

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1. Identifikasi Permasalahan

Dalam sebuah kota jumlah mikrolet tergantung pada luas area kota tersebut. Salah satu kota yang banyak menggunakan alat transportasi adalah kota Malang. Kota Malang merupakan kota terbesar kedua di Jawa Timur dan termasuk salah satu kota yang menjadi tempat tujuan masyarakat untuk pendidikan dan pariwisata serta berbagai aktivitas masyarakat lainnya. Dengan banyaknya aktivitas yang dilakukan di kota Malang tentu saja banyak masyarakat yang menggunakan mikrolet untuk mendukung aktivitas sehari-harinya.

Di kota Malang, mikrolet sangat mudah sekali ditemukan. Untuk mengetahui tujuan utama mikrolet, masyarakat dapat melihat simbol tujuan yang biasanya terletak pada bagian depan dan belakang mikrolet. Salah satu contohnya adalah AL yang merupakan simbol tujuan Arjosari-Landungsari. Daftar tujuan mikrolet dapat dilihat di lampiran. Berdasarkan hasil survey di Dinas Perhubungan kota Malang, kota Malang menyediakan 25 trayek/jurusan mikrolet untuk mendukung aktivitas masyarakatnya.

Di dalam sebuah mikrolet tidak terdapat jalan dan tempat-tempat umum yang dilewati mikrolet. Jika ingin mengetahui jalan atau tujuan yang dilewati mikrolet, pengguna mikrolet dapat melihat di terminal atau bertanya pada supir mikrolet secara langsung. Tetapi kebanyakan pengguna mikrolet menaiki mikrolet dari tepi jalan bukan dari terminal sehingga masyarakat tidak mengetahui jalan dan tempat-tempat umum yang dilewati mikrolet. Dengan banyaknya jumlah

trayek/jurusan mikrolet di kota Malang tentu saja pengguna mikrolet tidak dapat menghafalkan satu per satu rute dari tiap trayek mikrolet dan menyebabkan kesulitan dalam menentukan mikrolet mana yang harus dipilih untuk menuju ke tempat tujuannya.

Bagi pengguna mikrolet yang sedang bertransportasi di kota Malang, masalah di atas tentu saja dapat merepotkan. Bisa saja masalah tersebut dapat membuat pengguna mikrolet tersesat jika salah memilih mikrolet. Dengan adanya permasalahan seperti ini perlu dibuat suatu aplikasi yang dapat memandu pengguna mikrolet dalam bertransportasi dengan mikrolet di kota Malang. Perangkat yang memungkinkan untuk membangun aplikasi tersebut adalah perangkat *mobile* karena perangkat komunikasi ini dapat dengan mudah dibawa kemana-mana.

3.2. Analisis Sistem

Dengan banyaknya aktifitas yang dilakukan masyarakat di kota Malang tentu saja mikrolet banyak menjadi pilihan utama bagi masyarakat dalam bertansportasi. Dengan banyaknya jumlah trayek/jurusan mikrolet di kota Malang tentu saja pengguna mikrolet tidak dapat menghafalkan satu per satu rute dari tiap trayek mikrolet. Ditambah dengan kurangnya pengetahuan masyarakat tentang jalan dan tempat-tempat umum yang dilewati mikrolet akan menyebabkan kesulitan dalam menentukan mikrolet mana yang harus dipilih untuk menuju ke tempat tujuannya.

Untuk melakukan penentuan lokasi suatu perangkat *mobile* terdapat banyak cara, namun yang sering digunakan adalah *Cell Identification* (Cell-ID) dan *Global Positioning System* (GPS). Pemanfaatan Cell-ID memiliki kelebihan yaitu tidak

membutuhkan perangkat tambahan karena memanfaatkan fasilitas penyedia jaringan yang memungkinkan seorang pelanggan seluler untuk mengetahui posisinya terhadap *Base Transfer Station* (BTS) terdekat. Namun akurasi dengan teknik Cell-ID ini sangat rendah yaitu berkisar 1-3 kilometer. GPS memiliki akurasi yang jauh lebih tinggi sehingga lebih cocok untuk diimplementasikan dalam sistem penentuan posisi meskipun teknik ini mengharuskan dibutuhkan perangkat tambahan. Ditambah dengan adanya pengembangan baru GPS yaitu *Assisted Gloval Positioning System* (A-GPS) akurasi yang didapat dalam menentukan posisi menjadi lebih akurat dengan bantuan BTS disekitarnya.

Sekarang ini banyak sekali *smartphone* dengan berbagai *platform* yang beredar di pasaran dan memiliki fitur yang semakin banyak. Dari sekian banyak *platform* yang beredar, hampir semua *smartphone* dengan *platform* Android telah tertanam fitur GPS bahkan fitur A-GPS pada *platform* Android keluaran terbaru. Fitur GPS inilah yang merupakan komponen utama untuk membangun aplikasi ini, oleh karena itu aplikasi yang akan dibangun sangat cocok dengan fitur yang ada pada *smartphone* Android.

Untuk membantu masyarakat dalam menentukan posisi dan navigasi menggunakan mikrolet, maka dapat memanfaatkan fitur GPS pada *smartphone* Android dengan mengembangkan sebuah aplikasi navigasi berbasis Android. Pencarian data dan pengolahan data yang dilakukan dengan cara merancang *database* dan membuat sistem. Data-data tersebut nantinya akan ditampung dan diolah oleh aplikasi sehingga memberikan informasi navigasi kepada *user*.

Dari uraian di atas, maka dapat ditarik kesimpulan bahwa untuk mempermudah menentukan posisi pengguna mikrolet dan memilih mikrolet, dapat

dikembangkan sebuah aplikasi navigasi berbasis Android yang dapat membantu menentukan posisi pengguna mikrolet dan memberikan navigasi pemilihan penggunaan transportasi umum mikrolet kepada penggunanya. Dengan adanya aplikasi ini, diharapkan membantu pengguna mikrolet dalam bertransportasi menggunakan mikrolet di kota Malang dengan memanfaatkan teknologi GPS yang ada pada *smartphone* Android.

Gambaran umum sistem yang ada dalam aplikasi ini secara garis besar adalah sebagai berikut :

1. GPS module yang terdapat pada *smartphone* Android akan menentukan posisi terhadap satelit GPS yang ada. Untuk dapat menentukan posisi *smartphone* maka dibutuhkan minimal tiga atau lebih sinyal satelit GPS.
2. Setelah posisi *smartphone* diketahui, data posisi ini bisa digunakan untuk melakukan navigasi dengan menggunakan mikrolet. Data koordinat yang didapat disesuaikan dengan data koordinat yang dilewati mikrolet dan data koordinat tujuan untuk diolah.
3. Dari data-data yang telah diolah oleh aplikasi, aplikasi akan menampilkan data-data tersebut dalam bentuk gambar rute navigasi di peta *google map* dan informasi berupa jalan dan petunjuk penggunaan mikrolet.

3.3. Identifikasi Kebutuhan

Berdasarkan hasil analisis diatas, maka aplikasi *mobile* yang dibuat harus dapat menentukan posisi penggunanya dan dapat memberikan navigasi dengan mikrolet kepada penggunanya. Untuk membantu menentukan posisi dapat digunakan fitur GPS yang terdapat pada *smartphone* Android. Navigasi kepada

pengguna dapat dilakukan dengan merancang aplikasi yang dapat mengolah data koordinat sehingga dapat memberikan informasi navigasi kepada pengguna.

Untuk mendukung kinerja aplikasi yang dibuat agar bekerja maksimal dibutuhkan spesifikasi *smartphone Android* minimal sebagai berikut:

1. *Smartphone* yang dilengkapi GPS atau A-GPS.

Smartphone harus memiliki GPS atau A-GPS karena aplikasi yang dibuat membutuhkan kemampuan GPS atau A-GPS dalam menentukan posisi.

2. *Smartphone* dengan *operating system* Android minimal Android 2.2 (*Froyo*).

Sekarang ini sudah banyak beredar *smartphone* yang dilengkapi *operating system* Android terbaru. Supaya aplikasi yang dibuat dapat berjalan lancar maka aplikasi dibuat dengan menggunakan standar Android 2.2.

