

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Bagian ini merupakan implementasi dari perancangan *game* yang telah dibuat sebelumnya dan disertai dengan pembahasan dari setiap proses yang dilakukan dan hasil yang didapat.

4.1.1 Implementasi *Game Design Document*

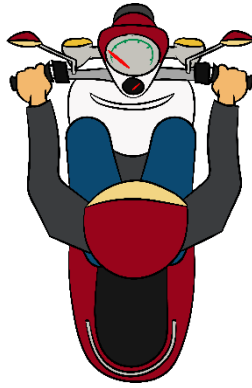
Implementasi *game design document* adalah implementasi dari rancangan *game* yang telah dibuat pada bab sebelumnya. Sehingga pada bagian ini akan menjelaskan hasil dari rancangan *game* yang telah dibuat menggunakan *game design document*.

4.1.1.1 Implementasi Model/Karakter *Game*

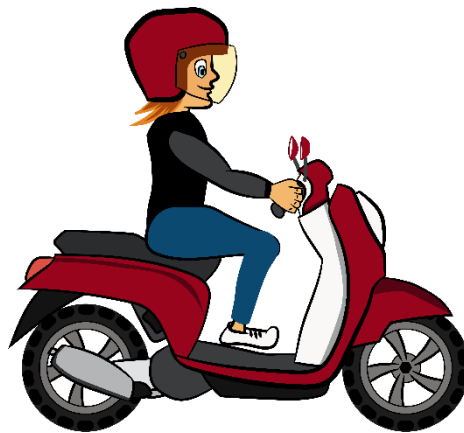
Implementasi karakter *game* ini akan menjelaskan pemodelan dan hasil dari rancangan objek penting menggunakan perangkat lunak yang digunakan dalam membuat “*Game Edukasi Berlalu Lintas Untuk Roda dua Berbasih Android*”. Berikut pembahasan implementasi model/karakter pada *game*:

a. **Objek Pengendara Sepeda Motor**

Objek Pengendara Sepeda Motor merupakan karakter utama yang digunakan pada *game* ini. Terdapat dua model pengendara sepeda motor yaitu tampak atas dan tampak samping. Model tampak atas digunakan pada *scene* dengan sudut pandang dari atas, sedangkan tampak samping digunakan pada *scene* dengan sudut pandang dari samping. Pembuatan objek pengendara sepeda motor dikerjakan menggunakan *software* Adobe Illustrator CC 2017. Untuk lebih jelasnya pemodelan pengendara sepeda motor dengan sudut pandang atas dapat dilihat pada Gambar 4.1. Pemodelan pengendara sepeda motor tampak samping dapat dilihat pada Gambar 4.2. Dua model yang telah dibuat akan diprogram menggunakan Unity 3D agar dapat dijalankan.



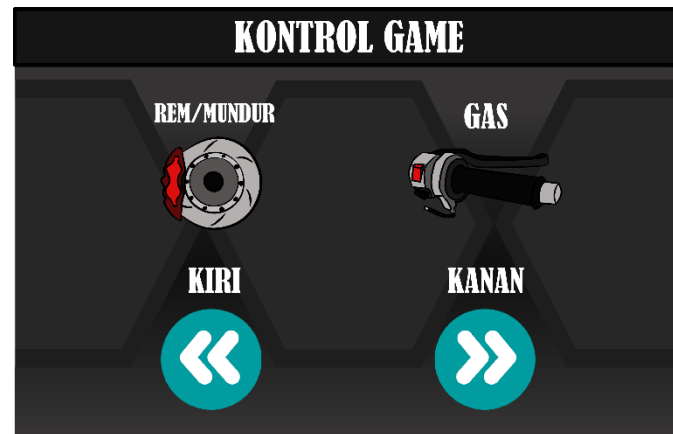
Gambar 4.1 Pengendara Sepeda Motor Dari Atas



Gambar 4.2 Pengendara Sepeda Motor Dari Samping

b. Objek Tombol Kontrol

Objek tombol kontrol digunakan sebagai tombol yang dapat digunakan sebagai kontrol karakter utama pada *game*. Tombol ini terdiri dari tombol maju, tombol rem, tombol ke kiri, dan tombol ke kanan. Untuk lebih jelas dapat dilihat pada Gambar 4.3. tombol maju berfungsi untuk akselerasi karakter utama dan tombol rem berfungsi untuk mengurangi kecepatan. Kontrol untuk membelokan ke kanan dan ke kiri menggunakan tombol ke kanan dan tombol kiri.



