

BAB III

METODOLOGI PENELITIAN

3.1 Metode Pengembangan Perangkat Lunak

Metode pengembangan perangkat lunak yang digunakan yaitu metode pengembangan sistem *Multimedia Development Life Cycle* (MDLC) Tahapan yang dilakukan dalam pengembangan aplikasi ini adalah sebagai berikut : *Concept, Design, Material Colecting, Assembly, Testing, dan Distribution.*

3.1.1 Concept (Pengkonsepan)

Tahap *concept* (pengkonsepan) yaitu menentukan tujuan aplikasi dan spesifikasi umum. Dasar aturan untuk perancangan juga ditentukan pada tahap ini, seperti analisis kebutuhan aplikasi, rancangan aplikasi, ukuran aplikasi, target, dapat terpenuhi seperti pada tahapan berikut ini :

1. Studi pustaka

Pengumpulan data dengan cara membaca, mempelajari dan memanfaatkan buku – buku yang tersedia di perpustakaan, artikel atau jurnal yang mendukung dalam penyusunan dan penulisan skripsi. Tujuan dari studi pustaka yaitu untuk menyusun dasar teori yang digunakan dalam melakukan penelitian.

2. Observasi

Melakukan pengamatan langsung terhadap data rambu - rambu lalu lintas yang ada di Dinas perhubungan Kota Metro.

3. Wawancara

Wawancara dilakukan secara langsung kepada staf bagian admin Dinas Perhubungan Kota Metro.

3.1.1.1 Analisis Kebutuhan Perangkat Lunak

Untuk membangun aplikasi visualisasi 3D, diperlukan beberapa jenis perangkat lunak, yaitu perangkat lunak pengolah gambar, perangkat lunak untuk mengolah objek 3D dan perangkat lunak pembangun aplikasi visualisasi 3D itu sendiri. Setelah mempelajari dan mempertimbangkan beberapa hal maka dipilihlah perangkat lunak sebagai berikut :

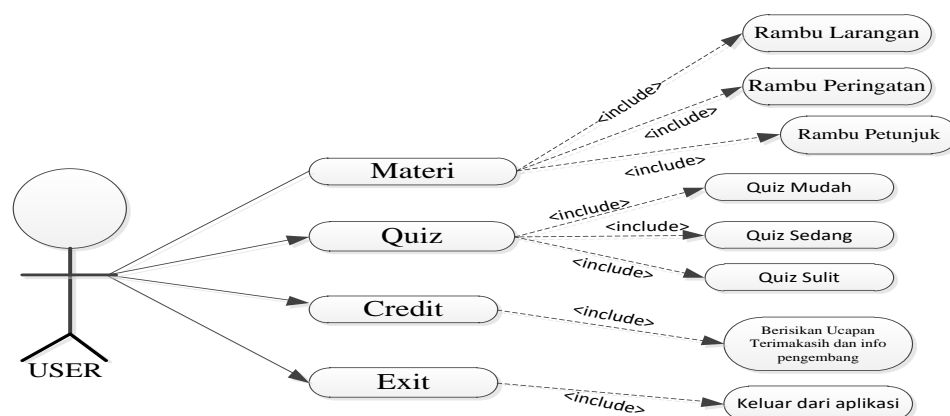
1. Sistem Operasi Windows 10
2. Google Sketchup Pro 2017
3. Unity 2019.2.2f1 (64-bit)

3.1.1.2 Rancangan Sistem Yang Diusulkan

Pada tahapan ini diuraikan tentang perancangan sistem yang akan dibuat untuk terwujudnya aplikasi yang diinginkan, dengan memodelkan permasalahan dalam bentuk diagram-diagram UML, diagram yang digunakan adalah *use case diagram* dan *activity diagram* karena lebih muda untuk dipahami. Berikut adalah penjelasan dari diagram - diagram UML yang digunakan :

1. Use Case Diagram

Use case diagram Dibawah ini menggambarkan sistem dari sudut pandang pengguna sistem tersebut (user) sehingga pembuatan *use case diagram* ini lebih dititik beratkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan alur atau urutan kejadian, sistem yang di usulkan akan di gambarkan dalam *use case diagram*, ditunjukkan pada Gambar 3.1.