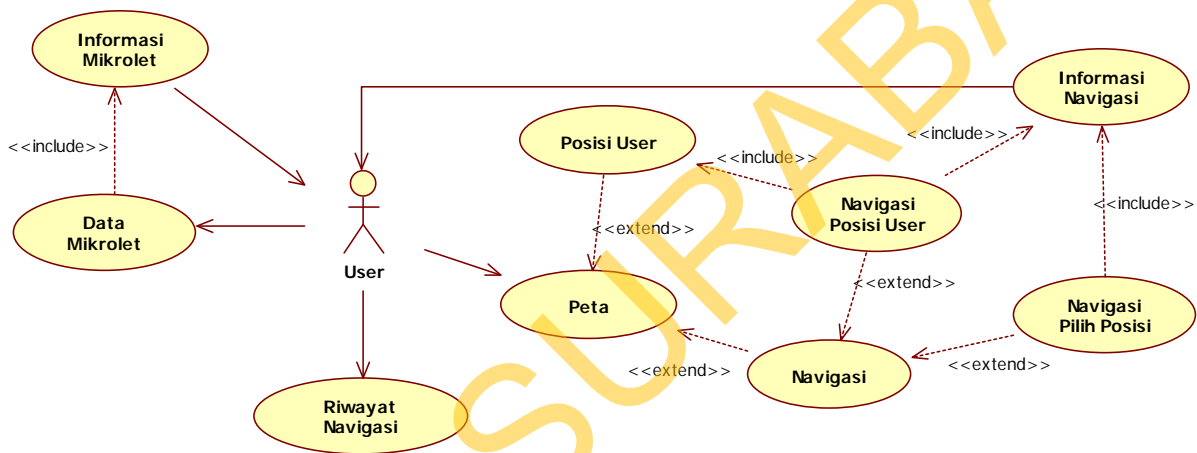
3. Koneksi internet pada *smartphone* Android.

Aplikasi yang dibuat menggunakan peta dari *google map* sehingga aplikasi selalu membutuhkan koneksi internet untuk mengakses *google map*.

3.4. Perancangan Sistem

3.4.1 Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. Gambar 3.1 menunjukkan *use case* pada Aplikasi Pemilihan Penggunaan Transportasi Umum Mikrolet di Malang Berbasis Android.



Gambar 3.1 Use Case Diagram Aplikasi Pemilihan Penggunaan Transportasi Umum Mikrolet di Malang Berbasis Android

Berikut adalah penjelasan singkat dari masing-masing *use case* yang dimiliki aplikasi pemilihan penggunaan transportasi umum mikrolet di Malang berbasis Android.

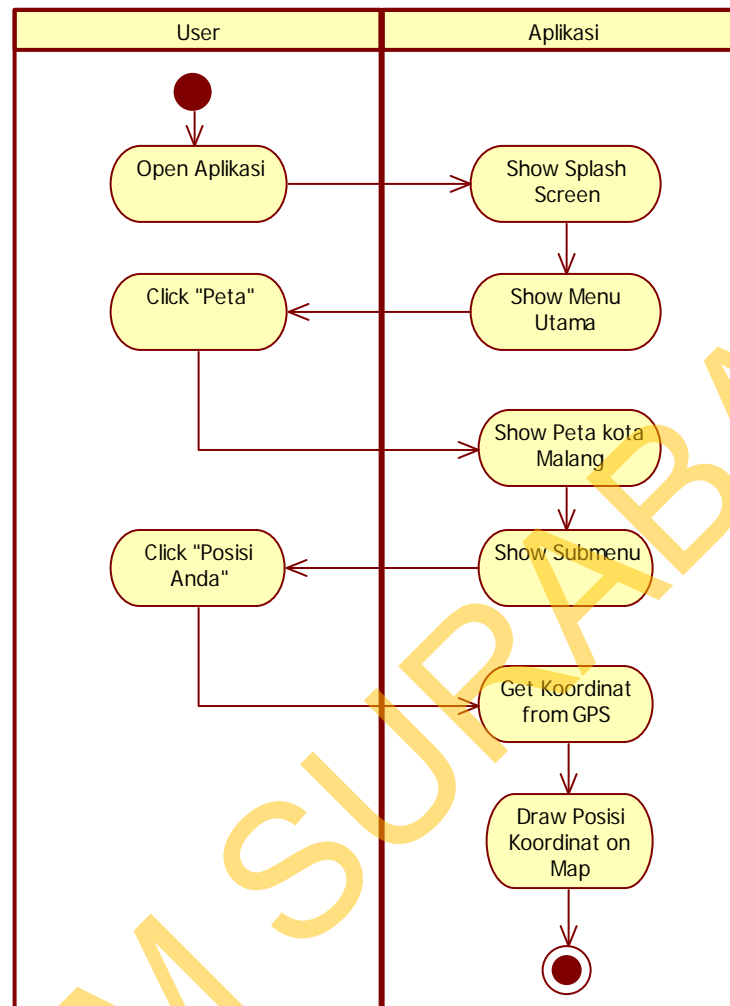
Tabel 3.1 Penjelasan Singkat *Use Case* Aplikasi Pemilihan Penggunaan Transportasi Umum Mikrolet di Malang Berbasis Android.

Nama <i>Use Case</i>	Penjelasan
Peta	Proses untuk menampilkan peta kota Malang. Pada proses ini akan ditampilkan beberapa opsi seperti navigasi dan posisi user.
Posisi User	Proses yang menangani penentuan posisi user dengan menggunakan koordinat yang didapat dari GPS pada <i>device</i> Android.
Navigasi	Proses untuk memulai navigasi
Navigasi Posisi User	Proses yang menangani navigasi berdasarkan posisi user berada.
Navigasi Pilih Posisi	Proses yang menangani navigasi berdasarkan posisi awal yang ditentukan manual oleh user.
Informasi Navigasi	Proses yang berfungsi untuk menampilkan informasi navigasi.
Data Mikrolet	Proses yang berfungsi untuk melihat data transportasi mikrolet yang ada di Kota Malang.
Informasi Mikrolet	Proses yang berfungsi untuk menampilkan informasi mikrolet.
Riwayat Navigasi	Proses yang berfungsi menampilkan daftar navigasi yang pernah dilakukan oleh user.

3.4.2 *Activity Diagram*

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity Diagram* adalah salah satu bentuk diagram UML yang mudah dimengerti karena diagram ini memiliki simbol yang menyerupai simbol *flowchart*, yang sangat berguna untuk menjelaskan langkah-langkah proses yang dilakukan.

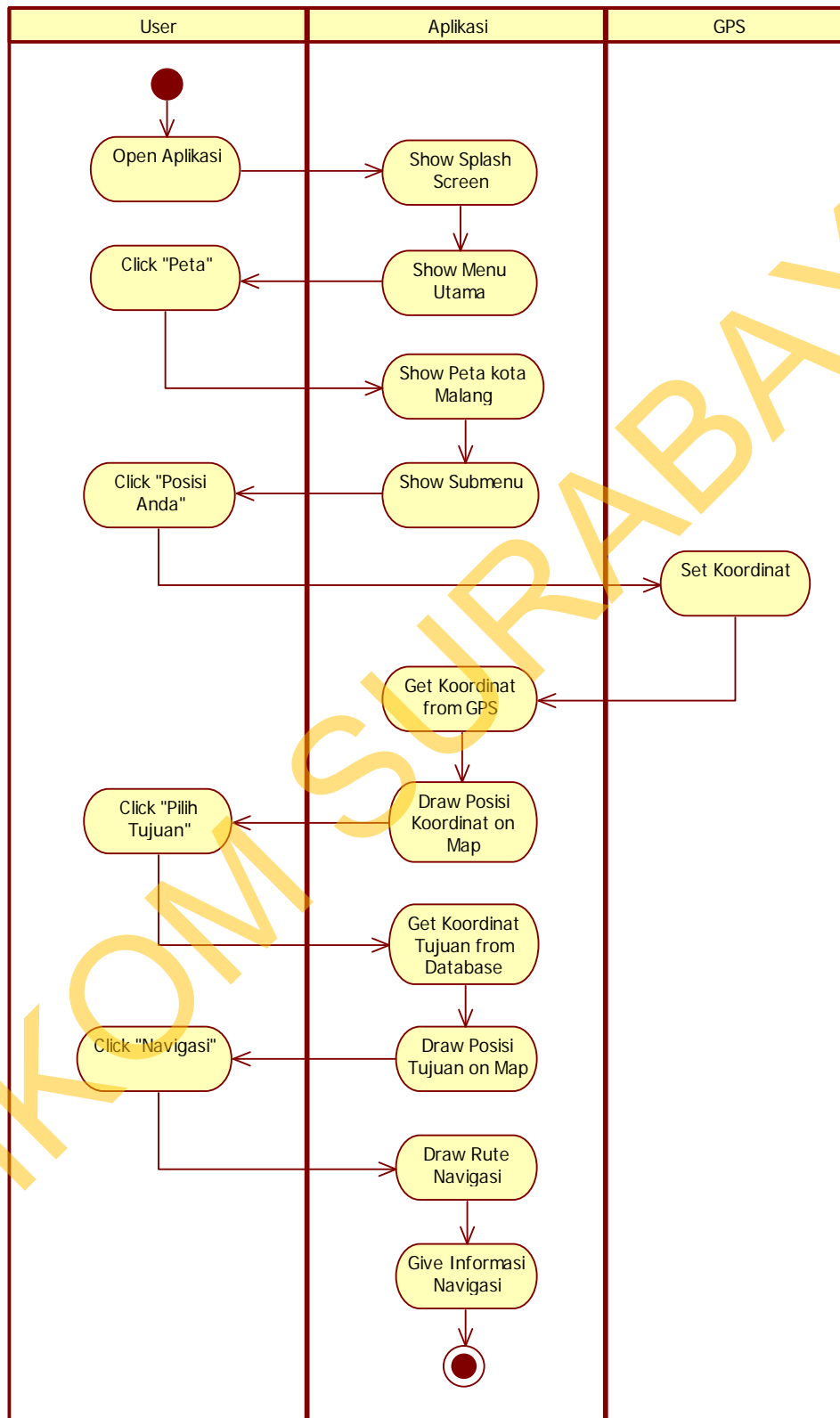
3.4.2.1 Activity Diagram Penentuan Posisi User



