran dari *use case* Mencari Tempat Berdasarkan Jalur Yang Diinginkan

Tabel 3.13 *Use Case* Scenario Mencari Tempat Berdasarkan Jalur Yang Diinginkan

Identifikasi	
Nomor	5
Nama	Mencari Tempat Berdasarkan Jalur Yang Diinginkan
Deskripsi	Pengguna melakukan pencarian tempat makan berdasarkan jalur yang diinginkan
Aktor	Pengguna
Skenario Utama	
Kondisi Awal	Halaman utama setelah berhasil *Login*
Aksi Aktor	**Reaksi Sistem**
1. Menekan menu "Cari Berdasarkan Jalur Yang Diinginkan"	
2. Memasukkan lokasi yang akan dituju.	3. Menampilkan nama dan lokasi tempat.
4. Menekan nama tempat	5. Menunujukkan jalan ke lokasi
Skenario Alternatif	
Aksi Aktor	**Reaksi Sistem**
1. Menekan menu "Cari Berdasarkan Jalur Yang Diinginkan"	

2. Memasukkan lokasi yang akan dituju dan juga nama makanan.	3. Menampilkan peta dan tujuan yang dituju, juga nama tempat.
4. Menekan nama tempat.	
	5. Menunjukkan jalur ke lokasi
Kondisi Akhir	Menampilkan nama dan lokasi tempat sepanjang jalur yang diinginkan.

6. *Use Case Scenario* Mencari Tempat Berdasarkan Nama Makanan

Skenario *use case* Mencari Tempat Berdasarkan Nama Makanan akan menjelaskan bagaimana alur proses pada diagram use case. Berikut adalah tabel penjabaran dari *use case* Mencari Tempat Berdasarkan Nama Makanan.

Tabel 3.14 *Use Case* Scenario Mencari Tempat Berdasarkan Nama Makanan

Identifikasi	
Nomor	6
Nama	Mencari Tempat Berdasarkan Nama Makanan.
Deskripsi	Pengguna melakukan pencarian tempat makan berdasarkan nama makanan yang di inginkan.
Aktor	Pengguna
Skenario Utama	
Kondisi Awal	Halaman awal setelah berhasil Login.
Aksi Aktor	**Reaksi Sistem**
1. Menekan menu "Cari Berdasarkan Makanan"	
2. Memasukkan nama makanan yang di inginkan	
	3. Menampilkan nama tempat
4. Menekan nama tempat	
	5. Menunjukkan lokasi nama tempat makan
Skenario Alternatif	
Aksi Aktor	**Reaksi Sistem**
1. Menekan menu "Cari Berdasarkan Makanan"	
2. Memilih hashtag	3. Menampilkan tweet sesuai dengan hashtag yang dipilih
Kondisi Akhir	Menampilkan tweet sesuai dengan hashtag

7. *Use Case Scenario* Melihat Informasi Trending

Skenario *use case* Melihat Informasi Trending akan menjelaskan bagaimana alur proses pada diagram *use case* Melihat Informasi Trending. Berikut adalah tabel penjabaran dari *use case* Melihat Informasi Trending.

Tabel 3.15 *Use Case* Scenario Melihat Informasi Trending

Identifikasi	
Nomor	7
Nama	Melihat Informasi Trending
Deskripsi	Pengguna melihat informasi makanan asia apa yang sedang trending di media sosial khususnya *Instagram*
Aktor	Pengguna
Skenario Utama	
Kondisi Awal	Halaman utama setelah berhasil *Login*
Aksi Aktor	**Reaksi Sistem**
1. Menggeser aplikasi di halaman utama sampai pada menu trending	
2. Memilih hashtag apa yang akan dicari	
	3. Menampilkan informasi makanan asia yang sedang trending di Instagram sesuai dengan hashtag yang sudah dimasukkan
Kondisi Akhir	Menampilkan informasi makanan asia yang sedang trending di Instagram sesuai dengan hashtag yang dipilih.

8. *Use Case Scenario* Edit Data

Skenario *use case* Edit Data akan menjelaskan bagaimana alur proses pada diagram *use case* Edit Data. Berikut adalah tabel penjabaran dari *use case* Edit Data.

Tabel 3.16 *Use Case* Scenario Edit Data

Identifikasi	
Nomor	8
Nama	Edit Data
Deskripsi	Admin melakukan pengeditan/perubahan terhadap data di dalam database. Ini akan dilakukan bila adanya data yang kurang atau adanya data yang harus diperbaiki.
Aktor	*Admin*
Skenario Utama	
Kondisi Awal	Halaman awal
Aksi Aktor	**Reaksi Sistem**
1. Admin Melakukan Login	
	2. Menampikan halaman utama
3. Menuju ke halaman data	
4. Mengedit data yang diinginkan	5. Merubah data
	6. Menyimpan data yang terbaru

Kondisi Akhir	Menampilkan data baru yang sudah di edit oleh admin

9. *Use Case Scenario* Menambah Data

Skenario *use case* Menambah Data akan menjelaskan bagaimana alur proses pada diagram *use case* Menambah Data. Berikut adalah tabel penjabaran dari *use case* Menambah Data.

Tabel 3.17 *Use Case* Scenario Menambah Data

Identifikasi	
Nomor	9
Nama	Menambah Data
Deskripsi	Admin akan menambah data ke database bila adanya sebuah data baru yang harus ditambahkan pada database
Aktor	*Admin*
Skenario Utama	
Kondisi Awal	Halaman awal
Aksi Aktor	**Reaksi Sistem**
1. Admin melakukan login	
	2. Menampilkan halaman utama
3. Menuju ke halaman data	
4. Menambah data yang dibutuhkan	5. Data ditambahkan
	6. Menyimpan data yang ada
Kondisi Akhir	Menampilkan data yang terbaru setelah dilakukan penambahan data yang baru

10. *Use Case Scenario* Menghapus Data

Skenario *use case* Menghapus Data akan menjelaskan bagaimana alur proses pada diagram *use case* Menghapus Data. Berikut adalah tabel penjabaran dari *use case* Menghapus Data.

Tabel 3.18 *Use Case* Scenario Menghapus Data

Identifikasi	
Nomor	10
Nama	Menghapus Data
Deskripsi	Admin akan menghapus data tidak penting yang berada pada database sistem
Aktor	*Admin*
Skenario Utama	
Kondisi Awal	Halaman awal
Aksi Aktor	**Reaksi Sistem**
1. Admin melakukan login	

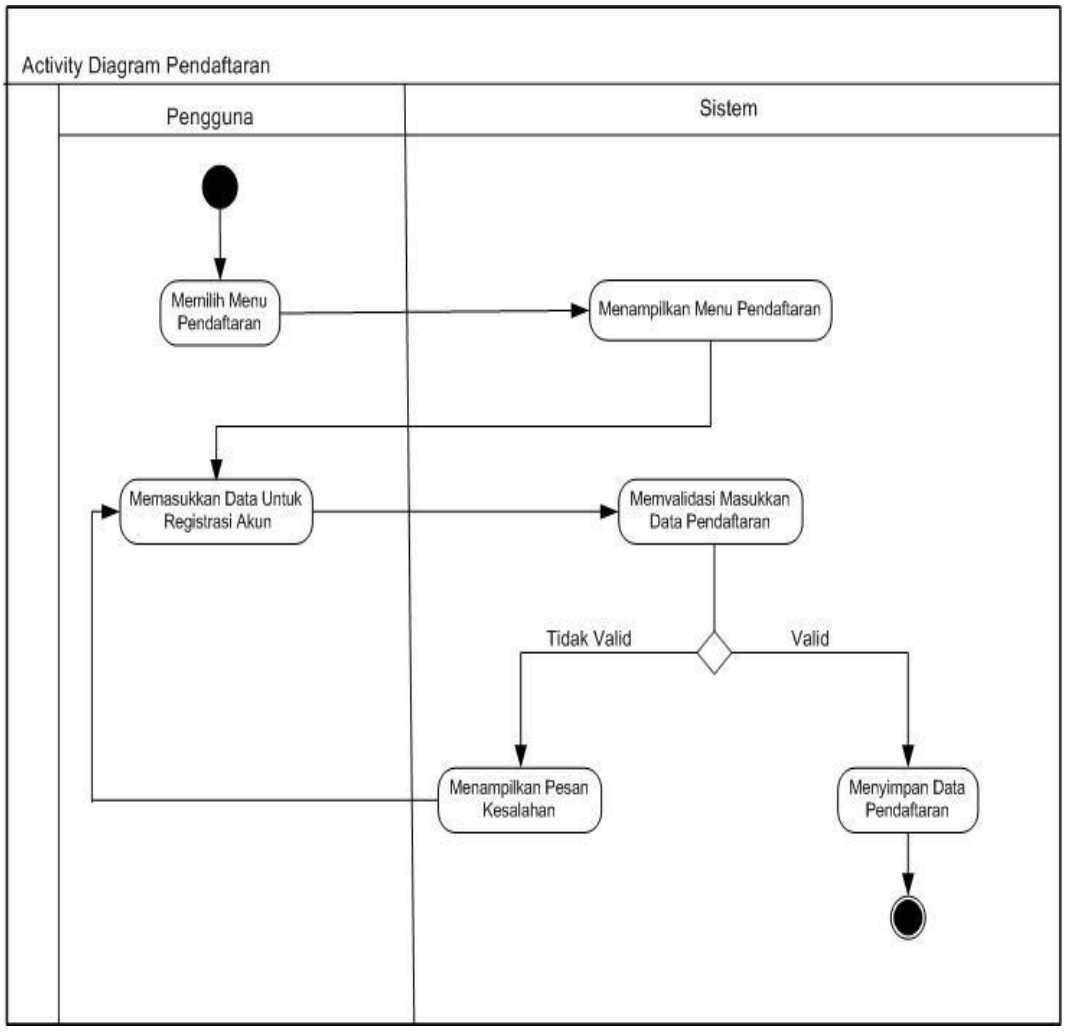
	2. Menampilkan halaman utama
3. Menuju ke halaman data	
4. Menghapus data yang sudah tidak diperlukan lagi	5. Data dihapus
	6. Menyimpan data yang ada
Kondisi Akhir	Menampilkan data terbaru setelah dilakukannya penghapusan data

3.1.6.1.2 *Activity Diagram*

Activity Diagram adalah sebuah diagram yang menggambarkan proses bisnis dan juga urutan aktifitas dari sebuah proses, digunakan pada saat business modeling untuk memperlihatkan urutan dari aktifitas proses bisnis. *Activity diagram* juga dapat menjadi alat untuk memahami proses dari sistem secara keseluruhan. Dibuatnya *activity diagram* akan sesuai berdasarkan *use case* pada *use case* diagram. Berikut adalah penjelasan dari masing-masing activity diagram.

1. *Activity Diagram* Pendaftaran

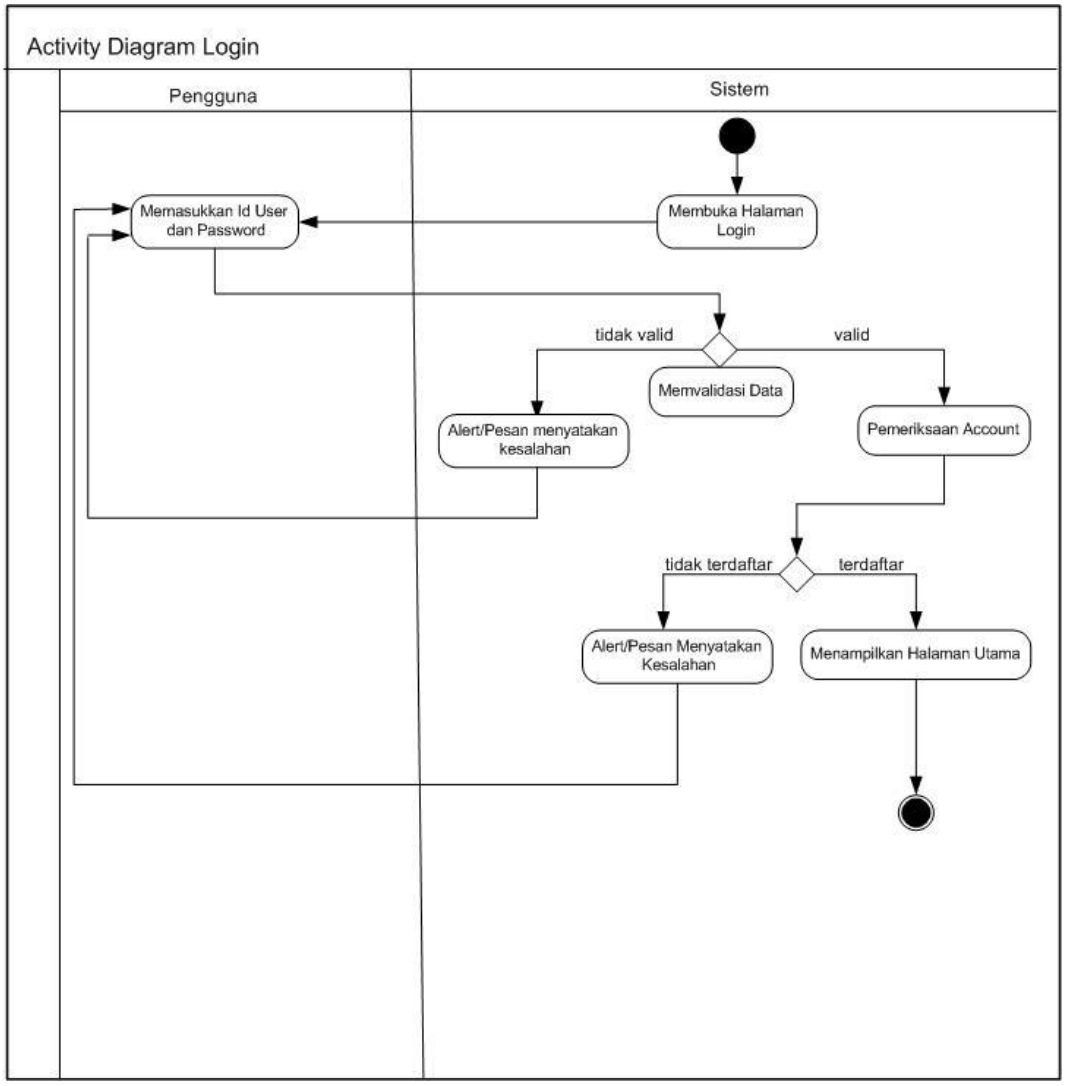
Berikut *activity diagram* dari proses pendaftaran pada aplikasi yang akan dibangun.



Gambar 3.12 *Activity* Diagram Pendaftaran

2. *Activity Diagram Login*

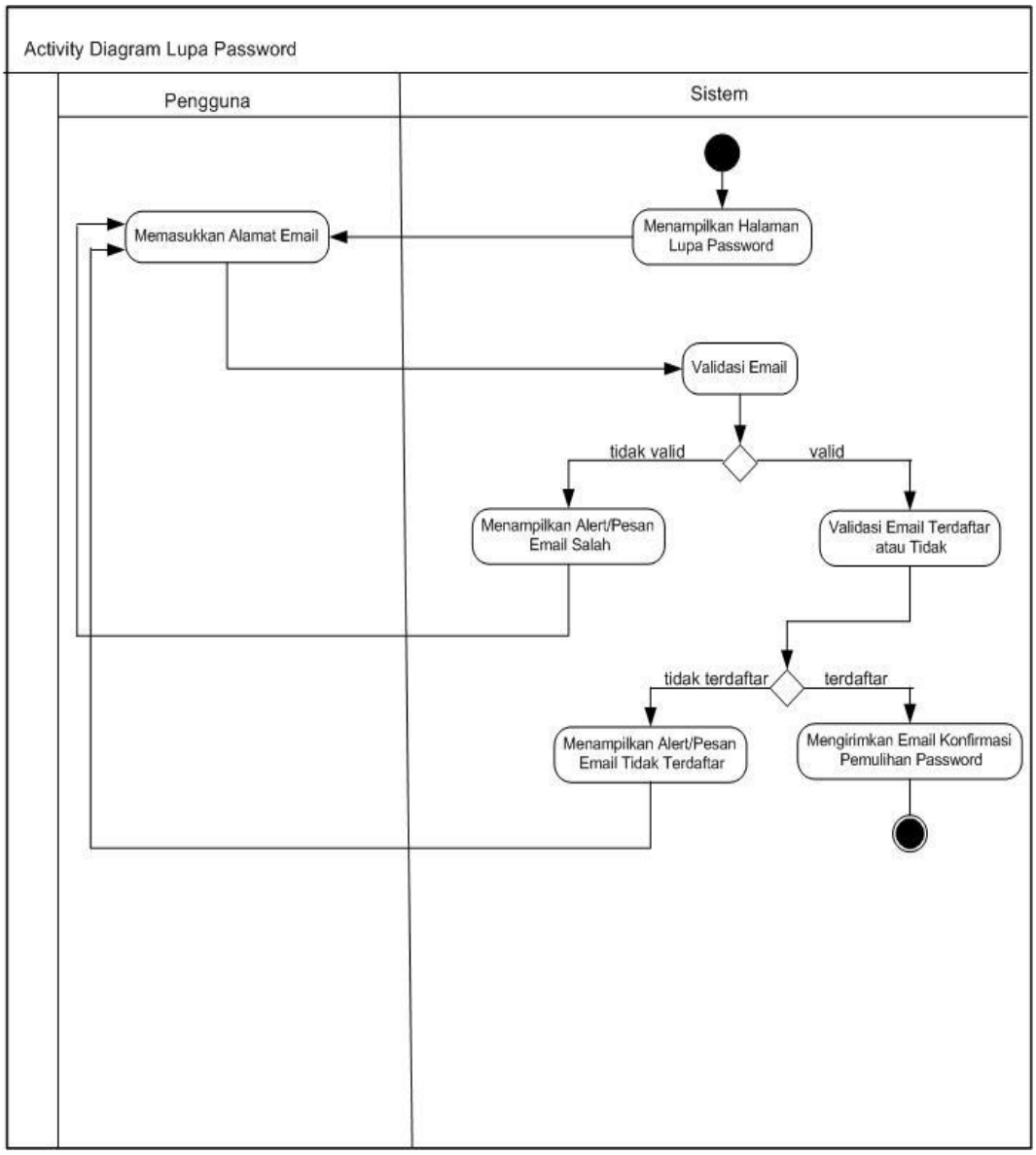
Berikut *activity diagram* dari proses *login* pada aplikasi yang akan dibangun.



Gambar 3.13 *Activity* Diagram Login

3. *Activity Diagram* Lupa *Password*

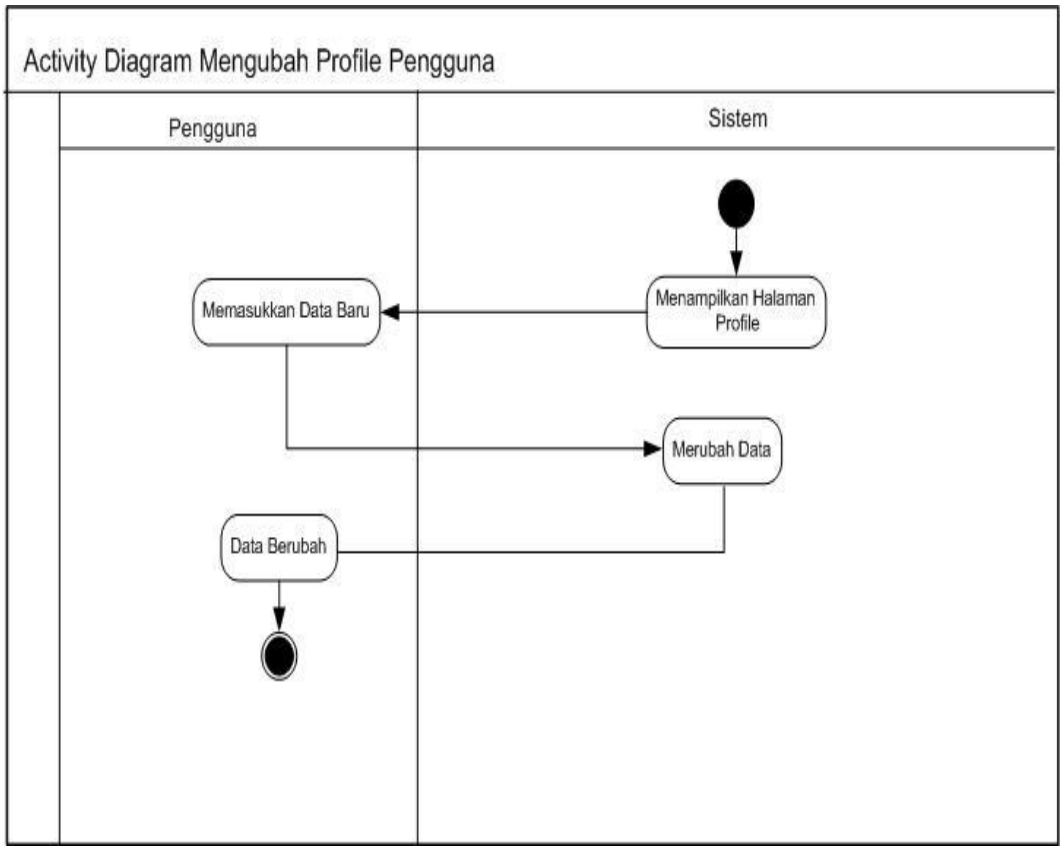
Berikut *activity diagram* dari proses lupa *password* pada aplikasi yang akan dibangun.