Gambar 4.3 Tombol Kontrol

c. Objek Tombol Menu

Objek tombol menu digunakan pada *scene* menu *game* sebagai tombol pilihan arena pada *game*. Terdapat dua tombol yang memiliki gambar tema arena pada *game*. Pembuatan objek tombol menggunakan *software* Adobe Illustrator CC 2017. untuk lebih jelasnya dapat dilihat pada Gambar 4.4 dan Gambar 4.5.



Gambar 4.4 Tombol Menu Arena Pedesaan



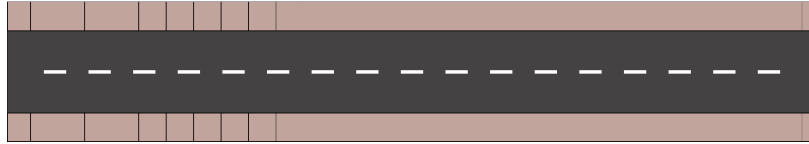
Gambar 4.5 Tombol Menu Arena Perkotaan

d. Objek Lintasan

Objek lintasan adalah objek yang digunakan untuk membuat lintasan berupa jalan dan objek lainnya. Pembuatan objek lintas menggunakan *software* Adobe Illustrator CC 2017. Berikut beberapa pemodelan objek lintasan yang telah dibuat:

1. Jalan Dengan Marka Garis Putus-putus

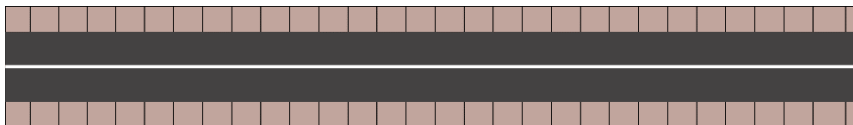
Objek ini merupakan gambar jalan dengan garis putus-putus di tengahnya. Arti dari garis putus-putus di tengah menandakan pengendara lain diperbolehkan mendahului melewati garis tersebut. Untuk lebih jelasnya dapat dilihat pada Gambar 4.6.



Gambar 4.6 Marka Garis Putus-putus

2. Jalan Dengan Marka Garis Tanpa Putus

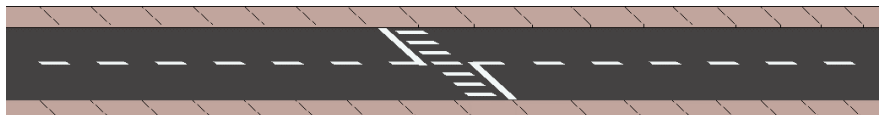
Objek ini merupakan gambar sebuah jalan dengan garis tidak putus di tengahnya. Arti dari garis yang tidak putus menandakan bahwa pada jalur tersebut pengendara dilarang melewati garis tersebut. Pada jalur ini pengendara dilarang mendahului kendaraan di depannya. Untuk lebih jelasnya dapat dilihat pada Gambar 4.7.



Gambar 4.7 Marka Garis Tanpa Putus

3. Penyebrangan Pejalan Kaki

Objek ini merupakan sebuah jalan yang dilengkapi dengan marka untuk penyebrangan pejalan kaki yang disebut dengan zebra cross. Pada jalur ini pengendara harus hati-hati untuk menghindari kemungkinan player menabarak pejalan kaki yang sedang menyebrang. Untuk lebih jelasnya dapat dilihat pada Gambar 4.8.

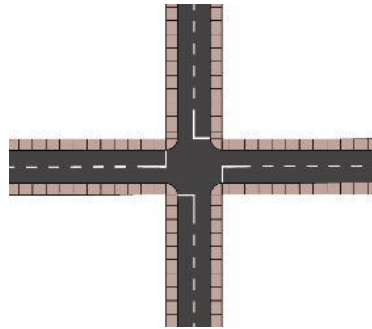


Gambar 4.8 Marka Penyebrangan Pejalan Kaki

4. Simpang Empat

Simpang empat merupakan objek jalan yang memiliki empat persimpangan jalan. Pada objek ini digunakan sebagai pengatur lalu lintas yang menggunakan lampu merah

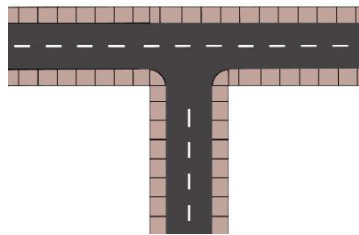
sebagai pengaturnya. Untuk melihat lebih jelas bentuk simpang empat pada *game* dapat dilihat pada Gambar 4.9.