Gambar 3.2 Activity Diagram Penentuan Posisi User

Pada gambar 3.2 alur proses *activity diagram* posisi user ini dimulai ketika user menekan tombol “Posisi Anda” pada submenu di dalam peta kota Malang. Setelah user menekan tombol “Posisi Anda”, aplikasi akan melakukan pengecekan apakah GPS yang terdapat pada *device* Android sudah me-*lock* posisi user. Jika posisi sudah terkunci, aplikasi akan mengambil data koordinat dari GPS tersebut. Setelah data koordinat didapat, aplikasi akan menggambar koodinat tersebut ke peta berupa simbol posisi user berada.

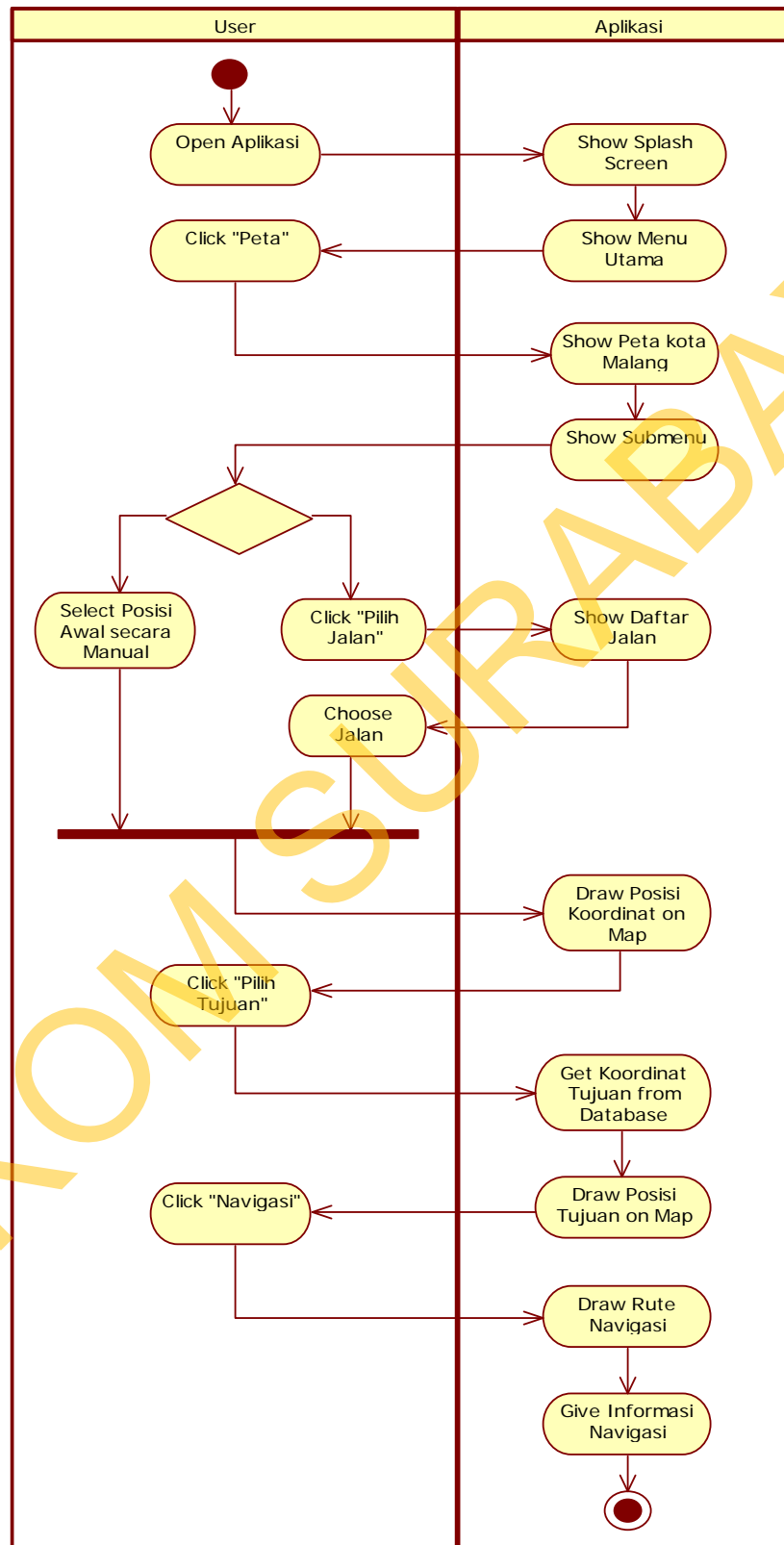
3.4.2.2 Activity Diagram Navigasi GPS



Gambar 3.3 Activity Diagram Navigasi GPS

Pada gambar 3.3 alur proses pada *activity diagram* navigasi GPS ini dimulai ketika user menekan tombol navigasi dengan posisi user sebagai posisi awalnya. Pertama kali yang dilakukan user adalah menekan tombol “Posisi Anda”, kemudian aplikasi akan mengambil data koordinat dari GPS yang terdapat pada *device* android. Aplikasi akan memberikan peringatan jika posisi user terlalu jauh dari jalan yang dilewati mikrolet yaitu nama jalan pada posisi awal tidak akan menampilkan nama jalan user berada. Setelah itu, user menekan tombol “Pilih Tujuan” untuk memilih tujuan yang diinginkan, setelah itu aplikasi akan menggambar simbol posisi tujuan yang dipilih. Kemudian user menekan tombol “Navigasi”, setelah itu aplikasi akan memproses data koordinat posisi user berada dan data koordinat tujuan. Kemudian aplikasi akan menggambar rute pada peta dan memberikan informasi mikrolet yang dinaiki dan jalan yang dilewati. Jika dalam navigasi terdapat pergantian mikrolet, aplikasi akan memberikan informasi jalan dimana user harus ganti mikrolet dan mikrolet yang dinaiki kemudian.

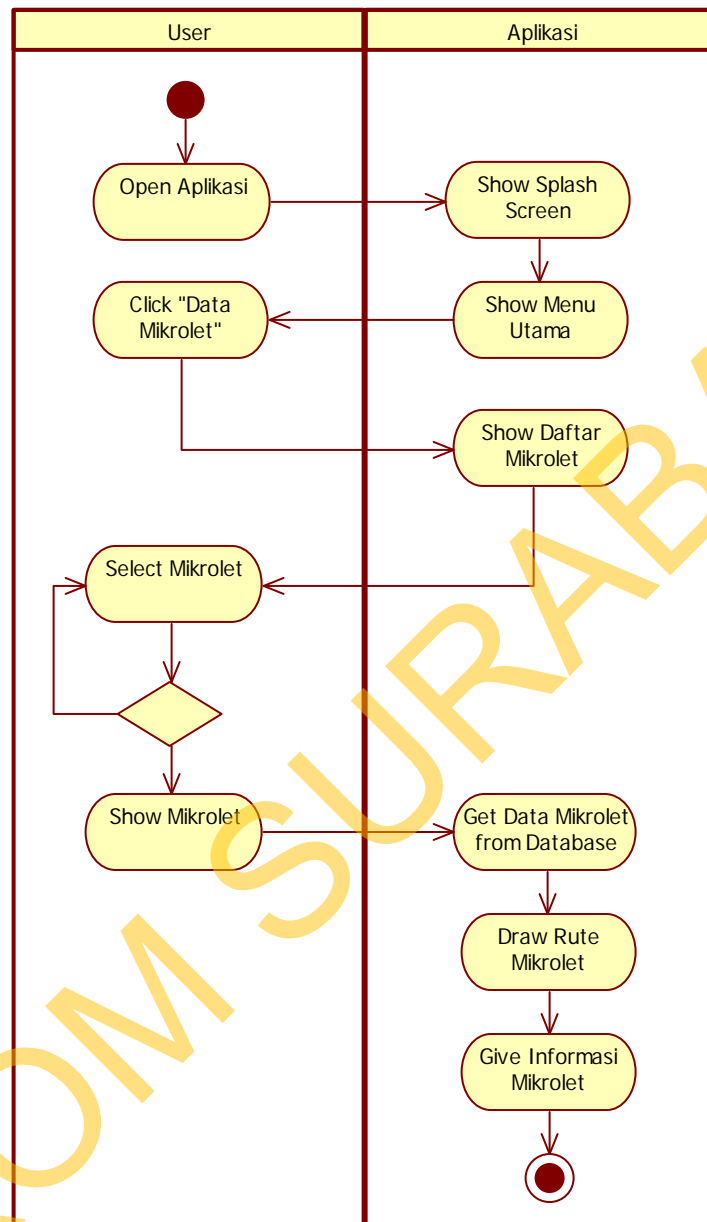
3.4.2.3 Activity Diagram Navigasi Manual