Gambar 3.14 *Activity* Diagram Lupa Password

4. *Activity Diagram* Mengubah *Profile* Pengguna

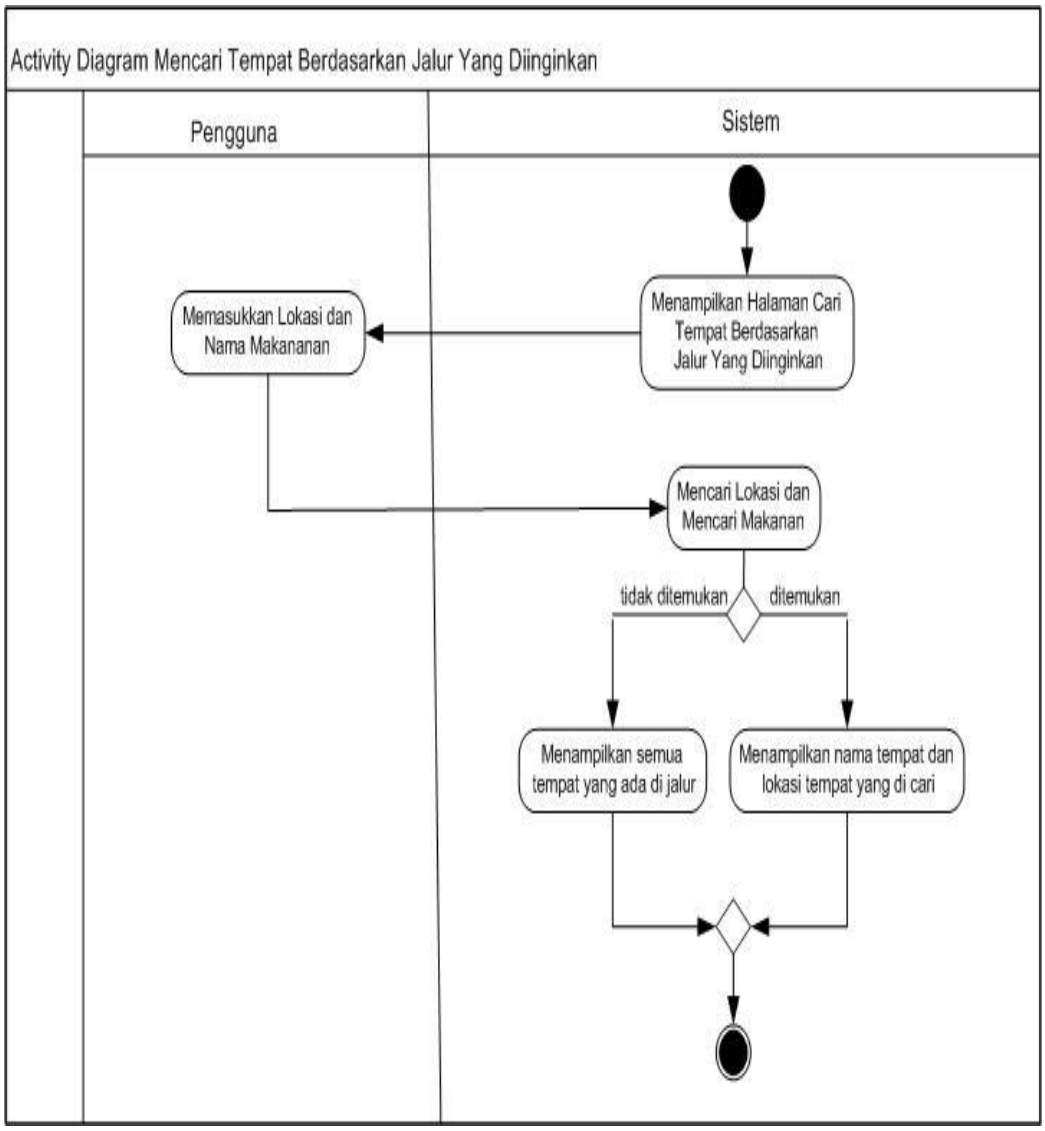
Berikut *activity diagram* dari proses mengubah *profile* pengguna pada aplikasi yang akan dibangun.



Gambar 3.15 *Activity* Diagram Mengubah Profile Pengguna

5. Activity Diagram Mencari Tempat Berdasarkan Jalur Yang Diinginkan

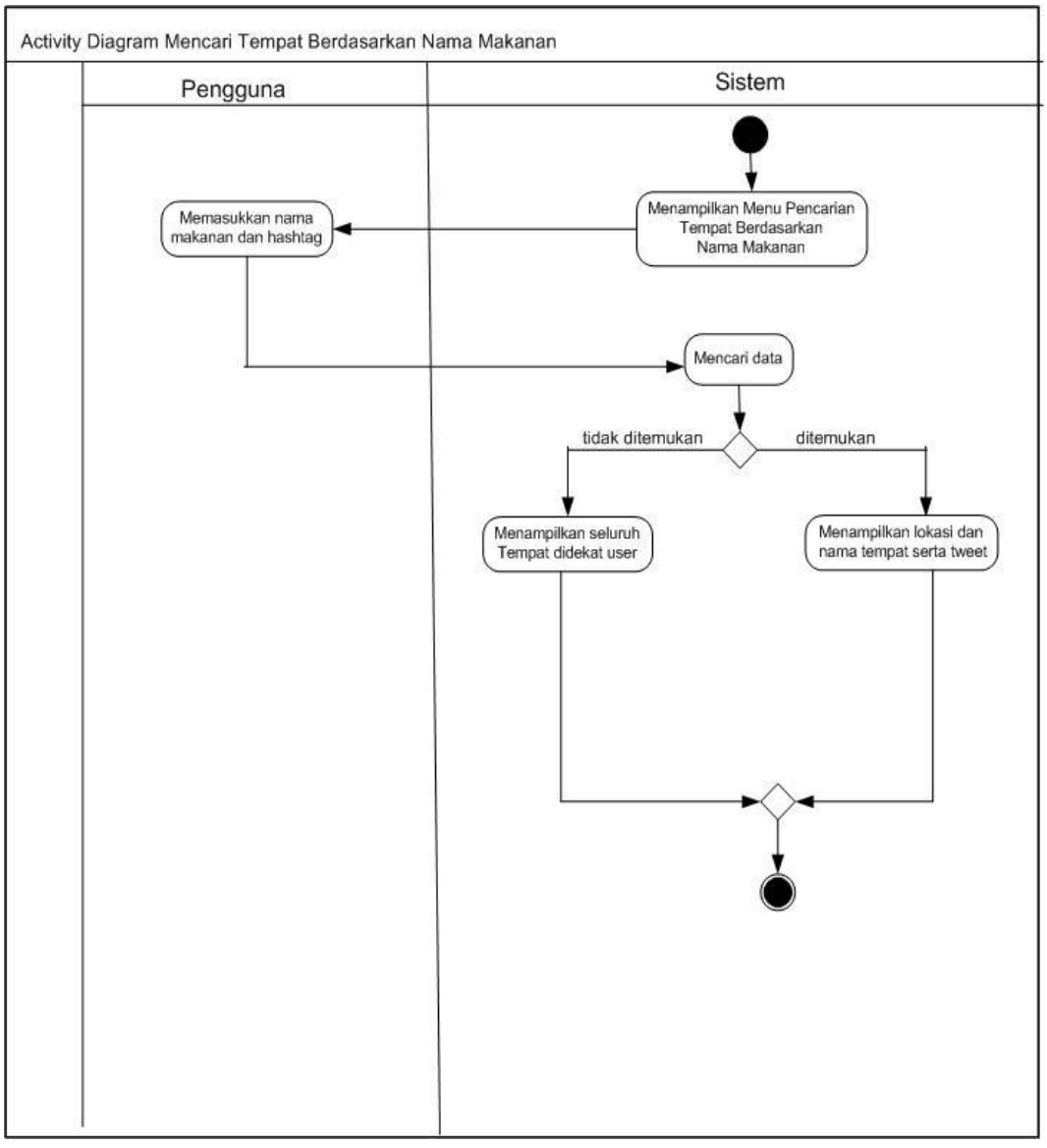
Berikut *activity diagram* dari proses mencari tempat berdasarkan jalur yang diinginkan pada aplikasi yang akan dibangun.



Gambar 3.16 *Activity* Diagram Tempat Berdasarkan Jalur Yang Diinginkan

6. *Activity* Diagram Mencari Tempat Berdasarkan Nama Makanan

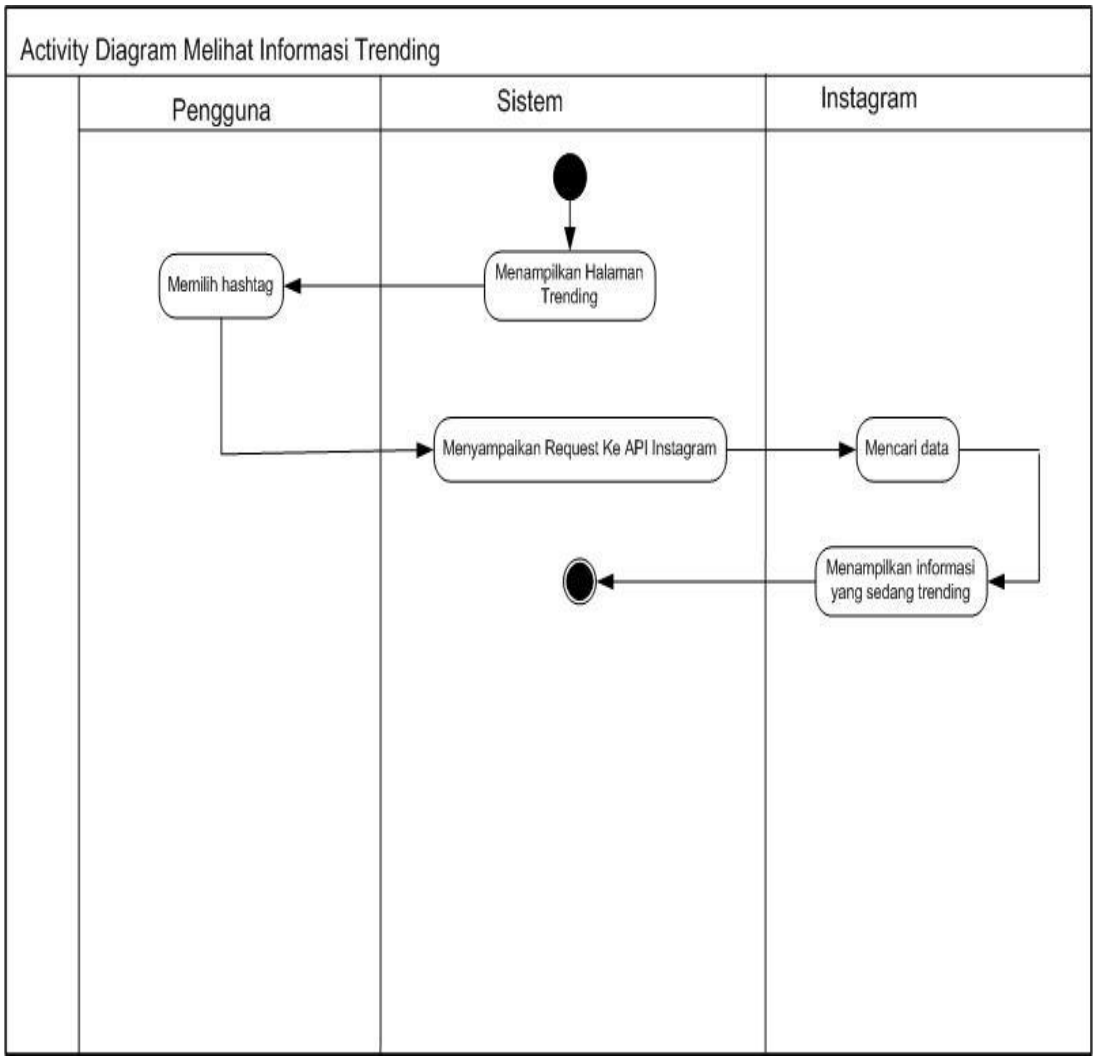
Berikut *activity diagram* dari proses mencari tempat berdasarkan nama makanan pada aplikasi yang akan dibangun ini.



Gambar 3.17 *Activity* Diagram Tempat Berdasarkan Nama Makanan

7. *Activity* Diagram Melihat Informasi Trending

Berikut *activity diagram* dari proses melihat informasi trending pada aplikasi yang akan dibangun.



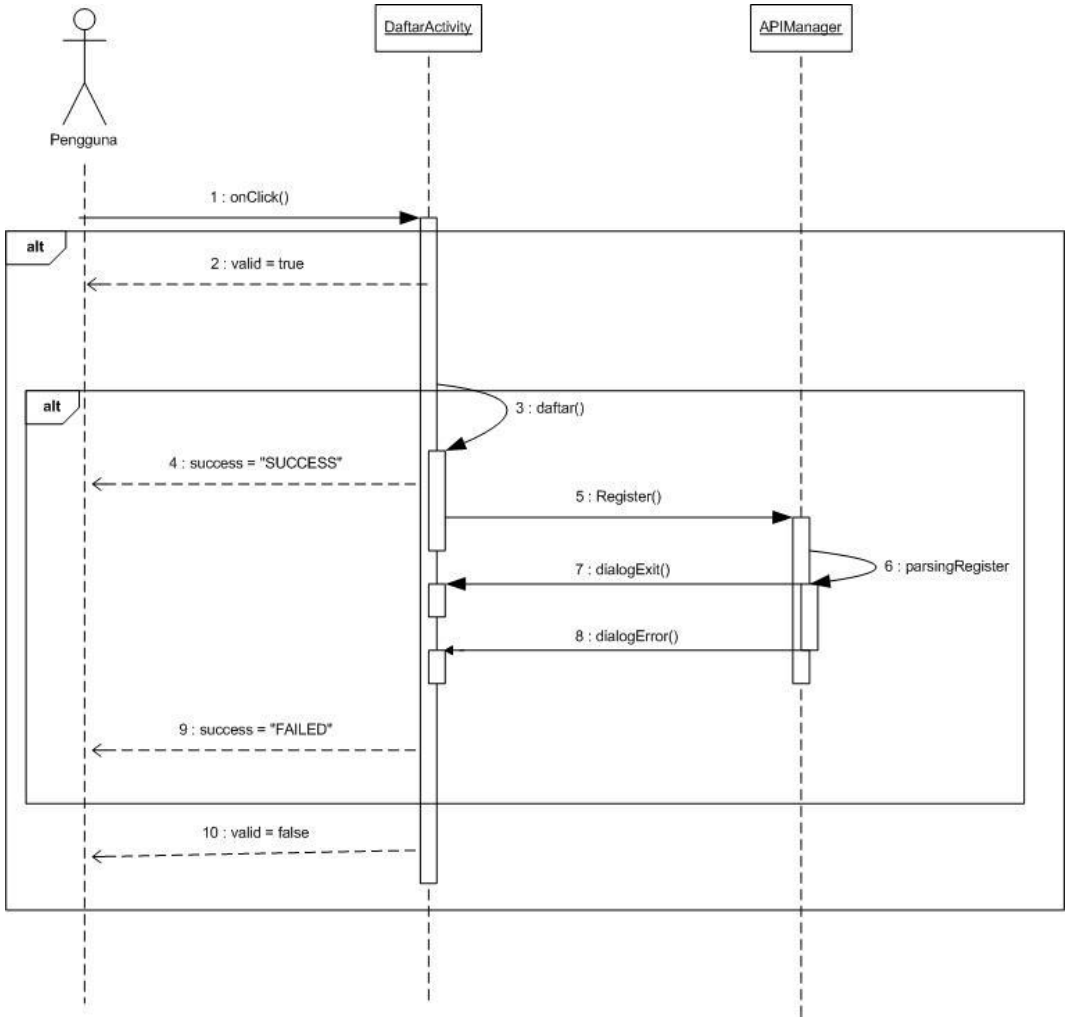
Gambar 3.18 *Activity* Diagram Melihat Informasi Trending

3.1.6.1.3 *Sequence Diagram*

Sequence Diagram adalah merupakan suatu gambaran interaksi yang menjelaskan bagaimana sebuah operasi yang dilakukan didalam aplikasi yang akan dibangun ini. Berikut ini merupakan sequence untuk aplikasi yang akan dibangun :

1. *Sequence* Diagram Pendaftaran

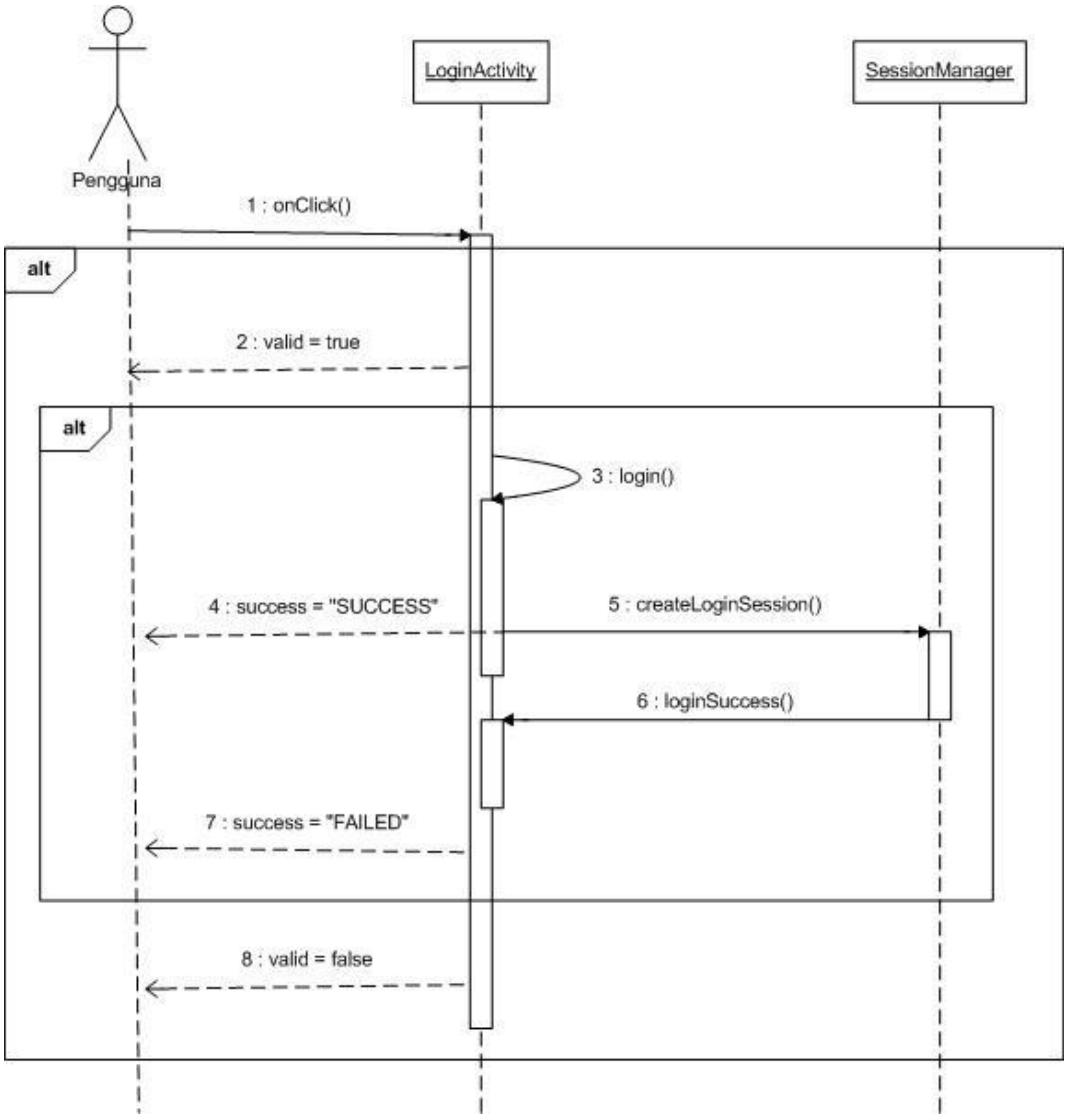
Berikut adalah gambaran interaksi antara actor dengan objek-objek yang ada pada saat proses pendaftaran. Sequence Diagram Perdaftara dapat dilihat pada gambar dibawah ini.