Gambar 3.1 Use Case Diagram Sistem

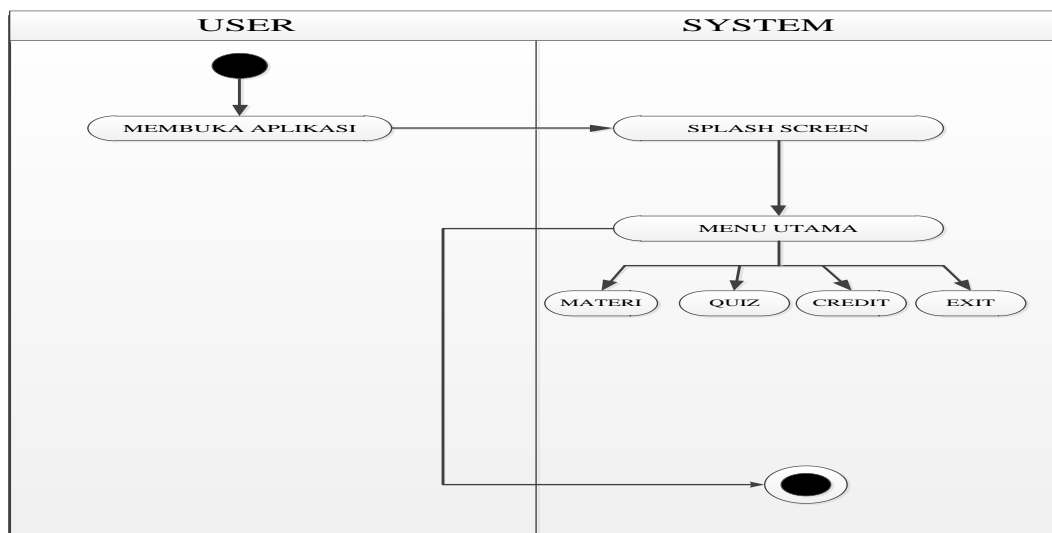
Dari gambar *use case* di atas dapat kita lihat bahwa pada saat *user* mengakses aplikasi, terdapat 4 menu utama yaitu : *Materi* , *Quiz* , *Credit* dan *Exit* . Pada pilihan menu *Materi* ,*user* akan di arahkan ke menu pilihan Materi yang disediakan kemudian *user* bisa memilih salah satu Materi yang diinginkan untuk memulai *Materi*, pada pilihan menu *Quiz*, *user* dapat memilih level yang diinginkan untuk memulai quiz. *Credit*,*user* akan diarahkan ke menu credit yang berisikan ucapan terimakasih dan info pengembang, terakhir pada pilihan menu *Exit* *user* dapat keluar dari aplikasi yang sedang dijalankan.

2. Activity Diagram

Activity Diagram digunakan untuk menggambarkan alur dari bagaimana suatu sistem mengawali, melakukan, dan mengakhiri proses tersebut bekerja. *Activity Diagram* yang diterapkan pada aplikasi Game Edukasi Pengenalan Rambu – Rambu Lalu Lintas Berbasis Android adalah sebagai berikut :