Gambar 3.4 Activity Diagram Navigasi Manual

Pada gambar 3.4 alur proses *activity diagram* navigasi manual ini dimulai ketika user memilih menu “Peta” pada menu utama. Setelah itu aplikasi akan menampilkan peta kota Malang dimana user dapat men-*tap* peta tersebut. Ketika peta di-*tap* aplikasi akan menggambar simbol untuk menunjukkan posisi awal. Jika user tidak menginginkan untuk men-*tap* peta, user dapat menggunakan opsi cari jalan untuk mencari jalan sebagai posisi awal user berada. Setelah menentukan posisi awal, maka aplikasi akan menampilkan daftar tujuan. Setelah user memilih tujuan yang diinginkan, aplikasi akan menggambar rute pada peta dan memberikan informasi mikrolet yang dinaiki dan jalan yang dilewati. Jika terdapat pergantian mikrolet, aplikasi akan memberikan informasi jalan dimana user harus ganti mikrolet.

3.4.2.4 Activity Diagram Data Mikrolet



Gambar 3.5 Activity Diagram Data Mikrolet

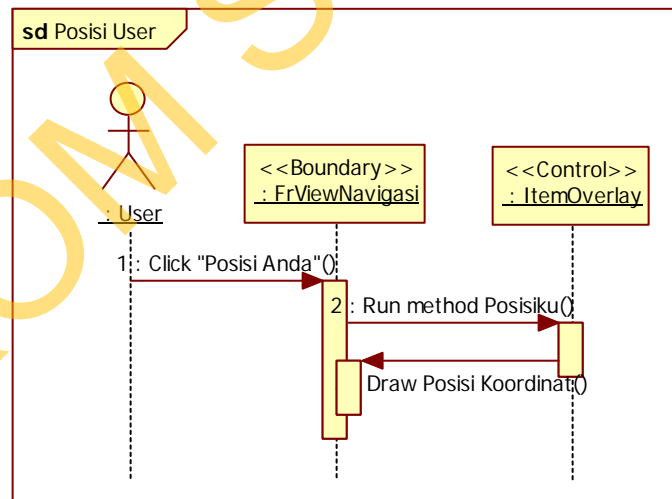
Pada gambar 3.5 alur proses pada *activity diagram* data mikrolet ini dimulai ketika user memilih menu “Data Mikrolet” pada menu utama. Setelah itu aplikasi akan menampilkan daftar mikrolet di kota Malang. Pada daftar mikrolet ini, user dapat memilih salah satu mikrolet untuk ditampilkan rute dan informasi jalan yang

dilewati. Setelah user memilih salah satu mikrolet, aplikasi akan memberikan konfirmasi apakah user ingin melihat rute mikrolet tersebut atau tidak. Jika tidak, user dapat memilih data mikrolet yang lain. Jika user memilih untuk menampilkan rute, maka aplikasi akan menampilkan rute mikrolet pada peta.

3.4.3 Sequence Diagram

Sequence Diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem berupa *message* yang digambarkan terhadap waktu. *Sequence Diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence Diagram* dapat menggambarkan jalannya suatu proses yang melibatkan objek dari *class* dalam aplikasi.

3.4.3.1 Sequence Diagram Posisi User

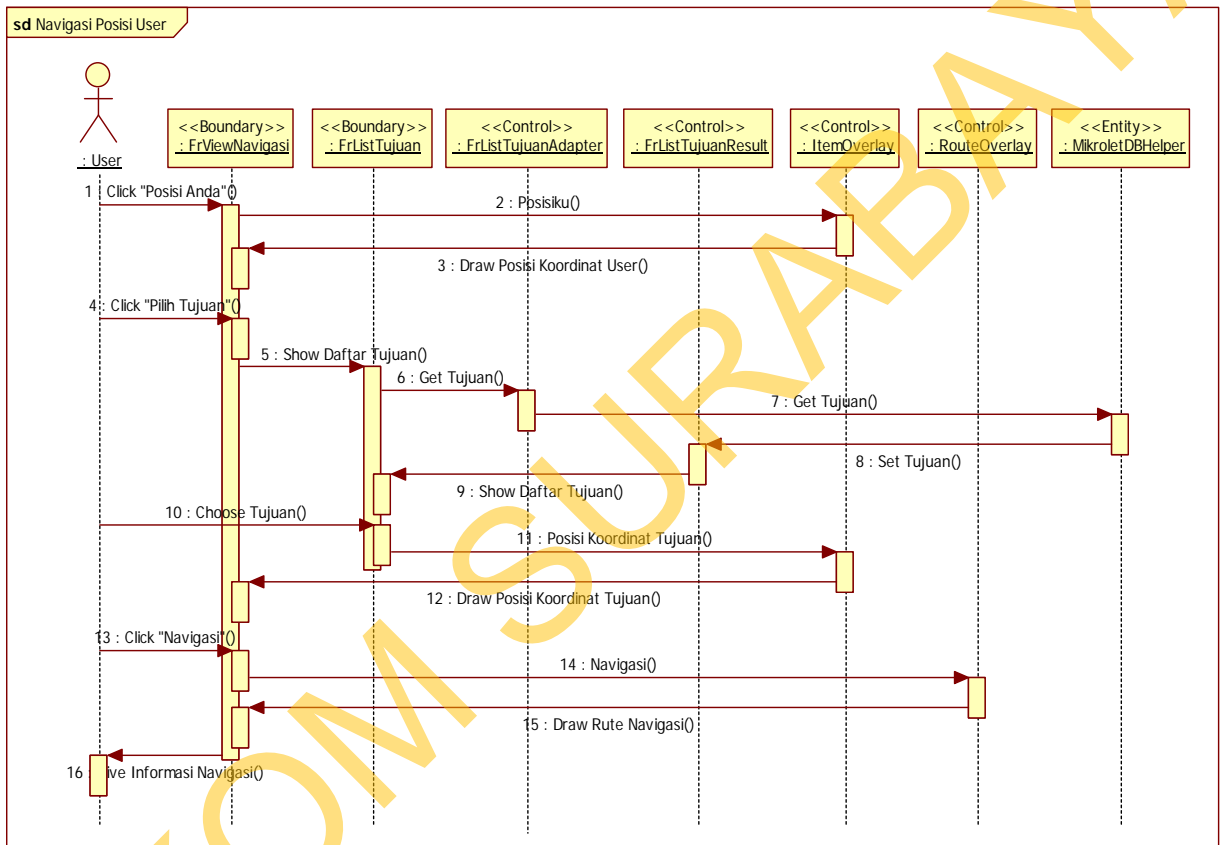


Gambar 3.6 *Sequence Diagram* Posisi User

Gambar 3.6 *Sequence diagram* posisi user menjelaskan bagaimana proses ketika user menekan tombol “Posisi Anda”. Setelah itu, aplikasi akan mengambil

data koordinat dari GPS. Setelah data koordinat didapat, aplikasi akan memanggil class *item overlay* untuk menggambar simbol pada peta sesuai koordinat dari GPS pada *device* Android.

