BAB 3
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Analisis sistem adalah tahapan dimana seorang peneliti atau pengembang menganalisis sebuah sistem yang bertujuan untuk mengidentifikasi permasalahan – permasalahan yang terdapat pada sebuah sistem yang sedang berjalan dan menentukan kebutuhan – kebutuhan dari sistem yang akan dibangun. Pada tahap analisis sistem ini terdapat beberapa analisis di dalamnya meliputi analisis masalah, analisis sistem yang sedang berjalan, analisis sistem yang akan dibuat, analisis arsitektur sistem, analisis teknologi yang digunakan, analisis kebutuhan non fungsional hingga analisis kebutuhan fungsional.

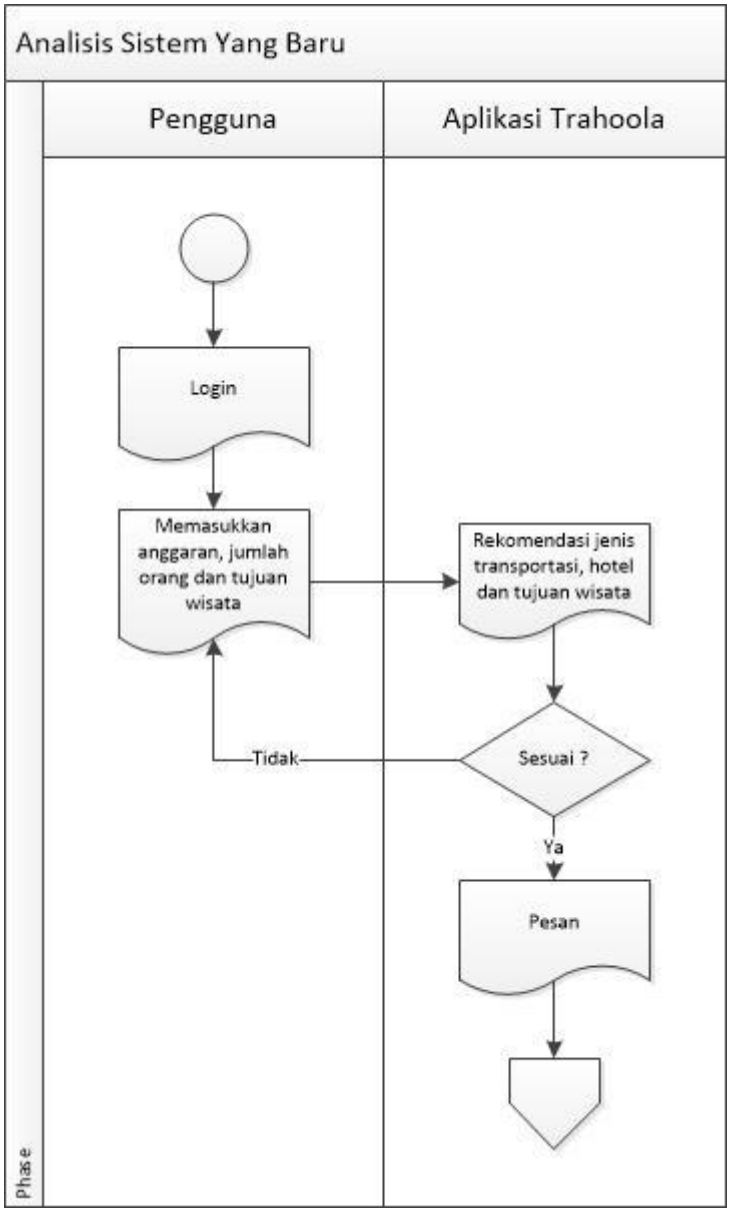
3.2 Analisis Masalah

Dalam melakukan perjalanan khususnya perjalanan wisata, pengguna sering kali mengalami kesulitan dalam mencari kebutuhan perjalanan wisata yang diinginkan. Ditambah dengan adanya keterbatasan dalam hal anggaran karena adanya berbagai pertimbangan untuk hal lain. Dengan rumusan masalah yang didapatkan dari berbagai cara pengumpulan data, maka adapun masalah-masalah yang ada diantaranya sebagai berikut :

1. Sulitnya pengguna dalam mencari informasi kebutuhan perjalanan wisata yang diinginkan berdasarkan anggaran yang dimiliki.
2. Belum tersedianya aplikasi yang bisa memberikan rekomendasi perjalanan wisata terhadap pengguna. Dengan tujuan bisa mendapatkan rekomendasi jenis transportasi, harga, rute, penginapan dan tujuan wisata agar kebutuhan tersebut bisa terpenuhi.
3. Sulitnya pengguna dalam mengetahui informasi detail dari transportasi, penginapan, dan tujuan wisata mulai dari harga, waktu, jenis transportasi dan jarak tempuh.

3.3 Analisis Prosedur Sistem yang Dibangun

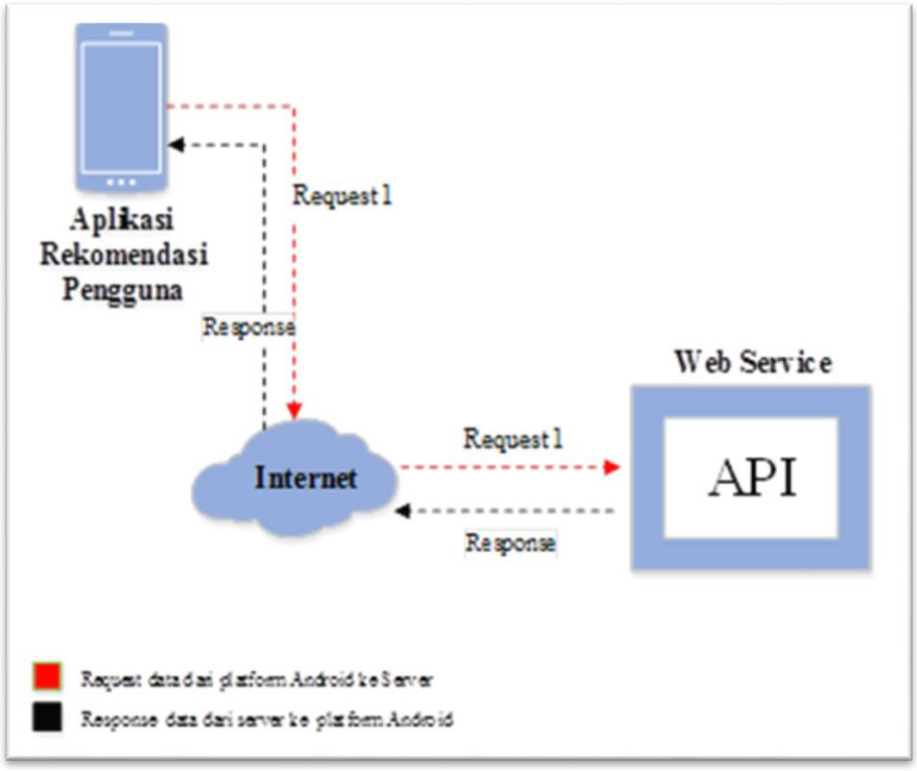
Analisis sistem yang baru berisi mengenai analisis yang berjalan nanti. Dengan cara-cara tertentu yang berbeda dengan sistem yang sedang berjalan saat ini. Dibandingkan terhadap sistem yang sedang berjalan sekarang. Dengan adanya sistem yang baru ini memberi kemudahan terhadap pengguna.



Gambar 3. 1 Prosedur Sistem yang Dibangun

3.4 Analisis Arsitektur Sistem

Analisis arsitektur sistem bertujuan untuk mengidentifikasi arsitektur yang dibangun berdasarkan *platform mobile*. *Platform mobile* merupakan sistem yang dipilih untuk pembangunan dari perangkat lunak ini. Arsitektur perangkat lunak pada *platform mobile* menggambarkan bagaimana perangkat lunak saling berinteraksi seperti diilustrasikan pada Gambar 3.2 Arsitektur Perangkat Lunak pada *Platform Android*. Gambar tersebut menggambarkan keseluruhan arsitektur sistem pada *platform mobile*.



Gambar 3. 2 Arsitektur Sistem Web

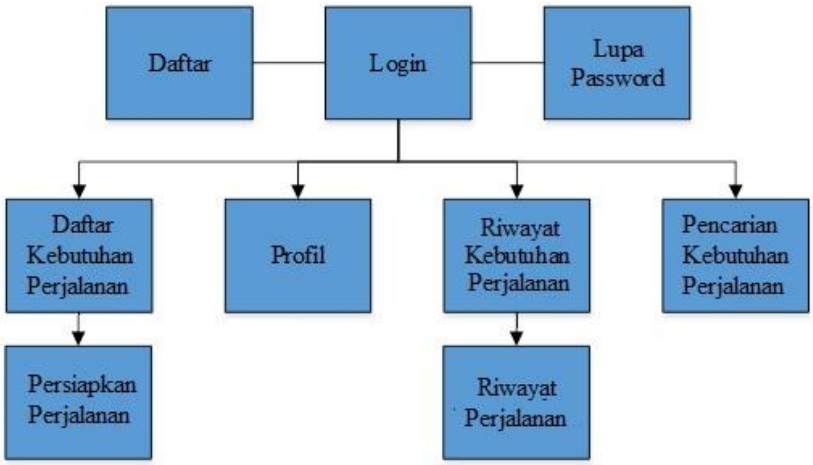
Berikut adalah deskripsi dari Gambar 3.2 Arsitektur Perangkat Lunak pada Platform Android :

1.5 Perangkat *mobile* pengguna melakukan koneksi ke internet untuk *Request1* data ke API

2.5 Setelah *Request1* data dari perangkat *mobile* pengguna sampai dan diterima oleh API, kemudian API melanjutkan *Request2* terhadap *server*.

3.5 Perancangan Arsitektur Menu

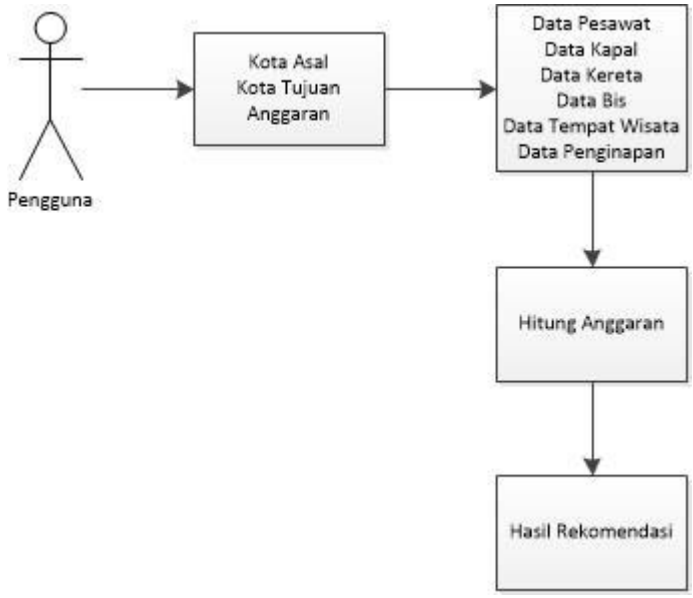
Perancangan arsitektur menu ini untuk menyediakan satu gambaran dari struktur menu program dari perangkat lunak yang dibuat. Tujuan perancangan ini adalah untuk membangun struktur dari program tersebut dan menggambarkan hubungan kendali diantara modul program. Adapun perancangan arsitektur menu pada perangkat lunak yang dibangun terdiri dari perancangan arsitektur menu pada *mobile*. Berikut adalah Gambar 3.3 Perancangan Arsitektur Menu yang menggambarkan arsitektur menu *mobile*.



Gambar 3.3 Perancangan Arsitektur Menu

3.6 Analisis Metode Rekomendasi

Analisis metode rekomendasi bertujuan untuk mengetahui alur pembangunan aplikasi Trahoola. Adapun metode yang digunakan sebagai berikut.



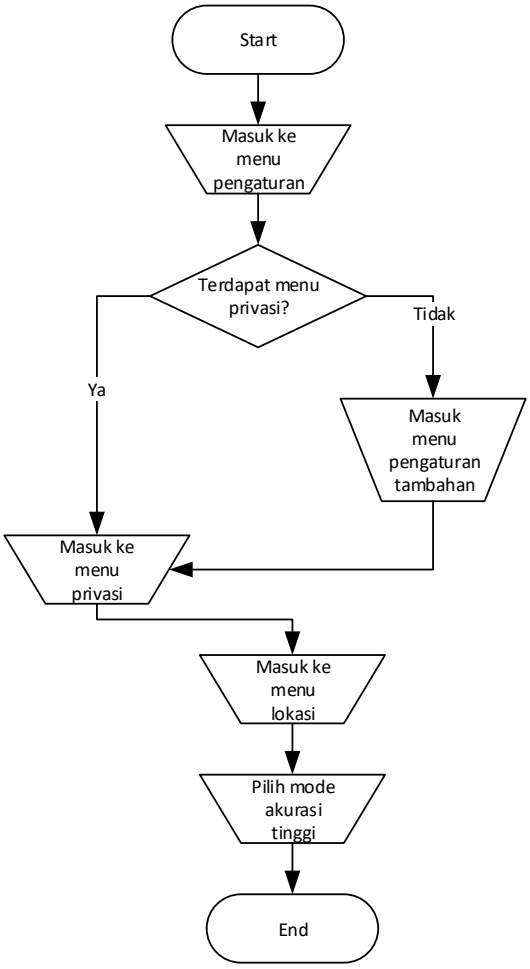
Gambar 3.4 Konsep Perancangan Aplikasi

3.7 Analisis Teknologi Yang Digunakan

Analisis teknologi bertujuan untuk mengetahui teknologi apa saja yang akan dipakai di dalam sebuah sistem yang dibangun. Adapun teknologi yang digunakan pada sistem yang dibangun antara lain :

3.7.1 GPS

Teknologi yang digunakan pada aplikasi Trahoola adalah teknologi GPS dengan tipe A-GPS. Teknologi ini akan menentukan posisi *smartphone* dengan akurasi yang tinggi. Alasan akurasi yang tinggi tersebut menjadi dasar peneliti memilih A-GPS sebagai GPS yang digunakan pada aplikasi yang dibangun. Pemanfaatan A-GPS pada aplikasi ini berguna untuk mendapatkan lokasi *smartphone* lebih cepat karena didukung oleh operator telekomunikasi yang digunakan pada *smartphone*. Penggunaan A-GPS ini tentu melibatkan biaya berlebih karena menggunakan koneksi internet untuk menggunakannya. Adapun alur untuk memanfaatkan A-GPS pada *smartphone* android guna memaksimalkan fungsionalitas pada aplikasi yang dibangun dapat dilihat pada Gambar berikut :



Gambar 3. 5 Alur Pemanfaatan A-GPS Pada *Smartphone* Android

3.7.2 Google *Maps* Android API

API (Application Programming Interface) yang dibesut oleh Google akan digunakan di dalam sistem yang dibangun. Kegunaan API ini adalah untuk mendapatkan layanan peta digital yang akan dipakai pada aplikasi. Pada API ini, layanan yang dapat digunakan selain menampilkan peta digital di dalam aplikasi adalah penggunaan *marker*, *poligon*, dan *overlay* ke peta dasar, serta mengubah tampilan area peta tertentu ke pengguna. Semua objek ini memberikan informasi tambahan tentang lokasi peta, dan memungkinkan interaksi pengguna dengan peta. Fitur-fitur tersebut digunakan oleh peneliti guna mendukung kebutuhan aplikasi.

Teknologi *Google Maps Android API* akan berguna pada pembangunan sistem dalam fungsional yang berkaitan dengan lokasi suatu objek di dalam peta digital. Teknologi ini senantiasa digunakan oleh peneliti untuk mendukung fungsionalitas sistem. Adapun pemanfaatan Google Maps Android API dapat dilihat pada Gambar 3.9 berikut :



Gambar 3. 6 Penggunaan Google *Maps* Android Pada Aplikasi Trahoola

Penggunaan Google *Maps* Android API pada sebuah aplikasi yang dibangun dilakukan dengan cara mendapatkan kunci API yang akan disematkan pada *file google_maps_api.xml* di dalam *project* Trahoola. Adapun langkah-langkah pemanfaatan Google *Maps* Android API pada sebuah aplikasi Trahoola sebagai berikut :

1. **Mendapatkan Kunci Google *Maps* API**

   Kunci Google *Maps* API dibutuhkan sebagai akses menggunakan layanan Google *Maps*. Kunci ini akan disematkan pada *file* google_maps_api.xml pada project Trahoola. file *goole_maps_api.xml* tersedia setelah peneliti membuat *maps activivy* di dalam *project* Trahoola. Berikut adalah cara mendapatkan kunci Google *Maps* API :

1. Menyalin tautan yang ada pada *file goole_maps_api.xml* . tautan ini diawali dengan baris kode seperti berikut :

```
"https://console.developers.google.com/...."
```

2. Masuk pada tautan tersebut melalui browser.
3. Mengikuti petunjuk untuk membuat kunci API.
4. Menyalin Kunci API tersebut untuk digunakan pada *project* Trahoola.

5. Menggunakan Kunci API

Implementasi daripada kunci API yang telah didapatkan pada halaman *console developer* android adalah dengan menyematkan kunci tersebut pada *file google_maps_api.xml* seperti pada baris kode berikut :

```
<string name="google_maps_key"templateMergeStrategy
="preserve" translatable="false">
    KUNCI API
</string>
```

6. Google *Maps* Pada Aplikasi Trahoola

Google *Maps* pada aplikasi Trahoola dibuat pada sebuah *activity* yang dimana di dalamnya terdapat aktivitas map untuk sebuah fungsionalitas tertentu. Adapun contoh implementasi Google *Maps* Android API pada aplikasi Trahoola dapat dilihat pada Tabel 3.1 berikut :

Tabel 3. 1 Baris Kode Google Maps pada Aplikasi Trahoola

1	**Public class** FragmentLocation **extends** Fragment **implements** OnMapReadyCallback, DirectionFinderListener {
2	
3	private GoogleMap mMap;
4	MapView **mapView;**
5	public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {
6	

```
7       view = inflater.inflate(R.layout.fragment_location,container,false);
8
9           Bundle args3 = getArguments();
10          tujuan = (String) args3.get("EXTRAS_ALAMAT");
11          nama= (String) args3.get("EXTRAS_NAMA");
12
13           goDirection();
14      return view;
15      }
16
17      @Override
18      public void onViewCreated(View view, @Nullable Bundle savedInstanceState) {
19          super.onViewCreated(view, savedInstanceState);
20
21          mapView = (MapView)view.findViewById(R.id.map);
22          if (mapView!=null){
23              mapView.onCreate(null);
24              mapView.onResume();
25              mapView.getMapAsync(this);
26          }
27      }
28
29      @Override
30      public void onMapReady(GoogleMap googleMap) {
31          mMap = googleMap;
32          LatLng hcmus = new LatLng(lat, lng);
33          mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(hcmus, 18));
34          originMarkers.add(mMap.addMarker(new MarkerOptions()
35              .title(nama)
36              .position(hcmus)));
37      }
```

Adapun penjelasan dari setiap langkah baris kode pada Tabel 3.1 dapat dilihat pada Tabel 3.2 berikut :

Tabel 3. 2 Penjelasan Kode Google *Maps* di Aplikasi Trahoola

Baris ke	Penjelasan
1	Merupakan nama *class Fragment Location* untuk halaman lokasi wisata yang mewarisi *Fragment* dan mengimplementasikan *OnMapReadyCallback* dan *DirectionFinderListener*
3-4	Pendefinisian variabel *GoogleMap* dan *MapView*
5-7	Pembuatan *method onCreateView* yang akan membuat suatu *Fragment* tersebut diciptakan dan juga untuk mengidentifikasi *layout* untuk *fragment* tersebut.
9-11	Mendapatkan *value* alamat dan nama wisata dari *activity* sebelumnya yang dipilih.
18-27	*Method onViewCreated*
30-37	*Method* untuk pembuatan *maps* pada aplikasi dengan mendefinisikan titik koordinat serta parameter-parameter di dalamnya seperti judul dan *marker* titik koordinat.

3.7.3 Google *Maps Direction* API

Selain *Google Maps Android API,* teknologi API besutan Google yang digunakan pada aplikasi yang dibangun adalah teknologi *Google Maps Direction API*. Teknologi API ini akan memberikan layanan berupa penentuan rute dari sebuah titik lokasi menuju titik lokasi lain. *Google Maps Direction API* ini digunakan untuk penentuan rute terhadap lokasi pengguna aplikasi menuju lokasi tempat wisata. Untuk menggunakan layanan Google *Maps Direction* API ini dilakukan dengan cara mengakses API melalui antarmuka HTTP dan permintaan (*request*) dibuat dengan *string* URL seperti baris kode berikut :

```
https://maps.googleapis.com/maps/api/directions/jso
n?origin=LokasiAwal&destination=LokasiTujuan
&key=KUNCI_API
```

Tabel 3. 3 Google Maps Direction pada Aplikasi Trahoola

```
1   private void sendRequest() {
2           String origin =
ed_position.getText().toString();
3
4           if (origin.isEmpty()) {
5               Toast.makeText(getActivity(), "Please
enter Your Position!", Toast.LENGTH_SHORT).show();
6               return;
7           }
8           String destination = tujuan;
9
10          try {
11              new DirectionFinder(this, origin,
destination).execute();
12          } catch (UnsupportedEncodingException e) {
13              e.printStackTrace();
14          }
15      }
16
17      @Override
18      public void onDirectionFinderStart() {
19          progressDialog =
ProgressDialog.show(getActivity(), "Please wait.",
20                  "Finding direction..!", true);
21
22          if (originMarkers != null) {
23              for (Marker marker : originMarkers) {
24                  marker.remove();
25              }
```

```
26          }
27
28          if (destinationMarkers != null) {
29              for (Marker marker :
destinationMarkers) {
30                  marker.remove();
31              }
32          }
33
34   if (polylinePaths != null) {
35              for (Polyline polyline:polylinePaths )
{
36                  polyline.remove();
37              }
38          }
39      }
```

3.7.4 Google *Places* API

Teknologi selanjutnya yang dimanfaatkan dalam aplikasi yang dibangun adalah *Google Places API*. *Google Places API* dimanfaatkan untuk mendapatkan permintaan mengenai informasi tempat-tempat yang ada di sekitar lokasi yang kita inisialisasi. Teknologi ini dimanfaatkan peneliti untuk menemukan tempat-tempat tujuan wisata sebagai salah satu fitur yang terdapat pada aplikasi yang dibangun.