Gambar 4.9 Simpang Empat

5. Simpang Tiga

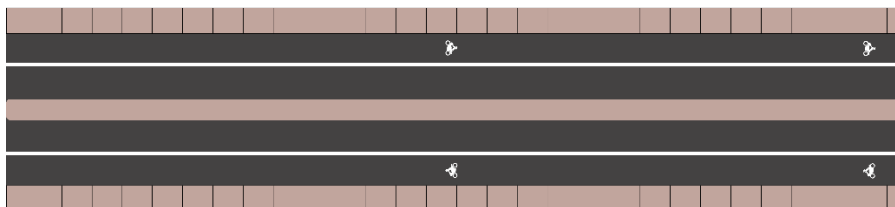
Simpang tiga adalah objek jalan yang memiliki tiga persimpangan. Objek ini digunakan untuk membentuk jalur yang digunakan pada *game*. Objek simpang tiga dapat dilihat pada Gambar 4.10.



Gambar 4.10 Simpang Tiga

6. Jalan Dengan Marka Jalur Khusus Roda Dua

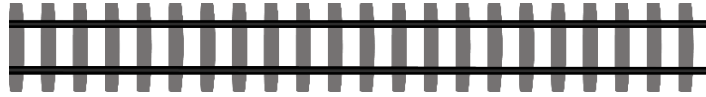
Objek ini adalah jalan yang memiliki pembeda antara jalur kendaraan roda empat dan jalur khusus roda dua seperti yang terlihat pada Gambar 4.11.



Gambar 4.11 Marka Jalur Khusus Roda Dua

7. Jalan Kereta Api

Jalan kereta api adalah objek yang digunakan untuk membatu pembuatan objek kereta api yang bergerak. Untuk lebih jelasnya bentuk jalan kereta api yang dibuat dapat dilihat pada Gambar 4.12.



Gambar 4.12 Jalan Kereta Api

8. Kereta Api

Objek kereta api digunakan sebagai rintangan bagi player di dalam *game*. Bentuk dari objek kereta api adalah kereta api barang dengan muatan batu bara. Objek kereta api akan dibuat tampak atas. Untuk lebih jelasnya dapat dilihat pada Gambar 4.13.



Gambar 4.13 Kereta Api

9. Pejalan Kaki

Bentuk dari objek pejalan kaki pada *game* ini adalah orang laki-laki yang akan dibuat bergerak menyebrangi jalan sebagai rintangan bagi *player*. Untuk lebih jelasnya bentuk dari objek pejalan kaki dapat dilihat pada Gambar 4.14.



Gambar 4.14 Pejalan Kaki

10. Lampu Merah

Lampu merah adalah objek yang digunakan pada persimpangan untuk mengatur lalu lintas pada *game* dan sebagai rintangan untuk *player*. Objek lampu merah yang telah dibuat dapat dilihat pada Gambar 4.15.



Gambar 4.15 Lampu Merah

e. Objek Lingkungan

Objek lingkungan adalah objek yang digunakan untuk membuat lingkungan pada setiap halaman arena pedesaan maupun arena perkotaan. Pembuatan objek lintas menggunakan *software* Adobe Illustrator CC 2017. Berikut beberapa pemodelan objek lingkungan yang telah dibuat:

1. Objek Rumput

Objek rumput digunakan untuk memberikan menghidupkan lingkungan pada setiap arena pada *game* ini. Untuk lebih jelasnya objek rumput dapat dilihat pada Gambar 4.16.