3.4.3.2 Sequence Diagram Navigasi GPS

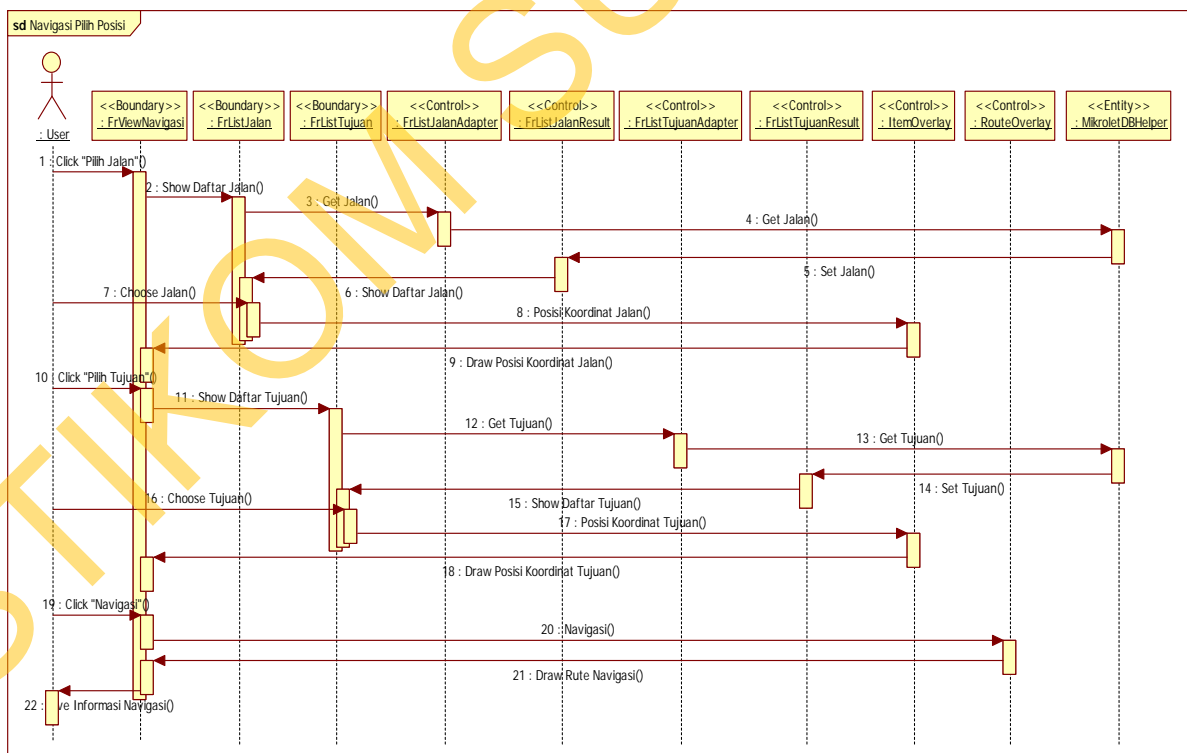


Gambar 3.7 Sequence Diagram Navigasi GPS

Sequence diagram navigasi GPS menjelaskan bagaimana proses ketika user melakukan navigasi dengan posisi awal adalah posisi user berdasarkan koordinat GPS. Pertama kali, user menekan tombol "Posisi Anda" kemudian aplikasi akan mengambil koordinat berdasarkan GPS pada *device* Android dan aplikasi memanggil class *itemOverlay* untuk menggambar simbol posisi user. Setelah itu,

aplikasi akan melakukan pengecekan apakah jalan tempat user berada dilewati mikrolet atau tidak. Jika dilewati mikrolet, maka akan muncul nama jalan tempat user berada. Kemudian user menekan tombol “Pilih Tujuan” untuk memilih tujuan yang diinginkan. Setelah tujuan dipilih aplikasi akan memanggil *class itemOverlay* untuk menggambar simbol tujuan yang dipilih. Kemudian user menekan tombol “Navigasi”, setelah itu aplikasi akan memanggil *class routeOverlay* untuk menggambar rute navigasi dari posisi user ke posisi tujuan. Aplikasi juga memberikan informasi mikrolet yang dinaiki dan jalan yang dilewati. Jika terdapat pergantian mikrolet, aplikasi juga memberikan informasi jalan dimana harus ganti mikrolet dan mikrolet apa yang dinaiki kemudian.

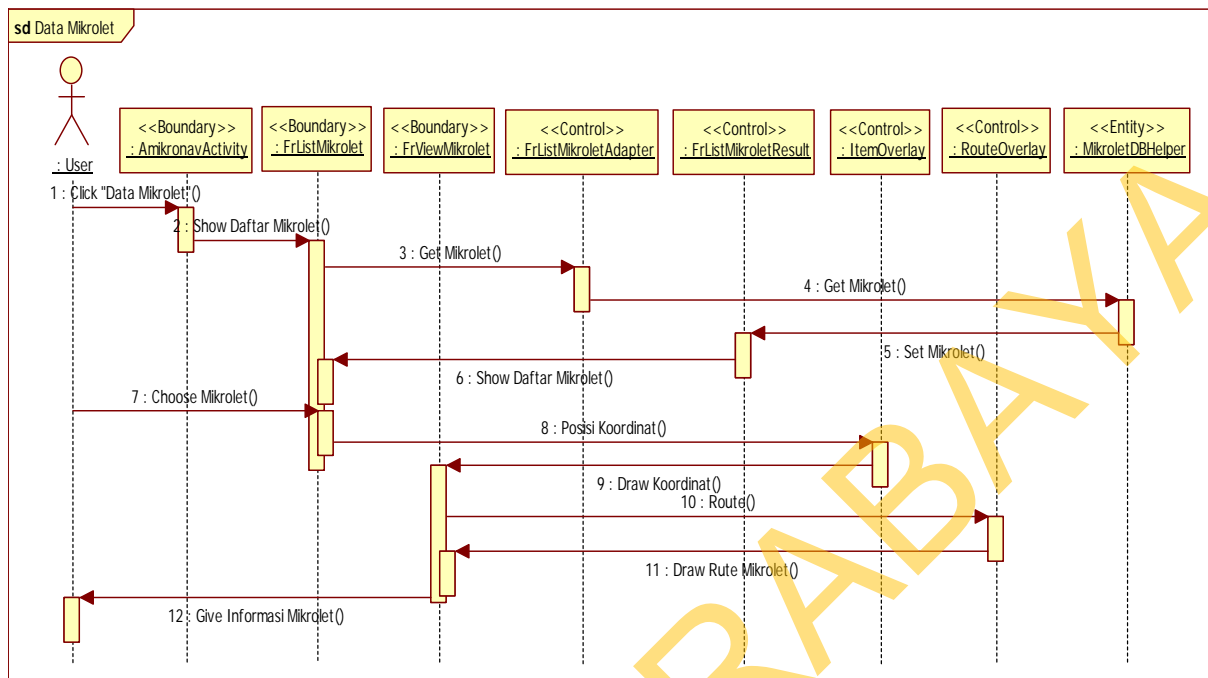
3.4.3.3 Sequence Diagram Navigasi Manual



Gambar 3.8 Sequence Diagram Navigasi Manual

Sequence diagram navigasi manual menjelaskan proses ketika user memilih menu “peta” pada menu utama. Kemudian user dapat menentukan posisi awalnya dengan cara menekan langsung pada peta atau menekan tombol “Pilih Jalan”, jika user menekan langsung pada peta user dapat secara bebas menentukan posisi awalnya. Tetapi jika user menekan tombol “Pilih Jalan”, user dapat memilih jalan yang diinginkan. Setelah jalan terpilih aplikasi akan memanggil *class itemOverlay* untuk menggambar simbol posisi jalan tersebut. Kemudian user menekan tombol “Pilih Tujuan” untuk menampilkan daftar tujuan. Setelah user memilih tujuan yang diinginkan, aplikasi akan memanggil *class itemOverlay* untuk menggambar simbol posisi tujuan tersebut. Kemudian user menekan tombol “Navigasi”, setelah itu aplikasi akan memanggil *class routeOverlay* untuk menggambar rute navigasi dari posisi user ke posisi tujuan. Aplikasi juga memberikan informasi mikrolet yang dinaiki dan jalan yang dilewati. Jika terdapat pergantian mikrolet, aplikasi juga memberikan informasi jalan dimana harus ganti mikrolet dan mikrolet apa yang dinaiki kemudian.