a) Activity Diagram Menu Utama

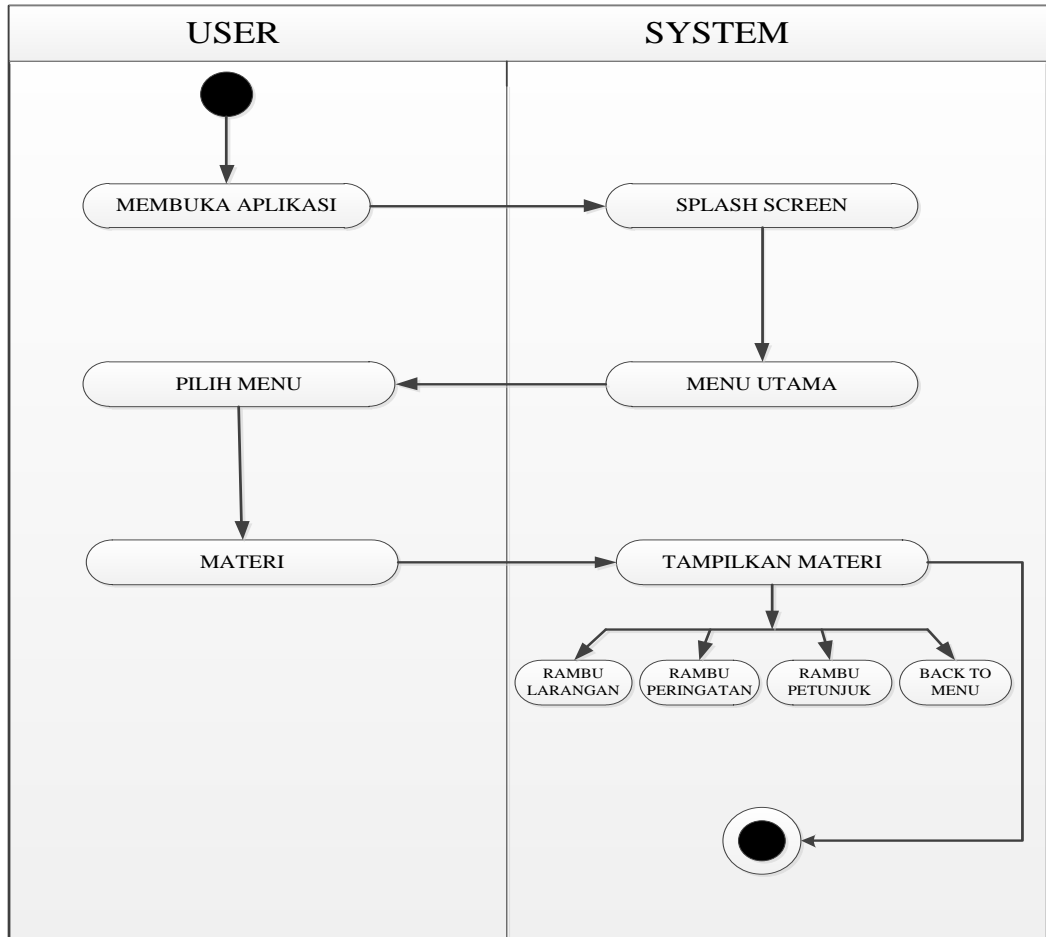
Activity Diagram Menu Utama merupakan aktivitas yang dilakukan *user* Mengoperasikan Aplikasi untuk masuk ke Tampilan Menu Utama . *Activity diagram* Tampilan Menu dapat dilihat pada Gambar 3.2.



Gambar 3.2 Activity Diagram Menu Utama

b) *Activity Diagram Materi*

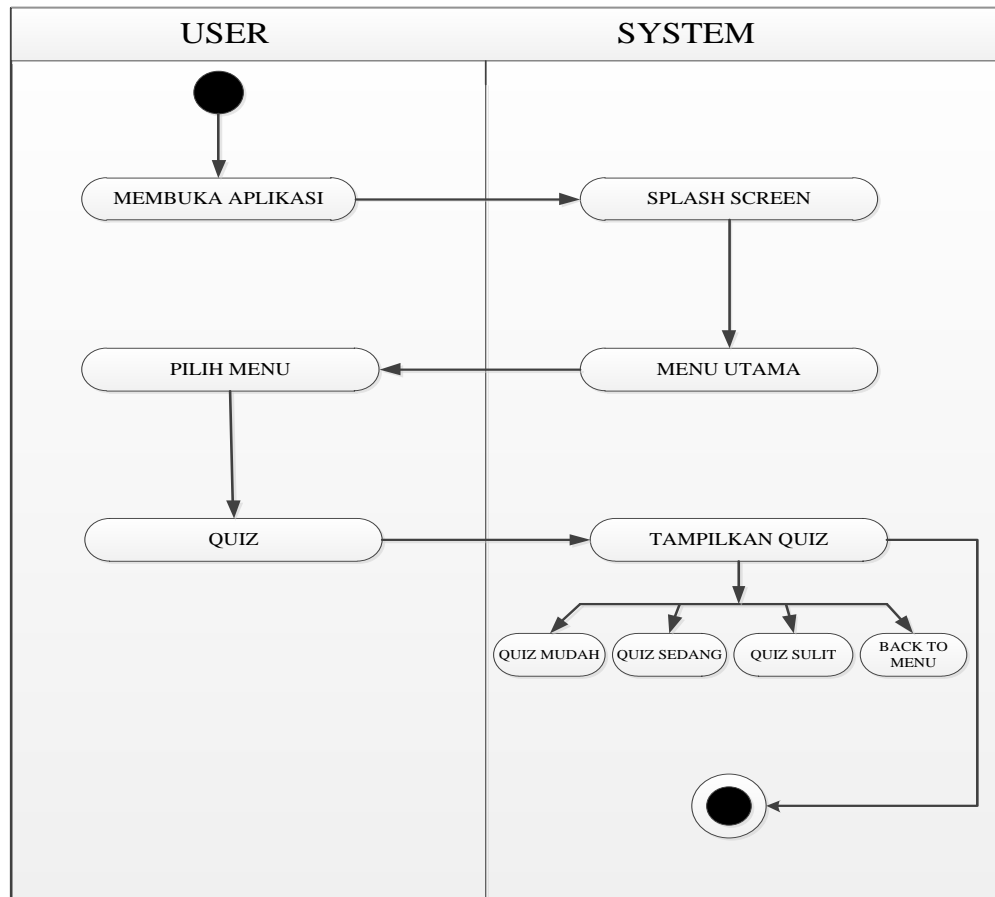
Activity Diagram Materi adalah aktivitas yang dilakukan user untuk membuka menu Materi. *Activity diagram Materi* dapat dilihat pada Gambar 3.3