Gambar 4.16 Rumput

2. Objek Pohon

Objek pohon digunakan untuk menghias pinggir jalan sehingga tampilan *game* terlihat lebih menarik dan tidak kosong. Objek pohon dibuat terlihat tampak atas,

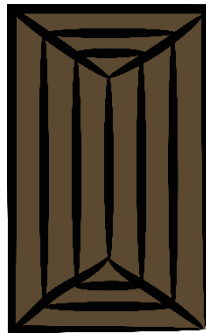
sehingga hanya daunnya saja yang terlihat. Untuk lebih jelasnya dapat dilihat pada Gambar 4.17.



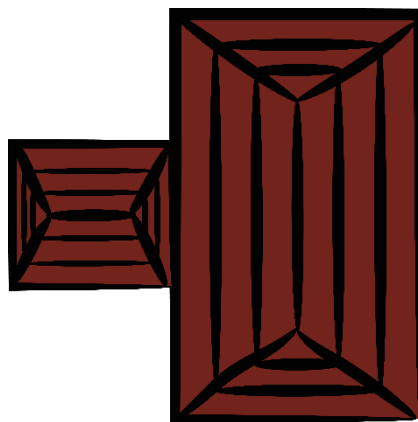
Gambar 4.17 Pohon

3. Objek Rumah

Objek rumah digunakan pada arena pedesaan saat tampilan terlihat dari atas. Objek rumah ditampilkan terlihat dari atas, sehingga hanya terlihat atap rumah saja. Objek rumah dapat dilihat pada Gambar 4.18 dan Gambar 4.19.



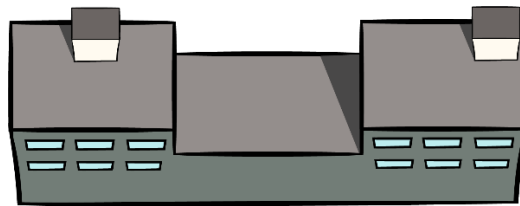
Gambar 4.18 Rumah1



Gambar 4.19 Rumah2

4. Objek Gedung

Objek gedung digunakan pada tampilan arena perkotaan. Objek gedung bertujuan untuk membuat lingkungan layaknya perkotaan. Objek gedung berupa gambar bangunan yang memiliki beberapa tingkat. Objek gedung dibuat tampak dari atas sehingga hanya beberapa sisi gedung yang dapat terlihat. Untuk lebih jelasnya dapat dilihat pada Gambar 4.20.



Gambar 4.20 Gedung

4.1.1.2 Implementasi Antarmuka

Implementasi antarmuka adalah implementasi dari *storyboard* yang telah dibuat di dalam *game design document* pada bab sebelumnya. Implementasi antarmuka berisikan penjelasan dari antarmuka yang dibuat untuk memenuhi kebutuhan analisis yang telah dilakukan sebelumnya. Implementasi antarmuka “*Game* Edukasi Berlalu Lintas Untuk Roda Dua Berbasis Android” adalah sebagai berikut:

a. Implementasi Antarmuka Halaman Utama

Halaman utama adalah *scene* pertama yang akan dilihat *player* ketika menjalankan aplikasi *game* ini. Tampilan halaman utama adalah gambar pengendara sepeda motor dengan latar belakang jalan yang di setiap sisinya terdapat gedung-gedung. Pada halaman utama terdapat dua tombol yaitu tombol “MAIN” dan tombol “KELUAR”. Ketika tombol “MAIN” ditekan berikutnya akan muncul *scene* halaman menu pada *game*. Tombol “KELUAR” digunakan untuk keluar dari aplikasi *game*. Untuk lebih jelasnya dapat dilihat pada Gambar 4.16.



Gambar 4.21 Halaman Awal

b. Implementasi Antarmuka Halaman Menu

Scene halaman menu akan tampil ketika player menekan tombol “MAIN” pada scene shalaman utama. Pada halaman menu terdapa dua pilihan arena yang bisa dimainkan oleh *player* yaitu arena pedesaan dan arena perkotaan. Arena pedesaan adalah lintasan dengan tema pedesaan dan arena perkotaan adalah lintasan dengan tema perkotaan. Pada halaman menu *player* dapat menggeser ke bawah atau ke atas untuk memilih arena. Terdapat tombol “home” pada kanan atas layar dengan lambang rumah yang berfungsi untuk kembali ke halaman utama. Untuk lebih jelasnya dapat dilihat pada Gambar4.17.