3.4.3.4 Sequence Diagram Data Mikrolet



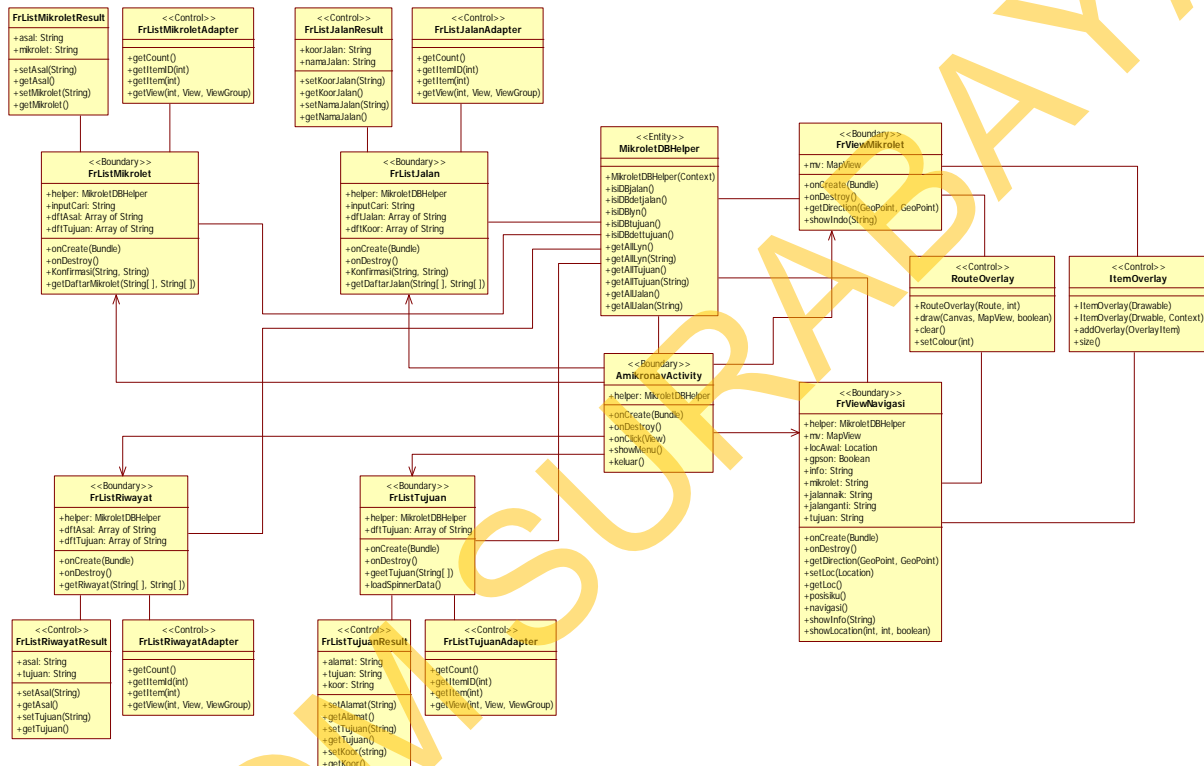
Gambar 3.9 Sequence Diagram Data Mikrolet

Sequence diagram data mikrolet menjelaskan bagaimana ketika user memilih menu “Data Mikrolet” pada menu utama. Setelah user memilih, aplikasi akan melakukan *load* data mikrolet dan menampilkannya pada daftar mikrolet. Disini user dapat memilih mikrolet mana yang ingin dilihat datanya. Ketika user mengklik salah satu nama mikrolet aplikasi akan menampilkan konfirmasi. Konfirmasi ini berisi dua tombol, tombol pertama berfungsi untuk menampilkan rute mikrolet sedangkan tombol kedua untuk keluar dari konfirmasi. Jika user memilih untuk menampilkan rute mikrolet, maka aplikasi akan memanggil *class routeoverlay* untuk menggambar rute mikrolet dan *class itemoverlay* untuk menggambar simbol-simbol yang dibutuhkan.

3.4.4 Class Diagram

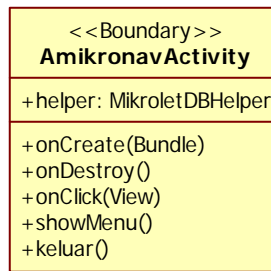
Class diagram menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan antar class. Gambar 3.10 menunjukkan class diagram

Aplikasi Pemilihan Penggunaan Transportasi Umum Mikrolet di Malang.



Gambar 3.10 Class Diagram Aplikasi Pemilihan Penggunaan Transportasi Umum Mikrolet di Malang

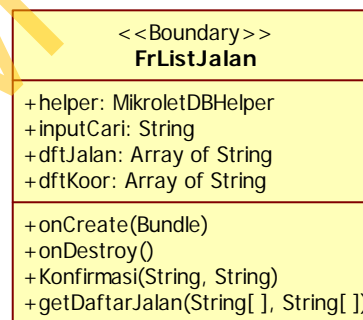
3.4.4.1 Class AmikronavActivity



Gambar 3.11 Class AmikronavActivity

Class AmikronavActivity merupakan *class boundary* yang digunakan sebagai class utama pada aplikasi. Pada *class* ini terdapat *method* *showMenu()* yang digunakan untuk membuat dan menampilkan menu utama aplikasi. Sedangkan *method* *keluar()* digunakan untuk fungsi keluar dari aplikasi. *Class* ini merupakan *class* induk yang dapat memanggil beberapa form lainnya.

3.4.4.2 Class FrListJalan

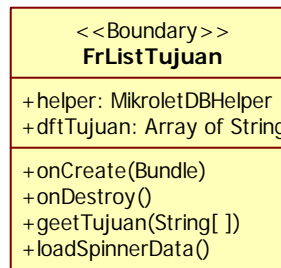


Gambar 3.12 Class FrListJalan

Class ListJalan merupakan *class boundary* yang digunakan sebagai penampil daftar jalan. Pada *class* ini terdapat *method* *getDaftarJalan()* untuk

mengambil data jalan dari database dan method *konfirmasi()* yang berfungsi untuk mengkonfirmasi jalan yang dipilih user.

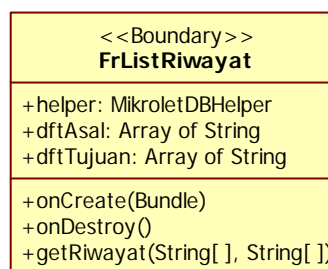
3.4.4.3 Class FrListTujuan



Gambar 3.13 Class ListTujuan

Class ListTujuan merupakan *class boundary* yang digunakan sebagai penampil daftar tujuan. Pada *class* ini terdapat method *getTujuan()* untuk mengambil data tujuan dari database dan method *loadSpinnerData()* untuk mengisi combobox berdasarkan penggolongan tujuan.

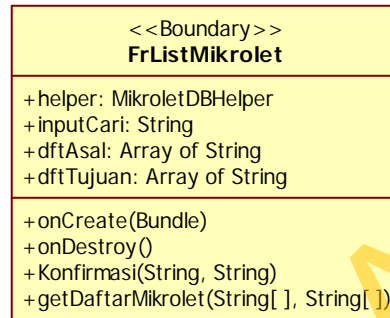
3.4.4.4 Class FrListRiwayat



Gambar 3.14 Class FrListRiwayat

Class ListRiwayat merupakan *class boundary* yang digunakan sebagai penampil daftar riwayat navigasi yang pernah dilakukan. Pada *class* ini terdapat method *getRiwayat()* untuk mengambil data riwayat dari database.

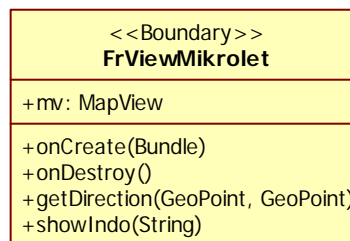
3.4.4.5 *Class* FrListMikrolet



Gambar 3.15 *Class* FrListMikrolet

Class ListMikrolet merupakan *class boundary* yang digunakan sebagai penampil daftar mikrolet. Pada *class* ini terdapat method *getDaftarMikrolet()* untuk mengambil data mikrolet dari database dan method *konfirmasi()* untuk mengkonfirmasi tujuan yang dipilih user.

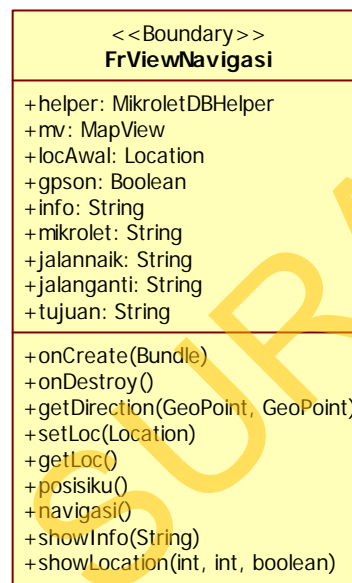
3.4.4.6 *Class* FrViewMikrolet



Gambar 3.16 *Class* FrViewMikrolet

Class FrViewMikrolet merupakan *class boundary* yang digunakan sebagai penampil rute mikrolet. Pada *class* ini terdapat method *getDirection()* untuk menggambar rute mikrolet pada peta. Sedangkan method *showInfo()* berfungsi untuk menampilkan informasi mikrolet.