Gambar 3.3 *Activity Diagram Materi*

c) *Activity Diagram Quiz*

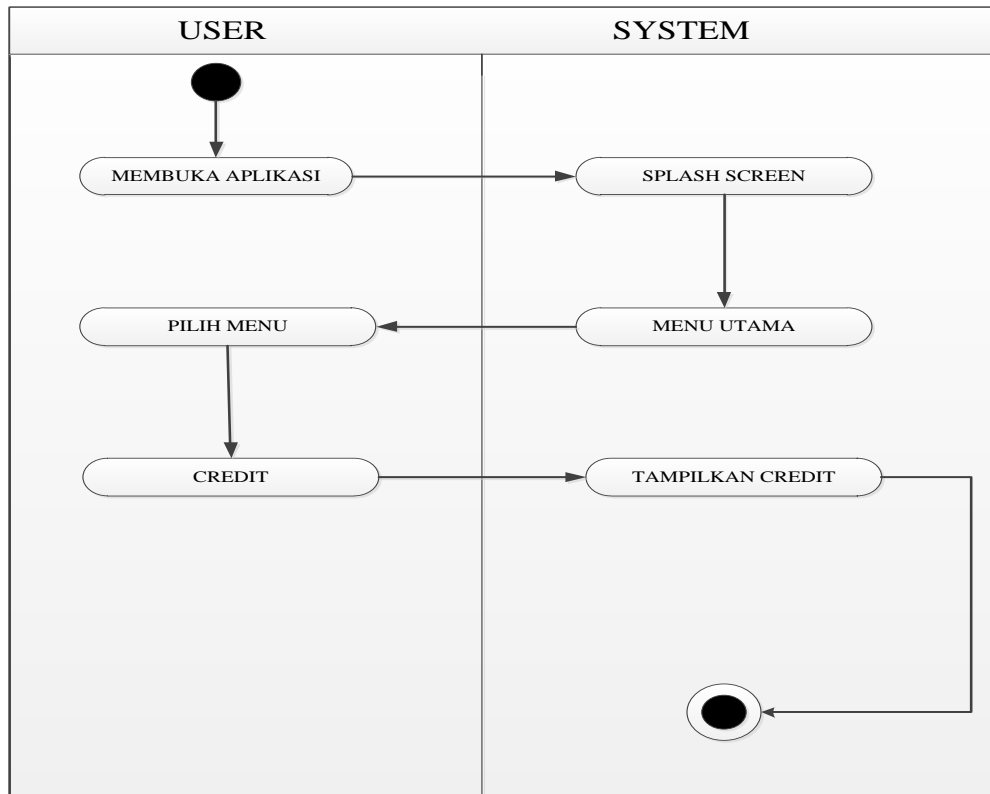
Activity Diagram Quiz adalah aktivitas yang dilakukan user untuk membuka menu Quiz. *Activity diagram Quiz* dapat dilihat pada Gambar 3.4