Selain itu juga, *Google Places API* digunakan untuk kebutuhan fitur *autocomplete* dari sebuah *field input* lokasi yang digunakan pada aplikasi yang dibangun. *Auto complete* berguna untuk memudahkan pengguna menemukan saran dari alamat yang diketikkan. Ini akan meminimalisir kesalahan pengetikan oleh pengguna. Adapun contoh baris kode *Google Places API* pada aplikasi Trahoola dapat dilihat pada Tabel 3.4 berikut :

Tabel 3. 4 Baris Kode Google Places API pada Aplikasi Trahoola

1	`public void findPlace(View view) {`
2	`    try {`
3	`        Intent intent = new PlaceAutocomplete.IntentBuilder (PlaceAutocomplete.MODE_FULLSCREEN)`
4	`                .setFilter(typeFilter)`
5	`                .build(this);`
6	`        startActivityForResult(intent, PLACE_AUTOCOMPLETE_REQUEST_CODE);`
7	`    } catch (GooglePlayServicesRepairableException e) {`
8	`        // TODO: Handle the error.`
9	`    } catch (GooglePlayServicesNotAvailableException e) {`
10	`        // TODO: Handle the error.`
11	`    }`
12	`}`
13	
14	
15	`@Override`
16	`protected void onActivityResult(int requestCode, int resultCode, Intent data) {`
17	`    if (requestCode == PLACE_AUTOCOMPLETE_REQUEST_CODE) {`
18	`        if (resultCode == RESULT_OK) {`
19	`            Place place = PlaceAutocomplete.getPlace(this, data);`
20	`        Origin = place.getName();`
21	`        } else if (resultCode == PlaceAutocomplete.RESULT_ERROR) {`
22	`            Status status = PlaceAutocomplete.getStatus(this, data);`
23	
24	`            Log.i(TAG, status.getStatusMessage());`
25	

26	} else if (resultCode == RESULT_CANCELED) {
27	Et_origin.setText(null);
28	}
29	}
30	}

Adapun penjelasan dari setiap langkah baris kode pada Tabel 3.3 dapat dilihat pada Tabel 3.4 berikut :

Tabel 3. 5 Penjelasan Baris Kode Google Places API

Baris ke	Penjelasan
1	Merupakan *method findPlace()* untuk menemukan tempat
2-12	Fungsi *try and catch* untuk mendapatkan tempat
3-7	Merupakan fungsi untuk membuat *autocomplete*
16	*Method* untuk menampilkan hasil
17-22	Merupakan fungsi mendapatkan hasil *autocomplete* dan menampilkannya dalam *command line*
23-27	Fungsi untuk menangani kesalahan yang mungkin terjadi

3.8 Analisis Spesifikasi Kebutuhan Perangkat Lunak

Spesifikasi kebutuhan perangkat lunak dari sistem yang dibangun terbagi menjadi dua yaitu spesifikasi kebutuhan fungsional dan spesifikasi kebutuhan non fungsional. Spesifikasi kebutuhan fungsional akan menjabarkan layanan sistem yang harus disediakan. Adapun spesifikasi kebutuhan fungsional dapat dilihat pada Tabel 3.30 berikut :

Tabel 3. 6 Spesifikasi Kebutuhan Fungsional

Kode SKPL	Spesifikasi Kebutuhan Fungsional
SKPL-F-001	Sistem *mobile* menyediakan fasilitas bagi pengguna untuk daftar ke dalam sistem.
SKPL-F-002	Sistem *mobile* menyediakan fasilitas bagi pengguna untuk lupa *password*.

Kode SKPL	Spesifikasi Kebutuhan Fungsional
SKPL-F-003	Sistem *mobile* menyediakan fasilitas bagi pengguna untuk *login* ke dalam sistem.
SKPL-F-004	Sistem *mobile* menyediakan fasilitas bagi pengguna untuk menampilkan masukkan anggaran.
SKPL-F-005	Sistem *mobile* menyediakan fasilitas bagi pengguna untuk menampilkan masukkan tujuan wisata.
SKPL-F-006	Sistem *mobile* menyediakan fasilitas bagi pengguna untuk menampilkan rekomendasi perjalanan yang dibutuhkan.
SKPL-F-007	Sistem *mobile* menyediakan fasilitas bagi pengguna untuk menampilkan detail rekomendasi perjalanan.
SKPL-F-008	Sistem *mobile* menyediakan fasilitas bagi pengguna untuk menampilkan info detail harga perjalanan.
SKPL-F-009	Sistem *mobile* menyediakan fasilitas bagi pengguna untuk menampilkan *history* perjalanan wisata.
SKPL-F-010	Sistem *mobile* menyediakan fasilitas bagi pengguna untuk pencarian mengenai perjalanan wisata.

Selain spesifikasi kebutuhan perangkat lunak, terdapat spesifikasi kebutuhan non fungsional. Spesifikasi kebutuhan non fungsional akan memberikan batasan layanan yang ditawarkan sistem seperti batasan waktu, batasan pengembangan proses standarisasi dan lain-lain. Adapun spesifikasi kebutuhan non fungsional dapat dilihat pada Tabel 3.7 berikut :

Tabel 3. 7 Spesifikasi Kebutuhan Non Fungsional

Kode SKPL	Spesifikasi Kebutuhan Perangkat Lunak
SKPL-NF-001	Sistem yang dibangun menggunakan sistem *mobile*
SKPL-NF-002	Sistem yang dibangun minimal menggunakan sistem operasi android versi 5.0 Lollipop
SKPL-NF-003	Sistem yang dibangun memanfaatkan fungsi GPS
SKPL-NF-004	Sistem dibangun dengan spesifikasi *hardware* yang memenuhi standar minimum kebutuhan

3.9 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional adalah analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan sistem baik itu kebutuhan perangkat keras, perangkat lunak maupun perangkat pikir (pengguna). Adapun kebutuhan non fungsional adalah sebagai berikut :

3.9.1 Analisis Kebutuhan Perangkat Keras

Analisis kebutuhan perangkat keras menunjukkan kebutuhan perangkat keras yang direkomendasikan pada tahap implementasi. Kebutuhan perangkat keras memiliki spesifikasi minimum seperti pada Tabel 3.26 berikut :

Tabel 3. 8 Kebutuhan Perangkat Keras

Sub Sistem Web	Sub Sistem Mobile
Prosesor @2.0 Ghz	*Smartphone* Android 5.0 *Lollipop*
HDD 500GB	RAM 1 GB
RAM DDR3 2GB	TFT LCD
VGA 1 GB	Jaringan GSM / CDMA
Resolusi Layar 1366x768 piksel	Data Koneksi HSDPA / EVDO
Koneksi Internet	GPS A-GPS

3.9.2 Analisis Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak menunjukkan kebutuhan perangkat lunak yang direkomendasikan pada tahap implementasi. Kebutuhan perangkat lunak memiliki spesifikasi minimum seperti pada Tabel 3.9 berikut :

Tabel 3. 9 Kebutuhan Perangkat lunak

Sub Sistem Web	Sub Sistem Mobile
Sistem Operasi Windows 7 atau diatasnya	Sistem Operasi Android 5.0 *Lollipop*
Browser Google Chrome, Firefox, Ms. Edge	
XAMPP v3.2.2	

3.9.3 Analisis Kebutuhan Pengguna

Analisis kebutuhan pengguna menunjukkan kebutuhan minimum karakteristik pengguna dalam mengoperasikan sistem yang dibangun. Adapun Kebutuhan pengguna untuk dapat mengoperasikan sistem ini dapat dilihat pada Tabel 3.10 berikut :

Tabel 3. 10 Kebutuhan Pengguna

No	Pengguna	Karakteristik
1	Administrator	1. Mampu mengoperasikan komputer dengan baik 2. Memiliki kemampuan mengolah *database*
2	Pengguna	1. Mampu mengoperasikan *smartphone* Android 2. Memahami bahasa Indonesia dan mengerti kata-kata dasar bahasa inggris

3.9.4 Analisis Kebutuhan Fungsional

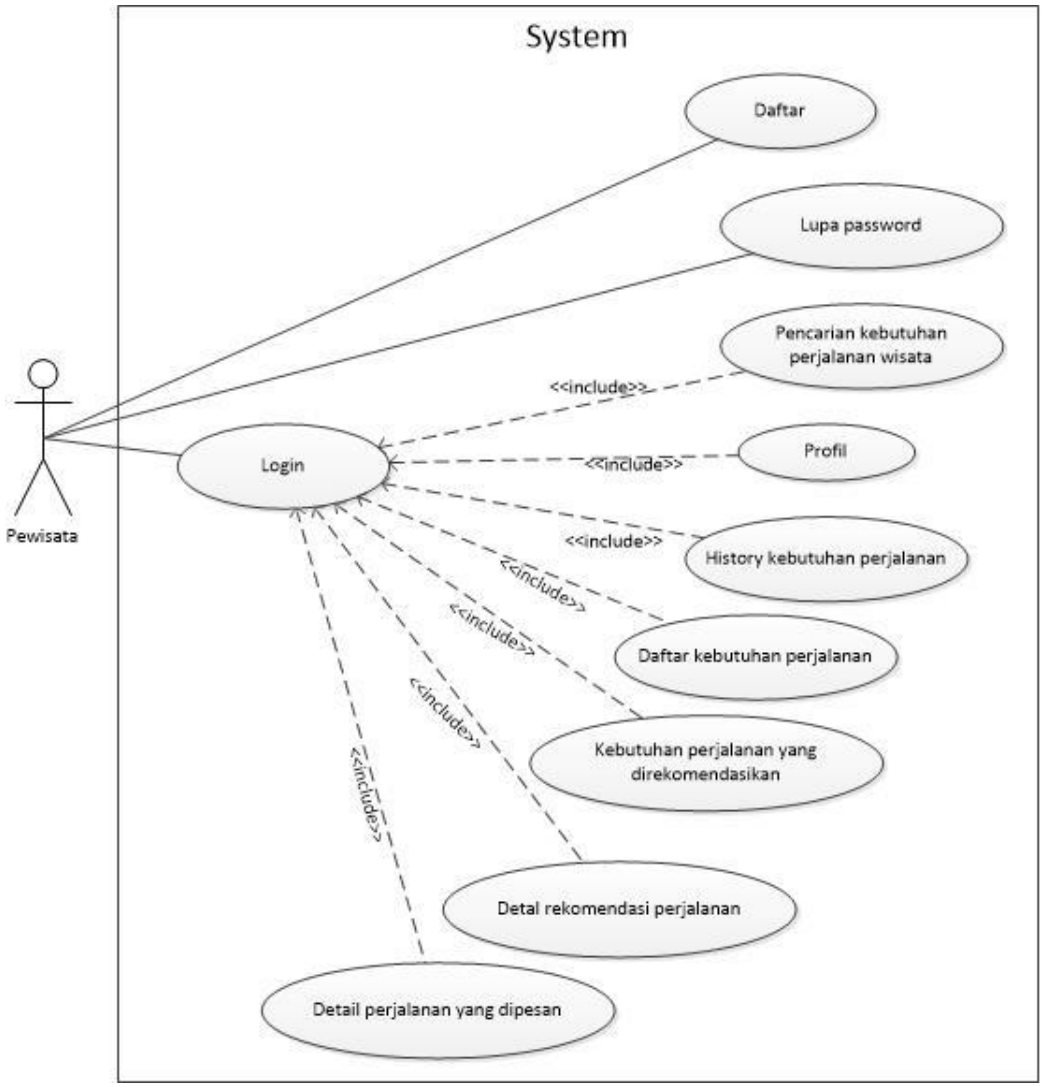
Analisis kebutuhan fungsional dapat didefinisikan sebagai perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satuan yang utuh dan berfungsi. Analisis kebutuhan fungsional pada sistem yang dibangun meliputi dua *platform* yaitu sistem *web* dan *mobile*. Pada platform web, analisis kebutuhan fungsional digambarkan dengan DFD (*Data Flow Diagram*) sedangkan analisis kebutuhan fungsional pada platform *mobile* digambarkan dengan Diagram UML (*Unified Modelling Language).*

3.10 Analisis Kebutuhan Fungsional

Metode analisis yang digunakan pada pembangunan sistem *mobile* adalah *OOAD (Object Oriented Analysis And Design).* Melalui Metode *OOAD* ini kebutuhan fungsional dilakukan dengan menggunakan diagram *UML*. Terdapat beberapa diagram pada *tools UML* yang umum digunakan untuk menggambarkan fungsional sistem antara lain *Use Case Diagram, Activity Diagram, Sequence Diagram dan Class Diagram.*

3.10.1 *Use Case* Diagram

Use Case Diagram digunakan untuk menggambarkan interaksi antara aktor dengan aktifitas yang terdapat pada sistem yang dibangun. Dengan menggambarkan *Use Case* Diagram, fungsional-fungsional yang terdapat pada sistem dapat dilihat secara singkat. Adapun *Use Case* Diagram pada sistem *mobile* yang dibangun dapat dilihat pada Gambar 3.7 berikut :



Gambar 3. 7 Use Case Diagram

3.10.2 Identifikasi Aktor

Identifikasi aktor digunakan untuk menjabarkan deskripsi peran dari sebuah aktor pada *use case diagram.* Adapun definisi aktor pada *use case* diagram sistem *mobile* dapat dilihat pada Tabel 3.11 berikut :

Tabel 3. 11 Definisi Aktor

No	Actor	Deskripsi
1	Pewisata	Aktor dengan peran ini memiliki tugas untuk menggunakan fungsionalitas pada level *user interface*.

3.10.3 Identifikasi *Use Case*

Identifikasi *use case* digunakan untuk menjabarkan deskripsi dari sebuah *use case* pada *use case* diagram beserta aktor yang melakukannya. Adapun definisi *use case* pada *use case* diagram sistem *mobile* dapat dilihat pada Tabel 3.12 berikut :

Tabel 3. 12 Definisi Use Case

No	*Use Case*	Deskripsi
1	Daftar	Sistem menampilkan *form* daftar untuk pengguna yang terdiri dari nama lengkap, *email*, *password*, dan ulangi *password*.
2	*Login*	Sistem menampilkan *form login* untuk pengguna yang terdiri dari *email* dan *password.*
3	Lupa *Password*	Sistem menampilkan masukkan lupa *password* untuk pengguna yang terdiri dari masukkan *email.*
4	Daftar kebutuhan perjalanan	Sistem menampilkan daftar dari perjalanan yang dibutuhkan yaitu jenis transportasi, penginapan dan tujuan wisata.
5	Masukkan anggaran	Sistem menampilkan masukkan anggaran untuk pengguna yang akan melakukan pemesanan.
6	Masukkan jumlah orang	Sistem menampilkan masukkan jumlah orang untuk pengguna yang akan melakukan perjalanan.
7	Masukkan tujuan wisata	Sistem menampilkan masukkan tujuan wisata untuk pengguna yang akan melakukan perjalanan.
8	Profil	Sistem menampilkan info profil dari pengguna.

No	*Use Case*	Deskripsi
9	Riwayat pemesanan	Sistem menampilkan detail rekomendasi perjalanan untuk pengguna.
10	Halaman *Logout*	Sistem menampilkan detail perjalanan yang dipesan oleh pengguna yang akan melakukan perjalanan.

3.10.4 *Use Case* Skenario

Use case skenario menjelaskan bagaimana skenario dari setiap proses yang ada pada setiap *use case*. Pada skenario *use case* ini dapat dijabarkan bagaimana alur dari setiap proses pada setiap *use case* berdasarkan interaksi antara aktor dan sistem. Adapun skenario *use case* untuk setiap *use case* sebagai berikut :

a. Skenario *Use Case* Daftar

Berikut adalah Tabel 3.13 *Requirement* A.1 yang menjelaskan kebutuhan dari *use case* daftar dan Tabel 3.14 Skenario *Use Case* Daftar yang menjelaskan skenario dari *use case* daftar.

Tabel 3. 13 Requirement A.1

Requirement A.1
Data Daftar yang dimasukan pengguna terdiri dari nama lengkap, *email*, *password*, dan ulangi *password*.

Tabel 3. 14 Skenario Use Case Daftar

Use Case Name	Daftar	
Related Requirements	Requirement A.1	
Goal Context	Pengguna baru berhasil melakukan daftar dalam system	
Preconditions	Form Daftar untuk pengguna baru ditampilkan	
Successful End Condition	Menampilkan halaman *login* perangkat lunak	
Failed End Condition	Menampilkan pesan kesalahan daftar	
Primary Actor	Pengguna	
Trigger	Pengguna memilih menu daftar	
Main Flow	Step	Action
	1	Pengguna memilih menu Daftar
	2	Pengguna mengisi data sesuai requirment A.1
	3	Sistem melakukan validasi form masukan pengguna
	4	Sistem melakukan pemeriksaan data akun baru
	5	Sistem menampilkan *activity* utama perangkat lunak
Extension	Step	Branching Action

	4.1	Nama lengkap, email, password, dan ulangi password
	4.2	Nama lengkap, email, password, dan ulangi password terisi

b. Skenario *Use Case* Lupa *Password*

Berikut adalah Tabel 3.15 *Requirement* A.2 yang menjelaskan kebutuhan dari *use case* lupa *password* dan Tabel 3.16 Skenario *Use Case* Lupa *Password* yang menjelaskan skenario dari *use case* lupa *password*.

Tabel 3. 15 Requirement A.2

Requirement A.2
Data Lupa Password yang dimasukan pengguna terdiri email..

Tabel 3. 16 Skenario Use Case Lupa Password

Use Case Name	Lupa Password	
Related Requirements	Requirement A.2	
Goal Context	Pengguna berhasil melakukan lupa password dalam system	
Preconditions	Form Lupa Password untuk pengguna ditampilkan	
Successful End Condition	Menampilkan activity utama perangkat lunak	
Failed End Condition	Menampilkan pesan kesalahan lupa password	
Primary Actor	Pengguna	
Trigger	Pengguna memilih menu lupa password	
Main Flow	Step	Action
	1	Pengguna memilih menu lupa password
	2	Pengguna mengisi data sesuai requipment A.2
	3	Pengguna memilih menu selesai/kirim
	4	Sistem melakukan validasi form masukan pengguna
	5	Sistem melakukan pemeriksaan data pengguna
	6	Sistem menampilkan activity utama perangkat lunak
Extension	Step	Branching Action
	4.1	Emai kosong
	4.2	email terisi
	5.1	Email yang dimasukkan terdaftar
	5.2	Email yang dimasukkan belum terdaftar

c. Skenario *Use Case Login*

Berikut adalah Tabel 3.17 *Requirement* A.3 yang menjelaskan kebutuhan dari *use case login* dan Tabel 3.18 Skenario *Use Case Login* yang menjelaskan skenario dari *use case login*.

Tabel 3. 17 *Requirement* A.3

Requirement A.3
Data *Login* yang dimasukkan pengguna terdiri *email* dan *password*

Tabel 3. 18 Skenario *Use Case* Login

Use Case Name	*Login*	
Related Requirements	*Requirement* A.3	
Goal Context	Pengguna berhasil melakukan *login* dalam *system*	
Preconditions	*Form Login* untuk pengguna ditampilkan	
Successful End Condition	Menampilkan *activity* utama perangkat lunak	
Failed End Condition	Menampilkan pesan kesalahan *login*	
Primary Actor	Pengguna	
Trigger	Pengguna memilih menu *login*	
Main Flow	Step	Action
	1	Pengguna mengisi data sesuai *requipment* A.3
	2	Pengguna memilih menu masuk
	3	Sistem melakukan validasi *form* masukan pengguna
	4	Sistem melakukan pemeriksaan data pengguna
	5	Sistem menampilkan *activity* utama perangkat lunak
Extension	Step	Branching Action
	3.1	Email dan *password* kosong
	3.2	email dan *password* terisi
	4.1	Email dan *password* yang dimasukkan benar
	4.2	Email dan *password* yang dimasukkan salah

d. Skenario *Use Case* Daftar Kebutuhan Perjalanan

Berikut adalah Tabel 3.19 Skenario *Use Case* Daftar Kebutuhan Perjalanan menjelaskan skenario dari *use case* daftar kebutuhan perjalanan.