Gambar 4.22 Halaman Menu

c. Implementasi Antarmuka Arena Pedesaan

Game1 akan berjalan setelah *player* memilih arena pedesaan pada *scene* halaman menu. Arena pedesaan adalah arena permainan yang memiliki tema pedesaan. Alur

dari arena pedesaan adalah *player* harus menjalankan karakter utama menuju sebuah toko yang bernama “TOMIRA” sebelum waktu yang sudah ditentukan habis. Pada awal *game*, *player* dibekali dengan skor yang berjumlah 100. Jika *player* melanggar rambu atau marka jalan yang terdapat pada Game1, skor pada *game* akan berkurang. Pada Game1 terdapat empat rambu dan 3 marka jalan yang digunakan sebagai rintangan pada arena pedesaan. Berikut adalah implementasi antarmuka dari arena pedesaan:

1. Implementasi Antarmuka Panel Pengenalan Kontrol *Game*

Panel pengenalan kontrol *game* adalah panel yang bertujuan memberikan informasi tentang tombol kontrol yang digunakan *player* untuk menjalankan karakter utama pada *game*. Pada panel ini terdapat tombol dengan arah panah kanan yang digunakan untuk menampilkan panel selanjutnya pada Game1. Untuk lebih jelasnya dapat dilihat pada Gambar 4.18.



Gambar 4.23 Pengenalan Kontrol *Game*

2. Implementasi Antarmuka Panel Pengenalan Rambu Arena Pedesaan

Panael pengenalan rambu adalah panel yang muncul ketika *player* menekan tombol panah kanan pada panel kontrol *game*. Panel pengenalan rambu berisi gambar disertai penjelasan setiap rambu lalu lintas dan marka jalan yang digunakan pada arena pedesaan. Panel rambu terbagi menjadi dua panel. Terdapat tombol panah kiri dan kanan pada pojok kanan bawah pada panel yang digunakan untuk melihat panel sebelumnya dan panel selanjutnya. Antarmuka panale pengenalan rambu dapat dilihat pada Gambar 4.19 dan Gambar 4.20.



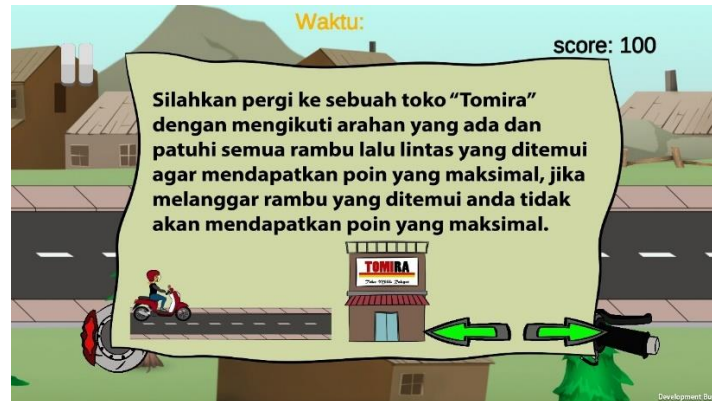
Gambar 4.24 Pengenalan Rambu Arena Pedesaan



Gambar 4.25 Pengenal Lanjut Rambu Arena Pedesaan

3. Implementasi Antarmuka Panel *gameplay* Arena Pedesaan

Panel *gameplay* adalah panel yang menjelaskan peraturan dan alur pada arena pedesaan. Terdapat tombol panah kiri dan panah kanan yang memiliki fungsi masing-masing. Tombol panah kiri untuk melihat panel sebelumnya dan tombol panah kanan digunakan untuk memulai arena pedesaan. Untuk lebih jelasnya dapat dilihat pada Gambar 4.21.



Gambar 4.26 *Gameplay* Arena Pedesaan

4. Implementasi Antarmuka Awal Arena Pedesaan

Setelah *player* melihat semua panel yang ada pada arena pedesaan maka selanjutnya *player* harus menjalankan karakter utama menggunakan tombol kontrol yang terletak di kiri bawah dan kanan bawah pada tampilan arena pedesaan sesuai dengan perintah dari *gameplay* Game1. Pada antarmuka ini terdapat waktu yang mulai terhitung mundur sebagai waktu yang diperluka *player* untuk menyelesaikan arena pedesaan. Pada bagian pojok kiri atas terdapat tombol paus yang digunakan untuk menghentikan sementara kita *game* berjalan. Untuk lebih jelasnya dapat dilihat pada Gambar 4.22.