3.4.4.7 Class FrViewNavigasi

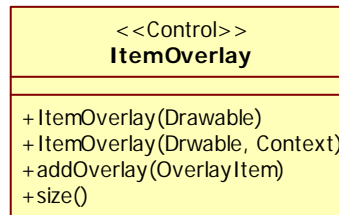


Gambar 3.17 *Class FrViewNavigasi*

Class FrViewNavigasi merupakan *class boundary* yang digunakan sebagai penampil peta pada aplikasi. Pada *class* ini terdapat berbagai macam opsi yang dapat digunakan oleh user. Atribut yang terdapat pada *class* ini digunakan untuk keperluan menggambar rute navigasi dan menampilkan informasi navigasi yang dibutuhkan user. Beberapa fungsi dari method *class* ini yaitu, *getDirection()* berfungsi untuk menggambar rute navigasi, *setLoc()* berfungsi untuk mengganti atribut *locAwal*, *getLoc()* berfungsi untuk mengambil nilai *locAwal*, *posisiku()* berfungsi untuk menggambar simbol posisi user berdasarkan koordinat dari GPS,

navigasi() merupakan method untuk melakukan proses navigasi, *showInfo()* berfungsi untuk menampilkan informasi navigasi dan *showLocation()* berfungsi untuk menggambar simbol-simbol pada peta.

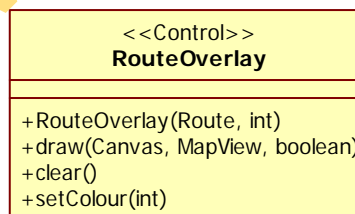
3.4.4.8 Class ItemOverlay



Gambar 3.18 Class ItemOverlay

Class ItemOverlay merupakan *class control* yang digunakan sebagai penggambar dari simbol-simbol yang ditampilkan pada peta. Pada *class* ini terdapat *method* yang berfungsi untuk menggambar simbol pada peta.

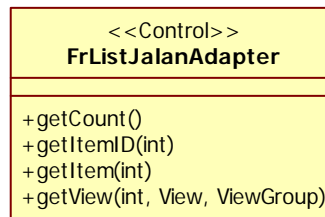
3.4.4.9 Class RouteOverlay



Gambar 3.19 Class RouteOverlay

Class RouteOverlay merupakan *class control* yang digunakan sebagai penggambar dari rute yang ditampilkan pada peta. Pada *class* ini terdapat *method* yang berfungsi untuk menggambar rute pada peta.

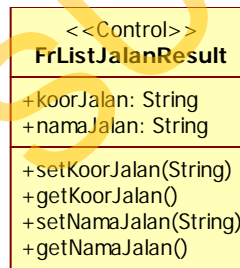
3.4.4.10 Class FrListJalanAdapter



Gambar 3.20 Class FrListJalanAdapter

Class FrListJalanAdapter merupakan *class control* yang berfungsi sebagai penghubung antara *class* FrListJalan dengan database. Class ini digunakan untuk mengatur penampilan tiap data jalan yang ditampilkan melalui *listbox*.

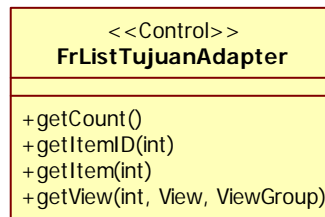
3.4.4.11 Class FrListJalanResult



Gambar 3.21 Class FrListJalanResult

Class FrListJalanResult merupakan *class control* yang berisi method *setter* dan method *getter*. Class ini berfungsi untuk mendukung *class* FrListJalan dan *class* FrListJalanAdapter.

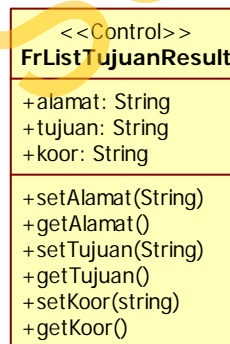
3.4.4.12 Class FrListTujuanAdapter



Gambar 3.22 Class FrListTujuanAdapter

Class FrListTujuanAdapter merupakan *class control* yang berfungsi sebagai penghubung antara *class FrListTujuan* dengan database. Class ini digunakan untuk mengatur penampilan tiap data tujuan yang ditampilkan melalui *listbox*.

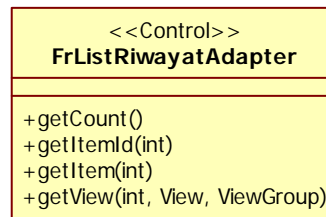
3.4.4.13 Class FrListTujuanResult



Gambar 3.23 Class FrListTujuanResult

Class FrListTujuanResult merupakan *class control* yang berisi method *setter* dan method *getter*. Class ini berfungsi untuk mendukung *class FrListTujuan* dan *class FrListTujuanAdapter*.

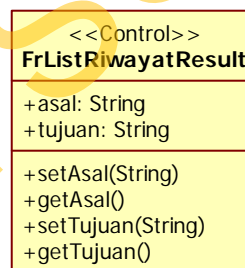
3.4.4.14 Class FrListRiwayatAdapter



Gambar 3.24 Class FrListRiwayatAdapter

Class FrListRiwayatAdapter merupakan *class control* yang berfungsi sebagai penghubung antara *class* FrListRiwayat dengan database. Class ini digunakan untuk mengatur penampilan tiap data riwayat navigasi yang ditampilkan melalui *listbox*.

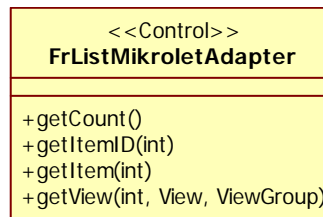
3.4.4.15 Class FrListRiwayatResult



Gambar 3.25 Class FrListRiwayatResult

Class FrListRiwayatResult merupakan *class control* yang berisi method *setter* dan method *getter*. Class ini berfungsi untuk mendukung *class* FrListRiwayat dan *class* FrListRiwayatAdapter.

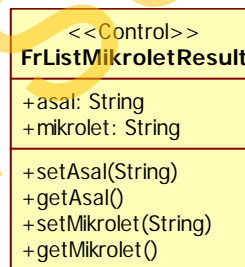
3.4.4.16 Class FrListMikroletAdapter



Gambar 3.26 Class FrListMikroletAdapter

Class FrListMikroletAdapter merupakan *class control* yang berfungsi sebagai penghubung antara *class* FrListMikrolet dengan database. Class ini digunakan untuk mengatur penampilan tiap data mikrolet yang ditampilkan melalui *listbox*.

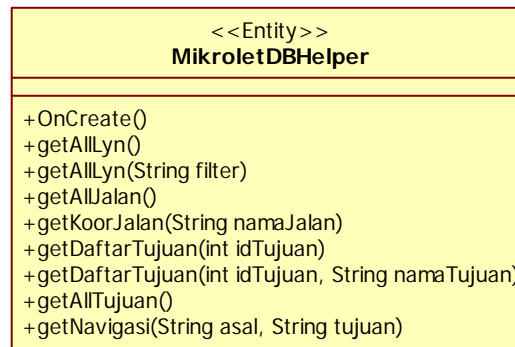
3.4.4.17 Class FrListMikroletResult



Gambar 3.27 Class FrListMikroletResult

Class FrListMikroletResult merupakan *class control* yang berisi method *setter* dan method *getter*. Class ini berfungsi untuk mendukung *class* FrListMikrolet dan *class* FrListMikroletAdapter.

3.4.4.18 Class MikroletDBHelper



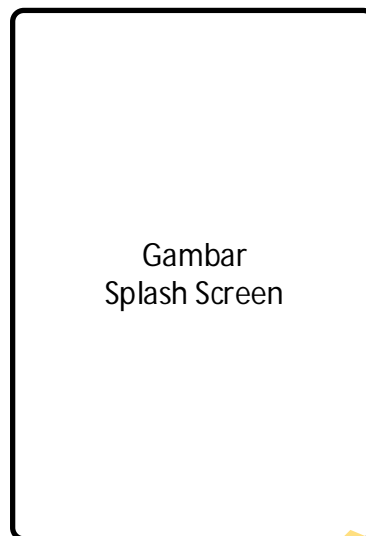
Gambar 3.28 Class MikroletDBHelper

Class MikroletDBHelper digunakan sebagai penghubung antara aplikasi dengan database. Pada class ini terdapat *method-method* yang dapat digunakan untuk mengirim *query* ke database.