Gambar 3.19 *Sequence* Diagram Pendaftaran

2. *Sequence* Diagram Login

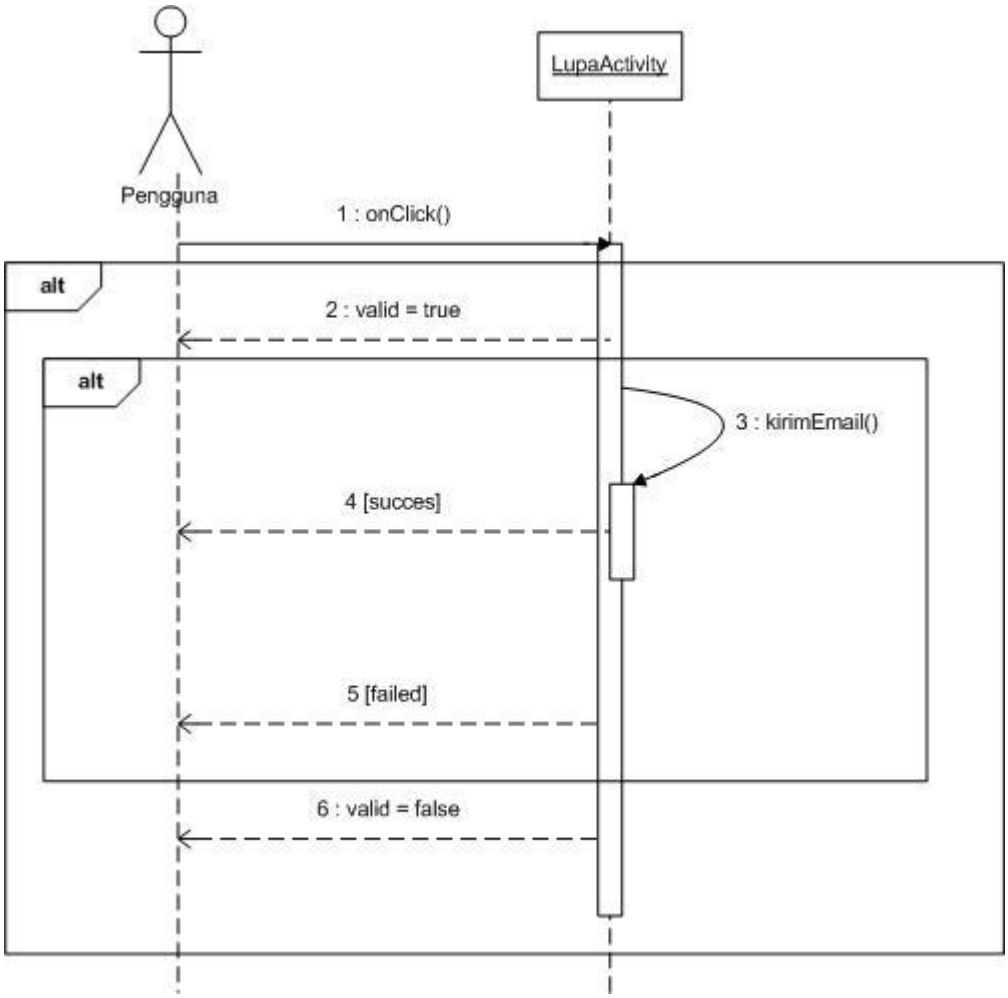
Berikut adalah gambaran interaksi antara actor dengan objek-objek yang ada pada saat proses login. *Sequence* Diagram Login dapat dilihat pada gambar dibawah ini.



Gambar 3.20 *Sequence* Diagram *Login*

3. *Sequence* Diagram Lupa Password

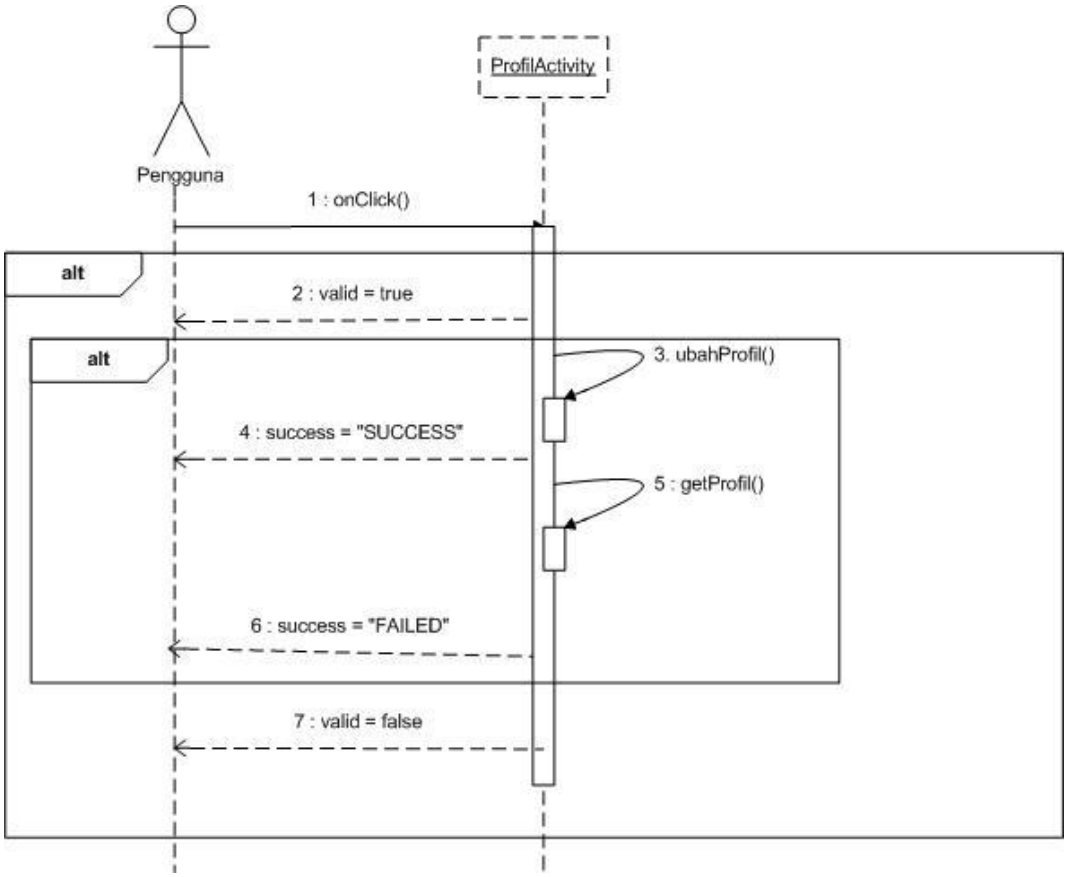
Berikut adalah gambaran interaksi antara actor dengan objek-objek yang ada pada saat proses lupa password. *Sequence* Diagram Lupa Password dapat dilihat pada gambar dibawah ini.



Gambar 3.21 *Sequence* Diagram Lupa Password

4. *Sequence* Diagram Mengubah *Profile* Pengguna

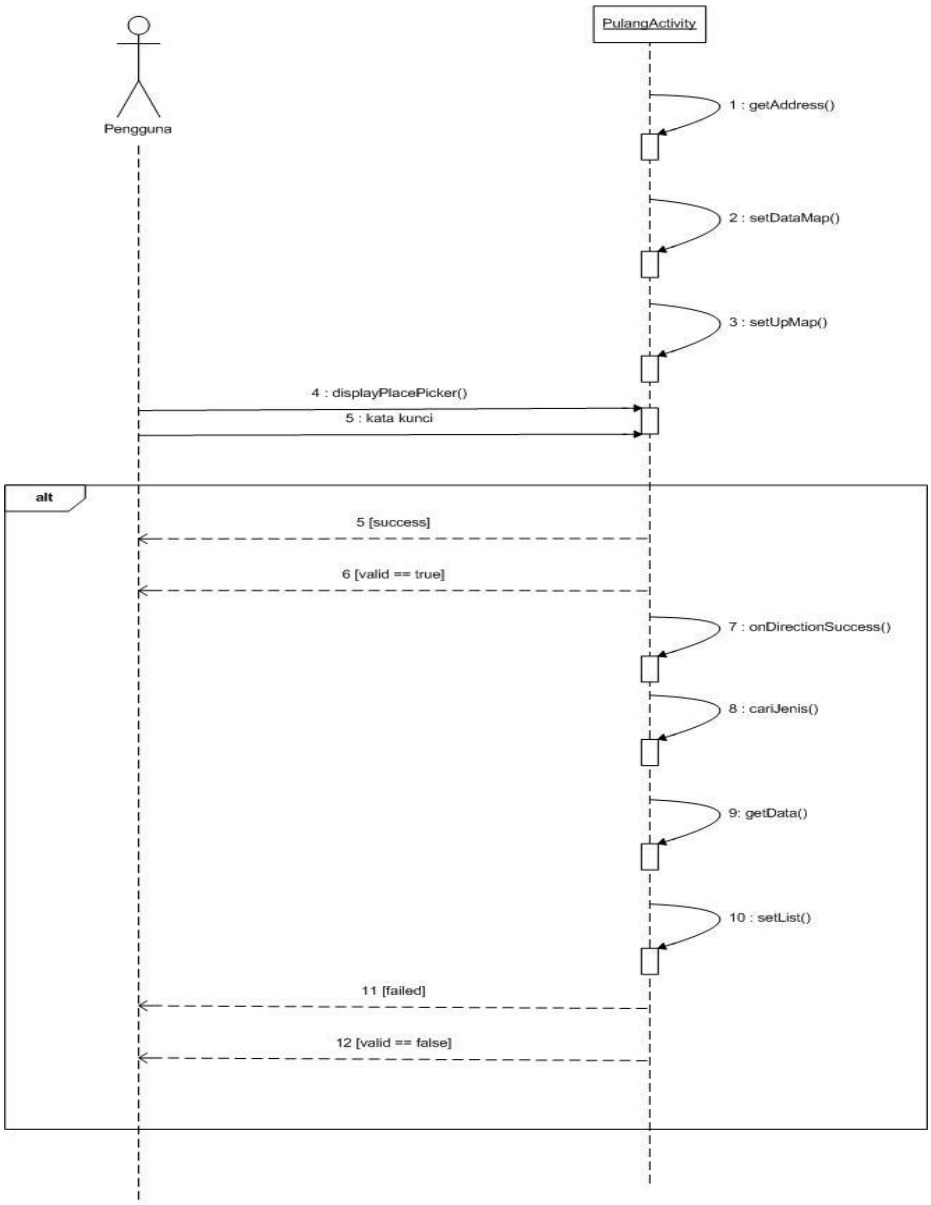
Berikut adalah gambaran interaksi antara actor dengan objek-objek yang ada pada saat proses mengubah profile. *Sequence* Diagram Mengubah Profile Pengguna dapat dilihat pada gambar dibawah ini.



Gambar 3.22 Sequence Diagram Mngubah Profile Pengguna

5. *Sequence* Diagram Mencari Tempat Berdasarkan Jalur Yang Diinginkan

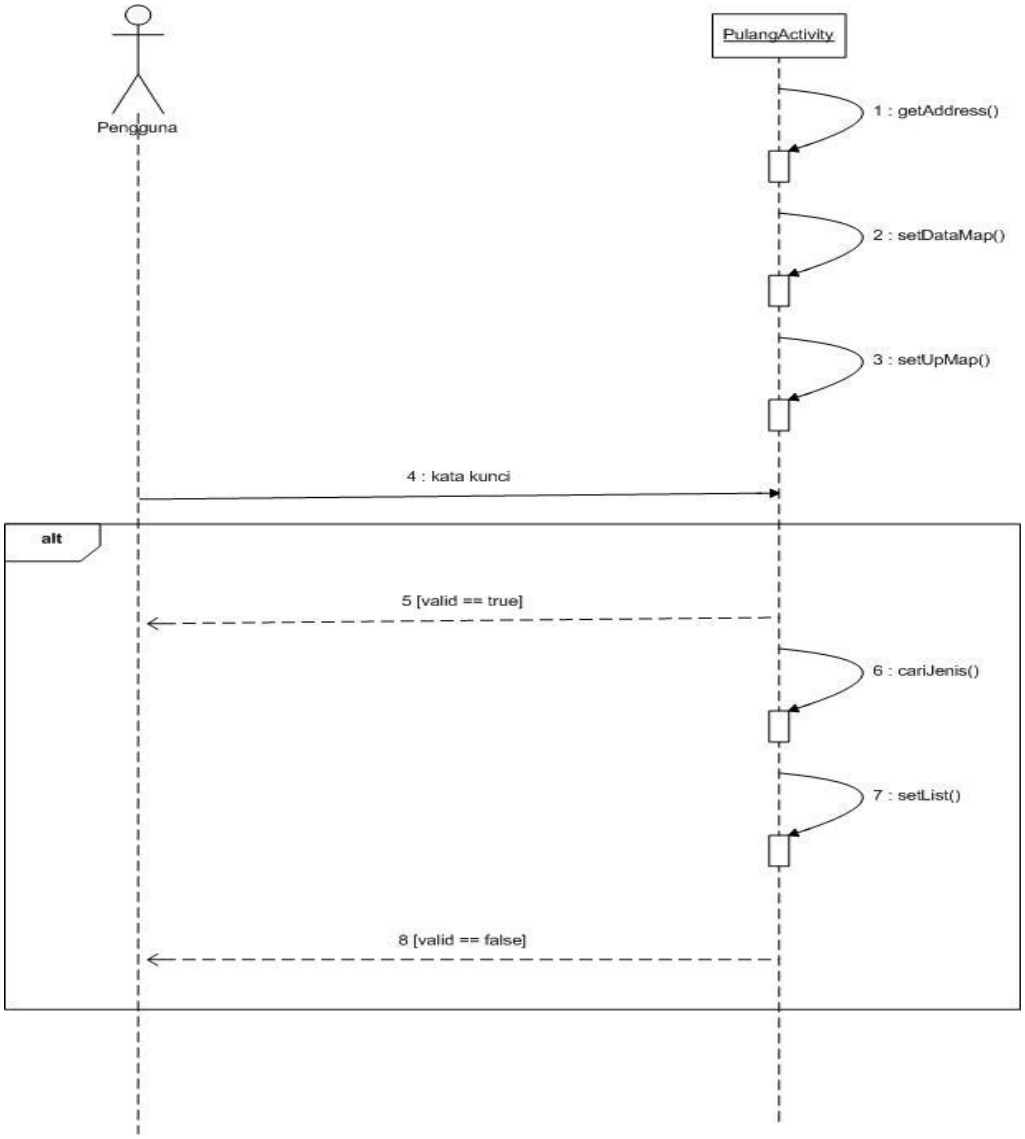
Berikut adalah gambaran interaksi antara actor dengan objek-objek yang ada pada saat proses mencari tempat berdasarkan jalur yang diinginkan. *Sequence* Diagram Mencari Tempat Berdasarkan Jalur Yang Diinginkan dapat dilihat pada gambar dibawah ini.



Gambar 3.23 *Sequence* Diagram Mencari Tempat Berdasarkan Jalur Yang Diinginkan

6. *Sequence* Diagram Mencari Tempat Berdasarkan Nama Makanan

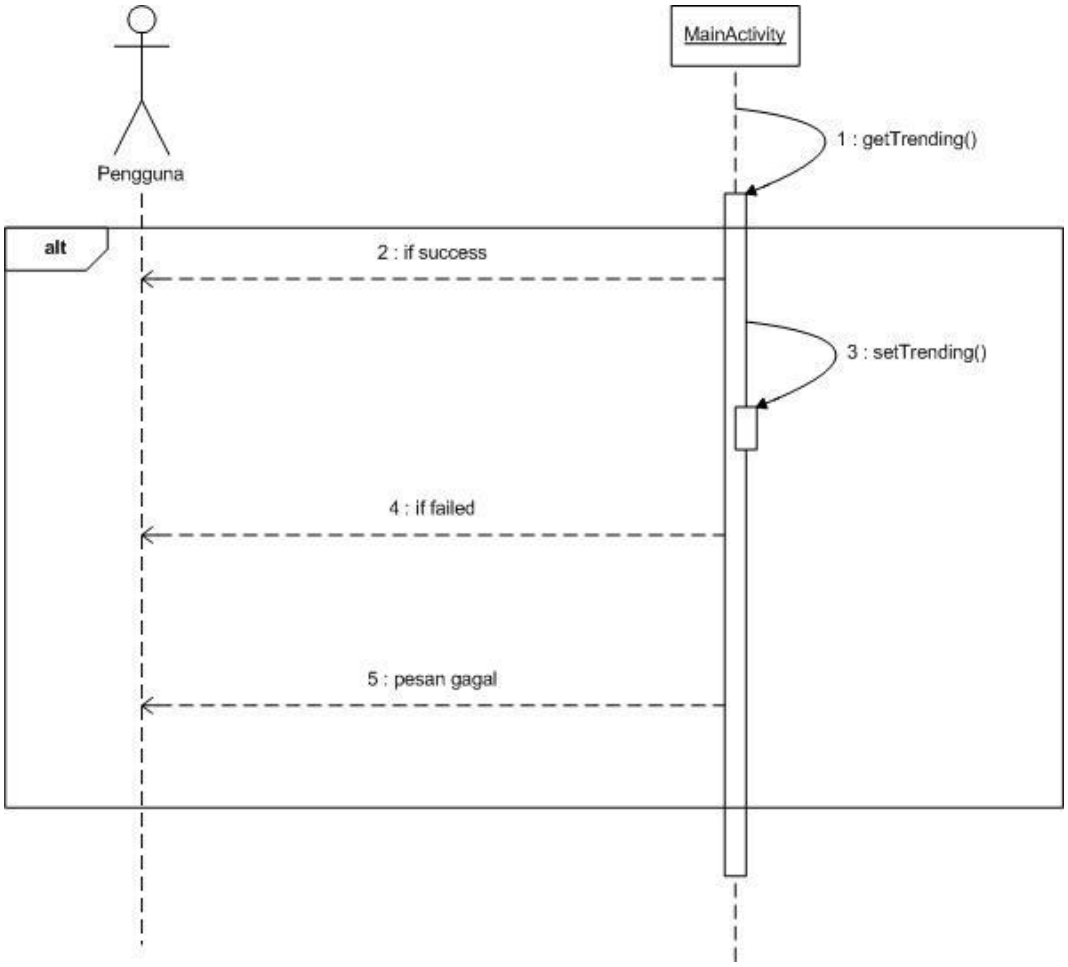
Berikut adalah gambaran interaksi antara actor dengan objek-objek yang ada pada saat proses mencari tempat berdasarkan nama makanan. *Sequence* Diagram Mencari Tempat Berdasarkan Nama Makanan dapat dilihat pada gambar dibawah ini.