Tabel 3. 19 Skenario *Use Case* Daftar Kebutuhan Perjalanan

Use Case Name	Daftar Kebutuhan Perjalanan	
Related Requirements	*Requirement* A.3	
Goal Context	Pengguna baru berhasil melakukan daftar kebutuhan perjalanan	
Preconditions	Menu daftar kebutuhan perjalanan untuk pengguna ditampilkan	
Successful End Condition	Menampilkan halaman daftar kebutuhan perjalanan	
Failed End Condition	Menampilkan pesan kesalahan *login*	
Primary Actor	Pengguna	
Trigger	Pengguna memilih menu *login*	
Main Flow	Step	Action
	1	Pengguna mengisi data sesuai requipment A.3
	2	Pengguna memilih menu masuk
	3	Sistem melakukan validasi *form* masukan pengguna
	4	Sistem melakukan pemeriksaan data pengguna
	5	Sistem menampilkan menu daftar kebutuhan perjalanan
Extension	Step	Branching Action
	3.1	Email dan *password* kosong
	3.2	Email dan *password* terisi

	4.1	Email dan *password* yang dimasukkan benar
	4.2	Email dan *password* yang dimasukkan salah

e. Skenario *Use Case* Masukkan Anggaran

Berikut adalah Tabel I3.20 *Requirement* A.4 yang menjelaskan kebutuhan dari *use case* masukkan anggaran dan Tabel 3.21 Skenario *Use Case* Masukkan Anggaran yang menjelaskan skenario dari *use case* masukkan anggaran.

Tabel 3. 20 *Requirement* A.4

Requirement A.4
Data Masukkan Anggaran yang dimasukkan pengguna terdiri anggaran yang dimliki.

Tabel 3. 21 Skenario *Use Case* Masukkan Anggaran

Use Case Name	Masukkan Anggaran	
Related Requirements	*Requirement* A.4	
Goal Context	Pengguna baru berhasil melakukan masukkan anggaran ke sistem	
Preconditions	*Form* masukkan anggaran untuk pengguna ditampilkan	
Successful End Condition	Menampilkan isian masukkan anggaran perangkat lunak	
Failed End Condition	Menampilkan pesan anggaran tidak boleh kosong	
Primary Actor	Pengguna	
Trigger	Pengguna memilih menu masukkan anggaran	
Main Flow	Step	Action
	1	Pengguna mengisi data sesuai *requipment* A.3
	2	Pengguna memilih menu masuk
	3	Sistem melakukan validasi *form* masukkan pengguna
	4	Sistem melakukan pemeriksaan data pengguna
	5	Sistem menampilkan menu daftar kebutuhan perjalanan
	6	Pengguna memilih menu lihat rekomendasi
	7	Pengguna memasukkan anggaran
Extension	Step	Branching Action
	3.1	Email dan *password* kosong
	3.2	Email dan *password* terisi
	4.1	Email dan *password* yang dimasukkan benar
	4.2	Email dan *password* yang dimasukkan salah
	7.1	Anggaran tidak boleh kosong

f. Skenario *Use Case* Masukkan Jumlah Orang

Berikut adalah Tabel 3.22 *Requirement* A.5 yang menjelaskan kebutuhan dari *use case* masukkan jumlah undangan dan Tabel 3.23 Skenario *Use Case* Masukkan Jumlah Orang yang menjelaskan skenario dari *use case* masukkan jumlah orang.

Tabel 3. 22 *Requirement* A.5

Requirement A.5
Data Masukkan Jumlah Orang yang dimasukkan pengguna terdiri jumlah undangan yang diinginkan.

Tabel 3. 23 Skenario *Use Case* Masukkan Jumlah Orang

Use Case Name	Masukkan Jumlah Orang	
Related Requirements	*Requirement* A.5	
Goal Context	Pengguna baru berhasil melakukan masukkan jumlah orang ke sistem	
Preconditions	*Form* masukkan jumlah orang untuk pengguna ditampilkan	
Successful End Condition	Menampilkan isian masukkan jumlah orang perangkat lunak	
Failed End Condition	Menampilkan pesan jumlah orang tidak boleh kosong	
Primary Actor	Pengguna	
Trigger	Pengguna memilih menu masukkan jumlah orang	
Main Flow	Step	Action
	1	Pengguna mengisi data sesuai *requipment* A.3
	2	Pengguna memilih menu masuk
	3	Sistem melakukan validasi *form* masukkan pengguna
	4	Sistem melakukan pemeriksaan data pengguna
	5	Sistem menampilkan menu daftar kebutuhan perjalanan
	6	Pengguna memilih menu lihat rekomendasi
	7	Pengguna masukkan jumlah orang
Extension	Step	Branching Action
	3.1	Email dan *password* kosong
	3.2	Email dan *password* terisi
	4.1	Email dan *password* yang dimasukkan benar
	4.2	Email dan *password* yang dimasukkan salah
	7.1	Jumlah orang tidak boleh kosong

g. Skenario *Use Case* Masukkan Tujuan Wisata

Berikut adalah Tabel 3.24 *Requirement* A.6 yang menjelaskan kebutuhan dari *use case* masukkan tujuan wisata dan Masukkan Tujuan Wisata yang menjelaskan skenario dari *use case* masukkan tujuan wisata.

Tabel 3. 24 *Requirement* A.6

Requirement A.6
Data Masukkan Tujuan Wisata yang dimasukan pengguna terdiri dari tujuan wisata yang diinginkan.

Tabel 3. 25 Skenario *Use Case* Pilih Tujuan Wisata

Use Case Name	Pilih Jenis Paket	
Related Requirements	*Requirement* A.6	
Goal Context	Pengguna baru berhasil melakukan pilih tujuan wisata ke sistem	
Preconditions	*Form* pilih tujuan wisata untuk pengguna ditampilkan	
Successful End Condition	Menampilkan isian pilih tujuan wisata pada perangkat lunak	
Failed End Condition	Menampilkan pesan pilih tujuan wisata harus dipilih	
Primary Actor	Pengguna	
Trigger	Pengguna memilih menu pilih tujuan wisata	
Main Flow	Step	Action
	1	Pengguna mengisi data sesuai *requipment* A.3
	2	Pengguna memilih menu masuk
	3	Sistem melakukan validasi *form* masukan pengguna
	4	Sistem melakukan pemeriksaan data pengguna
	5	Sistem menampilkan menu daftar kebutuhan wisata
	6	Pengguna memilih menu lihat rekomendasi
	7	Pengguna melakukan pilih tujuan wisata
Extension	Step	Branching Action
	3.1	Email dan *password* kosong
	3.2	Email dan *password* terisi
	4.1	Email dan *password* yang dimasukkan benar
	4.2	Email dan *password* yang dimasukkan salah
	7.1	Pilih jenis paket harus dipilih

h. Skenario *Use Case* Menampilkan Barang Rekomendasi

Berikut adalah Tabel 3.26 *Requirement* A.7 yang menjelaskan kebutuhan dari *use case* menampilkan perjalanan yang direkomendasikan dan Tabel 3.27 Skenario *Use Case* Menampilkan Kebutuhan Rekomendasi Perjalanan yang menjelaskan skenario dari *use case* menampilkan perjalanan yang direkomendasikan.

Tabel 3. 26 *Requirement* A.7

Requirement A.7
Data Menampilkan Rekomendasi Perjalanan yang dimasukkan pengguna terdiri dari masukkan anggaran, masukkan jumlah orang, dan masukkan tujuan.

Tabel 3. 27 Skenario *Use Case* Menampilkan Rekomendasi Perjalanan

Use Case Name	Menampilkan Rekomendasi Perjalanan
Related Requirements	*Requirement* A.7
Goal Context	Pengguna berhasil melakukan proses lihat rekomendasi perjalanan
Preconditions	*Form* rekomendasi perjalanan untuk pengguna ditampilkan
Successful End Condition	Menampilkan halaman rekomendasi perjalanan pada perangkat lunak
Failed End Condition	Menampilkan pesan kesalahan masukkan anggaran, jumlah orang, dan masukkan tujuan harus diisi

Primary Actor	Pengguna	
Trigger	Pengguna memilih menu proses	
Main Flow	Step	Action
	1	Pengguna memilih menu lihat rekomendasi
	2	Pengguna mengisi data sesuai *requipment* A.7
	3	Sistem melakukan validasi *form* masukkan pengguna
	4	Sistem melakukan pemeriksaan data masukkan
	5	Sistem menampilkan halaman rekomendasi perjalanan
Extension	Step	Branching Action
	2.1	masukkan anggaran, jumlah orang, dan masukkan tujuan
	2.2	masukkan anggaran, jumlah orang, dan masukkan tujuan harus terisi

i. Skenario *Use Case* Menampilkan Detail Rekomendasi Perjalanan

Berikut adalah Tabel 3.28 Skenario *Use Case* Menampilkan Detail Rekomendasi Perjalanan yang menjelaskan skenario dari *use case* menampilkan detail rekomendasi perjalanan.

Tabel 3. 28 Skenario *Use Case* Menampilkan Detail Barang Rekomendasi

Use Case Name	Menampilkan Detail Rekomendasi Perjalanan	
Related Requirements	*Requirement* A.7	
Goal Context	Pengguna berhasil melakukan lihat detail rekomendasi perjalanan	
Preconditions	*Form* detail rekomendasi perjalanan untuk pengguna ditampilkan	
Successful End Condition	Menampilkan halaman detail rekomendasi perjalanan pada perangkat lunak	
Failed End Condition	Menampilkan pesan kesalahan masukkan anggaran, jumlah orang dan masukkan tujuan harus diisi	
Primary Actor	Pengguna	
Trigger	Pengguna memilih menu proses	
Main Flow	Step	Action
	1	Pengguna memilih menu lihat rekomendasi
	2	Pengguna mengisi data sesuai *requirement* A.7
	3	Sistem melakukan validasi *form* masukkan pengguna
	4	Sistem melakukan pemeriksaan data masukkan
	5	Sistem menampilkan halaman rekomendasi perjalanan
	6	Pengguna memilih detail rekomendasi perjalanan
Extension	Step	Branching Action
	2.1	masukkan anggaran, jumlah orang, dan masukkan tujuan
	2.2	masukkan anggaran, jumlah orang, dan masukkan tujuan harus terisi

j. Skenario *Use Case* Info Detail Perjalanan Yang Dipesan

Berikut adalah Tabel 3.29 Skenario *Use Case* Info Detail Perjalanan yang Dipesan menjelaskan skenario dari *use case* info detail perjalanan yang dipesan.

Tabel 3. 29 Skenario *Use Case* Info Detail Perjalanan Dipesan

Use Case Name	Menampilkan Detail Perjalanan Dipesan	
Related Requirements	*Requirement* A.7	
Goal Context	Pengguna berhasil melakukan lihat perjalanan dipesan	
Preconditions	*Form* detail perjalanan dipesan untuk pengguna ditampilkan	
Successful End Condition	Menampilkan halaman detail perjalanan dipesan pada perangkat lunak	
Failed End Condition	Menampilkan pesan kesalahan masukan anggaran, jumlah undangan, dan pilih jenis paket harus diisi	
Primary Actor	Pengguna	
Trigger	Pengguna memilih menu proses	
Main Flow	Step	Action
	1	Pengguna memilih menu lihat rekomendasi
	2	Pengguna mengisi data sesuai *requipment* A.7
	3	Sistem melakukan validasi *form* masukkan pengguna
	4	Sistem melakukan pemeriksaan data masukkan
	5	Sistem menampilkan halaman rekomendasi perjalanan
	6	Pengguna memilih detail rekomendasi perjalanan
	7	Pengguna memilih detail perjalanan dipesan
Extension	Step	Branching Action
	2.1	masukkan anggaran, jumlah orang, dan masukkan tujuan
	2.2	masukkan anggaran, jumlah orang, dan masukkan tujuan harus terisi

k. Skenario *Use Case* Profil

Berikut adalah Tabel 3.30 Skenario *Use Case* Profil yang menjelaskan skenario dari *use case* profil.

Tabel 3. 30 Skenario *Use Case* Profil

Use Case Name	Profil	
Related Requirements	*Requirement* A.3	
Goal Context	Pengguna berhasil melakukan lihat profil	
Preconditions	Menu profil untuk pengguna ditampilkan	
Successful End Condition	Menampilkan halaman profil untuk pengguna	
Failed End Condition	Menampilkan pesan kesalahan *login*	
Primary Actor	Pengguna	
Trigger	Pengguna memilih menu profil	
Main Flow	Step	Action
	1	Pengguna mengisi data sesuai *requipment* A.3
	2	Pengguna memilih menu masuk

	3	Sistem melakukan validasi *form* masukkan pengguna
	4	Sistem melakukan pemeriksaan data pengguna
	5	Sistem menampilkan menu profil
Extension	Step	Branching Action
	3.1	Email dan *password* kosong
	3.2	Email dan *password* terisi
	4.1	Email dan *password* yang dimasukkan benar
	4.2	Email dan *password* yang dimasukkan salah

l. Skenario *Use Case* Riwayat Kebutuhan Perjalanan

Berikut adalah Tabel 3.31 Skenario *Use Case* Riwayat Kebutuhan Perjalanan yang menjelaskan skenario dari *use case* riwayat kebutuhan perjalanan.

Tabel 3. 31 Skenario *Use Case* Riwayat Kebutuhan Perjalanan

Use Case Name	Riwayat Kebutuhan Perjalanan	
Related Requirements	*Requirement* A.3	
Goal Context	Pengguna berhasil melakukan lihat riwayat kebutuhan perjalanan	
Preconditions	Menu riwayat kebutuhan perjalanan untuk pengguna ditampilkan	
Successful End Condition	Menampilkan halaman riwayat kebutuhan perjalanan untuk pengguna	
Failed End Condition	Menampilkan pesan kesalahan *login*	
Primary Actor	Pengguna	
Trigger	Pengguna memilih menu riwayat kebutuhan perjalanan	
Main Flow	Step	Action
	1	Pengguna mengisi data sesuai *requirement* A.3
	2	Pengguna memilih menu masuk
	3	Sistem melakukan validasi *form* masukkan pengguna
	4	Sistem melakukan pemeriksaan data pengguna
	5	Sistem menampilkan menu riwayat kebutuhan perjalanan
Extension	Step	Branching Action
	3.1	Email dan *password* kosong
	3.2	Email dan *password* terisi
	4.1	Email dan *password* yang dimasukkan benar
	4.2	Email dan *password* yang dimasukkan salah

m. Skenario *Use Case* Pencarian Kebutuhan Perjalanan

Berikut adalah Tabel 3.32 Skenario *Use Case* Pencarian Kebutuhan Perjalanan yang menjelaskan skenario dari *use case* pencarian kebutuhan perjalanan.

Tabel 3. 32 Skenario *Use Case* Pencarian Kebutuhan Perjalanan

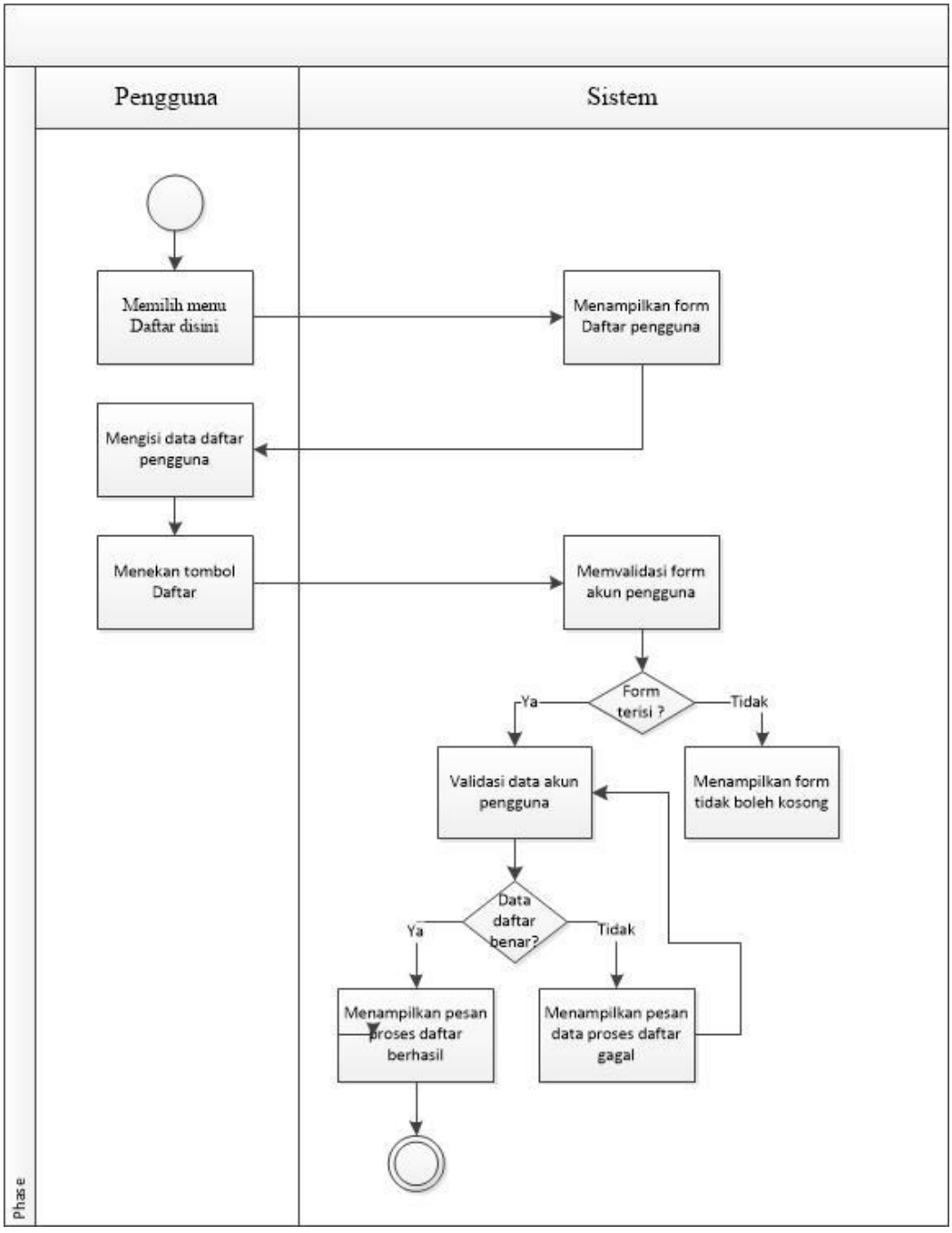
Use Case Name	Pencarian Kebutuhan Perjalanan	
Related Requirements	Requirement A.3	
Goal Context	Pengguna berhasil melakukan lihat pencarian kebutuhan perjalanan	
Preconditions	Menu riwayat kebutuhan perjalanan untuk pengguna ditampilkan	
Successful End Condition	Menampilkan halaman riwayat kebutuhan perjalanan untuk pengguna	
Failed End Condition	Menampilkan pesan kesalahan *login*	
Primary Actor	Pengguna	
Trigger	Pengguna memilih menu riwayat kebutuhan perjalanan	
Main Flow	Step	Action
	1	Pengguna mengisi data sesuai *requirement* A.3
	2	Pengguna memilih menu masuk
	3	Sistem melakukan validasi *form* masukkan pengguna
	4	Sistem melakukan pemeriksaan data pengguna
	5	Sistem menampilkan menu pencarian kebutuhan perjalanan
Extension	Step	Branching Action
	3.1	Email dan *password* kosong
	3.2	Email dan *password* terisi
	4.1	Email dan *password* yang dimasukkan benar
	4.2	Email dan *password* yang dimasukkan salah

3.10.5 *Activity* Diagram

Activity Diagram digunakan untuk menggambarkan alur proses dari skenario yang telah dirancang pada setiap *use case*. Dengan *acticity* diagram dapat terlihat interaksi antara aktor dengan sistem yang dibangun. Adapun beberapa *acitvity* diagram pada setiap *use case* sebagai berikut :

a) *Activity* Diagram Daftar

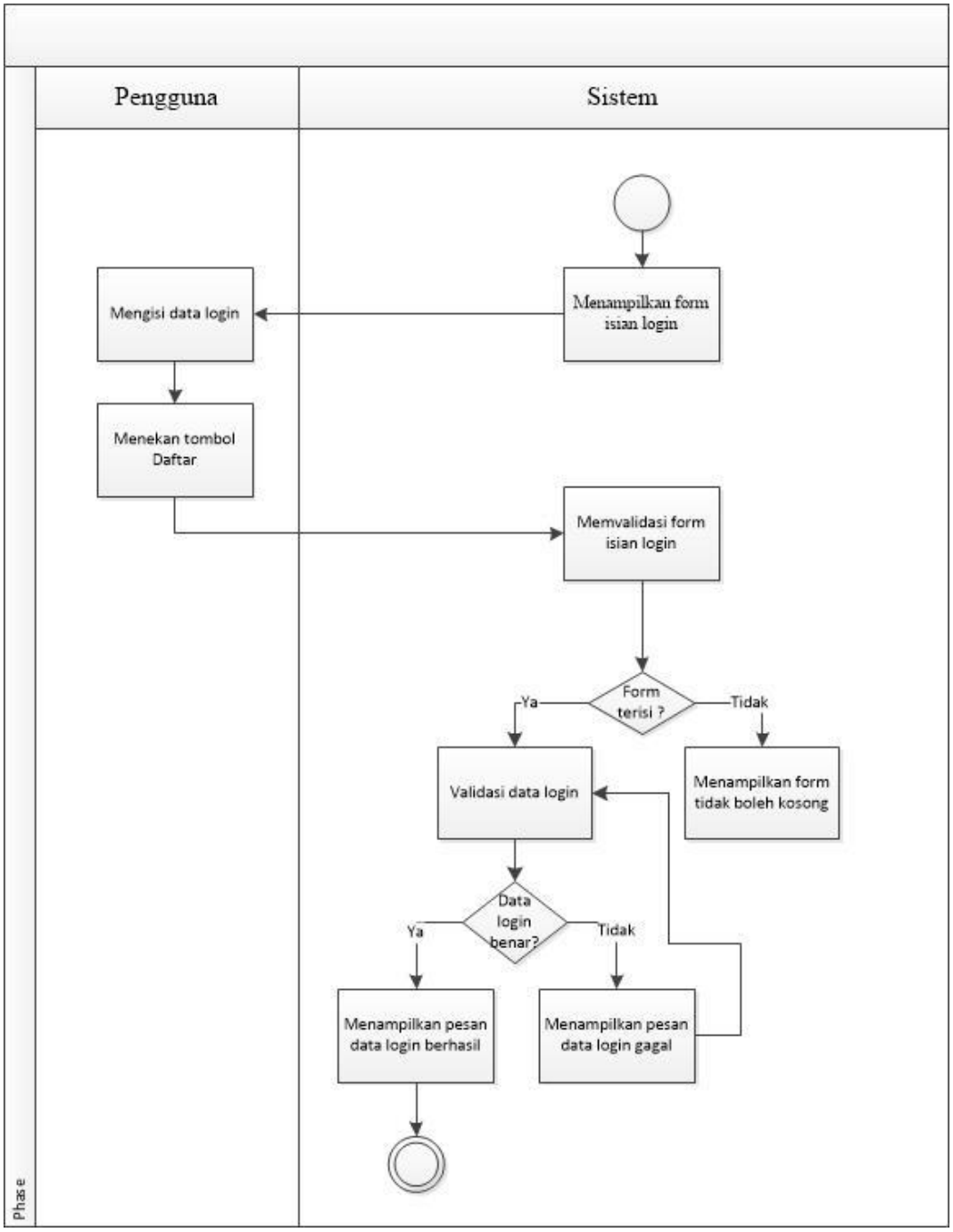
Berikut adalah Gambar 3.8 *Activity* Diagram Daftar yang menggambarkan *activity* diagram daftar.