Gambar 3.4 *Activity Diagram Quiz*

d) *Activity Diagram Credit*

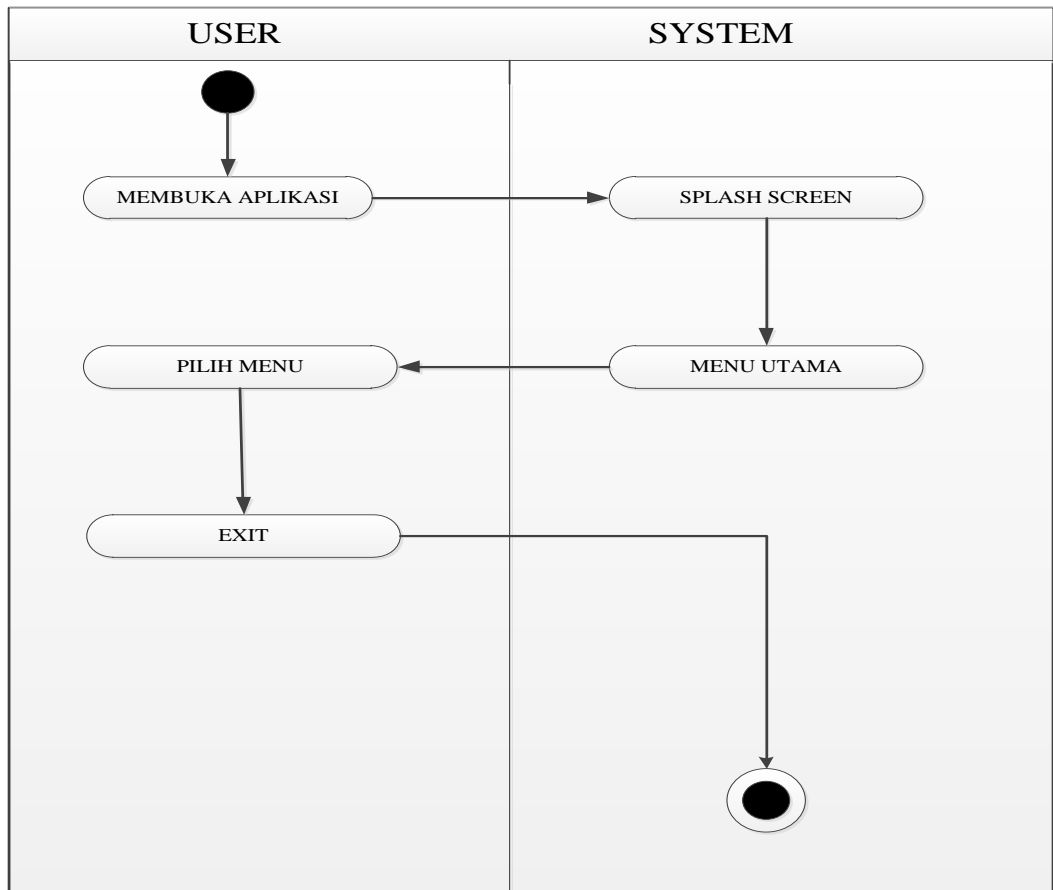
Activity Diagram Credit adalah aktivitas yang dilakukan user untuk membuka menu Credit. *Activity diagram Quiz* dapat dilihat pada Gambar 3.5



Gambar 3.5 *Activity Diagram Credit*

e) *Activity Diagram Exit*

Activity Diagram Exit adalah aktivitas yang dilakukan user untuk keluar dari aplikasi, *Activity diagram Exit* dapat dilihat pada Gambar 3.6

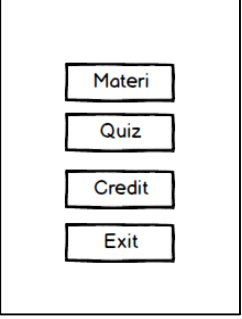





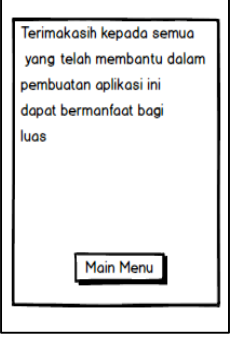
Gambar 3.6 *Activity Diagram Exit*

3.1.2 Desain

Desain adalah tahap merancang tampilan (*Interface*) aplikasi dan kebutuhan atau bahan yang dibutuhkan untuk membuat aplikasi tersebut. Pada Tahapan ini perancangan yang dibuat menggunakan metode *storyboard*. Penggunaan *storyboard* bermanfaat bagi pembuat, pengembang, dan pemilik multimedia. Bagi pembuat multimedia, *storyboard* merupakan pedoman dari aliran pekerjaan yang harus dilakukan. Bagi pengembang dan pemilik multimedia, *storyboard* merupakan *visual test* yang pertama – tama dari gagasan dimana secara keseluruhan dapat dilihat apa yang akan disajikan. Berikut *storyboard* dari aplikasi yang akan dibuat:

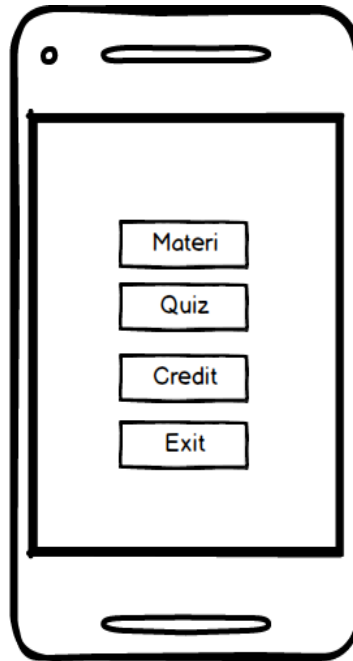
Tabel 3.1 *Storyboard* Aplikasi Yang Diusulkan

Halaman	Keterangan	Sketsa
Layout menu	Pada <i>layout</i> ini terdapat 4 pilihan Menu diantara Materi , quiz , credit dan exit	
<i>Layout</i> Pilihan Materi	Pada <i>layout</i> ini terdapat 4 pilihan Menu diantara Rambu larangan, Rambu Perintah, Rambu Petunjuk dan main menu	

<p><i>Layout materi</i></p>	<p>Pada <i>layout</i> ini saat pengguna memilih menu materi maka akan tampil materi rambu lalu lintas beserta artinya</p>	
<p><i>Layout Quiz</i></p>	<p>Pada <i>layout</i> ini pengguna disajikan beberapa pertanyaan tentang rambu – rambu lalu lintas</p>	
<p>Layout Credit</p>	<p>Pada <i>Layout</i> ini berisikan ucapan terimakasih kepada semua pihak yang telah membantu dalam pembuatan aplikasi ini</p>	

3.2.1.1 Rancangan Desain *Interface* Menu utama

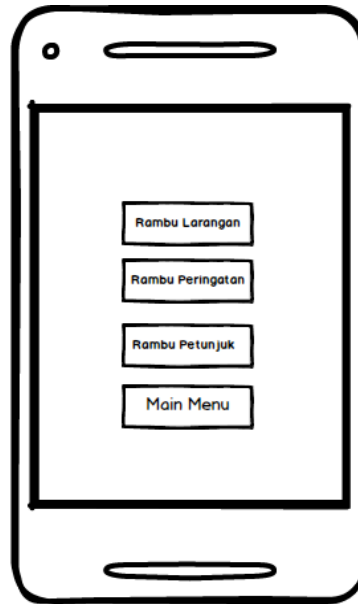
Halaman Menu utama merupakan halaman pertama yang akan ditampilkan saat pengguna mengakses aplikasi ini. Halaman ini terdiri dari menu *Materi* , *Quiz* *Credit* menu *Exit* untuk keluar dari aplikasi. Rancangan halaman menu utama dapat dilihat pada gambar 3.7



Gambar 3.7 Rancangan Desain *Interface* Menu Utama

3.2.1.2 Rancangan Desain *Interface* Pilihan *Materi*

Ketika pengguna menekan button *Materi* maka pengguna akan di arahkan ke halaman yang berisi materi yang telah disediakan, pengguna bisa memilih salah satu *button* tipe materi yang diinginkan untuk memulai *materi*. Rancangan halaman menu *materi* dapat dilihat pada gambar 3.8.



Gambar 3.8 Rancangan Desain *Interface Materi*

3.2.1.3 Rancangan Desain *Interface quiz*

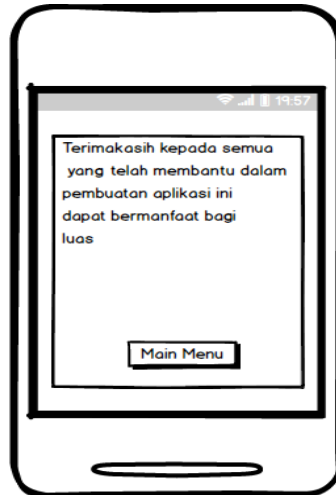
Ketika pengguna menekan button *Quiz* maka pengguna akan di arahkan ke halaman Quiz yang berisi level quiz yang telah disediakan, pengguna bisa memilih salah satu *Level* yang diinginkan untuk memulai *Quiz*. Rancangan halaman menu *Quiz* dapat dilihat pada gambar 3.9.