Gambar 4.27 Antarmuka Awal Arena Pedesaan

5. Implementasi Rambu Penyeberangan Arena Pedesaan

Saat *player* menjumpai rambu penyeberangan yang terlihat adalah terdapat 3 karakter pejalan kaki yang berjalan menyeberangi jalan raya secara berurutan. *Player* dianjurkan untuk menunggu semua pejalan kaki selesai menyeberangi jalan sebelum

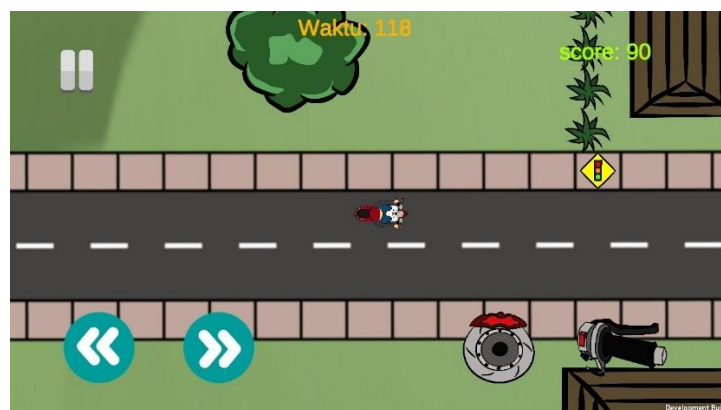
melanjut kan perjalanan pada Arena Pedesaan. *Player* akan mendapat sangsi pengurang skor jika tidak berhenti keti ada pejalan kaki yang menyeberang. Untuk lebih jelasnya dapat dilihat pada Gambar 4.23.



Gambar 4.28 Implementasi Rambu Penyebrangan

6. Implementasi Antarmuka Tampilan Atas Pada Arena Pedesaan

Tampilan atas pada arena pedesaan bertujuan untuk memudahkan dalam penggunaan rambu pada arena pedesaan. Pada tampilan ini, semua objek pada arena pedesaan ditampilkan tampak dari atas, mualai dari karakter utama hingga objek lain yang digunakan pada arena pedesaan, seperti yang terlihat pada Gambar 4.24.

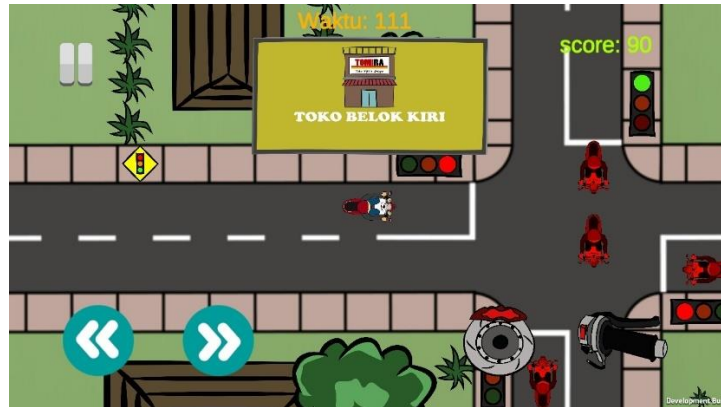


Gambar 4.29 Tampilan Atas Arena Pedesaan

7. Implementasi Rambu Lampu Merah Arena Pedesaan

Lampu merah pada arena pedesaan ditempatkan pada simpang 4 yang digunakan sebagai rintangan untuk *player*. Pada rintang ini, *player* harus menunggu lampu hijau menyala dan mengikuti pentunjuk arah yang muncul untuk melanjutkan perjalanan

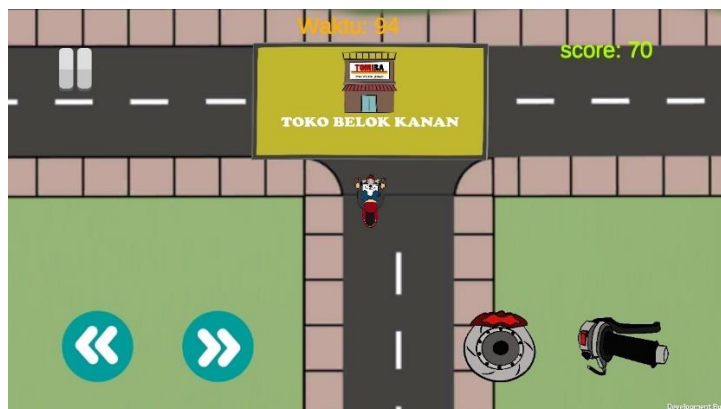
pada arena pedesaan. Implementasi rambu merah pada arena pedesaan dapat dilihat pada Gambar 4.25.