Gambar 3.8 *Activity* Diagram Daftar

b) *Activity* Diagram Login

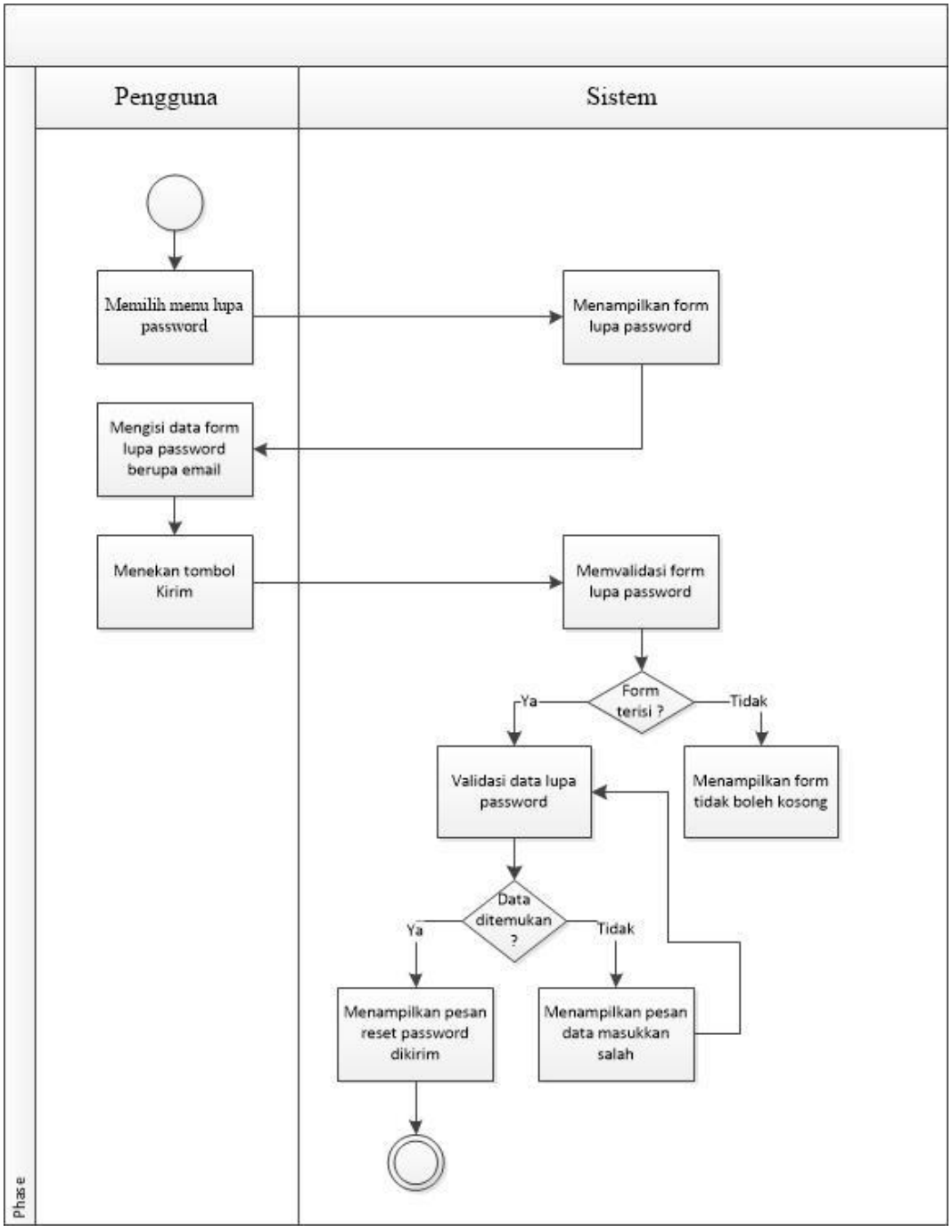
Berikut adalah Gambar 3.9 *Activity* Diagram *Login* yang menggambarkan *activity* diagram *login*.



Gambar 3.9 *Activity* Diagram *Login*

c) *Activity* Diagram Lupa *Password*

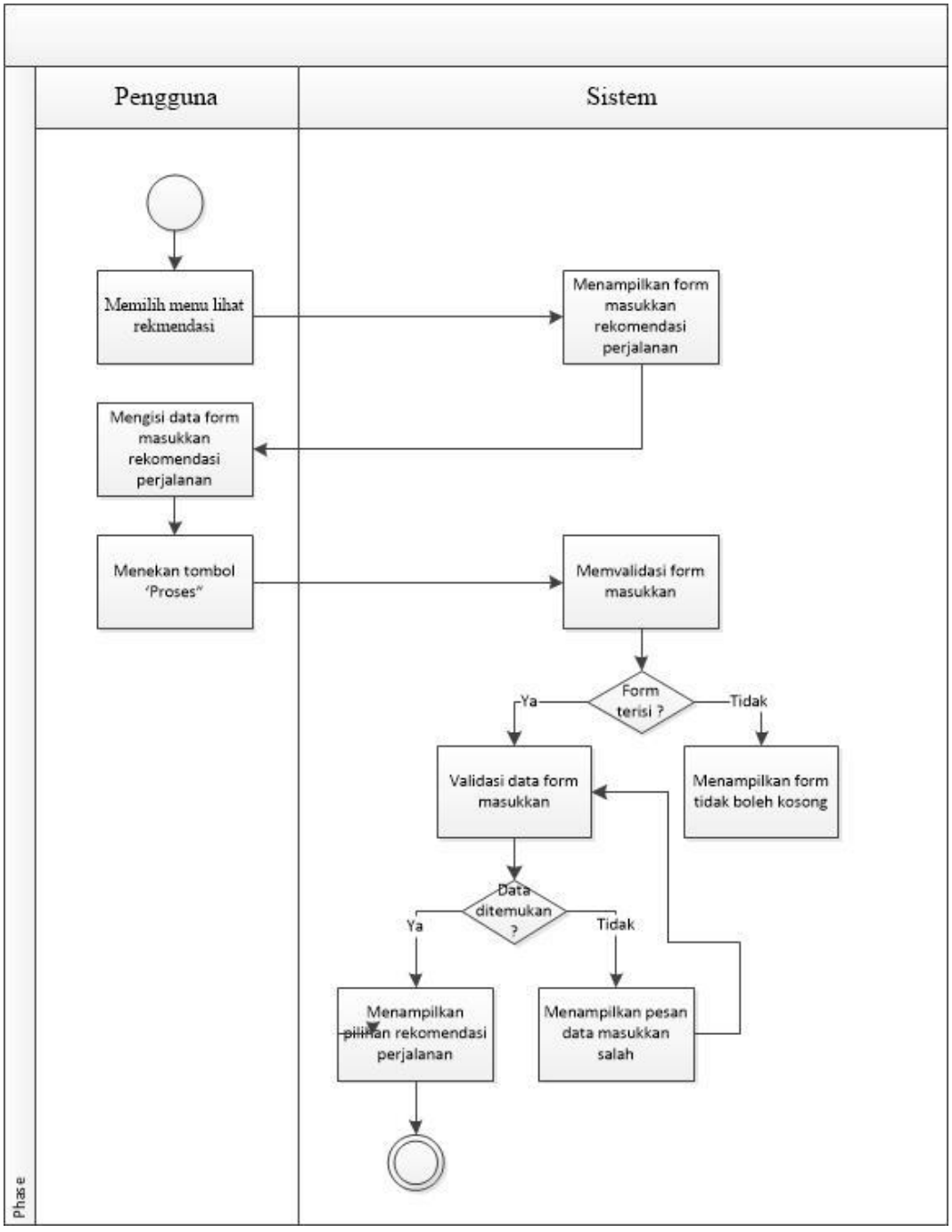
Berikut adalah Gambar 3.10 *Activity* Diagram Lupa *Password* yang menggambarkan *activity* diagram lupa *password*.



Gambar 3.10 *Activity* Diagram Lupa *Password*

d) *Activity* Diagram Daftar Kebutuhan Perjalanan

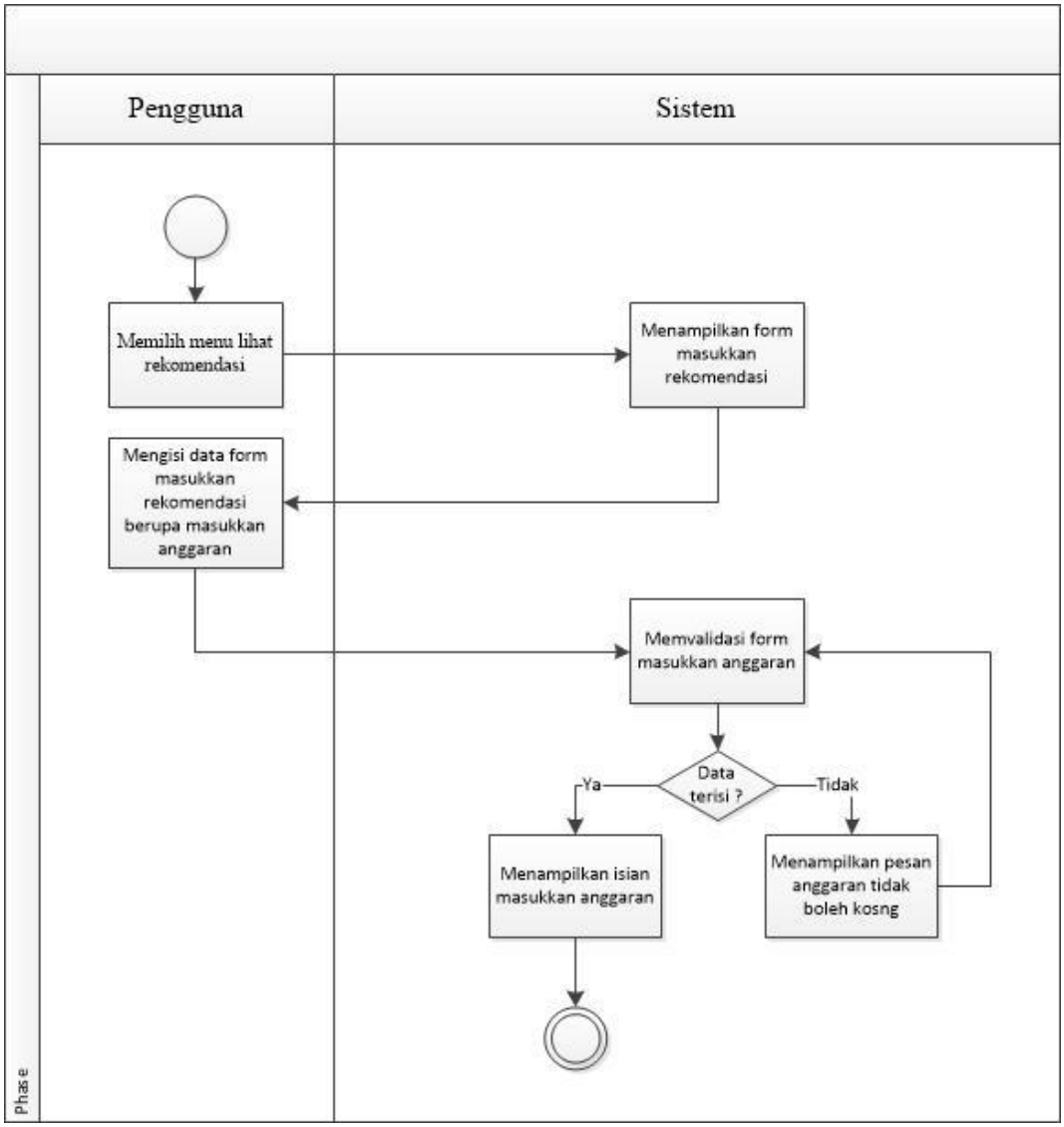
Berikut adalah Gambar 3.11 *Activity* Diagram Daftar Kebutuhan Perjalanan yang menggambarkan *activity* diagram daftar kebutuhan perjalanan.



Gambar 3.11 *Activity* Diagram Daftar Kebutuhan Perjalanan

e) *Activity* Diagram Masukkan Anggaran

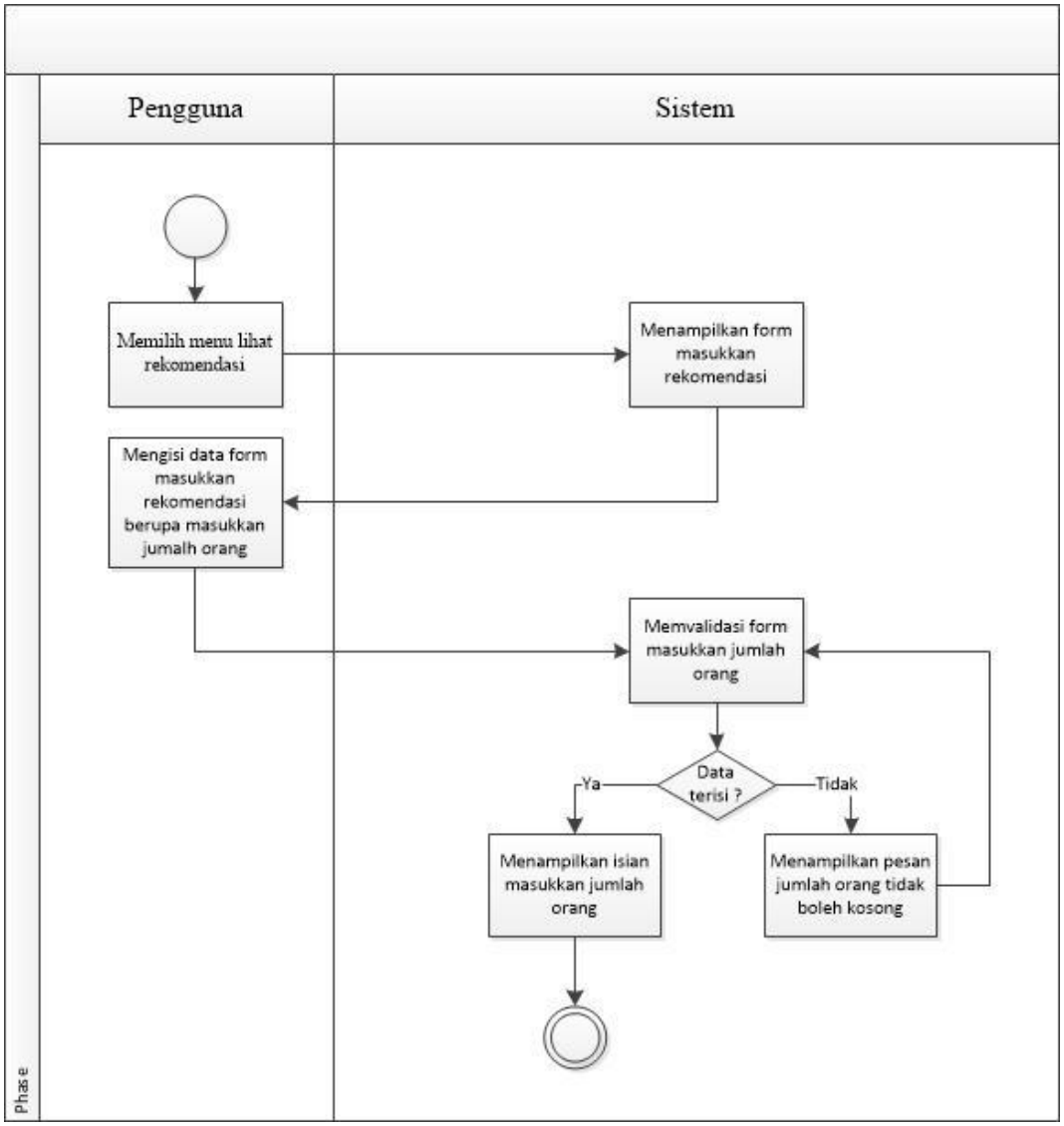
Berikut adalah Gambar 3.12 *Activity* Diagram Masukkan Anggaran yang menggambarkan *activity* diagram masukkan anggaran.



Gambar 3.12 *Activity* Diagram Masukkan Anggaran

f) *Activity* Diagram Masukkan Jumlah Orang

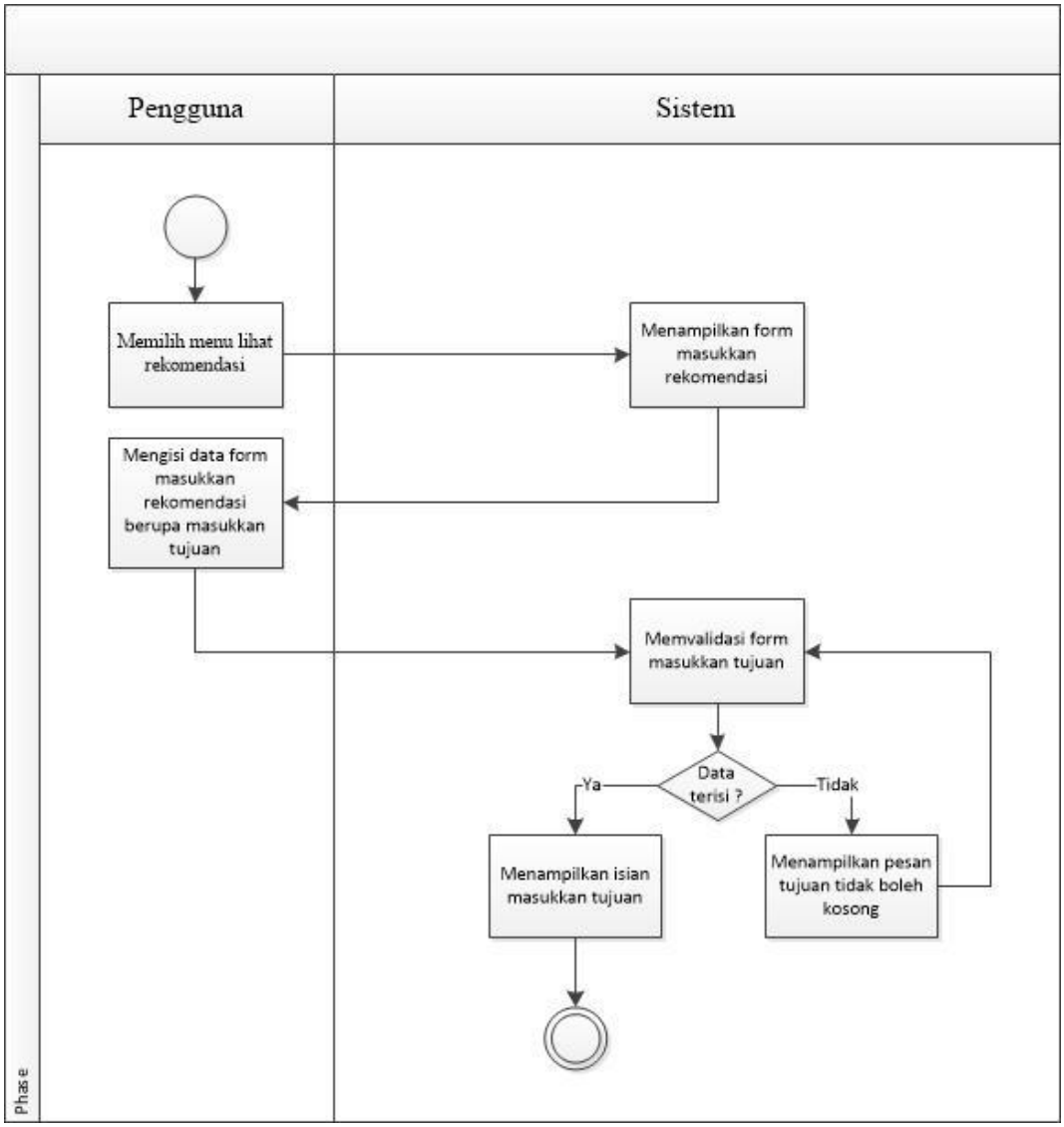
Berikut adalah Gambar 3.13 *Activity* Diagram Masukkan Jumlah Orang yang menggambarkan *activity* diagram masukkan jumlah orang.