Gambar 3.9 Rancangan Desain *Interface Quiz*

3.2.1.4 Rancangan Desain Interface *Credit*

Halaman *Credit* merupakan halaman yang akan ditampilkan saat pengguna menekan *button Credit*. Halaman ini berisi ucapan terima kasih dan info pengembang, Rancangan halaman menu *Credit* dapat dilihat pada gambar 3.10.



Gambar 3.10 Rancangan Desain *Interface Credit*

3.1.3 *Material Collecting*

Material collecting adalah tahap dimana pengumpulan bahan-bahan yang diperlukan dalam pembuatan aplikasi ini. Bahan yang dikumpulkan adalah gambar, foto digital, background dan *image-image* pendukung lain. Pada prakteknya, tahap ini bisa dilakukan secara paralel dengan tahap *assembly*. Sebagian besar pengambilan data dilakukan di Dinas Perhubungan Kota Metro. Data yang telah di dapat dari dinas perhubungan kota metro maka dilakukan pembuatan objek gambar rambu lalu lintas menggunakan adobe photoshop tahapan terakhir adalah memasukan asset gambar kedalam *software unity 3D* untuk tahapan pembuatan aplikasi,

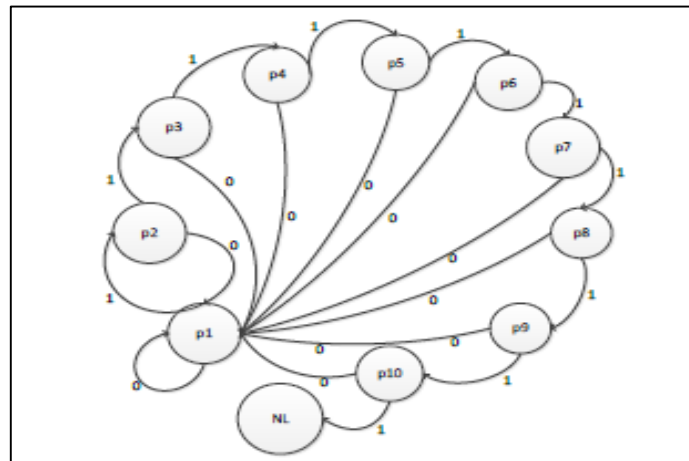
3.1.4 *Assembly*

Tahap *Assembly* adalah tahap pembuatan aplikasi dimana semua objek dan semua bahan multimedia dibuat dalam satu aplikasi. Pembuatan aplikasi ini berdasarkan tahap desain yang telah dirancang sebelumnya yang kemudian dibuat di tahap *assembly* ini. Algoritma yang digunakan Dalam Game ini dengan menggunakan

Finite State Machine (FSM) berjalan dengan mendapatkan aturan benar dan salah, ketika kondisi benar kemudian akan ditampilkan lanjutan state atau pertanyaan lanjutan dan selain dari jawaban benar maka akan melakukan pengulangan state / mengu-langi pertanyaan. Berikut diagram rancangan game dengan *Finite State Machine* (FSM) :

Inisialisasi input : Salah: 0 → Kembali

Benar: 1 → Next State



Gambar 3.11 Diagram *Finite State Machine* (FSM)

Dari diagram FSM diatas dapat dituliskan ke dalam tabel transisi sebagai berikut :

Tabel 3.2 Transisi *Finite State Machine* (FSM)

Current State	Next State	
	0	1
P1	P1	P2
P2	P1	P3
P3	P1	P4
P4	P1	P5
P5	P1	P6
P6	P1	P7
P7	P1	P8
P8	P1	P9
P9	P1	P10
P10	P1	NL
NL	-	-

Penjelasan tabel :