Gambar 4.30 Lampu Merah Arena Pedesaan

8. Implementasi Informasi Petunjuk Arah Pada Simpang Tiga

Informasi petunjuk arah pada simpang tiga akan muncul ketika karakter utama yang dikontrol oleh *player* mendekati persimpang jalan. Informasi petunjuk arah bertujuan untuk membantu *player* dengan memberikan arahan jalan menuju tempat tujuan. Implementasi petunjuk arah dapat dilihat pada Gambar 4.26.

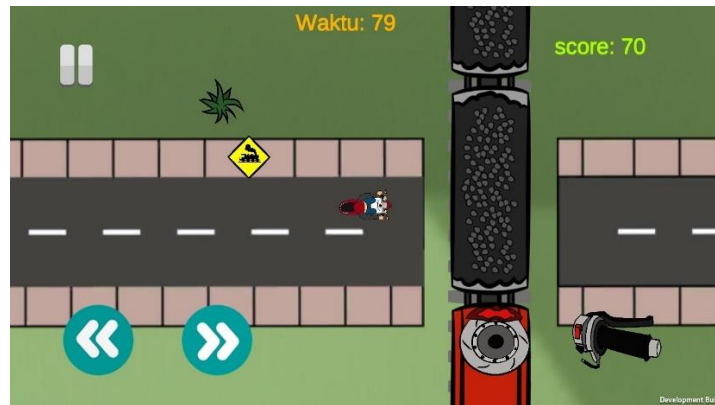


Gambar 4.31 Petunjuk Arah Simpang Tiga

9. Implementasi Tampilan Rambu Perlintasan Kereta Api

Rambu perlintasan kereta api digunakan sebagai rintangan bagi *player*. Rambu perlintasan kereta api yang digunakan pada *game* ini adalah rambu perlintasan yang tidak memiliki palang pintu kereta api. Ketika *player* mendekati perlintasan kereta api harus berhati-hati karena sewaktu-waktu kereta api bisa saja melintas dengan cepat. Jika

karakter utama tertabak kereta api maka kondisi *game* akan menjadi “*GAME OVER*”. Untuk lebih jelasnya dapat dilihat pada Gambar 4.27.



Gambar 4.32 Rambu Perlintasan Kereta Api

10. Implementasi Rambu Banyak Aktifitas Pejalan Kaki

Rambu ini hampir sama dengan rambu penyebrangan pejalankaki tetapi pada rambu ini tidak dilengkapi dengan marka *zebra cross* sehingga *player* harus hati-hati setelah melewati rambu agar tidak menabrak pejalan kaki yang menyebrang. Jika *player* menabrak pejalan kaki maka *player* mendapat sanksi pengurangan skor yang dimiliki. Implementasi ini dapat dilihat pada Gambar 4.28.



Gambar 4.33 Rambu Banyak Aktifitas Pejalan Kaki

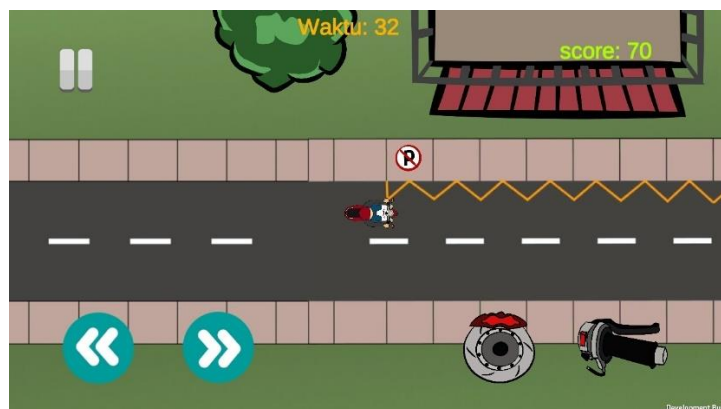
11. Implementasi Antarmuka Depan Toko

Implementasi ini adalah implementasi ketika *player* sampai di depan toko. Pada saat *player* sudah sampai di depan toko makan akan muncul panel dengan dua pilihan perintah parkir. Pilihan yang ada pada panel adalah parkir di depan toko atau lanjut