Gambar 3.13 *Activity* Diagram Masukkan Jumlah Orang

g) *Activity* Diagram Masukkan Tujuan

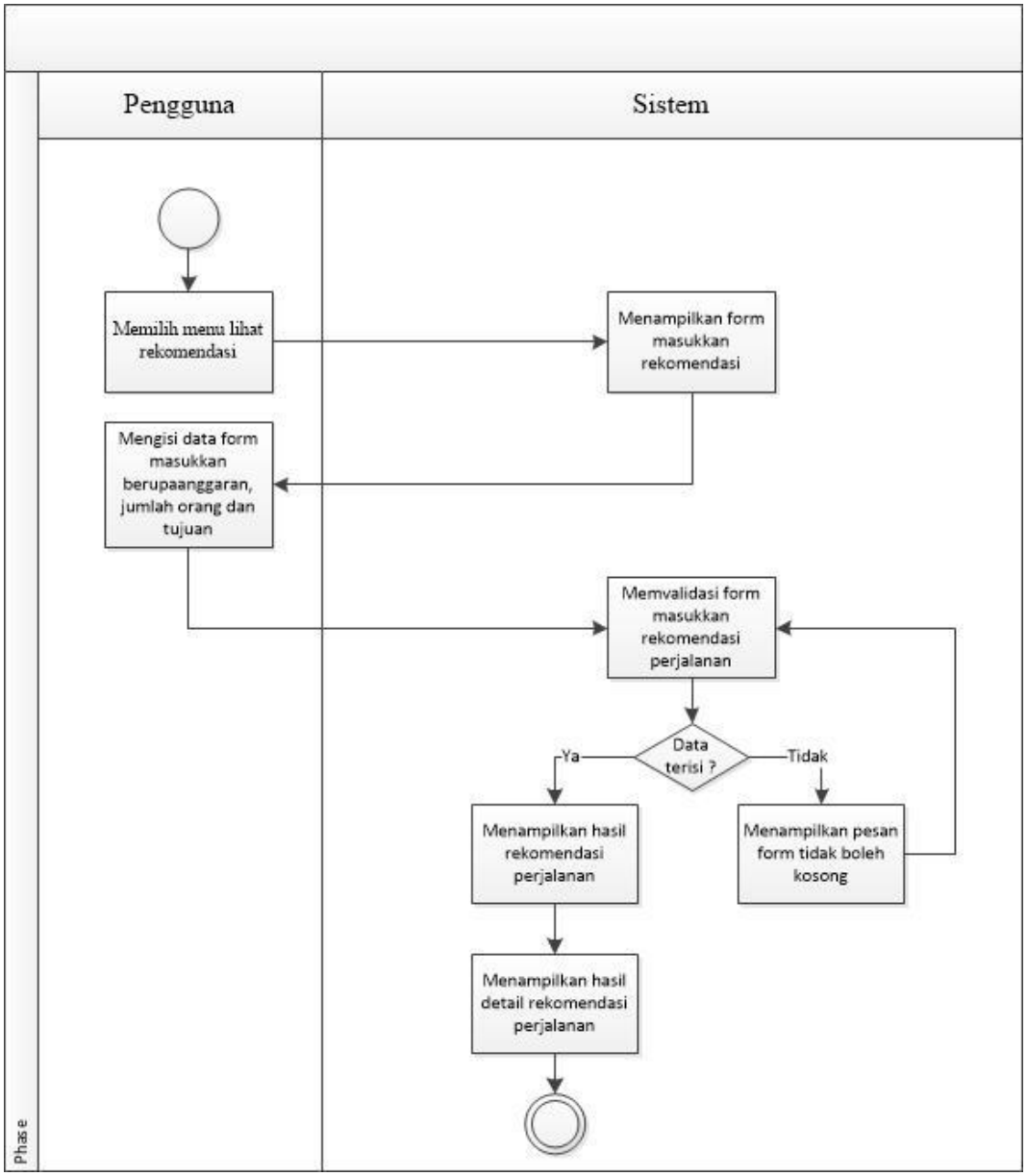
Berikut adalah Gambar 3.14 *Activity* Diagram Masukkan Tujuan menggambarkan *activity* diagram masukkan tujuan.



Gambar 3.14 *Activity* Diagram Masukkan Tujuan

h) *Activity* Diagram Menampilkan Rekomendasi Perjalanan

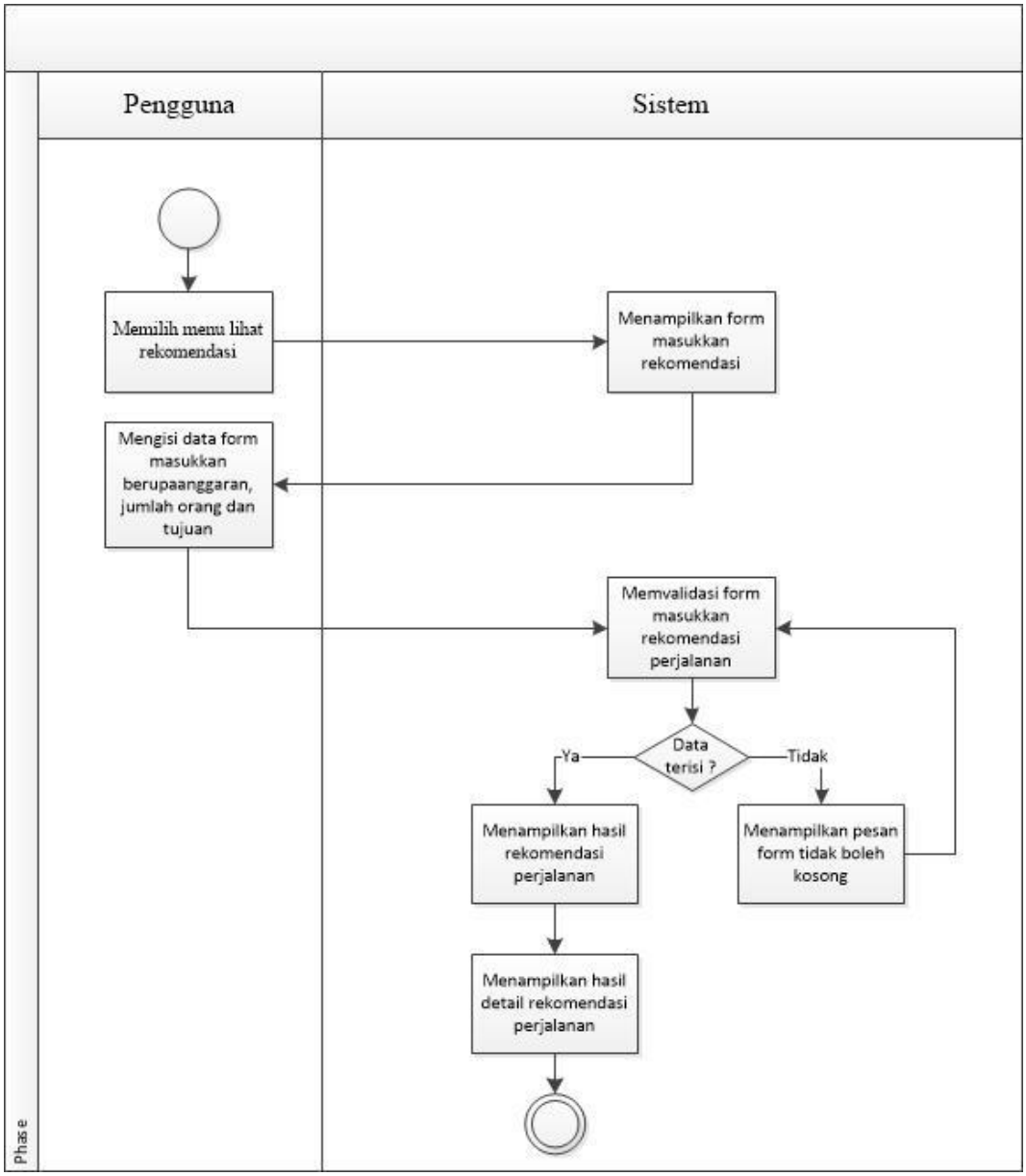
Berikut adalah Gambar 3.15 *Activity* Diagram Menampilkan Rekomendasi Perjalanan yang menggambarkan *activity* diagram menampilkan detail barang rekomendasi.



Gambar 3.15 *Activity* Diagram Menampilkan Rekomendasi Perjalanan

i) *Activity* Diagram Menampilkan Detail Rekomendasi Perjalanan

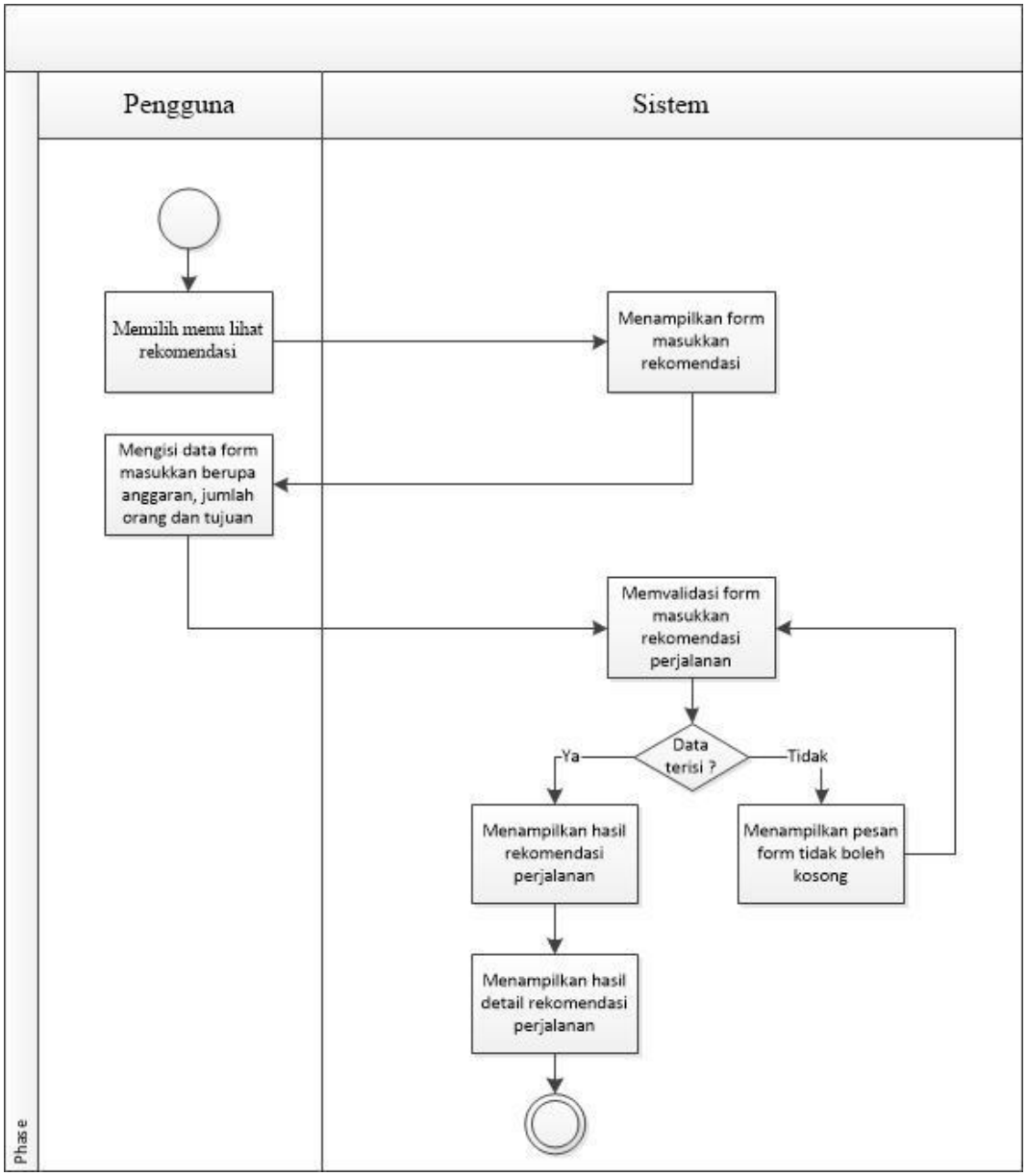
Berikut adalah Gambar 3.16 *Activity* Diagram Menampilkan Detail Rekomendasi Perjalanan yang menggambarkan *activity* diagram menampilkan detail rekomendasi perjalanan.



Gambar 3.16 *Activity* Diagram Menampilkan Detail Rekomendasi Perjalanan

j) *Activity* Diagram Info Detail Perjalanan Yang Dipesan

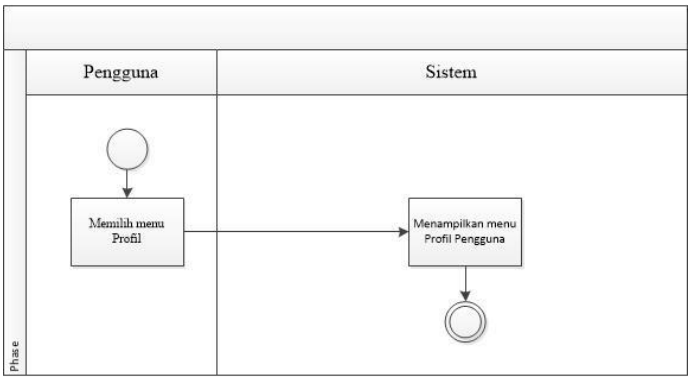
Berikut adalah Gambar 3.17 *Activity* Diagram Info Detail Perjalanan yang Dipesan yang menggambarkan *activity* diagram info detail perjalanan dipesan.



Gambar 3.17 *Activity* Diagram Info Detail Perjalanan Yang Dipesan

k) *Activity* Diagram Profil

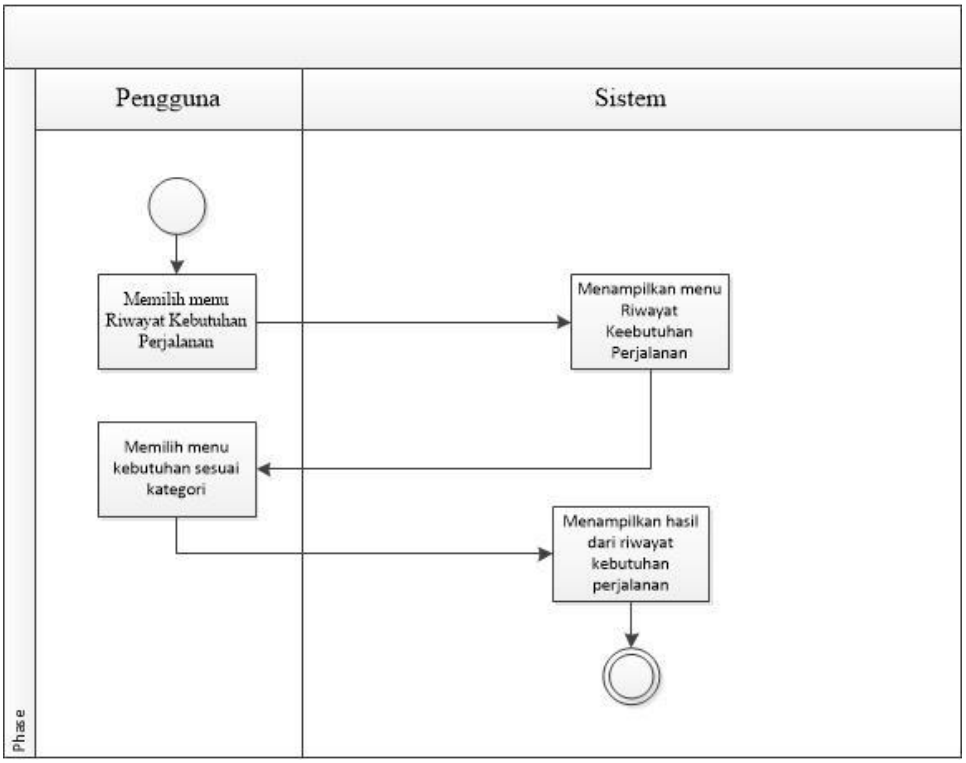
Berikut adalah Gambar 3.18 *Activity* Diagram Profil yang menggambarkan *activity* diagram profil.



Gambar 3.18 *Activity* Diagram Profil

l) *Activity* Diagram Riwayat Kebutuhan Perjalanan

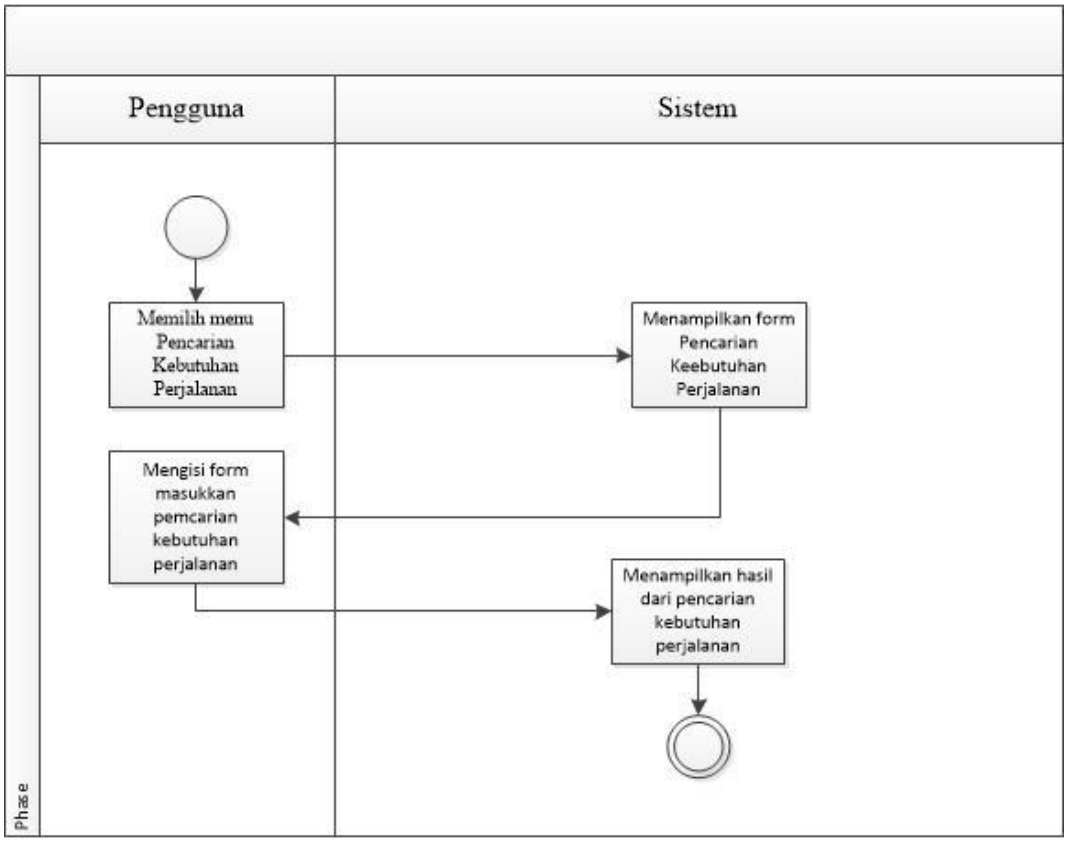
Berikut adalah Gambar 3.19 *Activity* Diagram Riwayat Kebutuhan Perjalanan yang menggambarkan *activity* diagram riwayat kebutuhan perjalanan.