Gambar 3.24 *Sequence* Diagram Mencari Tempat Berdasarkan Nama Makanan

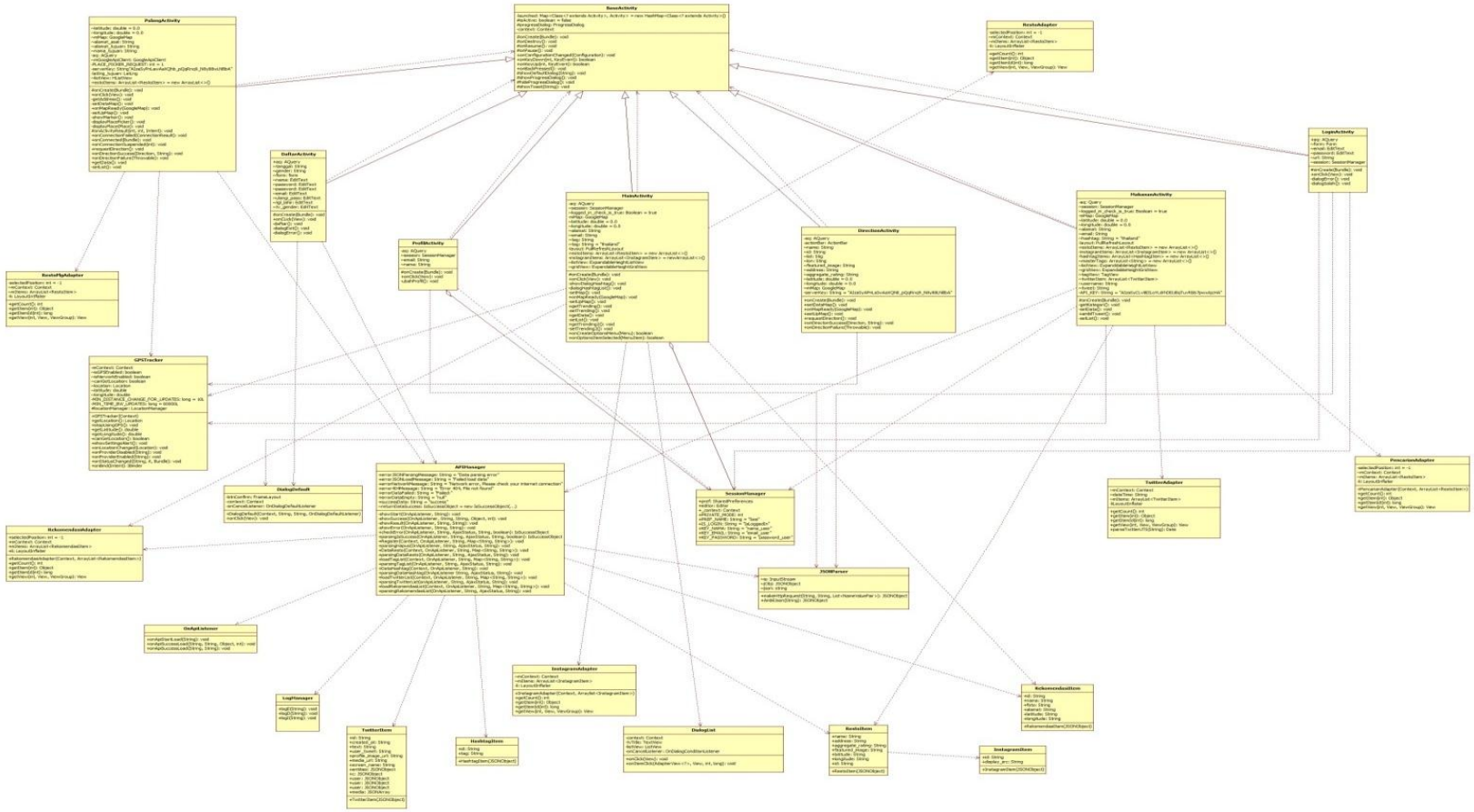
7. *Sequence* Diagram Melihat Informasi Trending

Berikut adalah gambaran interaksi antara actor dengan objek-objek yang ada pada saat proses melihat info trending. *Sequence* Diagram Melihat Informasi Trending dapat dilihat pada gambar dibawah ini.



Gambar 3.25 *Sequence* Diagram Melihat Informasi Trending

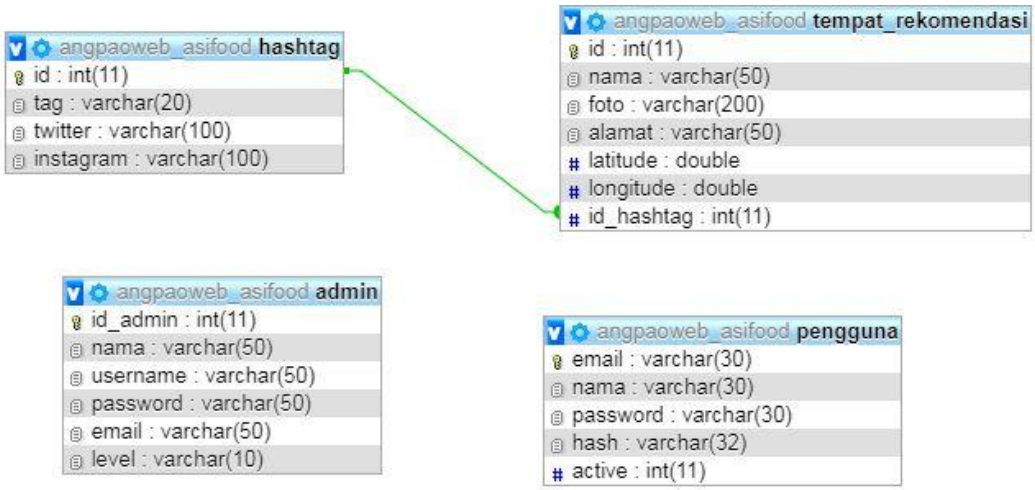
3.1.6.1.4 Diagram Class



Gambar 3. 26 Diagram Class

3.1.6.1.5 Skema Relasi

Skema Relasi merupakan sebuah rangkaian dua buah tabel yang saling berhubungan pada sistem basis data. Skema relasi untuk pembangunan aplikasi pencarian tempat makan kuliner asia dapat dilihat pada Gambar 3.26 Skema Relasi.



Gambar 3.27 Skema Relasi

3.1.6.1.6 Struktur Tabel

Struktur tabel merupakan gambaran detail yang berisi nama column, type, null, default. Tabel yang ada pada sistem basis data pembangunan sistem ini adalah pengguna, favorit, tempat_rekomendasi, dan hashtag. Berikut adalah penjelasan struktur tabel dari masing-masing tabel yang telah disebutkan.

1. Tabel pengguna

Berikut adalah Tabel 3.19 Struktur Tabel pengguna yang digunakan pada sistem basis data perangkat lunak.

Tabel 3.19 Struktur Tabel pengguna

Column	Type	Length	Null
email (*Primary*)	varchar	30	No
nama	varchar	30	No
password	varchar	30	No
hash	varchar	30	No
active	varchar	30	No

2. Tabel tempat_rekomendasi

Berikut adalah Tabel 3.20 Struktur Tabel tempat_rekomendasi yang digunakan pada sistem basis data perangkat lunak.

Tabel 3.20 Struktur Tabel tempat_rekomendasi

Column	Type	Length	Null
id (*Primary*)	int	11	No
nama	varchar	50	No
foto	varchar	200	No
alamat	varchar	50	No
latitude	double	-	No
longitude	double	-	No
id_hashtag	int	11	No

3. Tabel hashtag

Berikut adalah Tabel 3.21 Struktur Tabel hashtag yang digunakan pada sistem basis data perangkat lunak.

Tabel 3.21 Struktur Tabel Hashtag

Column	Type	Length	Null
id (*Primary*)	int	11	No
tag	varchar	20	No

4. Tabel admin

Berikut adalah Tabel 3. Struktur Tabel admin yang digunakan pada sistem basis data perangkat lunak.

Tabel 3.22 Struktur Tabel admin

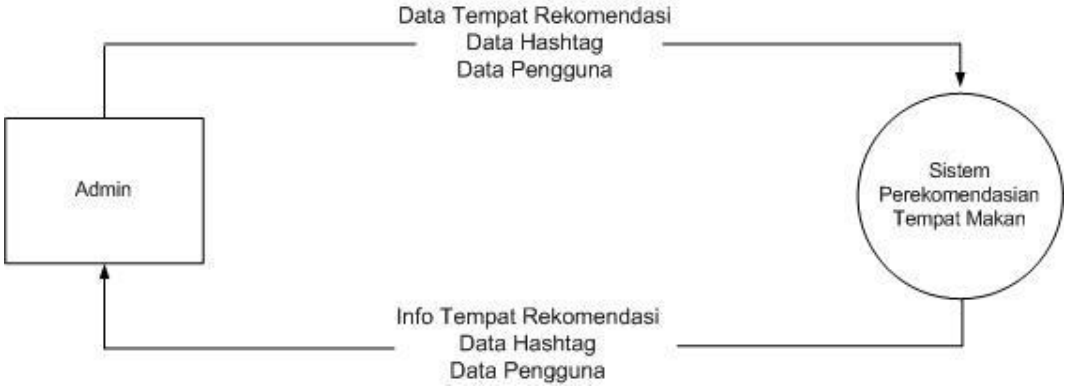
Column	Type	Length	Null
id_admin(*Primary*)	int	11	No
nama	varchar	20	No
username	varchar	50	No
password	varchar	50	No
email	varchar	50	No
level	varchar	10	No

3.1.6.2 Analisis Kebutuhan Fungsional Pada Paltform Web

Analisis yang akan dilakukan pada platform web adalah berupa analisis secara terstruktur. Berikut adalah beberapa analisis yang dilakukan :

3.1.6.2.1 Diagram Konteks

Analisis dengan menggunakan diagram konteks akan berupa suatu cara menggambarkan suatu sistem secara garis besar dan keseluruhan. Berikut adalah diagram konteks yang akan ditunjukkan pada Gambar 3.27 Diagram Konteks.



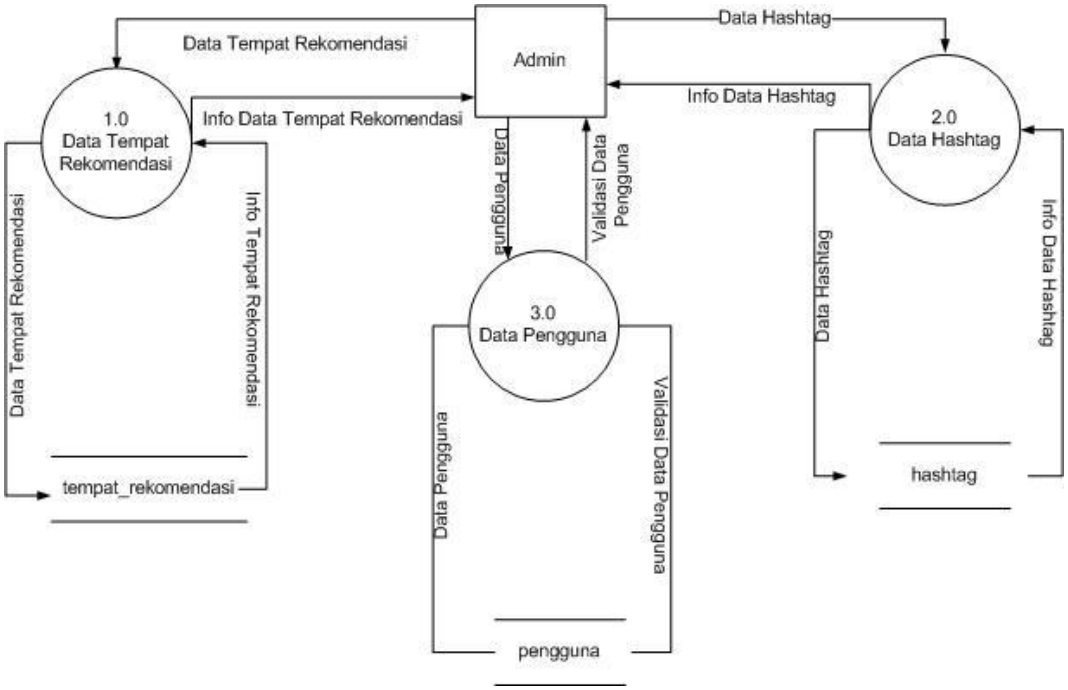
Gambar 3.28 Diagram Konteks

3.1.6.3.2 *Data Flow Diagram* (DFD)

Data Flow Diagram adalah alat yang memungkinkan professional sistem untuk meggambarkan sistem suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data. Berikut adalah Data Flow Diagram pada pembanunan sistem *platform website*

1. DFD Level 1

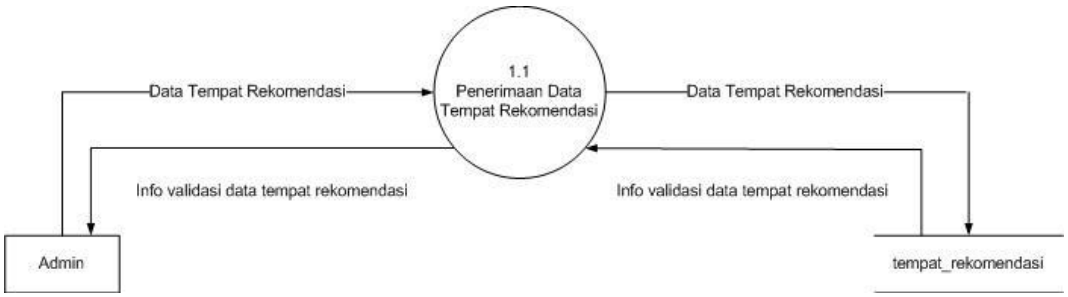
Berikut adalah data flow diagram level 1 yang bisa dilihat pada gambar dibawah ini.



Gambar 3.29 DFD Level 1

2. DFD Level 1.1 Tambah Data Tempat Rekomendasi

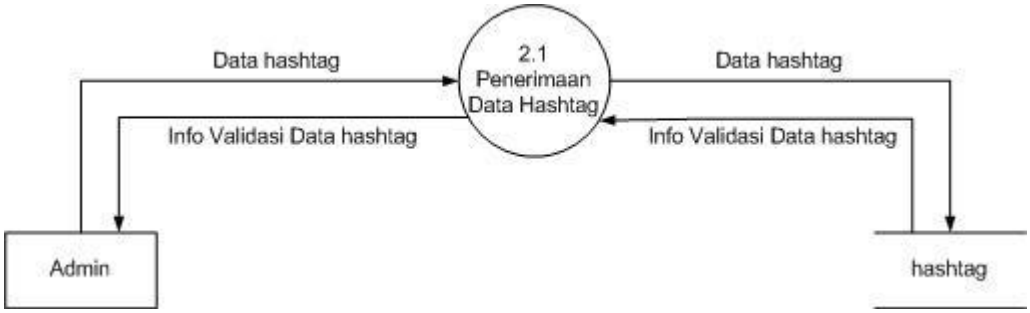
   Berikut adalah data flow diagram level 1.1 yang bisa dilihat pada gambar dibawah ini.



Gambar 3.30 DFD Level 1.1

3. DFD Level 2.1 Tambah Data Hashtag

   Berikut adalah data flow diagram level 2.1 yang bisa dilihat pada gambar dibawah ini.



Gambar 3.31 DFD Level 2.1

3.1.6.2.3 Spesifikasi Proses

Spesifikasi proses berfungsi sebagai gambaran proses bagaimana model aliran yang ada pada masing-masing *Data Flow Diagram* (DFD). Berikut adalah spesifikasi proses pada aplikasi *platform web*.

1. Spesifikasi Proses DFD Level 1

Spesfikasi proses yang terjadi pada DFD level 1 disini diperuntukan sebagai gambaran proses bagaimana model aliran yang ada pada DFD Level 1 berkerja.

Tabel 3.23 Data Flow Diagram Level 1

No	Proses	Keterangan
1	No. Proses	1.0
	Nama Proses	Data Tempat Rekomendasi
	Source	Admin
	Input	Data Tempat Rekomendasi
	Output	Info Tempat Rekomendasi
	Tujuan	Admin
	Logika Proses	1. Admin memasukkan data tempat rekomendasi pada halaman yang sudah ditentukan 2. Sistem melakukan validasi terhadap data yang sudah dimasukkan kepada sistsem
2	No. Proses	2.0
	Nama Proses	Data Hashtag
	Source	Admin
	Input	Data Hashtag
	Output	Informasi Hashtag
	Tujuan	Admin
	Logika Proses	1. Admin memasukkan data hashtag yang berupa nama negara 2. Sistem melakukan validasi terhadap data yang sudah dimasukkan kepada sistem
3	No. Proses	3.0
	Nama Proses	Data Pengguna
	Source	Admin
	Input	Admin melakukan penghapusan terhadap data pengguna bila adanya pengguna yang tidak aktif

	Output	Informasi Bahwa Data Berhasil Dihapus
	Tujuan	Admin
	Logika Proses	1. Admin melakukan penghapusan terhadap data pengguna yang sudah tidak diperlukan atau pengguna yang tidak aktif 2. Sistem memvalidasi data yang sudah diahpus tersebut dengan memberi permberitauan bahwa data telah dihapus dari sistem

2.Spesfikasi Proses DFD Level 1.1

Spesifikasi proses yang terjadi pada DFD level 1.1 disini diperuntukan sebagai gambaran proses bagaimana model aliran yang ada pada DFD Level 1.1 berkerja.

Tabel 3. 24 Data Flow Diagram Level 1.1

No	Proses	Keterangan
1	No. Proses	1.1
	Nama Proses	Penerimaan Data Tempat Rekomendasi
	Source	Admin
	Input	Data “nama”, “foto”, “alamat”, “latitude”, “longitude”
	Output	Data Tempat Rekomendasi
	Tujuan	Admin
	Logika Proses	1. Admin memasukkan beberapa data yang dibutuhkan untuk data tempat rekomendasi 2. Sistem memvalidasi hasil data masukkan tersebut

3. Spesifikasi Proses DFD Level 2.1

Spesifikasi proses yang terjadi pada DFD level 1.1 disini diperuntukan sebagai gambaran proses bagaimana model aliran yang ada pada DFD Level 1.1 berkerja.

Tabel 3. 25 Data Flow Diagram Level 2.1

No	Proses	Keterangan
1	No. Proses	2.1
	Nama Proses	Penerimaan Data Hashtag
	Source	Admin
	Input	Data “tag”
	Output	Data Hashtag
	Tujuan	Admin
	Logika Proses	1. Admin memasukkan data hahstag berupa nama negara yang dibutuhkan 2. Sistem melakukan validasi terhadap data yang telah dimasukkan tersebut

3.1.6.2.4 Kamus Data

Kamus data adalah deskripsi untuk menjelaskan data-data yang ada dalam DFD. Kamus data yang ada didalam sistem yang dibangun bisa dilihat pada tabel dibawah ini.

Tabel 3. 26 Kamus Data

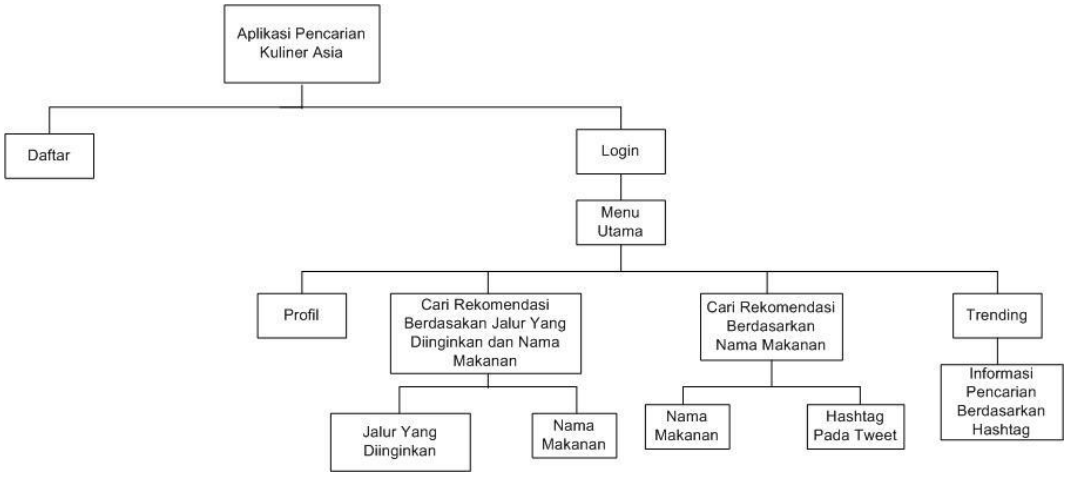
Nama Aliran Data	Data Tempat Rekomendasi
Digunakanan Pada Proses	Proses 1.0
Deskripsi	Berisi data-data tempat rekomendasi
Struktur Data	nama + foto + alamat + latitude + longitude nama [A-Z|a-z] foto [A-Z|a-z] alamat [A-Z|a-z|0-100] latitude [0-100] longitude [0-100]
Nama Aliran Data	Data Hashtag
Digunakanan Pada Proses	Proses 2.0
Deskripsi	Berisi data hashtag
Struktur Data	tag tag [a-z]

3.1.6.4 Struktur Menu

Perancangan struktur menu dilakukan untuk memberikan gambaran pemakaian aplikasi dan juga gambaran menu apa saja yang bisa diakses. Perancangan struktur menu untuk aplikasi yang dibagun ini dapat dilihat sebagai berikut :

1. Struktur Menu *Platform Mobile*

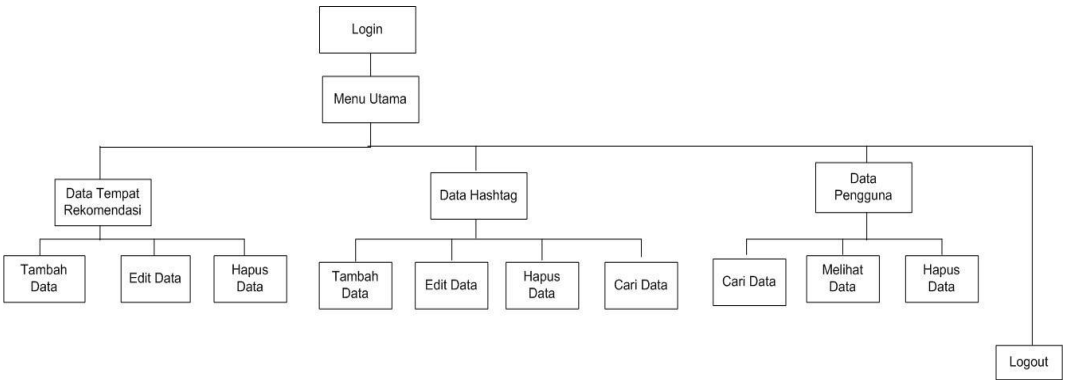
   Berikut adalah gambar Struktur Menu *Platform Mobile*.