1. P1 merupakan pertanyaan 1 : jika pertanyaan 1 dapat dijawab dengan benar, maka *next state* adalah pertanyaan 2 dan jika pertanyaan 1 di-jawab dengan jawaban salah maka akan kembali ke pertanyaan 1.
2. P2 merupakan pertanyaan 2 : jika pertanyaan 2 dapat dijawab dengan benar, maka *next state* adalah pertanyaan 3 dan jika pertanyaan 2 di-jawab dengan jawaban salah maka akan kembali ke pertanyaan 1.
3. P3 merupakan pertanyaan 3 : jika pertanyaan 3 dapat dijawab dengan benar, maka *next state* adalah pertanyaan 4 dan jika pertanyaan 3 di-jawab dengan jawaban salah maka akan kembali ke pertanyaan 1.
4. P4 merupakan pertanyaan 4 : jika pertanyaan 4 dapat dijawab dengan benar, maka *next state* adalah pertanyaan 5 dan jika pertanyaan 4 di-jawab dengan jawaban salah maka akan kembali ke pertanyaan 1.
5. P5 merupakan pertanyaan 5 : jika pertanyaan 5 dapat dijawab dengan benar, maka *next state* adalah pertanyaan 6 dan jika pertanyaan 5 di-jawab dengan jawaban salah maka akan kembali ke pertanyaan 1.
6. P6 merupakan pertanyaan 6 : jika pertanyaan 6 dapat dijawab dengan benar, maka *next state* adalah pertanyaan 7 dan jika pertanyaan 6 di-jawab dengan jawaban salah maka akan kembali ke pertanyaan 1.
7. P7 merupakan pertanyaan 7 : jika pertanyaan 7 dapat dijawab dengan benar, maka *next state* adalah pertanyaan 8 dan jika pertanyaan 7 di-jawab dengan jawaban salah maka akan kembali ke pertanyaan 1.
8. P8 merupakan pertanyaan 8 : jika pertanyaan 8 dapat dijawab dengan benar, maka *next state* adalah pertanyaan 9 dan jika pertanyaan 8 di-jawab dengan jawaban salah maka akan kembali ke pertanyaan 1.
9. P9 merupakan pertanyaan 9 : jika pertanyaan 9 dapat dijawab dengan benar, maka *next state* adalah pertanyaan 10 dan jika pertanyaan 9 di-jawab dengan jawaban salah maka akan kembali ke pertanyaan 1.
10. P10 merupakan pertanyaan 10 : jika pertanyaan 10 dapat dijawab dengan benar, maka *next state* adalah level selanjutnya dan jika pertanyaan 10 dijawab dengan jawaban salah maka akan kembali ke pertanyaan 1.
11. NL merupakan next level

3.1.5 Testing

Testing merupakan pengujian aplikasi merupakan tahap selanjutnya setelah program atau aplikasi perangkat lunak selesai dalam pembuatannya. Pengujian aplikasi dilakukan dengan menggunakan metode *black-box testing*. Pengujian yang akan dilakukan yaitu dengan menguji lama waktu *loading* atau *respon time* dari masing masing halaman yang terdapat pada aplikasi. Pengujian dilakukan dengan menggunakan tiga buah perangkat yang berbeda yang memiliki spesifikasi dengan kriteria rendah, sedang dan tinggi dilihat dari segi perangkat kerasnya. Dapat dilihat pada gambar dibawah ini;

Tabel 3.3 Pengujian Perangkat

DEVICE I	DEVICE II	DEVICE III
Android Versi 6.0 (<i>Marshmallow</i>)	Android Versi 7.0 (<i>Nougat</i>)	Android Versi 8.0 (<i>Oreo</i>)
❖ Qualcomm MSM8956 Snapdragon 650	❖ Qualcomm MSM8917 Snapdragon 425	❖ Qualcomm SDM636 Snapdragon 636 (14 nm)
❖ Hexa-core (4x1.4 GHz Cortex-A53 & 2x1.8 GHz Cortex-A72)	❖ Quad-core 1.4 GHz Cortex-A53	❖ Octa-core 1.8 GHz Kryo 260
❖ Display 5.5 inch	❖ Display 5.0 inch	❖ Display 5.99

3.1.6 Distribution

Tahapan dimana aplikasi disimpan dalam suatu media penyimpanan. Pada prakteknya aplikasi yang sudah berhasil dibuat dan sudah lulus uji selanjutnya dimasukkan ke *play store* sebagai pemenuhan semua tahapan yang ada pada versi Luther-Sutopo. Jika media penyimpanan tidak cukup untuk menampung aplikasinya, maka dilakukan kompresi terhadap aplikasi tersebut.

3.2 Proses Kerja Aplikasi

Proses kerja aplikasi ini dapat dijalankan secara *offline* atau dapat dijalankan tanpa menggunakan paket data internet. Jika terdapat pembaharuan dari aplikasi ini, maka *play store* akan mengirimkan notifikasi untuk segera melakukan pembaharuan.