Gambar 3.19 *Activity* Diagram Riwayat Kebutuhan Perjalanan

m) *Activity* Diagram Pencarian Kebutuhan Perjalanan

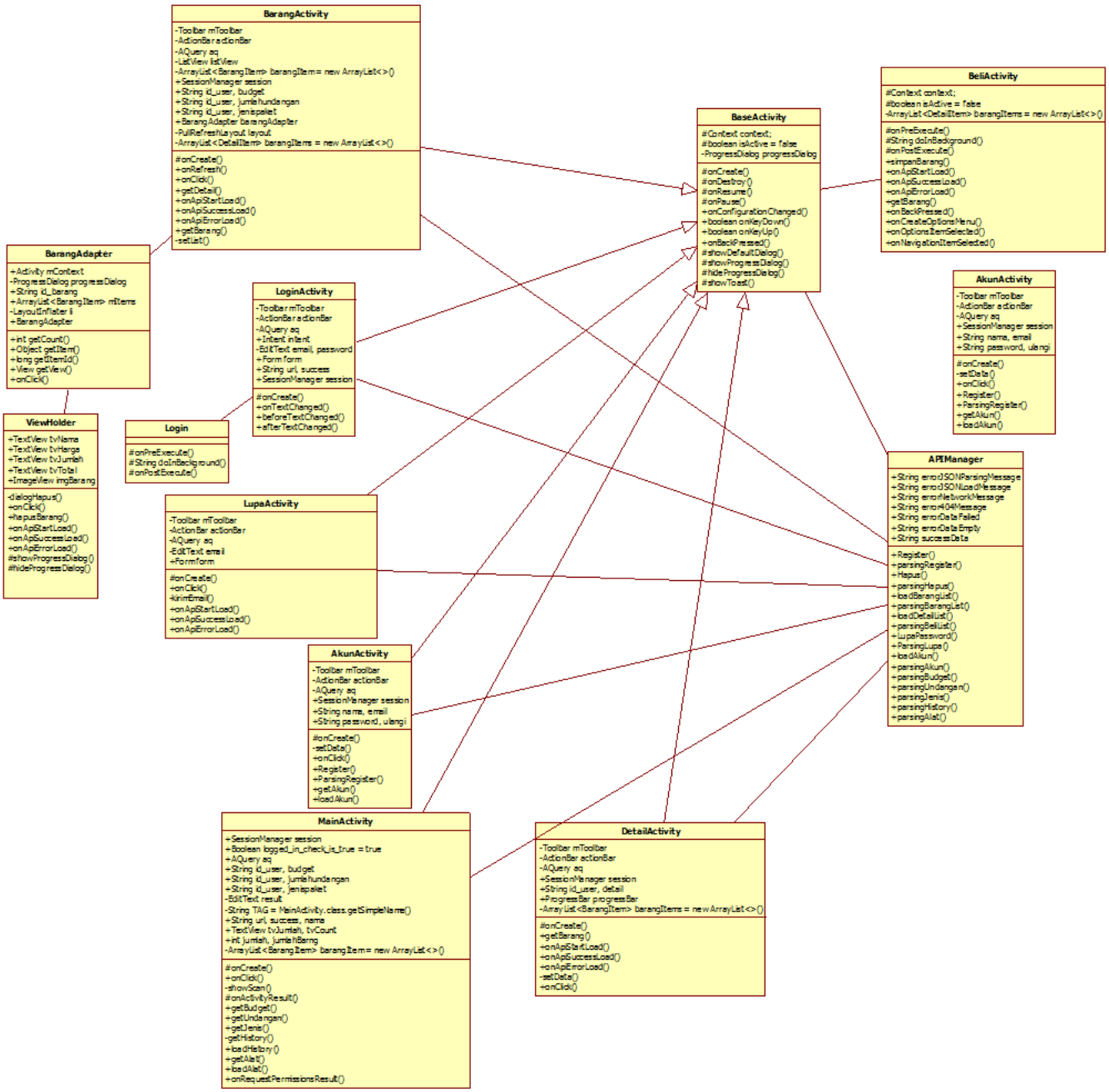
Berikut adalah Gambar 3.20 *Activity* Diagram Pencarian Kebutuhan Perjalanan yang menggambarkan *activity* diagram pencarian kebutuhan perjalanan.



Gambar 3.20 *Activity* Diagram Pencarian Kebutuhan Perjalanan

3.10.6 *Class* Diagram

Class Diagram digunakan untuk menggambarkan *class-class* yang terlibat dalam analisis sebuah sistem yang akan dibangun. Dengan *class diagram* struktur dan deskripsi *class* serta hubungan antar *class* akan terlihat dengan jelas beserta *method-method* setiap *class*-nya. Adapun *class diagram* dari sistem yang dibangun dapat dilihat pada Gambar berikut :



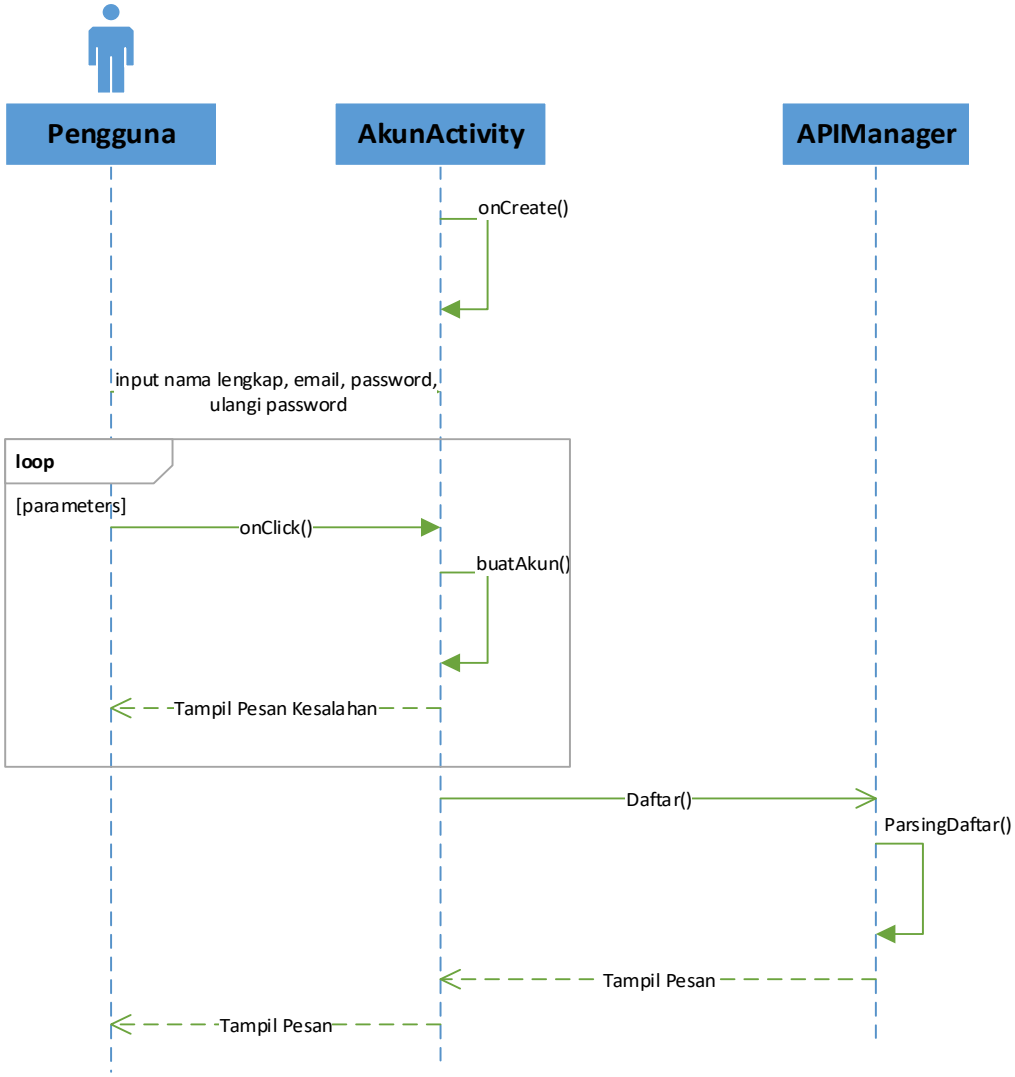
Gambar 3. 21 Class Diagram

3.10.7 *Sequence* Diagram

Sequence Diagram digunakan untuk menggambarkan interaksi antar objek dan mengindikasikan komunikasi diantara objek-objek tersebut. Diagram ini akan menunjukkan pesan-pesan yang bertukar antar objek yang melakukan suatu tugas/*method* pada objek tersebut. Terdapat beberapa *sequence* diagram pada sistem yang dibangun sebagai berikut :

a) *Sequence* Diagram Daftar

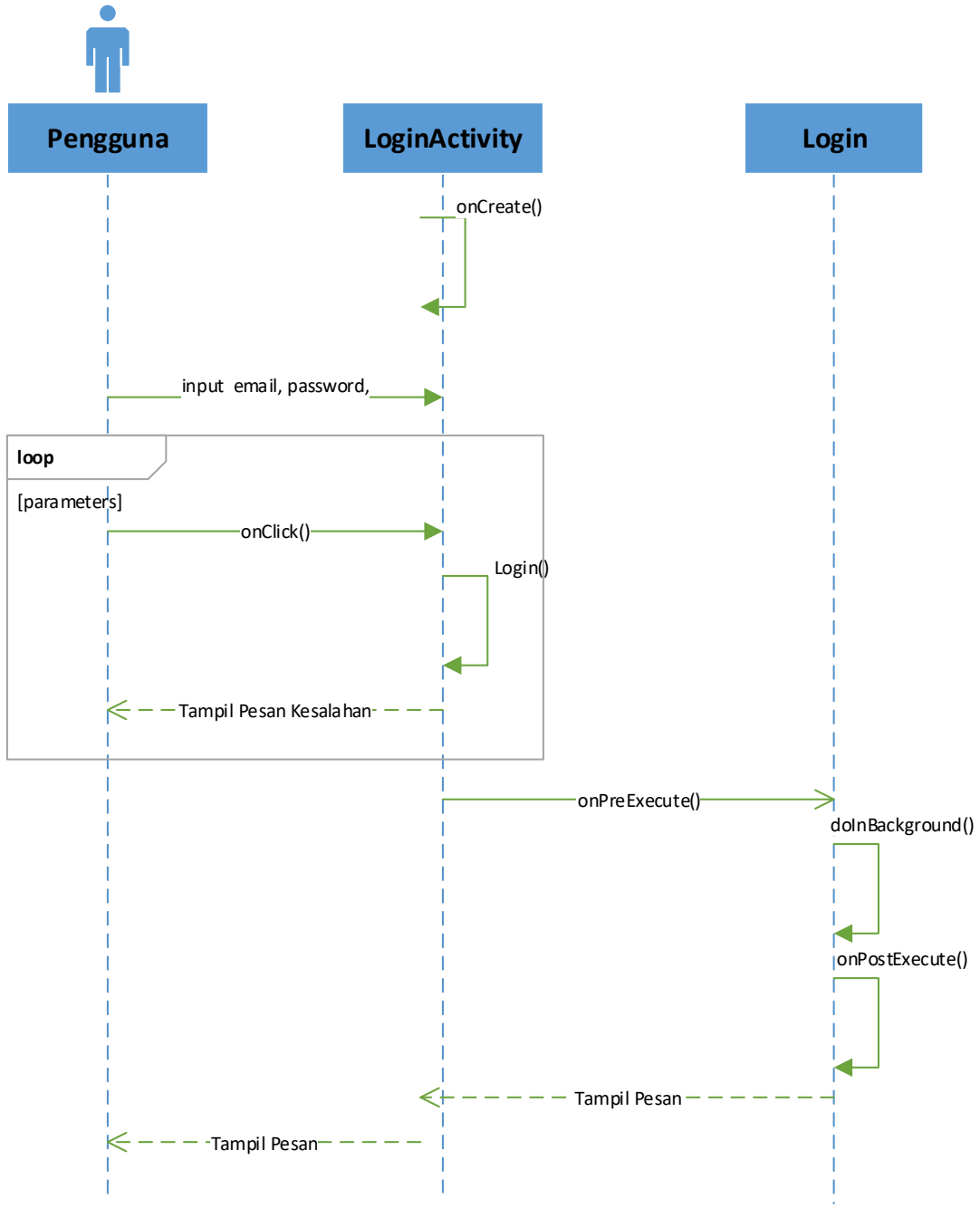
Berikut adalah Gambar 3.22 *Sequence* Diagram Daftar yang menggambarkan *sequence* diagram daftar pengguna.



Gambar 3.22 Sequence Diagram Daftar

b) *Sequence* Diagram *Login*

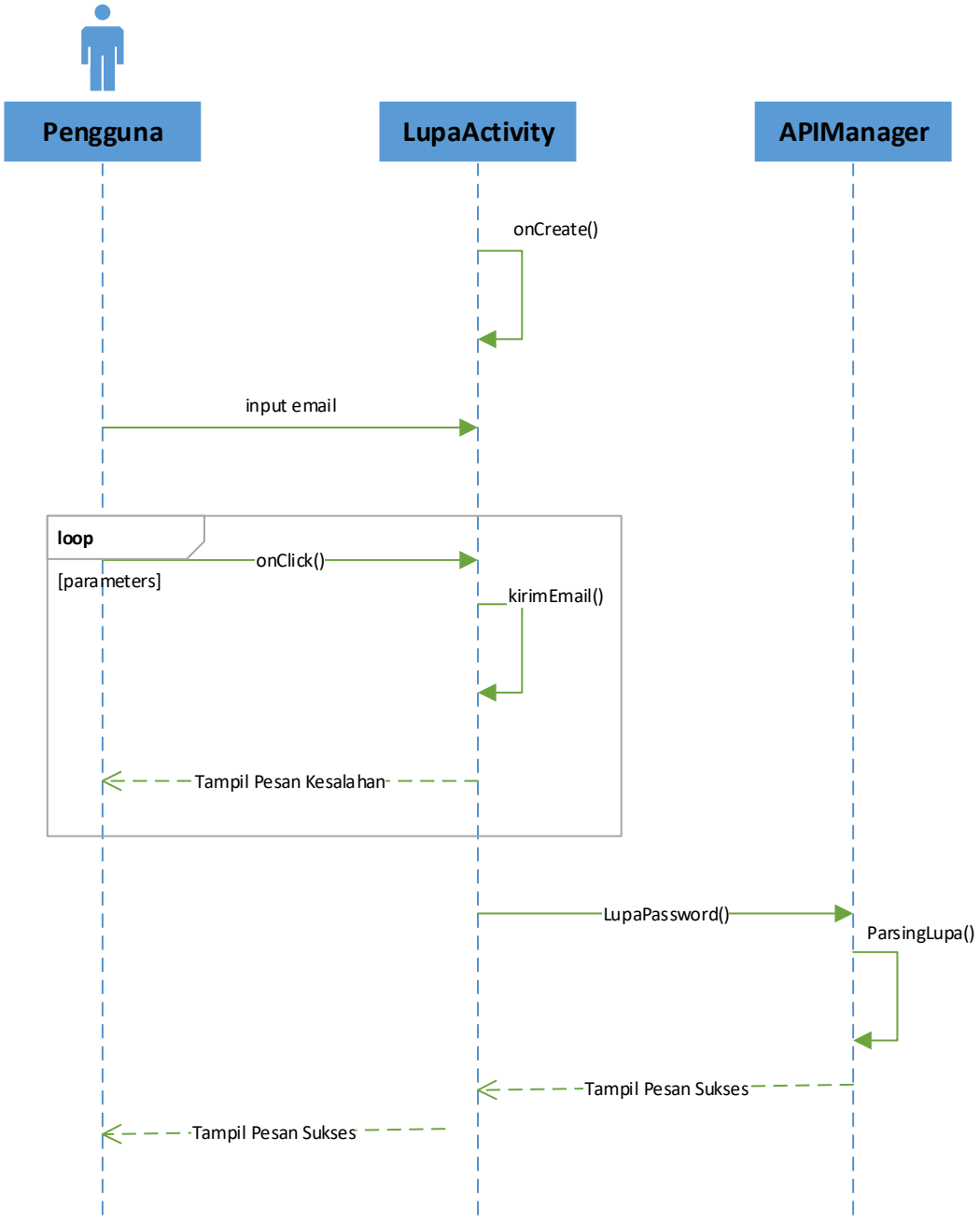
Berikut adalah Gambar 3.23 *Sequence* Diagram *Login* yang menggambarkan *sequence* diagram *login* pengguna.



Gambar 3.23 *Sequence* Diagram *Login*

c) *Sequence* Diagram Lupa *Password*

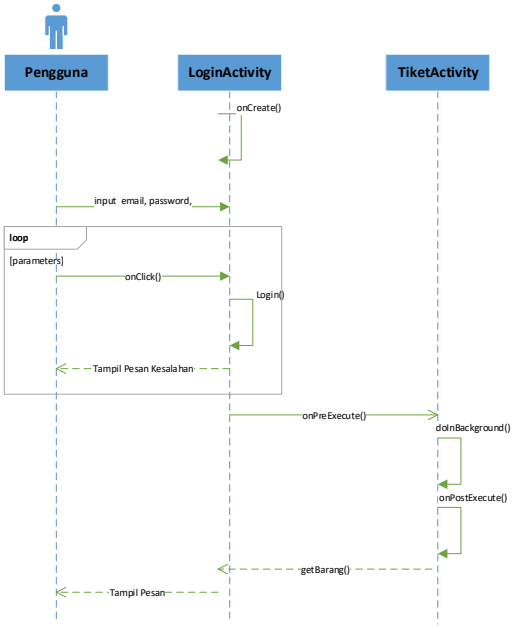
Berikut adalah Gambar 3.24 *Sequence* Diagram Lupa *Password* yang menggambarkan *sequence* diagram lupa *password* pengguna.



Gambar 3.24 *Sequence* Diagram Lupa *Password*

d) *Sequence* Diagram Pemesanan Tiket

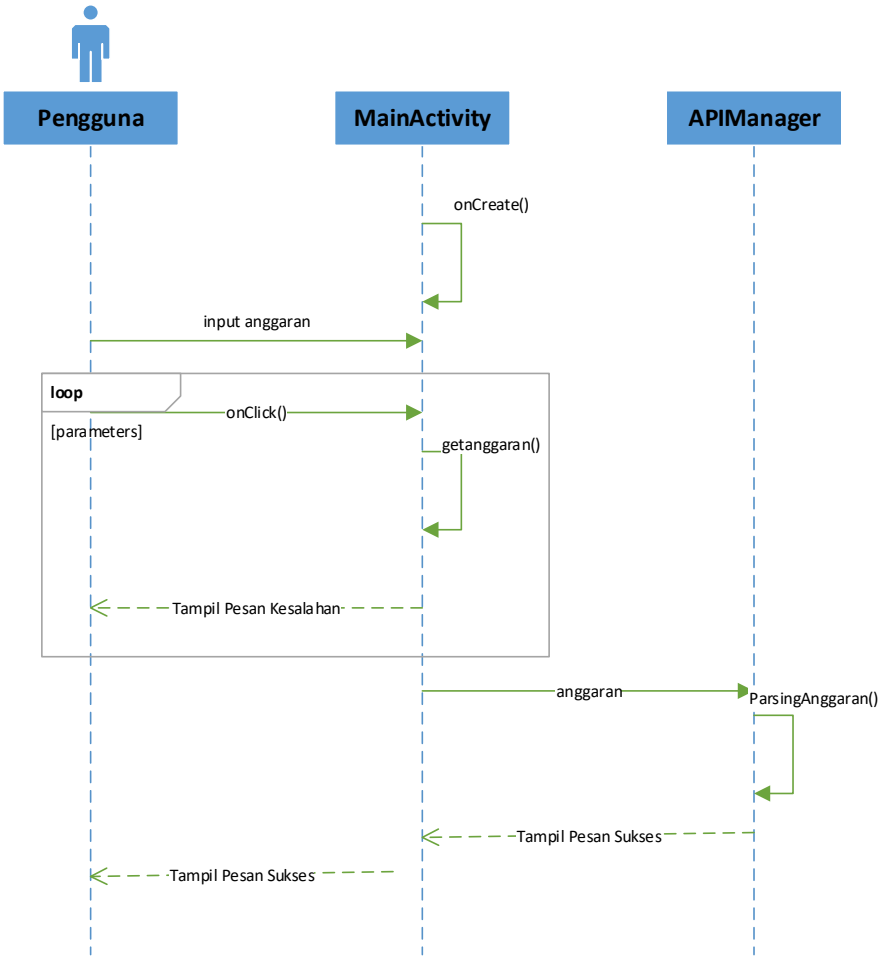
Berikut adalah Gambar 3.25 *Sequence* Diagram Pemesanan Tiket yang menggambarkan *sequence* diagram pemesanan tiket pengguna.



Gambar 3.25 *Sequence* Diagram Tiket yang Dipesan

e) *Sequence* Diagram Masukan Anggaran

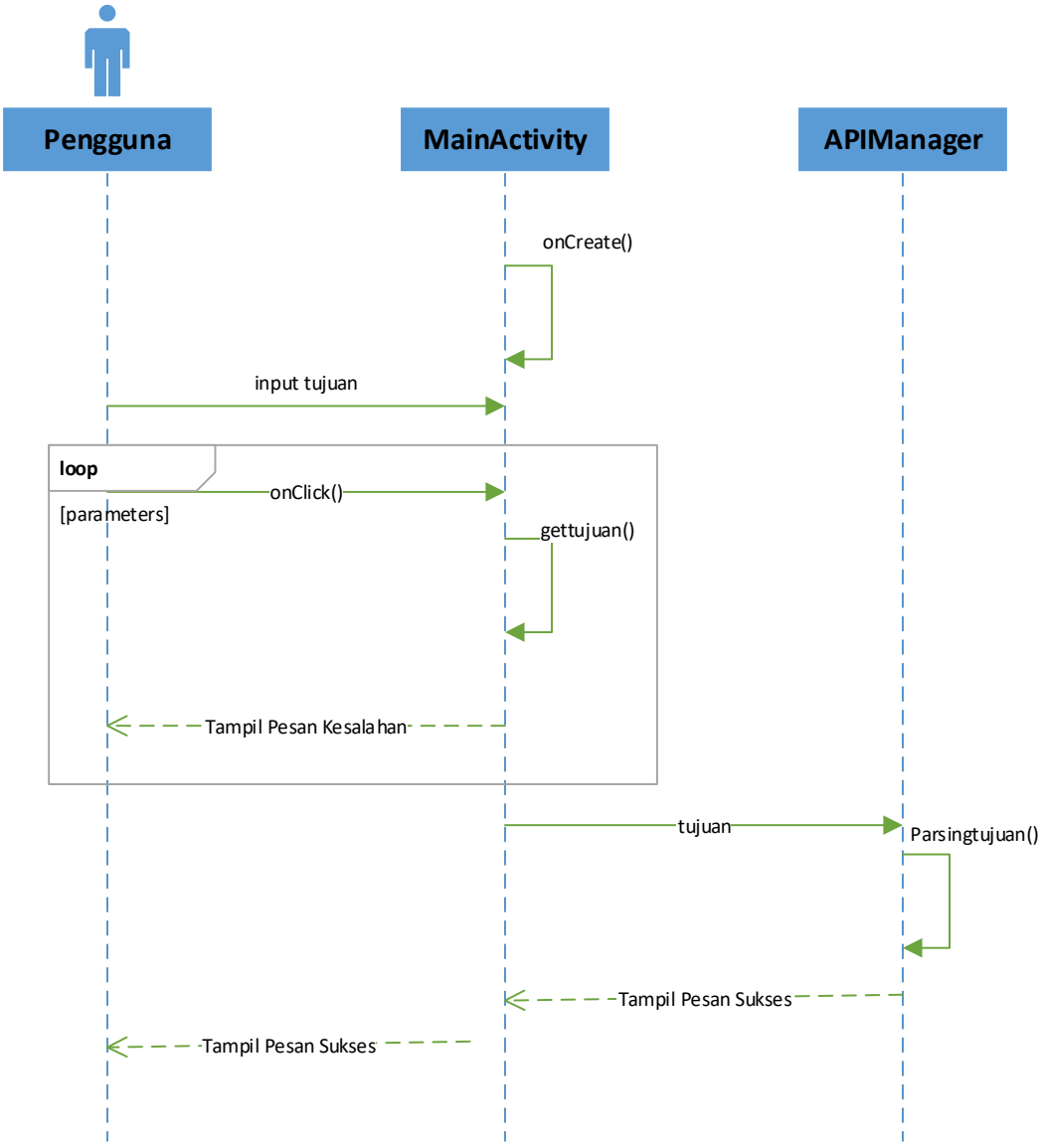
Berikut adalah Gambar 3.26 *Sequence* Diagram Masukan Anggaran yang menggambarkan *sequence* diagram masukan anggaran pengguna.