Gambar 3. 32 Struktur Menu *Platform Mobile*

2. Struktur Menu *Platform Web*

   Berikut adalah gambar Struktur Menu *PlatformWeb*



Gambar 3.33 Struktur Menu *Platform Web*

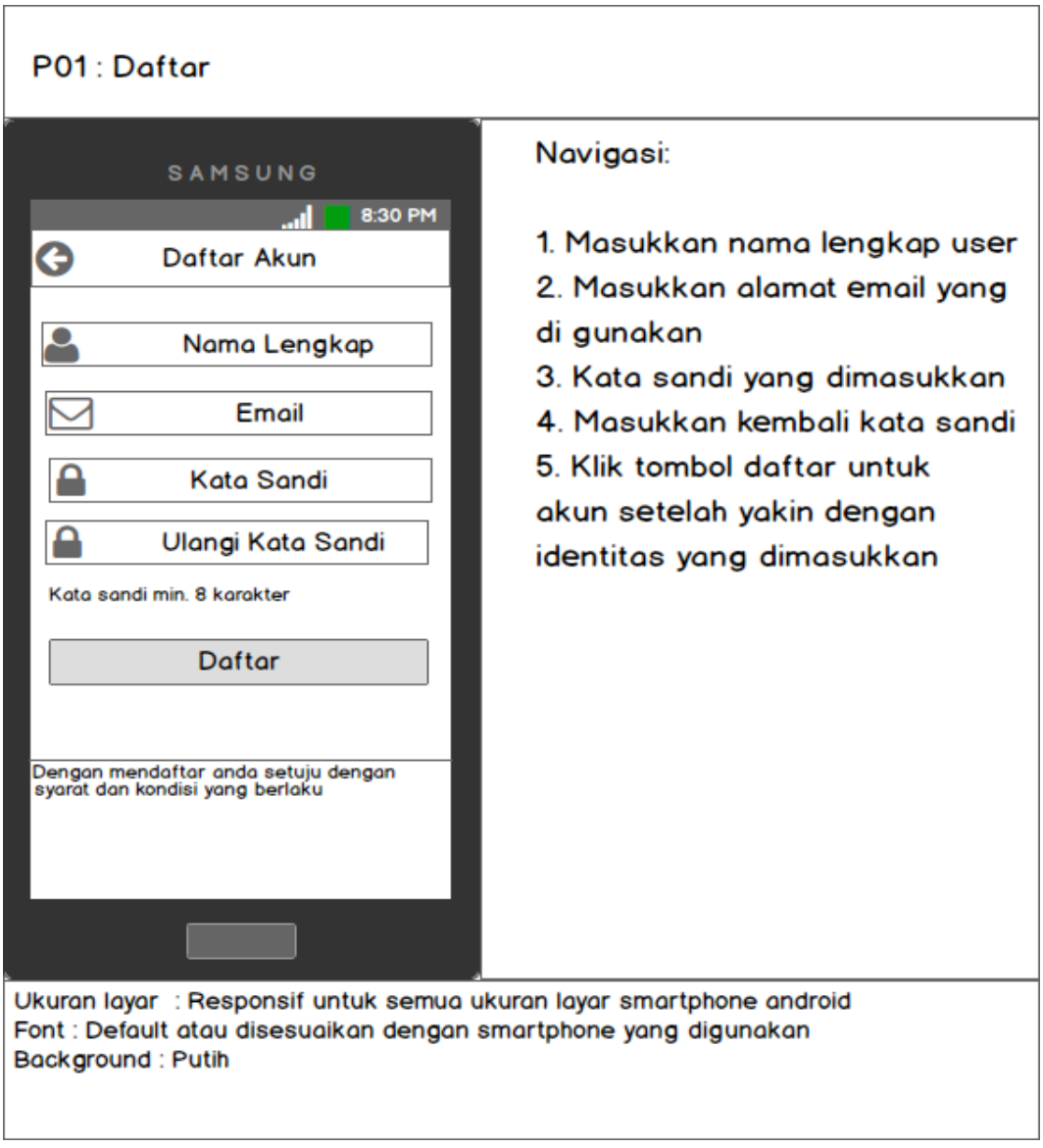
3.1.6.5 Perancangan Antarmuka

Perancangan antarmuka merupakan penjelasan atau gamabaran tentang aplikasi yang nantinya akan dibangun. Dari perancangan antarmuka ini nantinya dapat diimplementasikan menjadi sebuah program. Program yang dimananya nanti dapat digunakan langsung oleh pengguna. Berikut adalah perancangan antarmuka untuk pembangunan aplikasi pencarian tempat makan kuliner asia di kota Bandung :

1. Perancangan Antarmuka Mobile Pengguna

   a. Perancangan Antarmuka Daftar

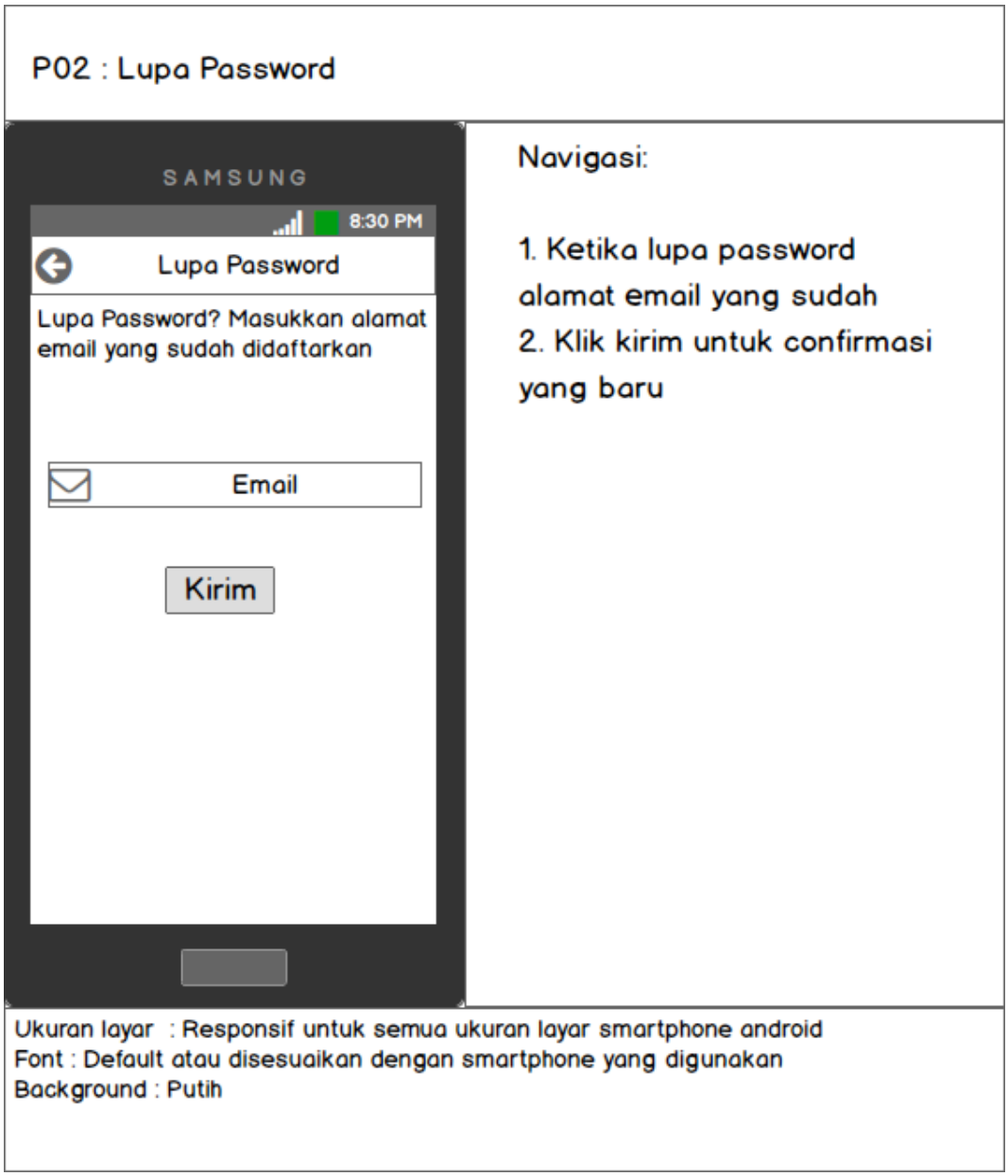
   Berikut adalah perancangan antarmuka untuk daftar yang dapat dilihat pada gambar dibawah ini.



Gambar 3.34 Antarmuka P01 Daftar

b. Perancangan Antarmuka Lupa Password

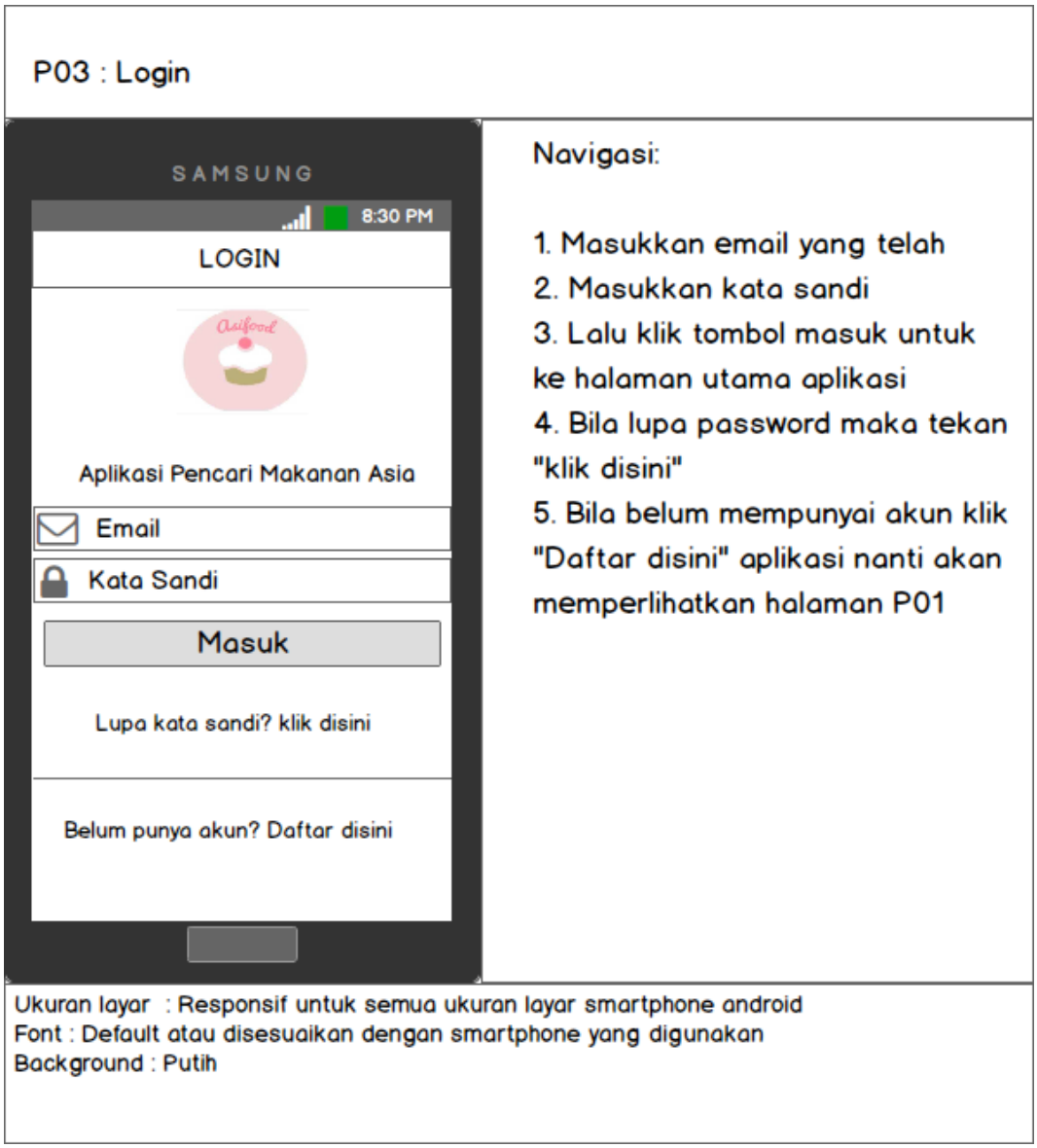
Berikut adalah perancangan antarmuka untuk lupa password yang dapat dilihat pada gambar dibawah ini.



Gambar 3.35 Antarmuka P03 Lupa *Password*

c. Perancangan Antarmuka *Login*

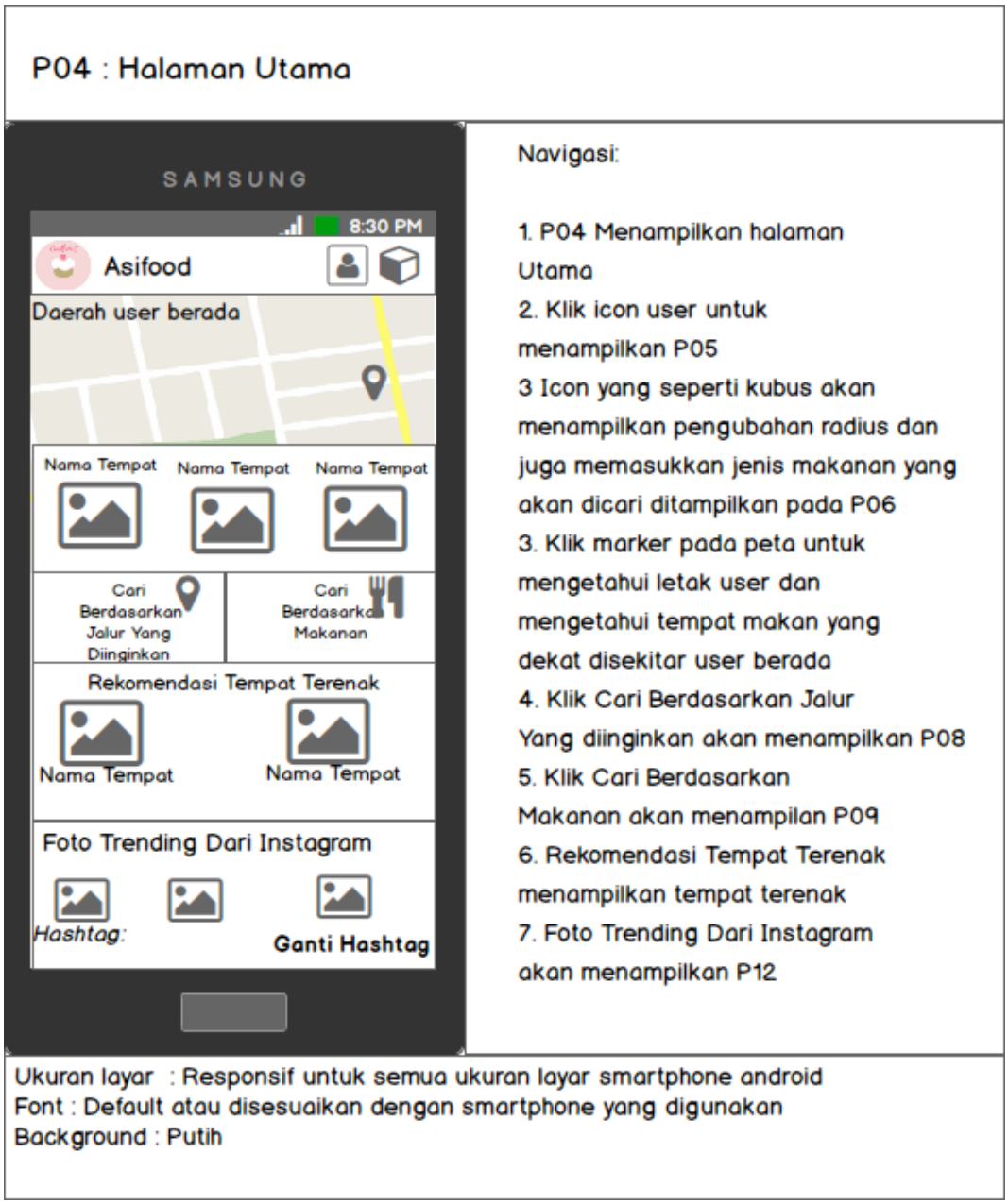
Berikut adalah perancangan antarmuka untuk login yang dapat dilihat pada gambar dibawah ini.



Gambar 3.36 Antarmuka P03 *Login*

d. Perancangan Antarmuka Halaman Utama

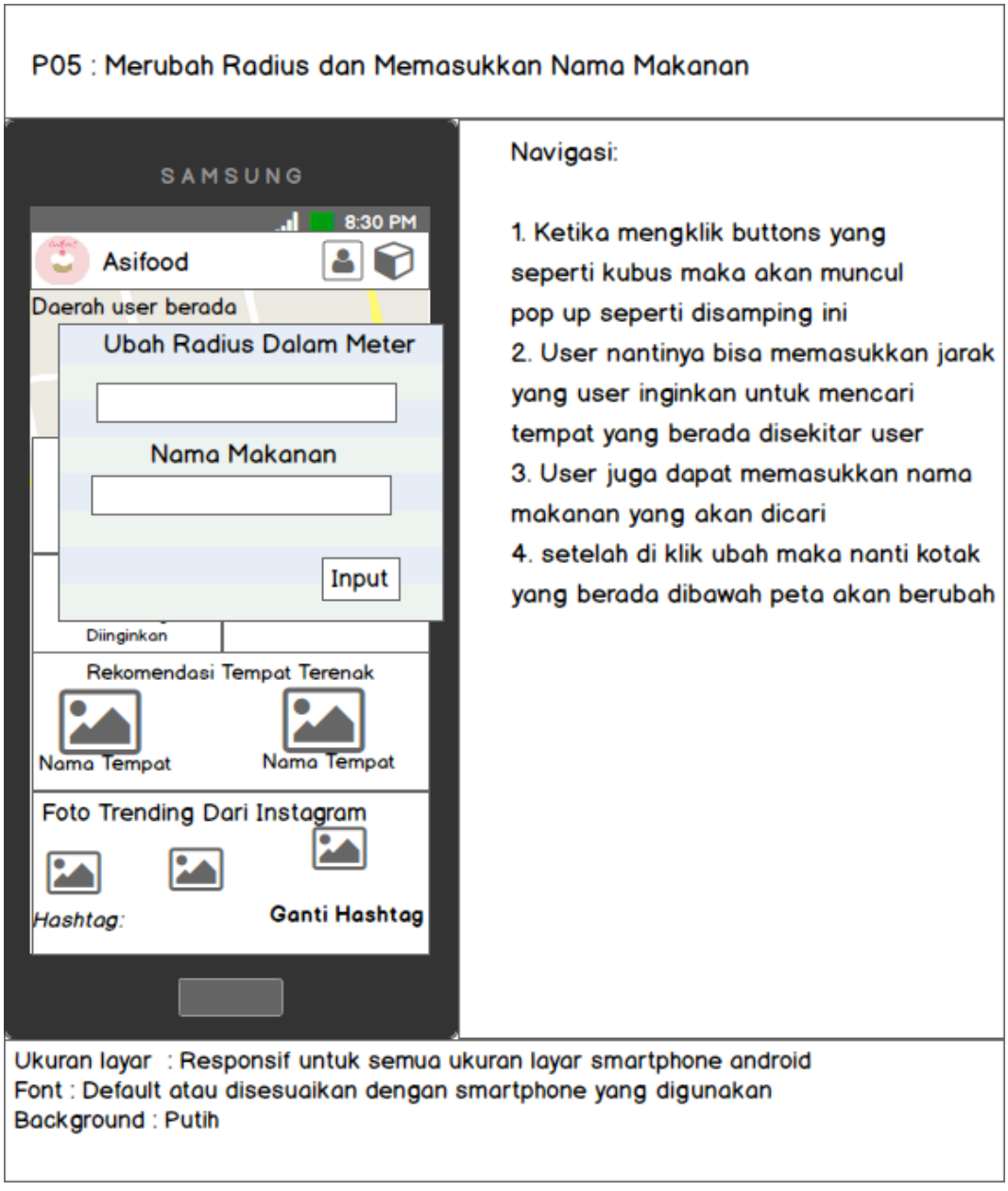
Berikut adalah perancangan antarmuka untuk halaman utama yang dapat dilihat pada gambar dibawah ini.