3.5. Desain Antarmuka

Perancangan antarmuka ini memvisualisasikan tampilan dari aplikasi. Perancangan antarmuka berfungsi untuk merancang antarmuka agar aplikasi dapat berinteraksi dengan user. Perancangan antarmuka Aplikasi Pemilihan Penggunaan Transportasi Umum Mikrolet di Malang Berbasis Android dibagi menjadi beberapa bagian sebagai berikut:

3.5.1 Rancangan Tampilan Splash Screen



Gambar 3.29 Rancangan Tampilan *Splash Screen*

Rancangan tampilan *splash screen* adalah tampilan aplikasi ketika pertama kali dijalankan. Pada tampilan ini hanya berisi gambar yang menggambarkan aplikasi. Form tampilan ini hanya dijalankan selama tiga detik.

3.5.2 Rancangan Menu Utama



Gambar 3.30 Rancangan Menu Utama

Rancangan menu utama adalah rancangan tampilan menu utama pada aplikasi. Pada menu utama terdapat beberapa tombol, tombol info berfungsi untuk menjelaskan fungsi tombol disebelah kirinya. Tombol peta digunakan untuk memanggil form peta. Tombol data mikrolet digunakan untuk melihat data mikrolet di kota Malang. Tombol riwayat navigasi digunakan untuk melihat daftar riwayat navigasi yang pernah dilakukan oleh user. Tombol petunjuk berfungsi untuk melihat beberapa fitur aplikasi. Tombol tentang berfungsi untuk melihat informasi tentang aplikasi. Dan tombol keluar berfungsi untuk keluar dari aplikasi.

3.5.3 Rancangan Form Peta

The image shows a wireframe for a map form. It consists of a rectangular frame containing several elements:

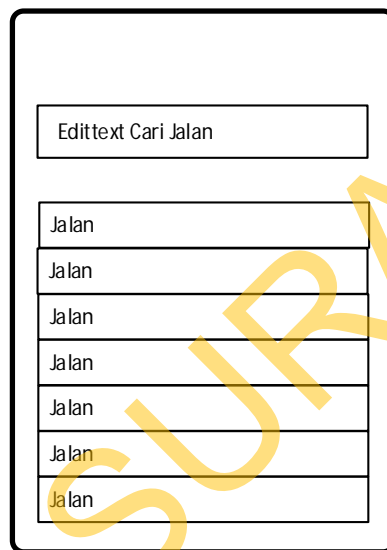
- Top row: Three input fields labeled "Posisi Awal", "Cari Jalan", and "Posisi Anda".
- Second row: Three input fields labeled "Posisi Tujuan", "Cari Tujuan", and "Navigasi".
- Center: A large rectangular area labeled "Mapview".
- Bottom left: A "Text" input field.
- Bottom center: A zoom control consisting of a minus sign "-" and a plus sign "+" in separate boxes.
- Bottom right: A compass rose icon with "N", "S", "E", and "W" labels.

Gambar 3.31 Rancangan Form Peta

Rancangan form peta merupakan tampilan peta utama aplikasi. Peta akan menunjukkan kota Malang. Pada form ini terdapat beberapa tombol, yaitu tombol cari jalan yang digunakan untuk memilih jalan, tombol cari tujuan yang digunakan

untuk memilih tujuan, tombol posisi anda untuk menentukan posisi user pada peta dan tombol navigasi untuk menggambar rute navigasi dari posisi awal ke posisi tujuan. Pada form ini juga terdapat dua buah *textbox*, *textbox* posisi awal sebagai nama jalan posisi awal dan *textbox* posisi tujuan sebagai nama tujuan.

3.5.4 Rancangan Form Daftar Jalan

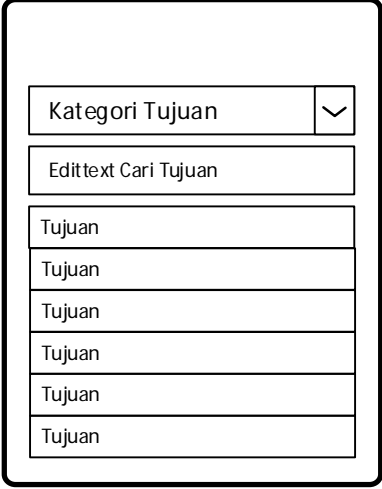


The image shows a wireframe of a form titled 'Rancangan Form Daftar Jalan'. It consists of a rectangular container with a black border. Inside the container, there is a text input field at the top with the placeholder text 'Edittext Cari Jalan'. Below this field is a list box containing seven entries, each labeled 'Jalan'.

Gambar 3.32 Rancangan Form Daftar Jalan

Rancangan form daftar jalan merupakan rancangan tampilan yang menampilkan daftar jalan. Pada form ini terdapat satu buah *textbox* dan sebuah *list*. *Textbox* digunakan sebagai pencari jalan sedangkan *list* digunakan sebagai penampil daftar jalan.

3.5.5 Rancangan Form Daftar Tujuan

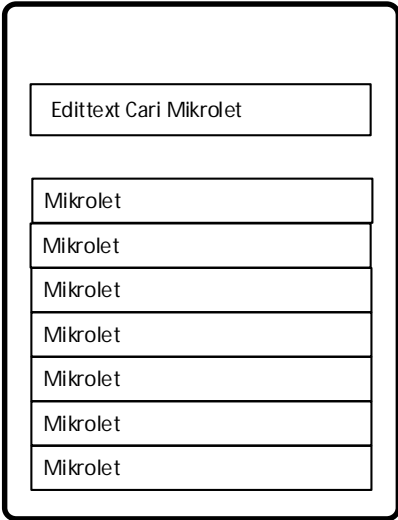


The image shows a wireframe for a 'Daftar Tujuan' (Goal List) form. It features a dropdown menu labeled 'Kategori Tujuan' with a downward arrow, a search input field labeled 'Edittext Cari Tujuan', and a list view containing seven entries, each labeled 'Tujuan'.

Gambar 3.33 Rancangan Form Daftar Tujuan

Rancangan form daftar tujuan adalah rancangan tampilan yang menampilkan daftar tujuan yang dapat dipilih oleh user. Rancangan ini berisi listview yang menampilkan data tujuan, sebuah *combobox* untuk memilih kategori tujuan dan sebuah *textbox* yang dapat digunakan untuk mencari tujuan yang diinginkan.

3.5.6 Rancangan Form Daftar Mikrolet



The image shows a wireframe for a 'Daftar Mikrolet' (Microlet List) form. It features a search input field labeled 'Edittext Cari Mikrolet' and a list view containing seven entries, each labeled 'Mikrolet'.

Gambar 3.34 Rancangan Form Daftar Mikrolet

Rancangan form daftar mikrolet adalah rancangan tampilan yang menampilkan daftar mikrolet yang ada di kota Malang. Rancangan ini berisi listview yang menampilkan data mikrolet dan sebuah *textbox* yang dapat digunakan untuk mencari mikrolet yang diinginkan.

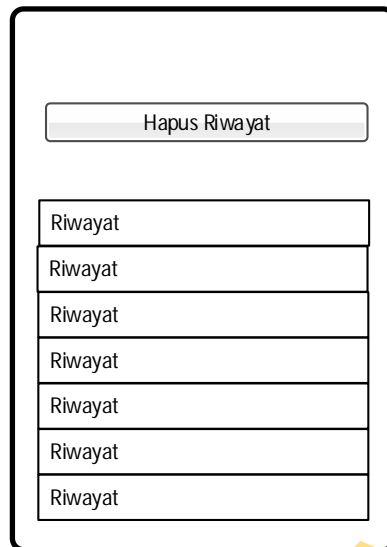
3.5.7 Rancangan Form View Mikrolet



Gambar 3.35 Rancangan Form View Mikrolet

Rancangan form view mikrolet merupakan tampilan aplikasi ketika menampilkan rute mikrolet yang dipilih oleh user. Pada form ini terdapat dua opsi tambahan yaitu, informasi yang digunakan untuk menampilkan informasi mikrolet dan map legenda untuk menampilkan legenda peta.

3.5.8 Rancangan Form Daftar Riwayat Navigasi



The image shows a wireframe of a 'Daftar Riwayat Navigasi' (Navigation History List) form. At the top, there is a button labeled 'Hapus Riwayat' (Delete History). Below the button is a vertical list of seven rectangular boxes, each containing the text 'Riwayat' (History), representing individual navigation entries.

Gambar 3.36 Rancangan Form Daftar Riwayat Navigasi

Rancangan form daftar riwayat navigasi merupakan tampilan form yang berfungsi untuk menampilkan daftar navigasi yang pernah dilakukan oleh user. Pada form ini terdapat satu buah *button* dan satu buah *list*. *Button* berfungsi untuk menghapus riwayat dan *list* berfungsi untuk menampilkan riwayat navigasi.