Gambar 3.26 *Sequence* Diagram Masukan Anggaran

f) *Sequence* Diagram Pilih Tujuan

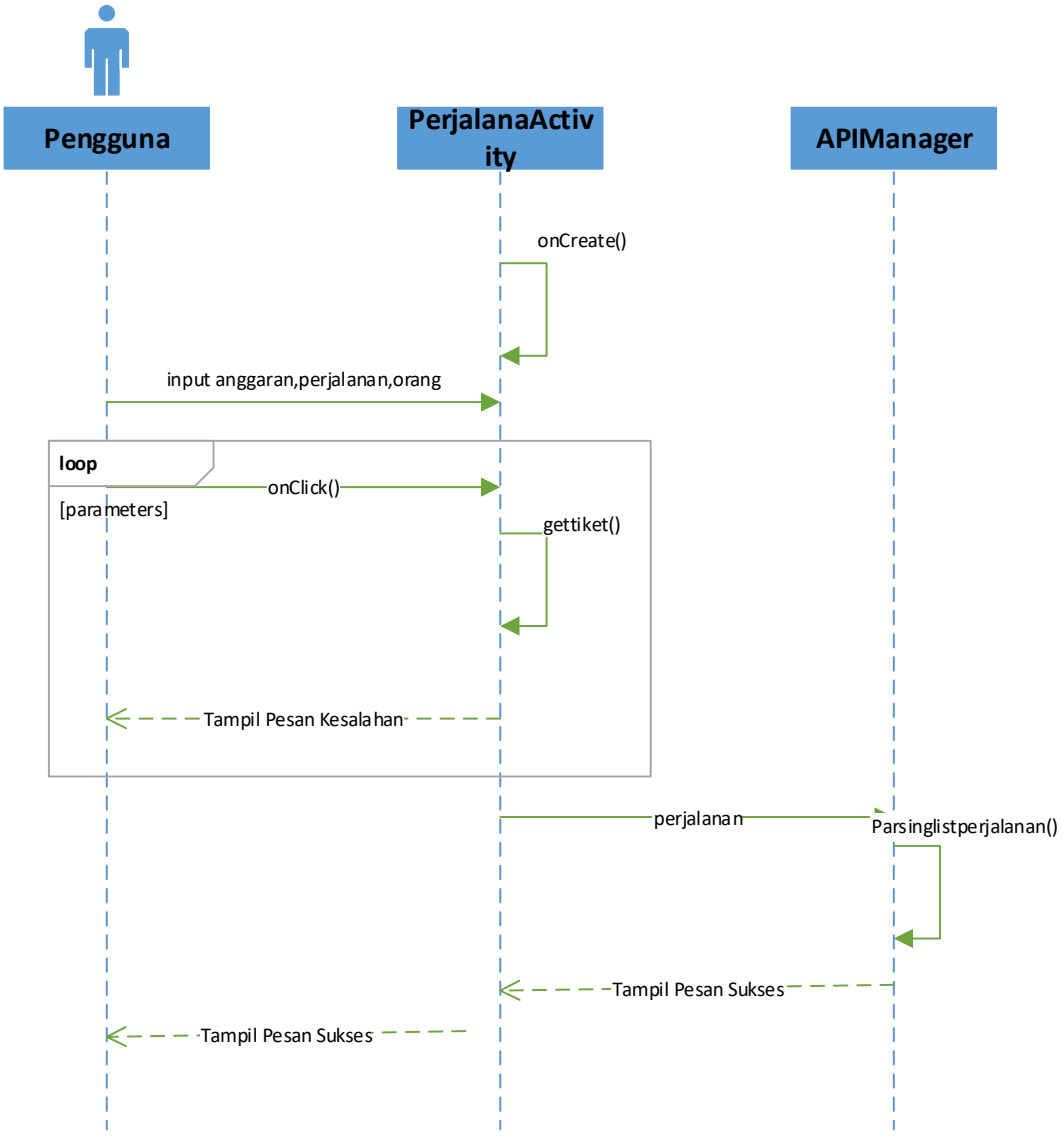
Berikut adalah Gambar 3.28 *Sequence* Diagram Pilih Tujuan yang menggambarkan *sequence* diagram pilih tujuan pengguna.



Gambar 3.27 *Sequence* Diagram Pilih Tujuan

g) *Sequence* Diagram Menampilkan Rekomendasi Perjalanan

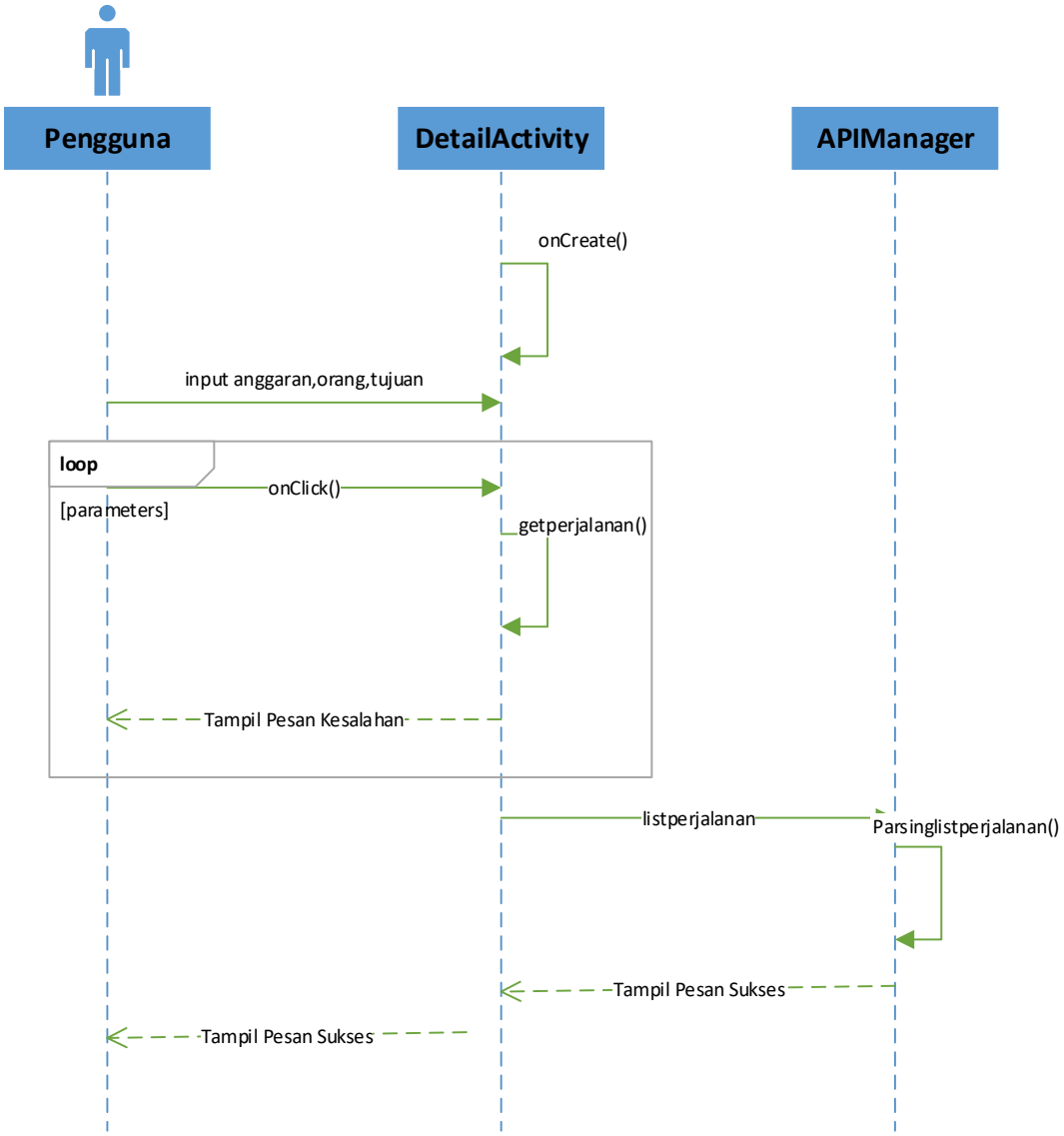
Berikut adalah Gambar 3.29 *Sequence* Diagram Menampilkan Rekomendasi Perjalanan yang menggambarkan *sequence* diagram menampilkan rekomendasi perjalanan pengguna.



Gambar 3.28 *Sequence* Diagram Menampilkan Rekomendasi Perjalanan

h) *Sequence* Diagram Menampilkan Detail Rekomendasi Perjalanan

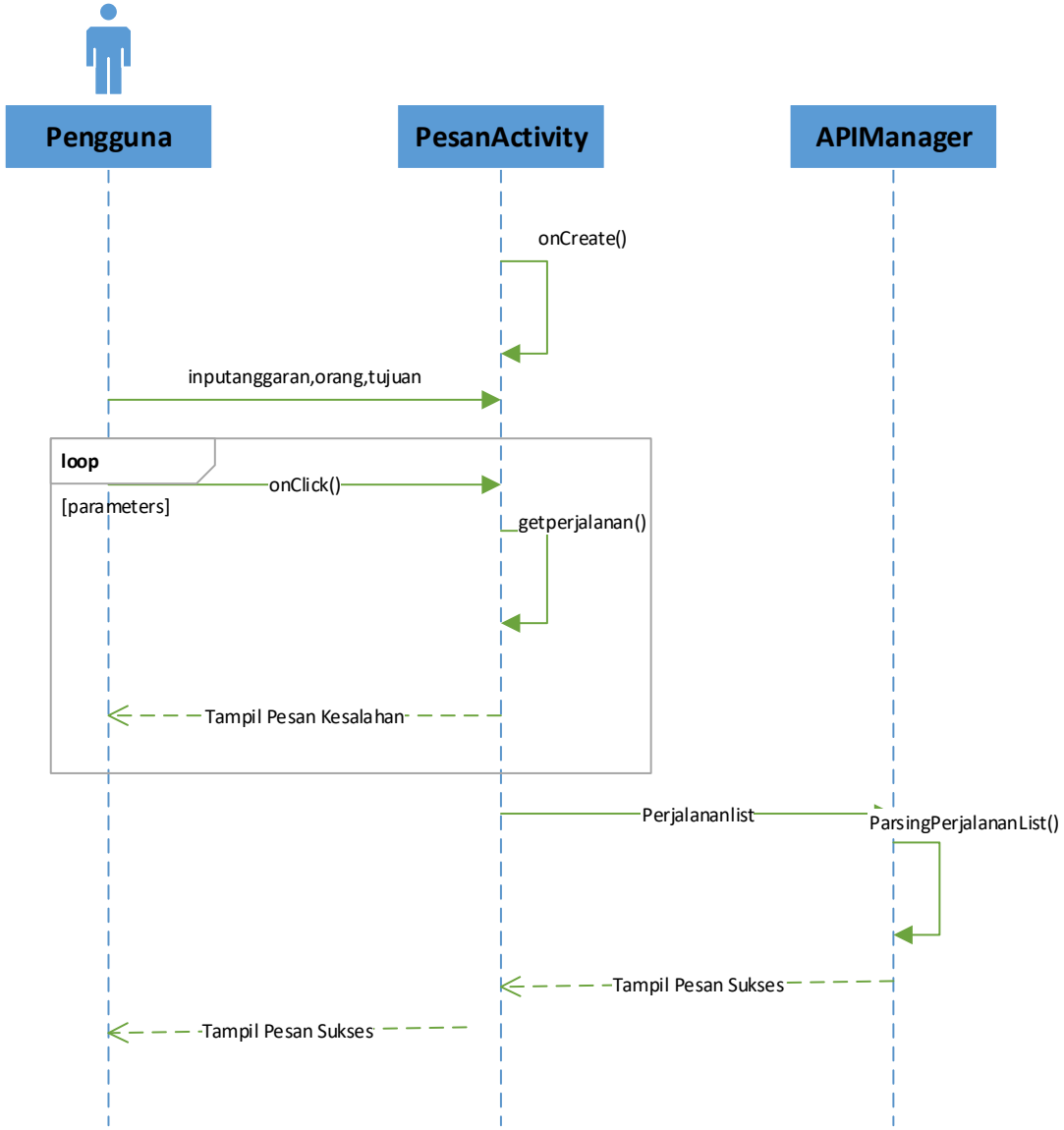
Berikut adalah Gambar 3.30 *Sequence* Diagram Menampilkan Detail Rekomendasi Perjalanan yang menggambarkan *sequence* diagram menampilkan detail rekomendasi perjalanan pengguna.



Gambar 3.29 *Sequence* Diagram Menampilkan Detail Rekomendasi Perjalanan

i) *Sequence* Diagram Info Detail Tujuan Perjalanan

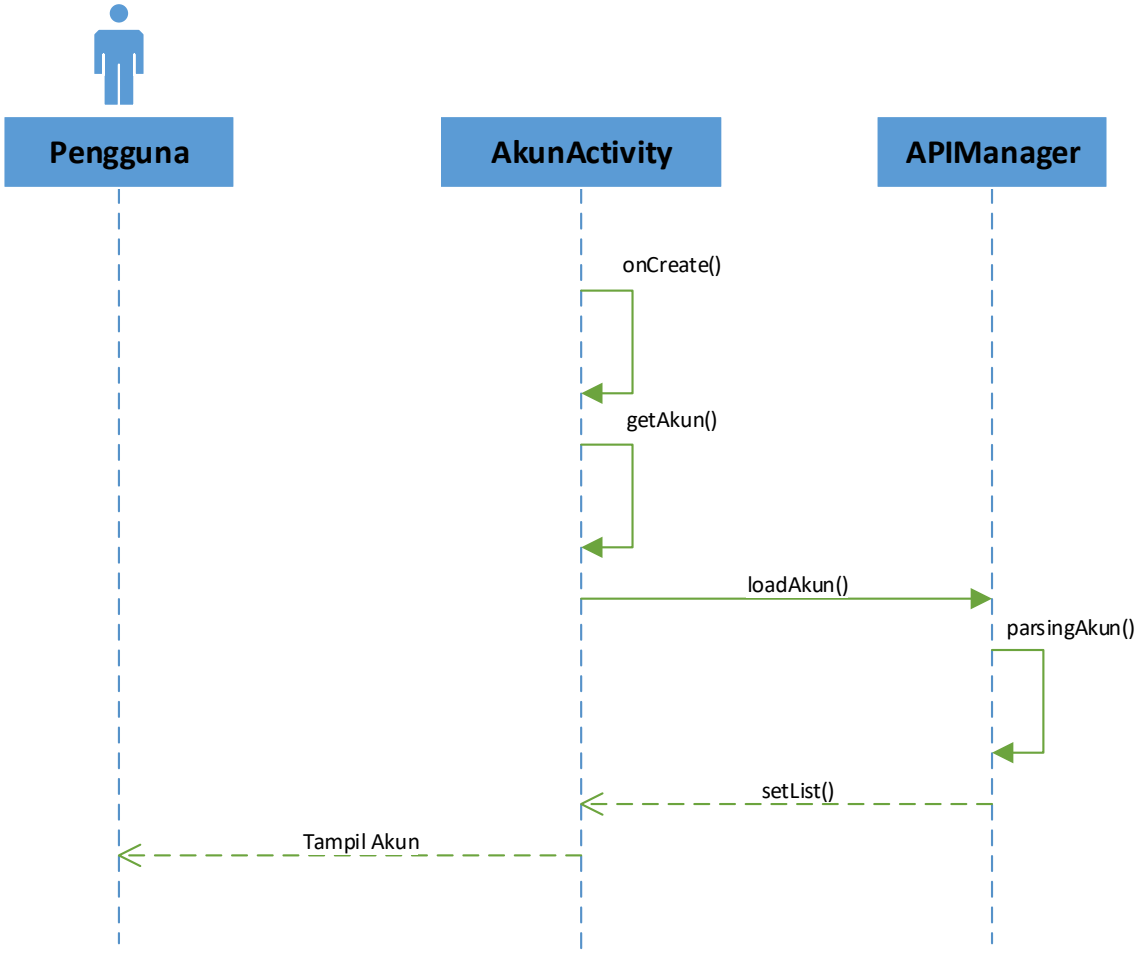
Berikut adalah Gambar 3.31 *Sequence* Diagram Info Detail Tujuan Perjalanan yang menggambarkan *sequence* diagram info detail tujuan perjalanan pengguna.



Gambar 3.30 *Sequence* Diagram Info Detail Tujuan Perjalanan

j) *Sequence* Diagram Profil

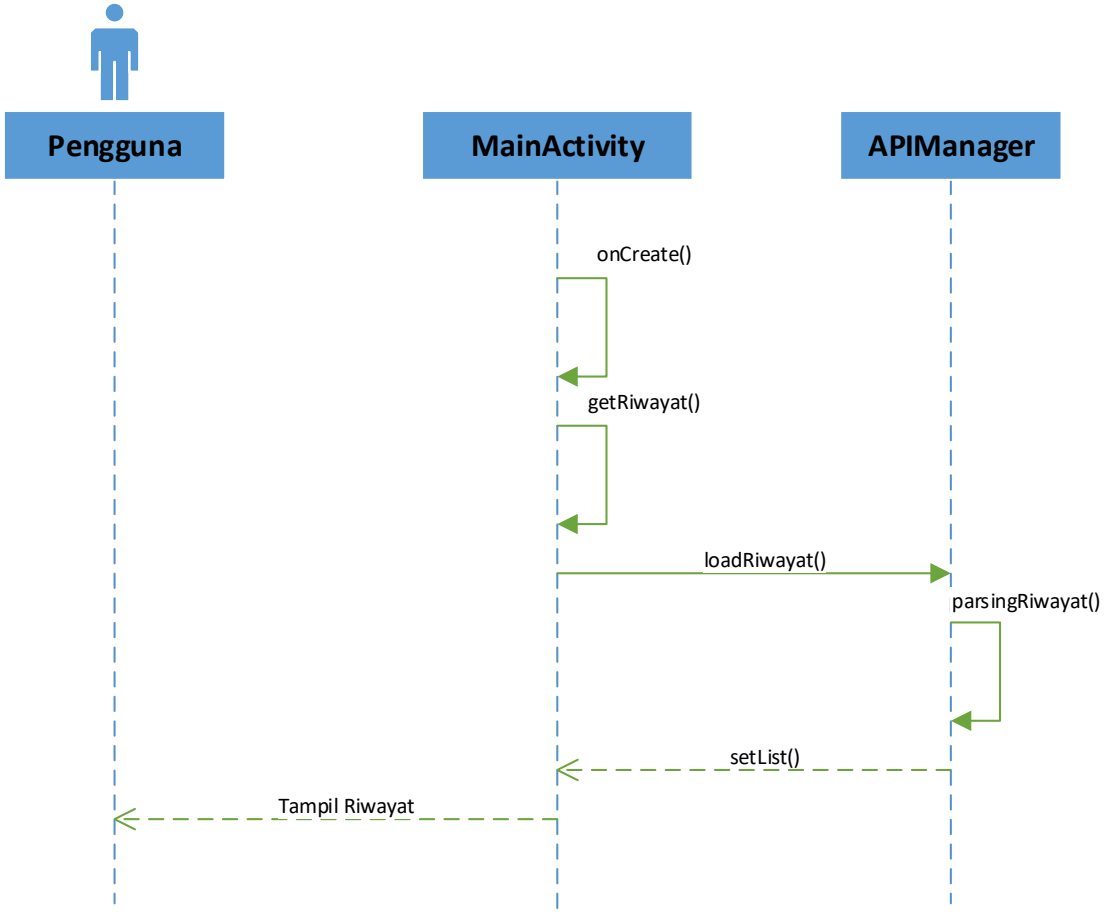
Berikut adalah Gambar 3.32 *Sequence* Diagram Profil yang menggambarkan *sequence* diagram profil pengguna.