Gambar 3.37 Antarmuka P04 Halaman Utama

e. Mengubah Radius Pencarian dan Memasukkan Nama Makanan

Berikut adalah perancangan antarmuka untuk mengubah radius dan memasukkan nama makanan untuk wilayah disekitar pengguna berada yang dapat dilihat pada gambar dibawah ini.



Gambar 3.38 Antarmuka P05 Merubah Radius dan Memasukkan Nama Makanan

f. Hasil Dari Merubah Radius dan Memasukkan Nama Makanan

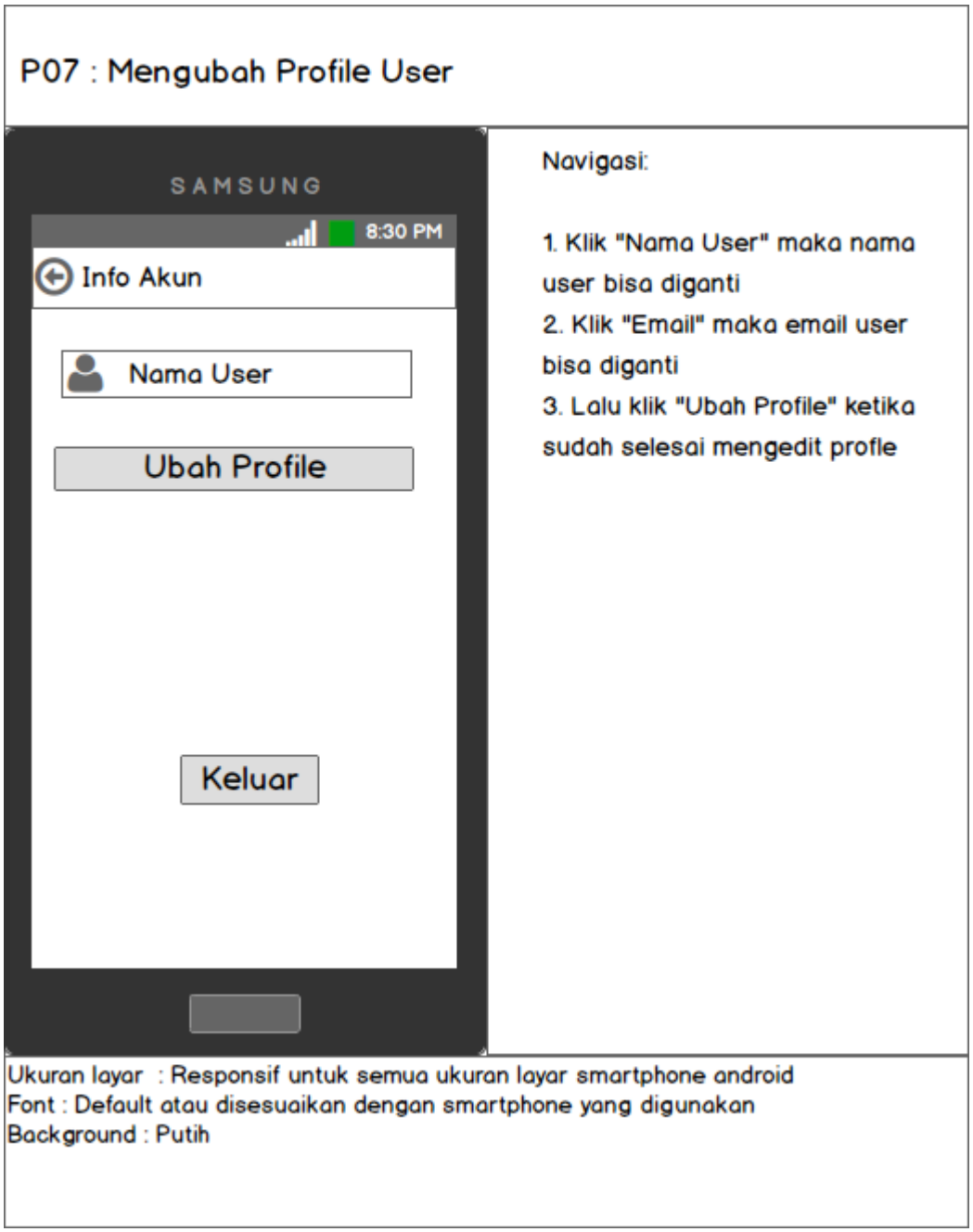
Berikut adalah perancangan antarmuka untuk hasil mengubah radius dan memasukkan nama makanan untuk wilayah disekitar pengguna berada yang dapat dilihat pada gambar dibawah ini.



Gambar 3.39 Antarmuka P06 Hasil Merubah Radius dan Memasukkan Nama Makanan

g. Perancangan Antarmuka Mengubah *Profile* Pengguna

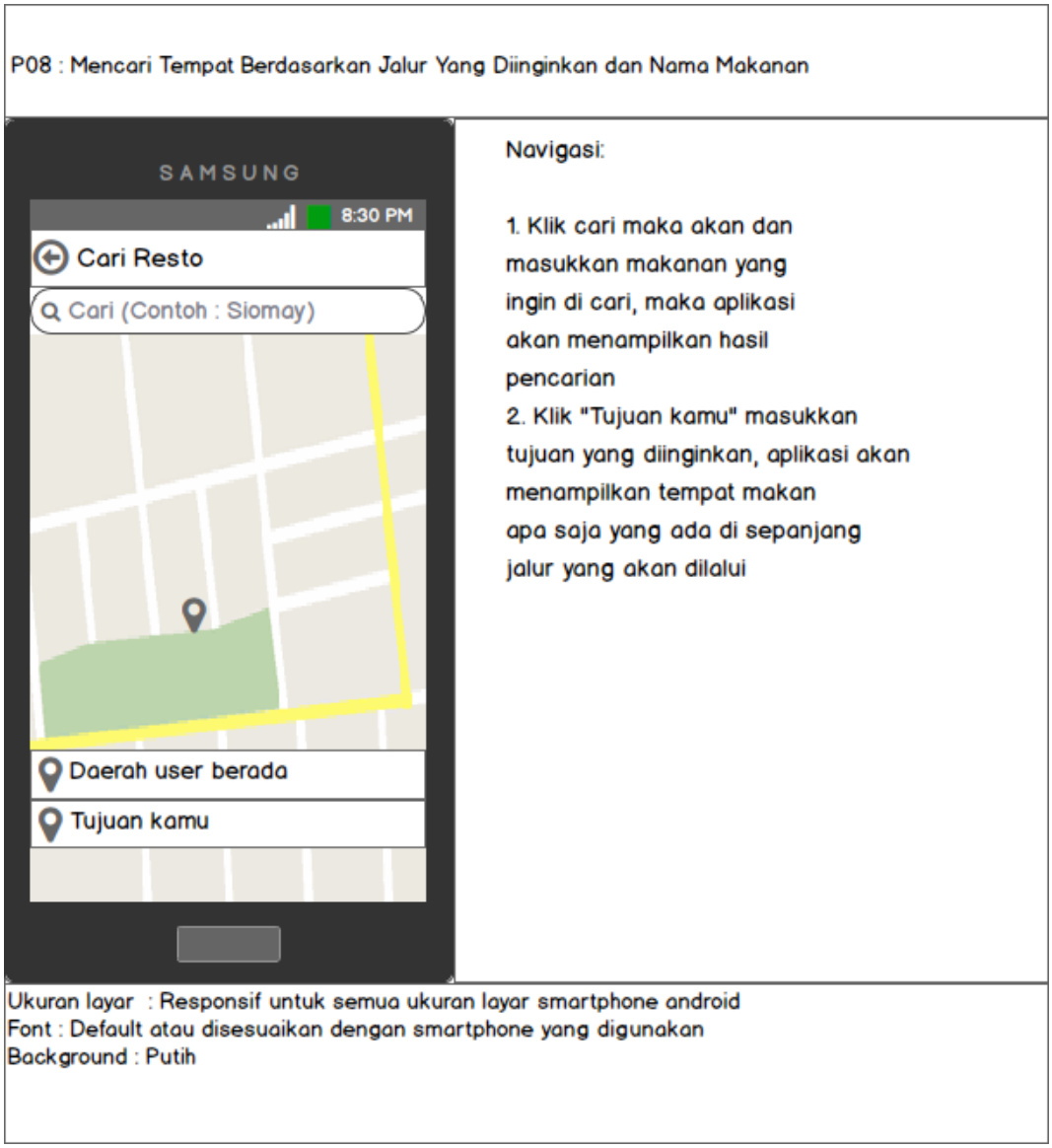
Berikut adalah perancangan antarmuka untuk daftar yang dapat dilihat pada gambar dibawah ini.



Gambar 3.40 Antarmuka P07 Mengubah *Profile* Pengguna

h. Perancangan Antarmuka Mencari Tempat Bersadasarkan Jalur Yang Diinginkan

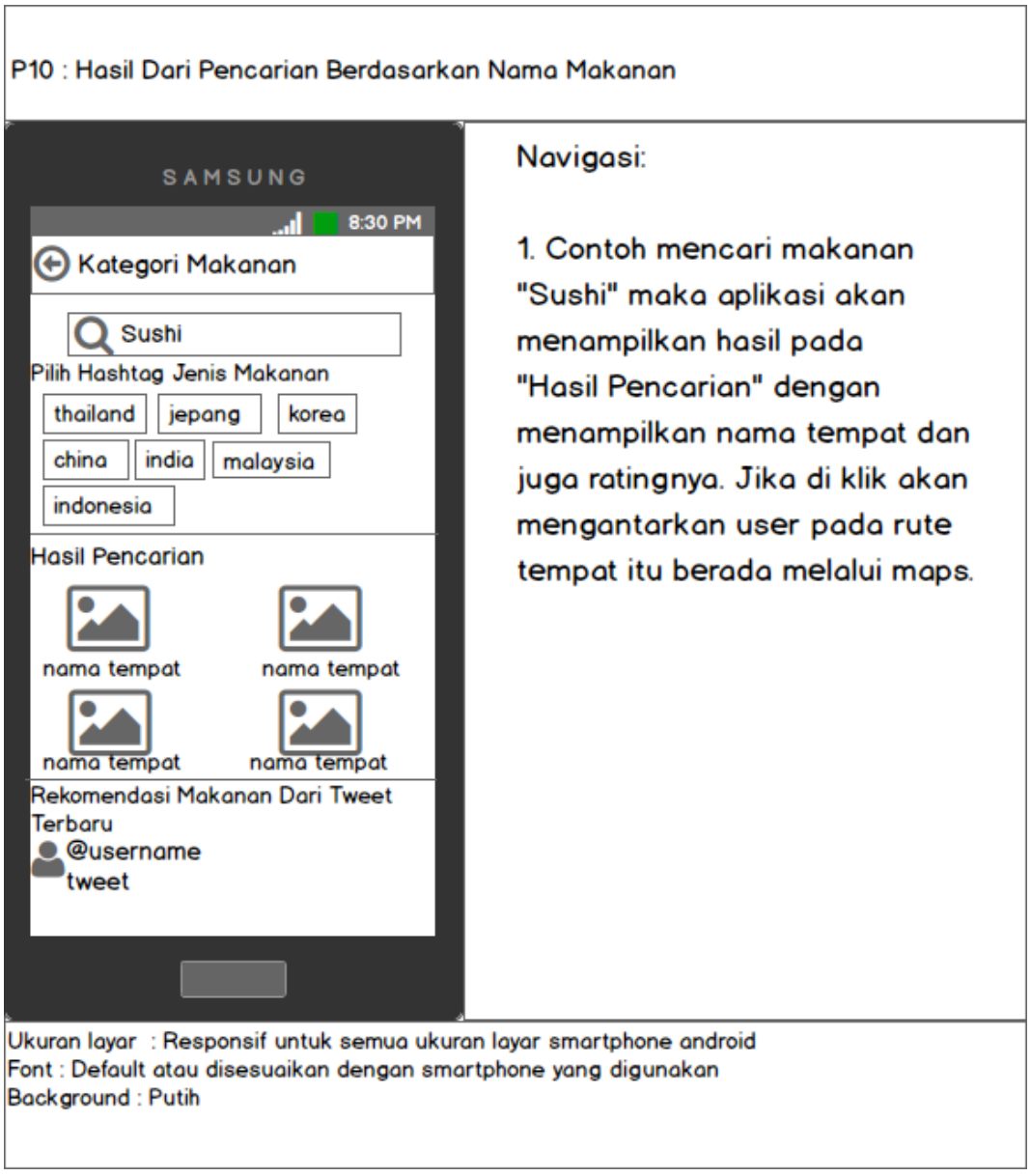
Berikut adalah perancangan antarmuka untuk daftar yang dapat dilihat pada gambar dibawah ini.



Gambar 3.41 Antarmuka P08 Mencari Tempat Berdasarkan Jalur Yang Diinginkan

i. Perancangan Antarmuka Mencari Tempat Berdasarkan Nama Makanan

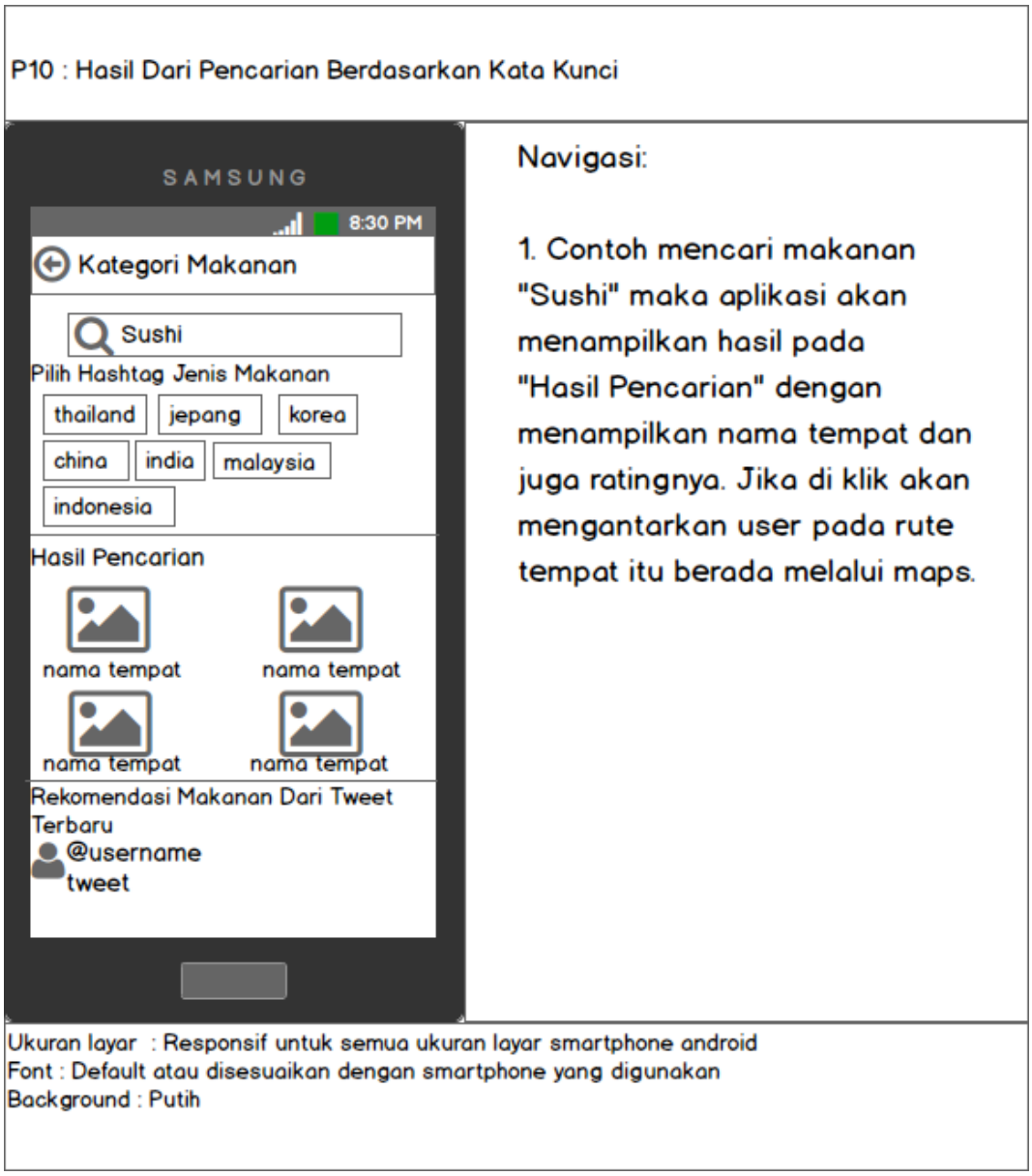
Berikut adalah perancangan antarmuka untuk daftar yang dapat dilihat pada gambar dibawah ini.



Gambar 3.42 Antarmuka P09 Mencari Tempat Berdasarkan Nama Makanan

j. Perancangan Antarmuka Hasil Dari Pencarian Berdasarkan Nama Makanan

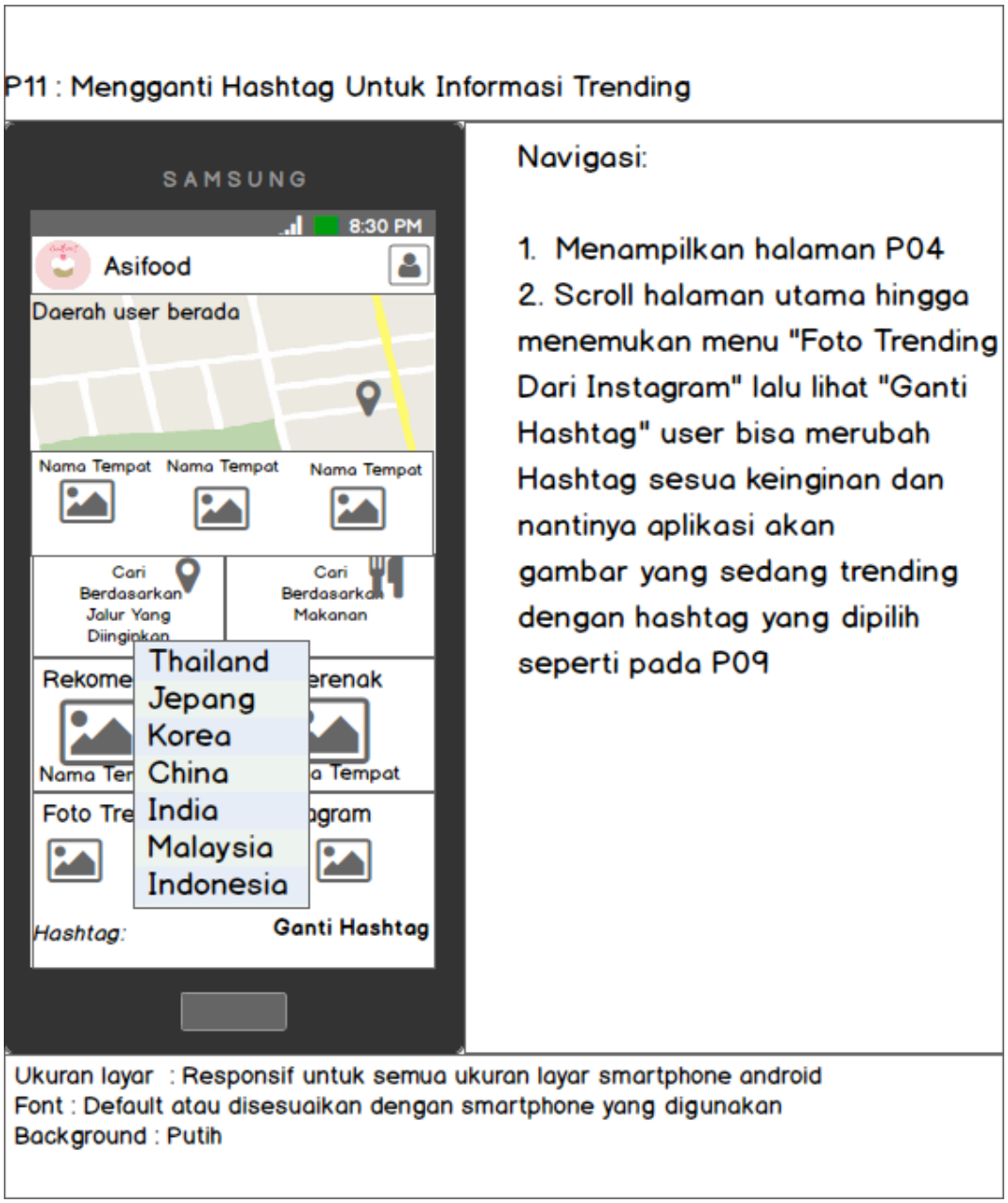
Berikut adalah perancangan antarmuka untuk daftar yang dapat dilihat pada gambar dibawa ini.



Gambar 3.43 Antarmuka P10 Hasil Dari Pencarian Berdasarkan Nama Makanan

k. Perancangan Antarmuka Mengganti Hashtag Untuk Informasi Trending

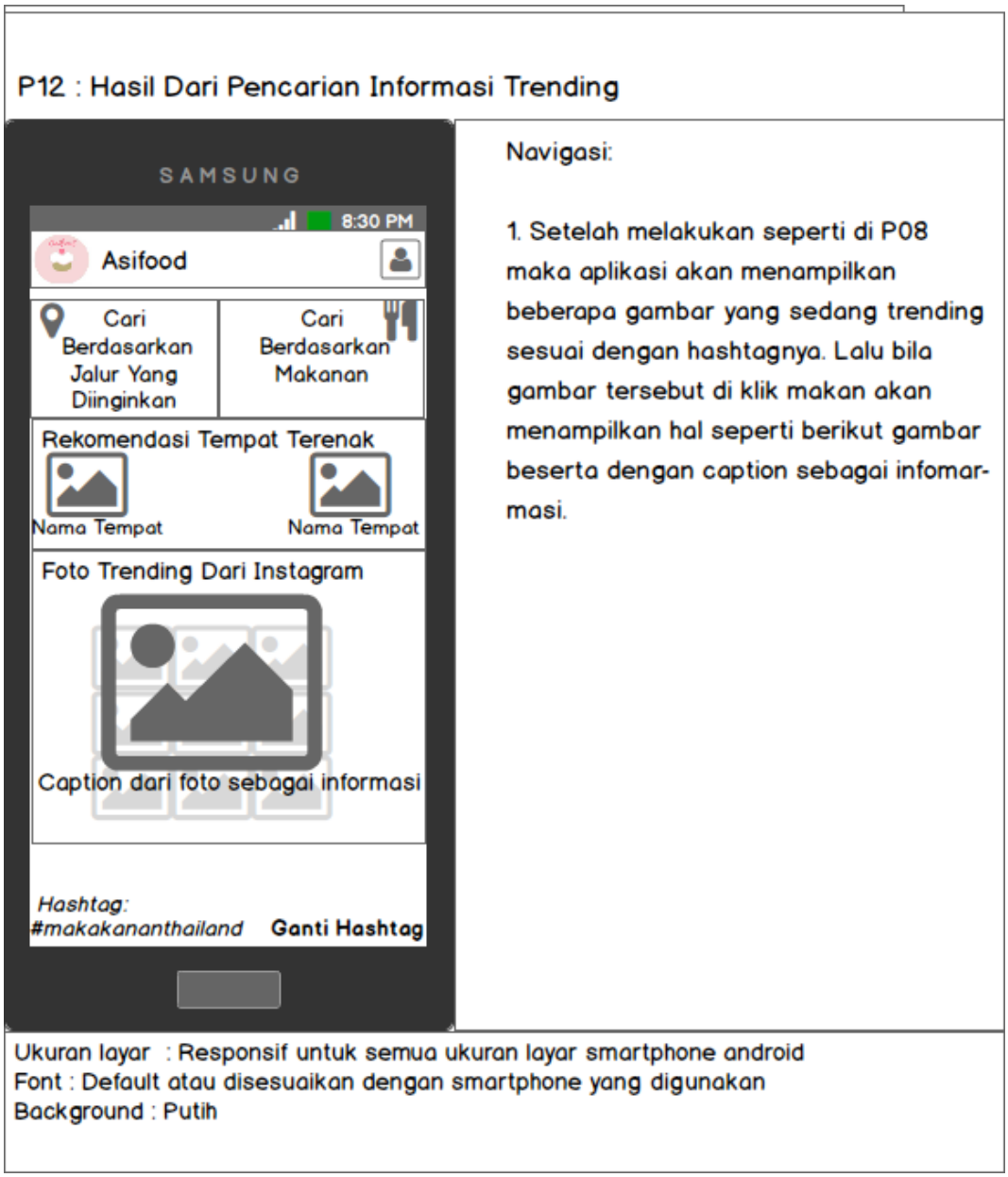
Berikut adalah perancangan antarmuka untuk daftar yang dapat dilihat pada gambar dibawah ini.



Gambar 3.44 Antarmuka P11 Mengganti *Hashtag* Untuk Informasi Trending

l. Perancangan Antarmuka Hasil Dari Pencarian Informasi Trending

Berikut adalah perancangan antarmuka untuk daftar yang dapat dilihat pada gambar dibawah ini.



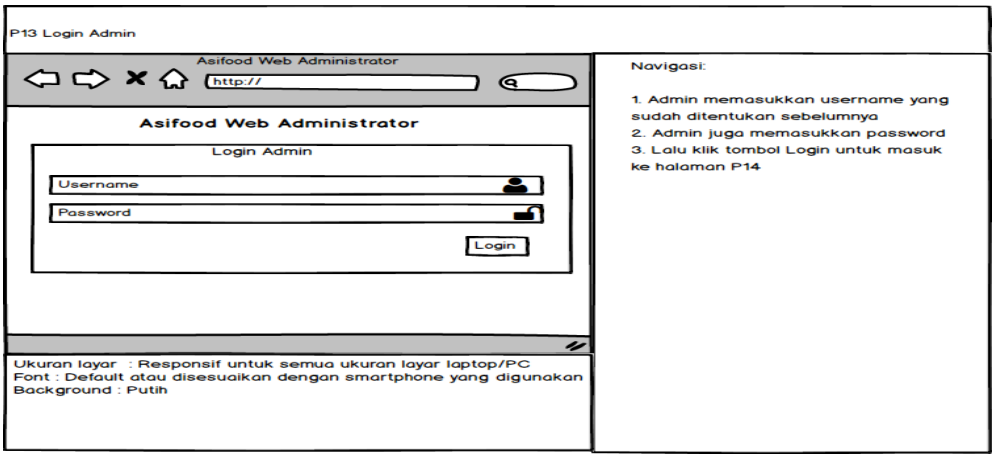
Gambar 3.45 Antarmuka P12 Hasil Dari Pencarian Informasi Trending

2. Perancangan Antarmuka *Platform Web* Admin

Berikut adalah perancangan antarmuka untuk *platform web* untuk admin :

a. Perancangan Antarmuka Login

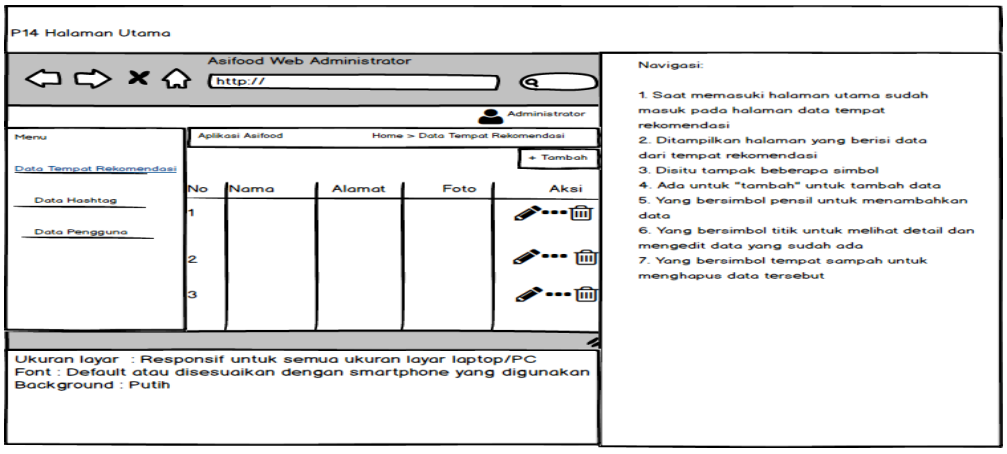
Berikut adalah perancangan antarmuka untuk login yang dapat dilihat pada gambar dibawah ini.



Gambar 3.46 Antarmuka P13 Login Web

b. Perancangan Antarmuka Halaman Utama *Web*

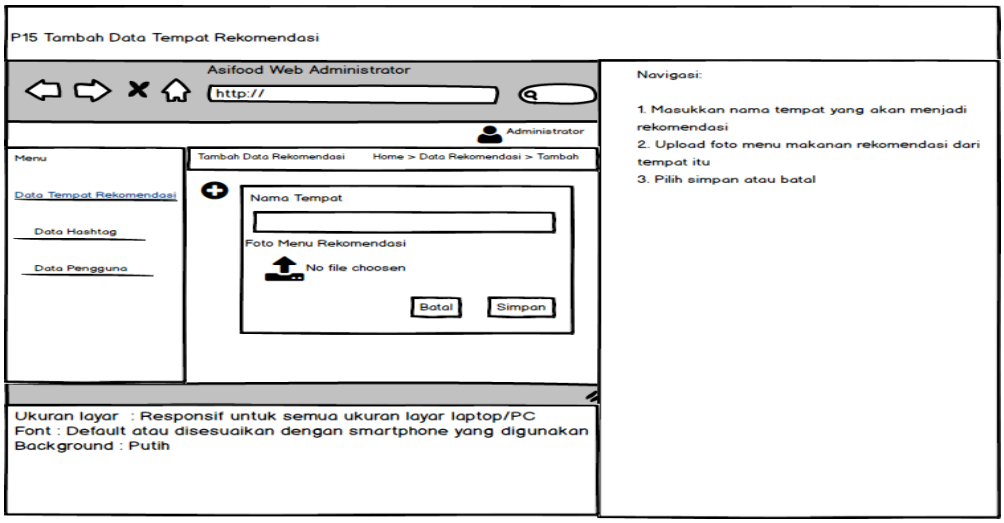
Berikut adalah perancangan antarmuka untuk daftar yang dapat dilihat pada gambar dibawah ini.



Gambar 3.47 Antarmuka P14 Halaman Utama *Web*

c. Perancangan Antarmuka Tambah Data Tempat Rekomendasi

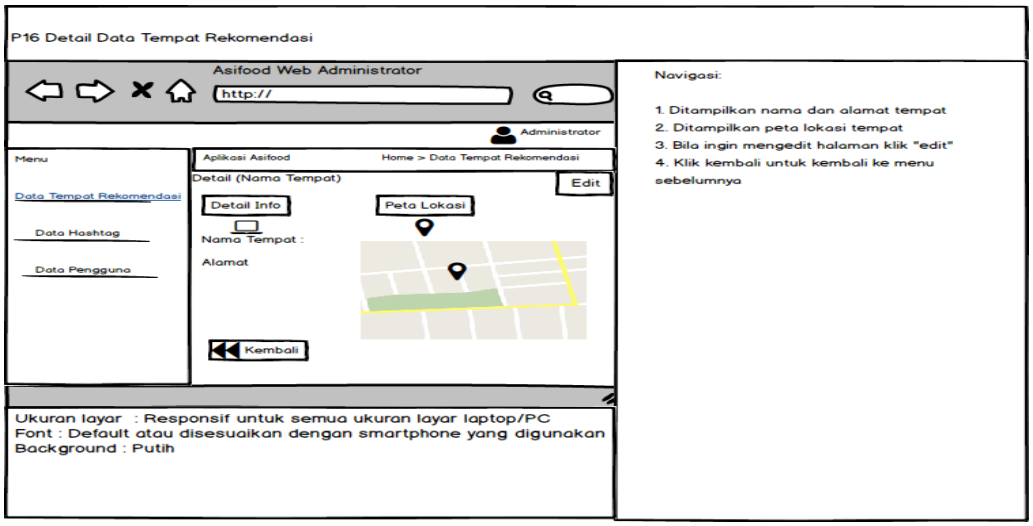
Berikut adalah perancangan antarmuka untuk daftar yang dapat dilihat pada gambar dibawah ini.



Gambar 3.48 Antarmuka P15 Tambah Data Tempat Rekomendasi

d. Perancangan Antarmuka Detail Data Tempat Rekomendasi

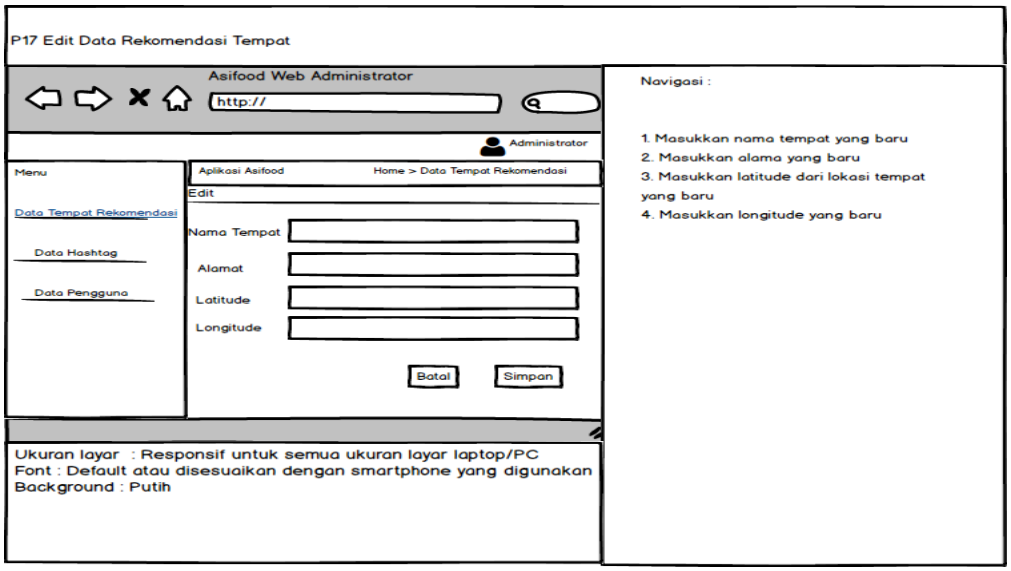
Berikut adalah perancangan antarmuka untuk daftar yang dapat dilihat pada gambar dibawah ini.



Gambar 3.49 Antarmuka P16 Detail Data Tempat Rekomendasi

e. Perancangan Antarmuka Edit Data Tempat Rekomendasi

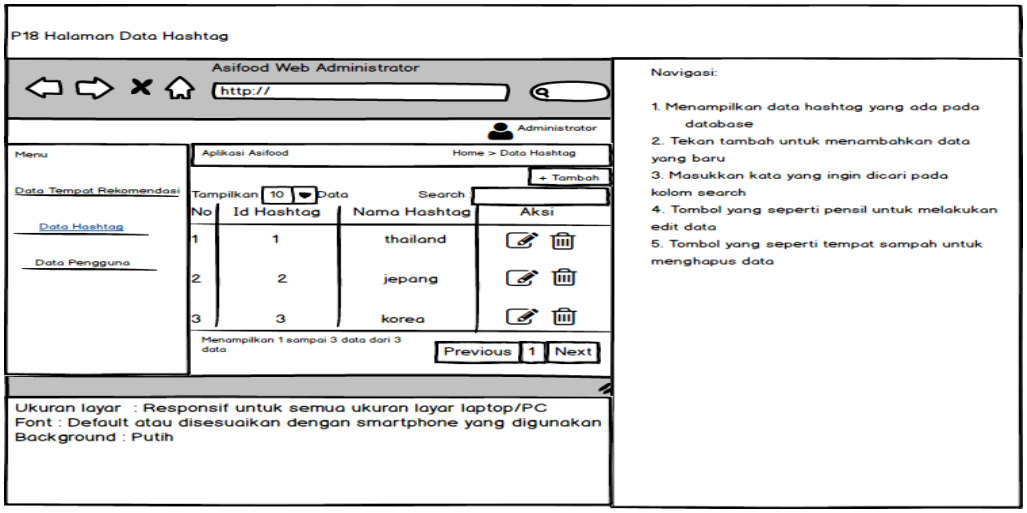
Berikut adalah perancangan antarmuka untuk daftar yang dapat dilihat pada gambar dibawah ini.



Gambar 3.50 Antarmuka P17 Edit Data Tempat Rekomendasi

f. Perancangan Antarmuka Halaman Data Hashtag

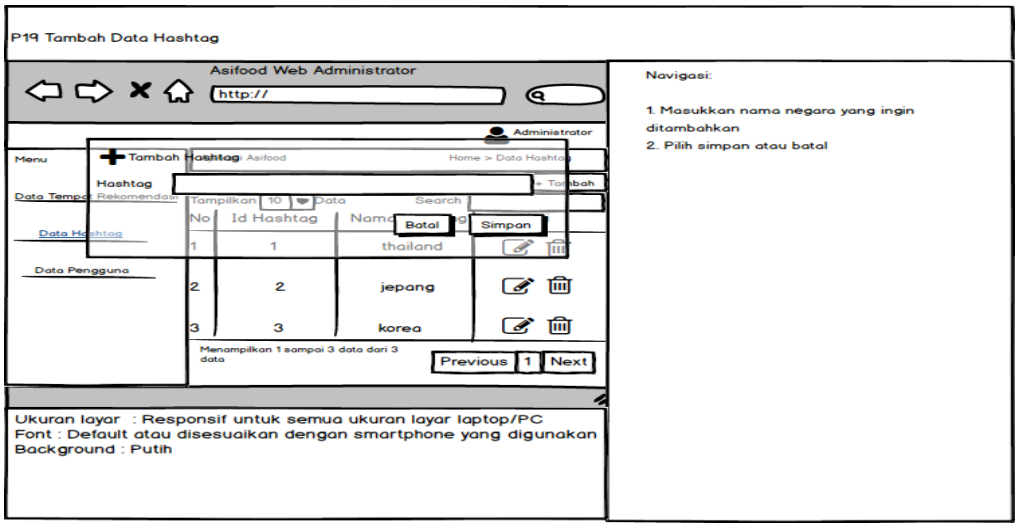
Berikut adalah perancangan antarmuka untuk daftar yang dapat dilihat pada gambar dibawah ini.