Gambar 3.31 *Sequence* Diagram Profil

k) *Sequence* Diagram Riwayat Perjalanan

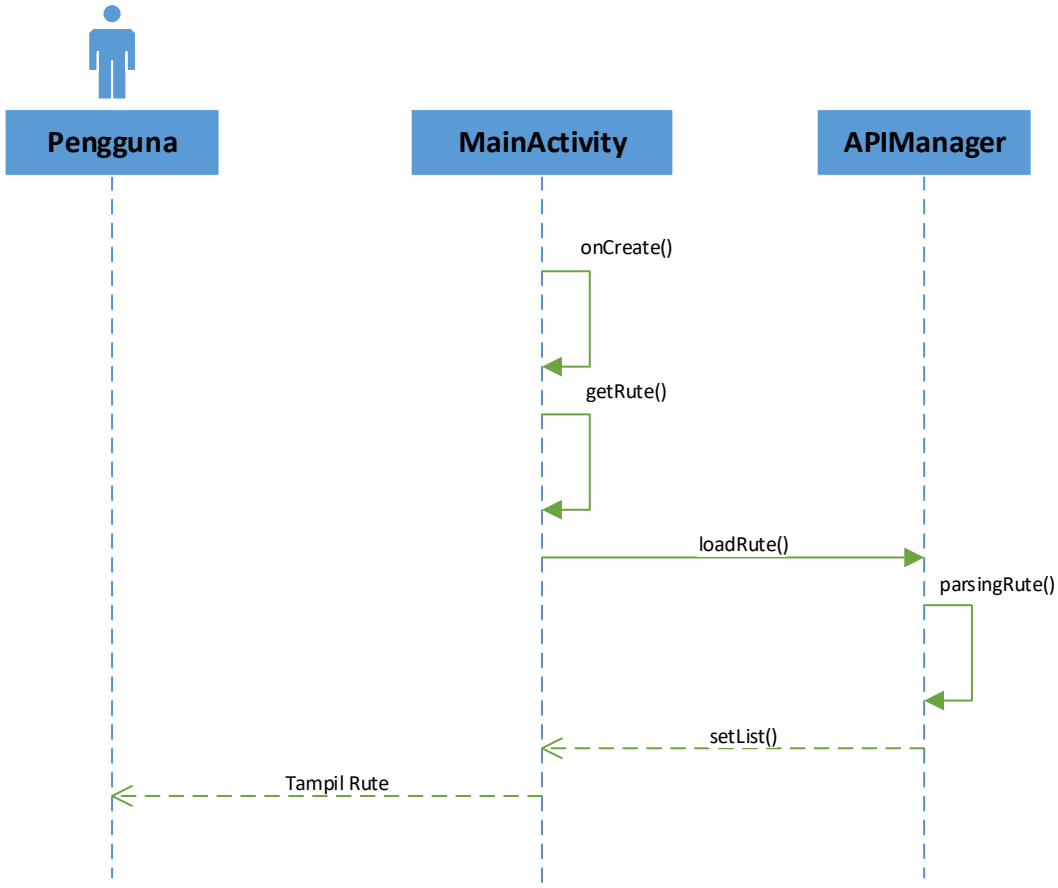
Berikut adalah Gambar 3.33 *Sequence* Diagram Riwayat Perjalanan yang menggambarkan *sequence* diagram riwayat perjalanan pengguna.



Gambar 3.32 *Sequence* Diagram Riwayat Perjalanan

l) *Sequence* Diagram Rute Perjalanan

Berikut adalah Gambar 3.34 *Sequence* Diagram Rute Perjalanan_yang menggambarkan *sequence* diagram rute perjalanan.



Gambar 3.33 *Sequence* Diagram Rute Perjalanan

3.11 Perancangan Sistem

Perancangan sistem adalah tahapan dimana seorang peneliti atau pengembang merancang sebuah sistem terkait dengan gambaran tentang hasil analisis sistem yang dilanjutkan dengan memikirkan bagaimana membentuk sistem tersebut. Perancangan ini bertujuan untuk memenuhi kebutuhan pengguna serta memberikan gambaran yang jelas atas hasil analisis sistem. Pada

tahap perancangan sistem ini terdapat beberapa perancangan meliputi perancangan basis data, perancangan struktur tabel, perancangan antarmuka, perancangan pesan hingga perancangan jaringan semantik.

3.11.1 Perancangan Struktur Tabel

Perancangan struktur tabel bertujuan untuk mengidentifikasi atribut-atribut yang ada pada suatu tabel. Perancangan struktur tabel didasarkan kepada perancangan basis data yang telah dirancang sebelumnya. Adapun perancangan struktur tabel pada sistem yang dibangun dapat sebagai berikut :

a. Berikut adalah Tabel 3.33 Struktur Tabel tbl_tiket yang mendeskripsikan tabel tiket.

Tabel 3. 33 Struktur Tabel tbl_tiket

Nama Field	Tipe Data	Panjang	Kunci	Keterangan
id_tiket	Int	11	Primary key	not null, auto increment
nama_tiket	Varchar	30		not null
harga_tiket	Int	11		not null
Id_budget	Int	11	Foreign key (`id_budget’) References ` db_angpao`.`tbl_budget` (`id_budget`) on delete cascade on update cascade	not null
Id_tujuan	Int	11	Foreign key (`id_undangan’) References ` db_angpao`.`tbl_undangan` (`id_undangan`) on delete cascade on update cascade	not null

Nama Field	Tipe Data	Panjang	Kunci	Keterangan
Id_tujuan	Int	11	Foreign key (`id_paket') References ` db_angpao`.`tbl_paket` (`id_paket`) on delete cascade on update cascade	not null

b. Berikut adalah Tabel 3.34 Struktur Tabel tbl_budget yang mendeskripsikan tabel *budget*.

Tabel 3. 34 Struktur Tabel tbl_budget

Nama Field	Tipe Data	Panjang	Kunci	Keterangan
id_budget	Int	11	Primary key	not null, auto increment
Nilai_budget	Int	11		not null,
Teks_budget	Varchar	20		not null

c. Berikut adalah Tabel III.30 Struktur Tabel tbl_undangan yang mendeskripsikan tabel tujuan.

Tabel 3. 35 Struktur Tabel tbl_tujuan

Nama Field	Tipe Data	Panjang	Kunci	Keterangan
id_tiket	Int	11	Primary key	not null, auto increment
Nilai_tiket	Int	11		not null,
Teks_tiket	Varchar	20		not null

3.11.2 Perancangan Antarmuka

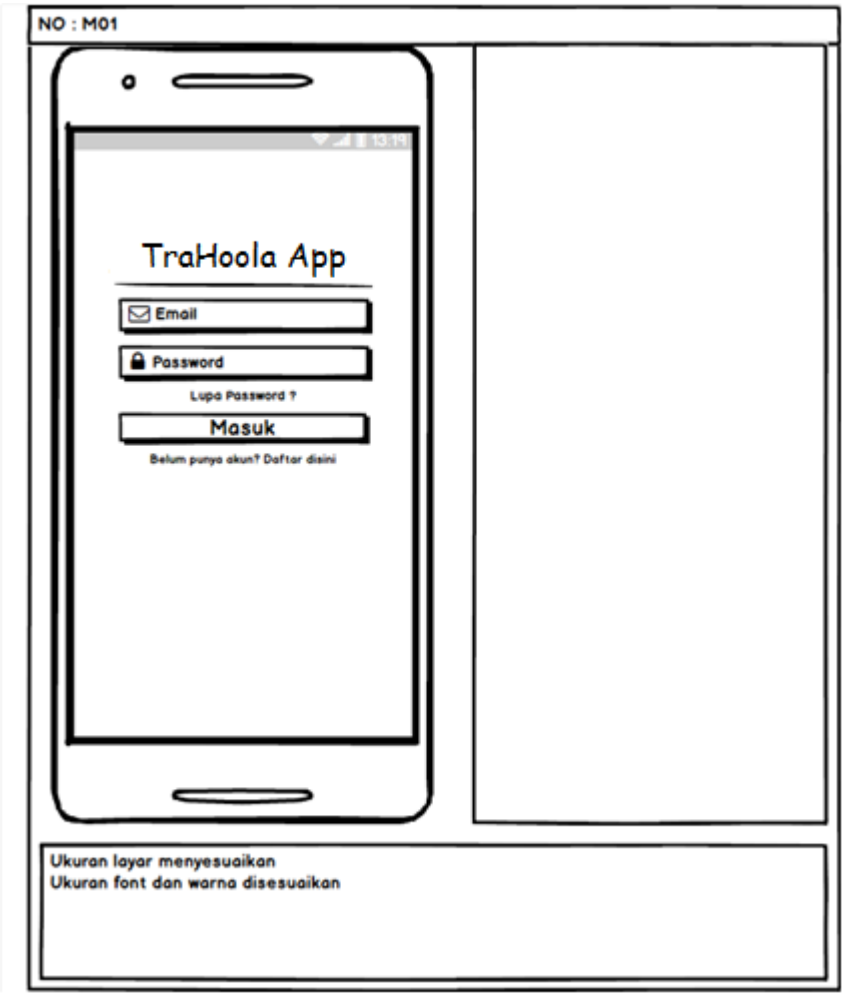
Perancangan antarmuka merupakan mekanisme komunikasi antara pengguna (*user*) dengan sistem. Antarmuka (*Interface*) dapat menerima informasi dari pengguna (*user*) dan memberikan informasi kepada pengguna (*user*) untuk membantu mengarahkan alur penelusuran masalah sampai ditemukan suatu solusi. Perancangan antarmuka pada sistem yang dibangun meliputi perancangan antarmuka sub sistem web dan sub sistem mobile.

3.11.3 Perancangan Antarmuka Sistem Mobile

Perancangan anrarmuka sub sistem mobile bertujuan untuk menggambarkan tampilan antarmuka pada aplikasi mobile dalam hal ini adalah aplikasi android. Perancangan antarmuka sub sistem ini berfokus pada bagaimana interaksi setiap komponen di dalam antarmuka.

1. **Antarmuka *Login***

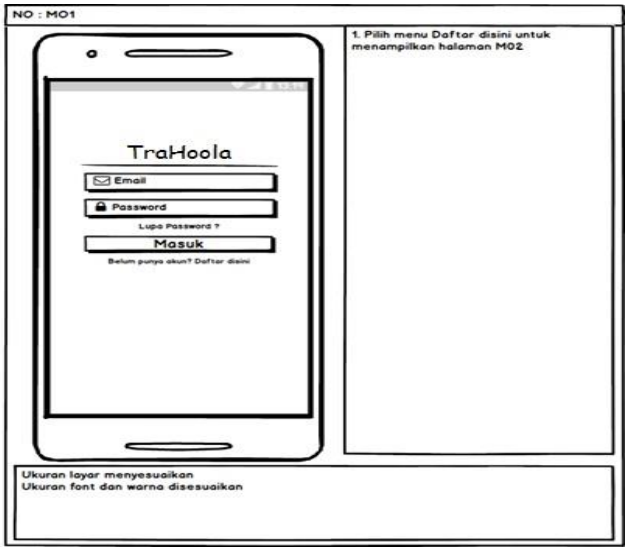
Pada antarmuka ini digambarkan bagaimana tampilan dari halaman *login* pada sub sistem *mobile* yaitu pada aplikasi android yang akan digunakan pengguna. Adapun antarmuka *login* dapat dilihat pada Gambar 3.36 berikut :



Gambar 3.34 Antarmuka Login

2. **Antarmuka *Sign Up***

Pada antarmuka ini digambarkan bagaimana tampilan dari halaman *sign up* pada sub sistem *mobile* yaitu pada aplikasi android yang akan digunakan pengguna. Adapun antarmuka *sign up* dapat dilihat pada Gambar 3.37 berikut :



Gambar 3.35 Antarmuka Sign Up

3.11.6 Perancangan Pesan

Perancangan pesan pada sistem *mobile* bertujuan untuk memberikan rancangan pesan penanganan kesalahan atas kesalahan yang mungkin dilakukan oleh pengguna. Terdapat beberapa perancangan pesan pada sub sistem *mobile* sebagai berikut :

1. **Pesan *error***

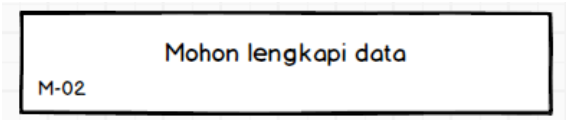
   Pesan ini akan muncul jika pengguna mengalami masalah pada *smartphone* miliknya seperti tidak terkoneksi internet, data *input* tidak valid ataupun kesalahan lainnya. Adapun perancangan pesan untuk pesan eror ini dapat dilihat pada Gambar 3.38 berikut :



Gambar 3.36 Pesan *Error*

2. **Pesan *field input* kosong**

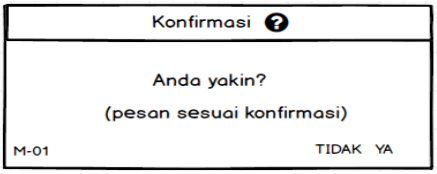
   Pesan ini akan muncul jika pengguna tidak mengisi *field input* yang seharusnya diisi. Adapun perancangan pesan untuk pesan eror ini dapat dilihat pada Gambar 3.39 berikut :



Gambar 3.37 Pesan Field Input Kosong

3. Pesan konfirmasi

Pesan ini akan muncul jika pengguna hendak melakukan suatu aksi yang membutuhkan konfirmasi ulang seperti *logout* dan penghapusan data. Adapun perancangan pesan untuk pesan eror ini dapat dilihat pada Gambar 3.40 berikut :

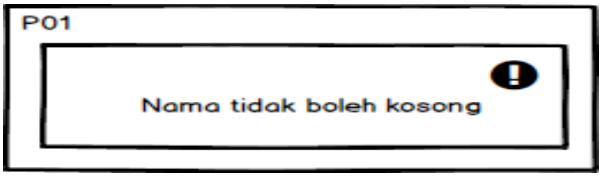


Gambar 3.38 Pesan Konfirmasi

4. Perancangan Pesan Kesalahan Pengisian

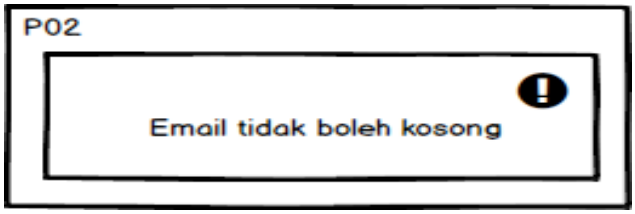
Perancangan pesan pada sub sistem *mobile* bertujuan untuk memberikan rancangan pesan penanganan kesalahan atas kesalahan yang mungkin dilakukan oleh pengguna. Terdapat beberapa perancangan pesan kesalahan pengisian sebagai berikut :

a) Perancangan Pesan Nama Lengkap Kosong



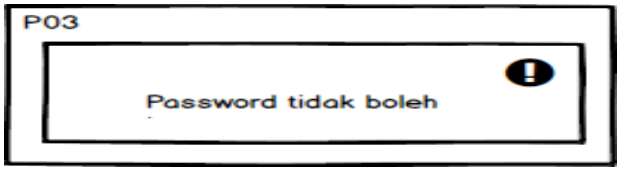
Gambar 3.39 Perancangan Pesan Nama Lengkap Kosong

b) Perancangan Pesan Email Kosong



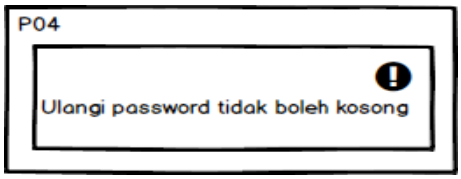
Gambar 3.40 Perancangan Pesan Email Kosong

c) Perancangan Pesan *Password* Kosong



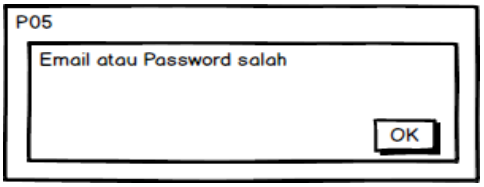
Gambar 3.41 Perancangan Pesan *Password* Kosong

d) Perancangan Pesan Ulangi *Password*



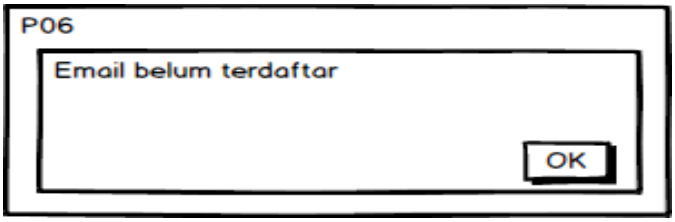
Gambar 3.42 Perancangan Pesan Ulangi *Password*

e) Perancangan Pesan Email atau *Password* Salah



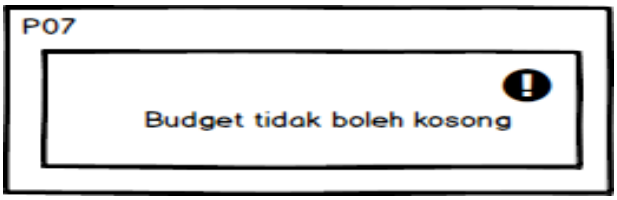
Gambar 3.43 Perancangan Pesan Email atau Password Salah

f) Perancangan Pesan *Email* Tidak Terdaftar



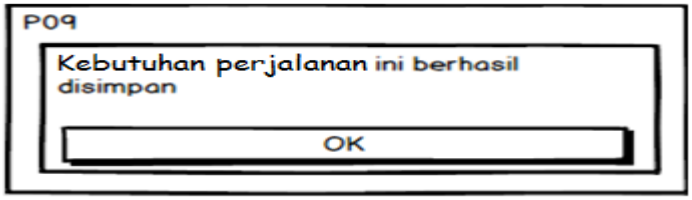
Gambar 3.44 Perancangan Pesan *Email* Tidak Terdaftar

g) Perancangan Pesan *Budget* Kosong



Gambar 3.45 Perancangan Pesan Budget Kosong

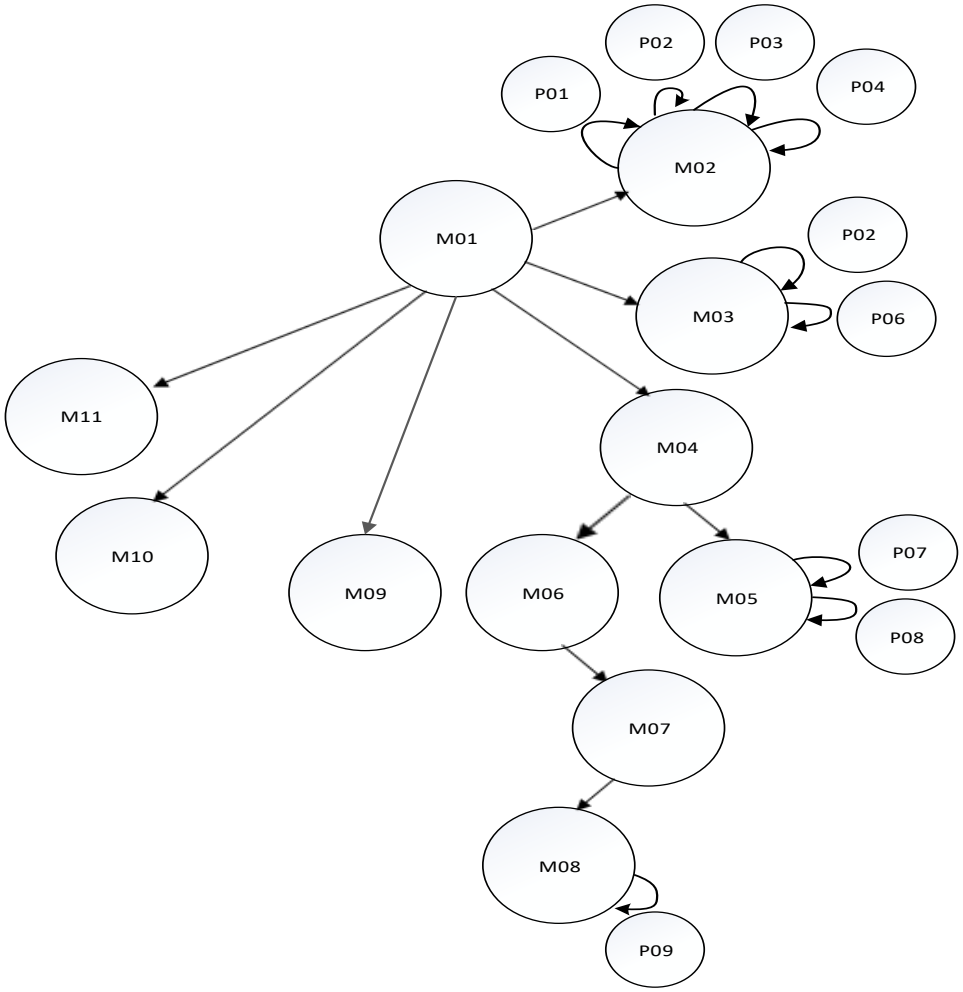
h) Perancangan Pesan Kebutuhan Perjalanan Disimpan



Gambar 3.46 Perancangan Pesan Kebutuhan Perjalanan Disimpan

3.11.7 Perancangan Jaringan Semantik

Perancangan jaringan semantik merupakan representasi hubungan antar *interface* (antarmuka) yang telah dibuat sebelumnya. Rancangan jaringan semantik ini akan menggambarkan alur setiap antarmuka dengan antarmuka yang lain.



Gambar 3.47 Jaringan Semantik