Gambar 3.51 Antarmuka P18 Halaman Data Hashtag

g. Perancangan Antarmuka Tambah Data Hashtag

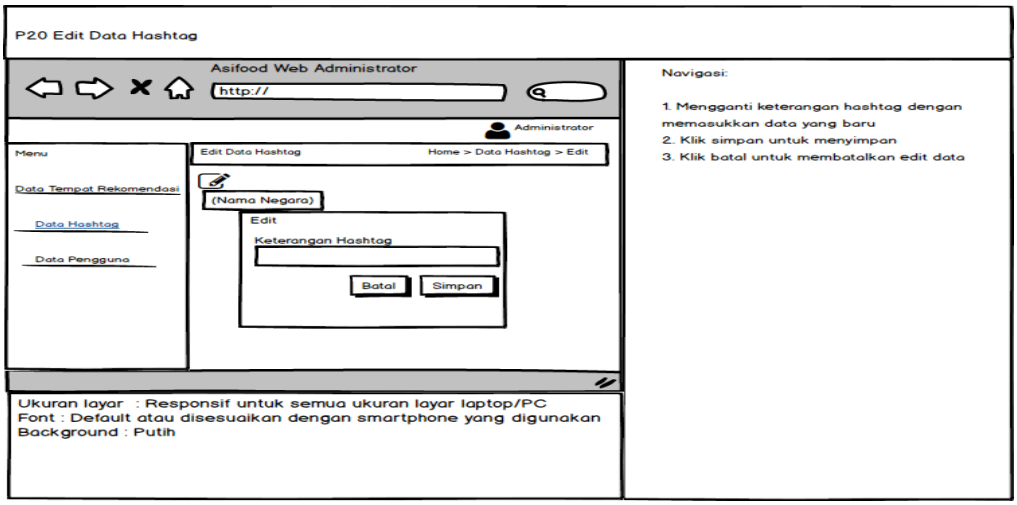
Berikut adalah perancangan antarmuka untuk daftar yang dapat dilihat pada gambar dibawah ini.



Gambar 3.52 Antarmuka P19 Tambah Data Hashtag

h. Perancangan Antarmuka Edit Data Hashtag

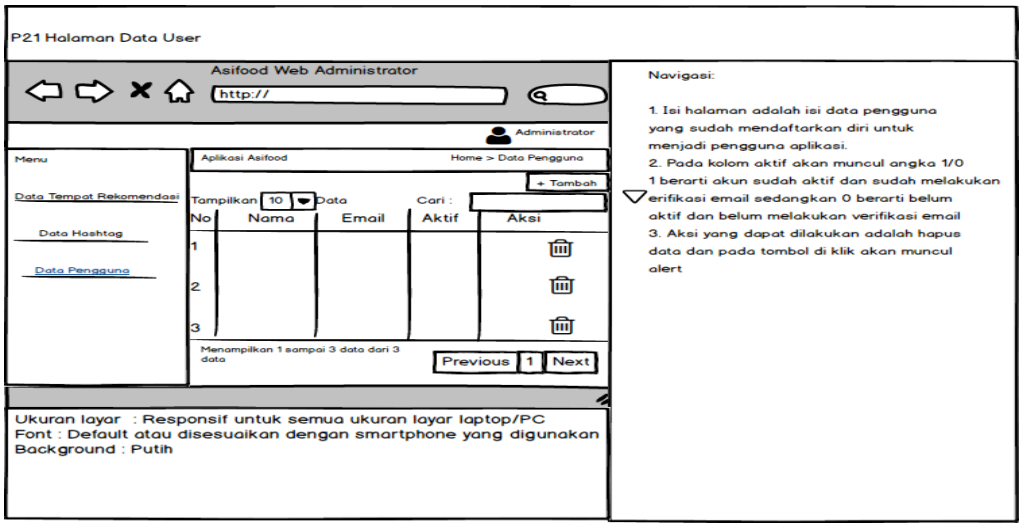
Berikut adalah perancangan antarmuka untuk daftar yang dapat dilihat pada gambar dibawah ini.



Gambar 3.53 Antarmuka P20 Edit Data Hashtag

i. Perancangan Antarmuka Halaman Data Pengguna

Berikut adalah perancangan antarmuka untuk daftar yang dapat dilihat pada gambar dibawah ini.



Gambar 3.54 Antarmuka P21 Halaman Data Pengguna

3.1.6.6 Perancangan Pesan

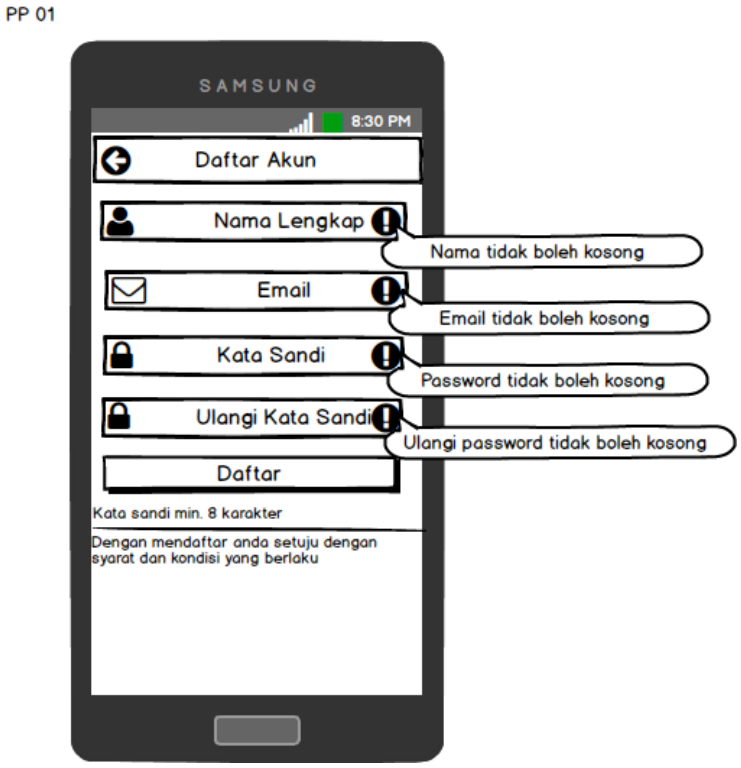
Perancangan pesan merupakan perancangan yang dibuat untuk memberi pesan yang berisi informasi apa saja yang akan ditampilkan pada saat pengguna melakukan sebuah perintah pada satu menu, berikut adalah perancangan pesan yang akan digunakan pada sistem aplikasi yang akan dibangun ini :

1. Perancangan Pesan Pada *Platform Mobile* Pengguna

   Berikut adalah perancangan antarmuka pesan *pada platform mobile* pengguna.

a. Perancangan Pesan Daftar

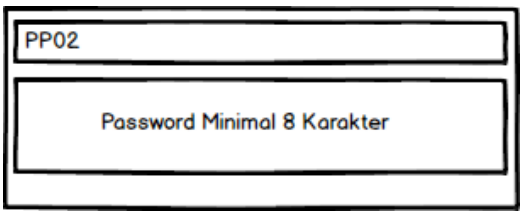
Berikut adalah perancangan pesan untuk menu daftar, disaat pengguna akan melakukan pendaftaran bila salah satu identitas tidak diisi makan akan muncul pesan seperti yang terlihat pada gambar dibawah ini.



Gambar 3.55 Perancangan Pesan PP01 Daftar

b. Perancangan Pesan Minimal *Password*

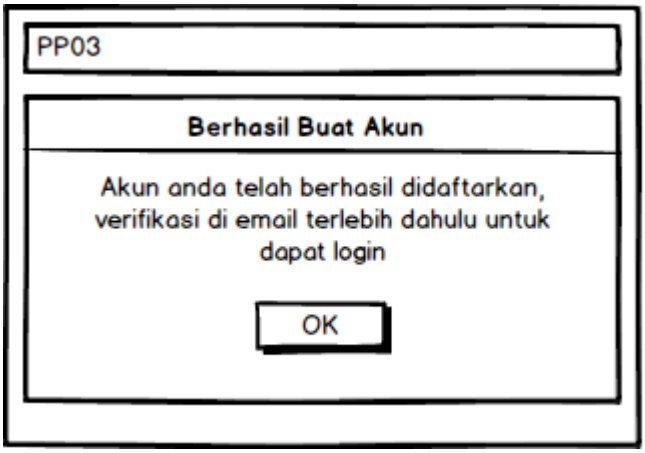
Berikut adalah perancangan pesan yang akan muncul jika disaat pengguna mengisi identitas untuk mendaftarkan akun mereka dan password diisikan kurang dari 8 karakter. Pesan tersebut seperti yang ada pada gambar dibawah ini.



Gambar 3. 56 Perancangan Pesan PP03 Minimal *Password*

c. Perancangan Pesan Berhasil Buat Akun

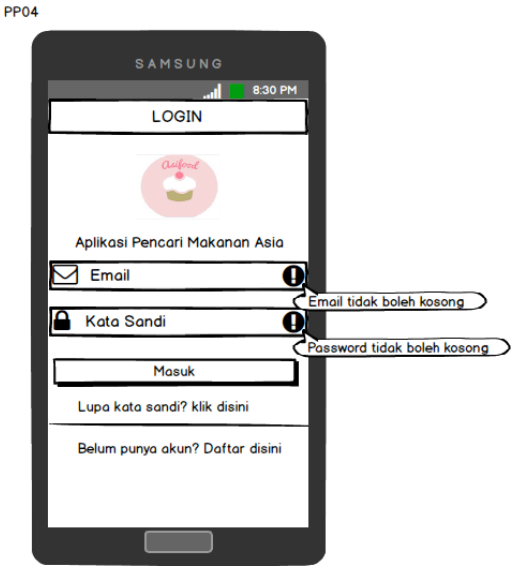
Berikut adalah perancangan pesan pada saat pengguna telah berhasil melakukan pendaftaran akun dan aplikasi meminta untuk melakukan verifikasi email agar pengguna dapat melakukan Login. Pesan tersebut dapat dilihat pada gambar dibawah ini.



Gambar 3.57 Perancangan Pesan PP03 Berhasil Buat Akun

d. Perancangan Pesan Login

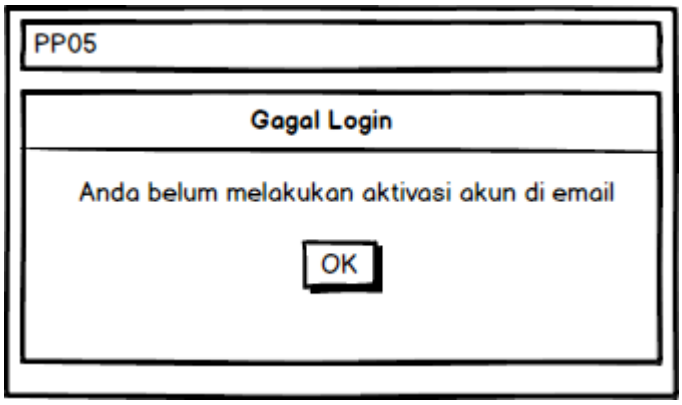
Berikut adalah perancangan pesan disaat pengguna akan melakukan Login saat email dan kata sandi tidak diisi maka akan muncul pesan seperti pada gambar dibawah ini.



Gambar 3.58 Perancangan Pesan PP04 *Login*

e. Perancangan Pesan Gagal Login

Berikut adalah perancangan pesan pada saat pengguna akan melakukan login tetapi akun belum melakukan aktivasi email. Pesan tersebut dapat dilihat pada gambar dibawah ini.



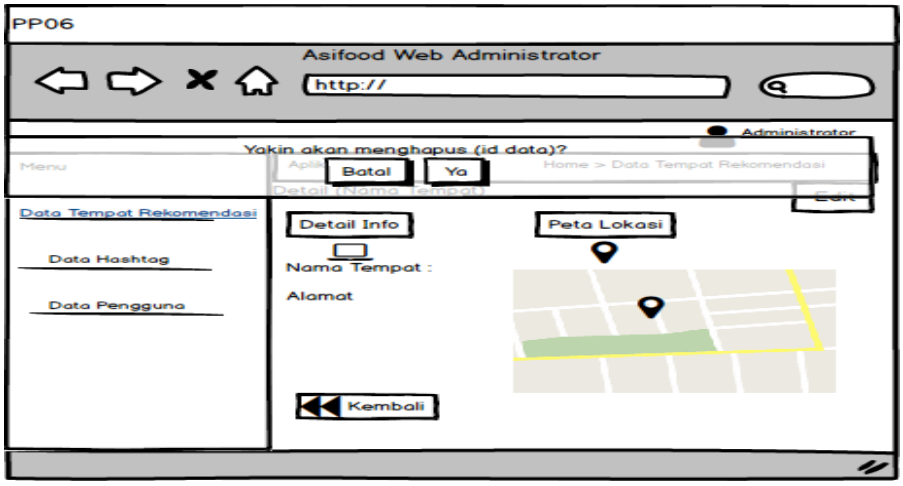
Gambar 3.59 Perancangan Pesan PP04 Gagal Login

2. Perancangan Pesan Pada *Platform Web* Admin

Berikut adalah perancangan antarmuka pesan pada *platform web* admin :

a. Perancangan Pesan Hapus Data Rekomendasi Tempat

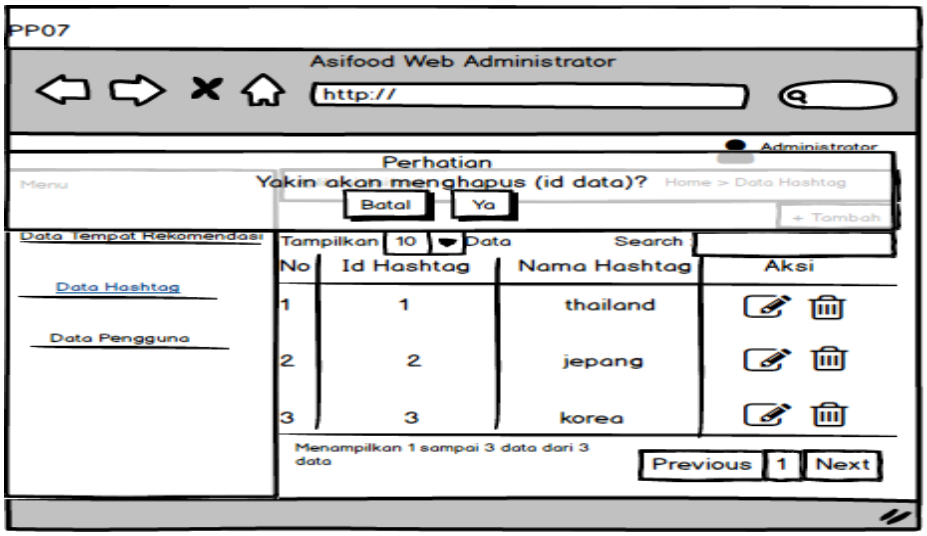
Berikut adalah perancangan pesan pada saat admin akan melakukan penghapusan terhadap data rekomendasi tempat. Pesan tersebut dapat dilihat pada gambar dibawah ini.



Gambar 3.60 Perancangan Pesan PP06 Hapus Data Rekomendasi Tempat

b. Perancangan Pesan Hapus Data Hashtag

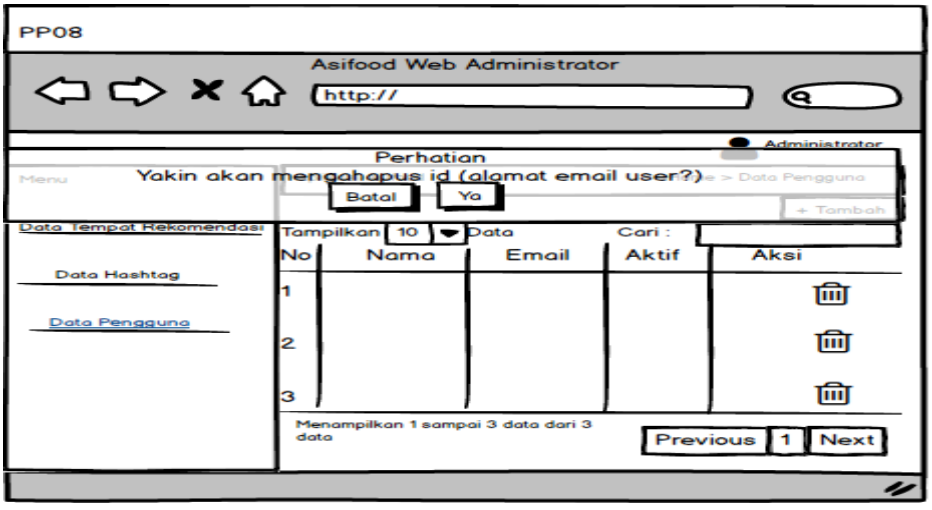
Berikut adalah perancangan pesan pada saat admin akan melakukan penghapusan terhadap data hashtag. Pesan tersebut dapat dilihat pada gambar dibawah ini.



Gambar 3.61 Perancangan Pesan PP07 Hapus Data Hashtag

c. Perancangan Pesan Data Pengguna

Berikut adalah perancangan pesan pada saat admin akan melakukan penghapusan terhadap data pengguna. Pesan tersebut dapat dilihat pada gambar dibawah ini



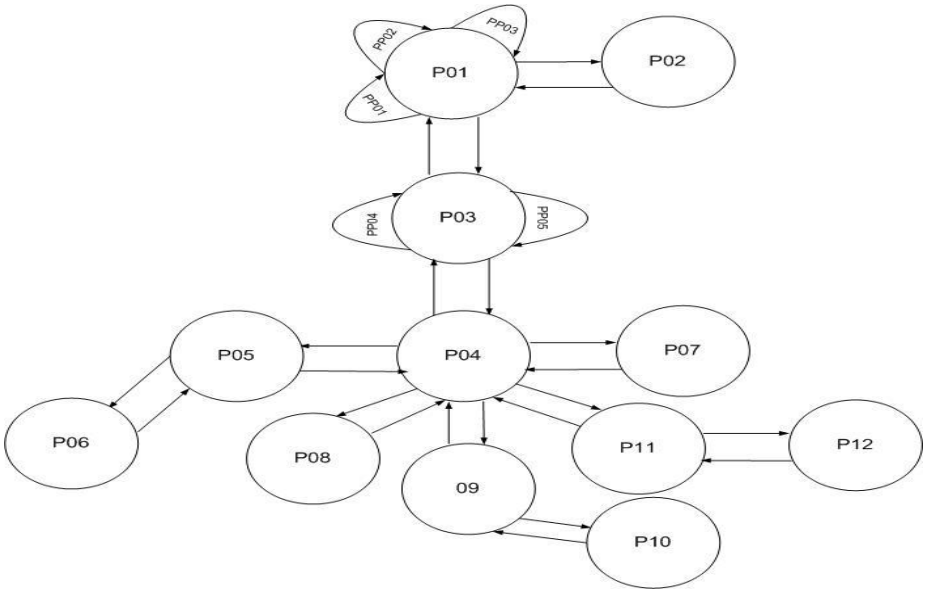
Gambar 3.62 Perancangan Pesan PP08 Hapus Data Pengguna

3.1.6.6 Jaringan Semantik

Jaringan semantik pada pembangunan aplikasi ini adalah sebagai berikut :

1. Jaringan Semantik Platform Mobile

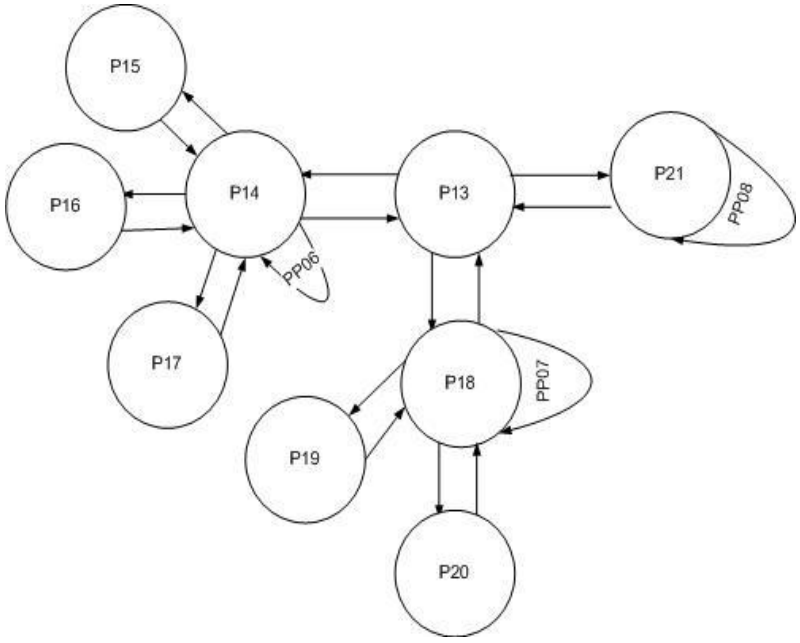
   Berikut adalah jaringan semantik pada *platform mobile* :



Gambar 3.63 Jaringan Semantik Platform *Mobile*

2. Jaringan Semantik

   Berikut adalah jaringan semantik pada *platform web*



Gambar 3.64 Jaringan Semantik Platform *Web*