menuju parkir, seperti yang terlihat pada Gambar 4.29. jika *player* memilih parkir di depan toko kondisi *game* akan dinyatakan selesai dengan konsekuensi *player* akan mendapatkan sangsi pengurang skor yang dia dapat. Antarmuka *game* pada saat di depan toko dapat dilihat pada Gambar 4.30.



Gambar 4.34 Pilihan Perintah Parkir Arena Pedesaan



Gambar 4.35 Depan Toko

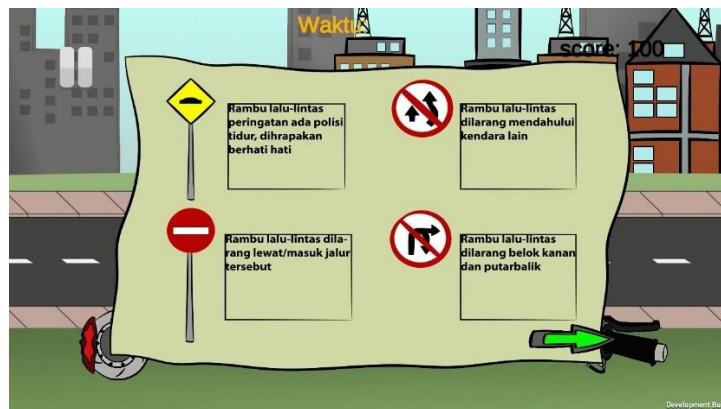
d. Implementasi Antarmuka Arena Perkotaan

Arena perkotaan akan berjalan setelah *player* memilih arena perkotaan pada *scene* halaman menu. Arena perkotaan adalah arena permainan yang memiliki tema perkotaan. Alur dari arena perkotaan adalah *player* harus menjalankan karakter utama menuju sebuah Universitas yang bernama "Universitas Kota" sebelum waktu yang sudah ditentukan habis. Pada awal *game*, *player* dibekali dengan skor yang berjumlah 100. Jika *player* melanggar rambu atau marka jalan yang terdapat pada arena perkotaan, skor pada *game* akan berkurang. Pada arena perkotaan terdapat

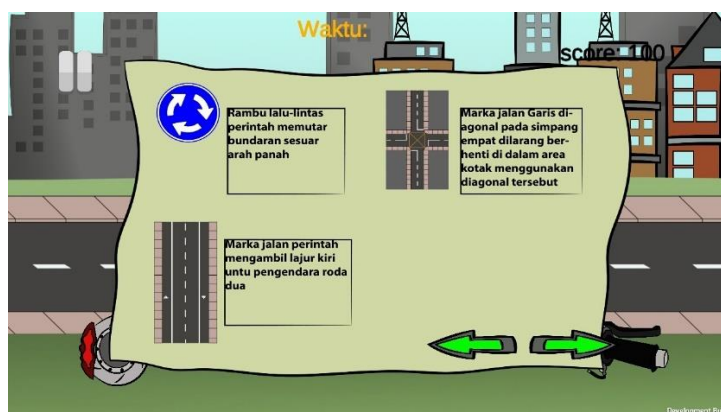
lima rambu dan dua marka jalan yang digunakan sebagai rintangan pada arena pedesaan. Berikut adalah implementasi anatar muka dari arena perkotaan:

1. Implementasi Panel Pengenalan Rambu Arena Perkotaan

Panael pengenalan rambu adalah panel yang muncul ketika *player* menekan tombol panah kanan pada panel kontrol *game*. Panel pengenalan rambu berisi gambar disertai penjelasan setiap rambu lalu lintas dan marka jalan yang digunakan pada arena perkotaan. Panel rambu terbagi menjadi dua panel. Terdapat tombol panah kanan pada pojok kanan bawah pada panel yang digunakan untuk melihat panel sebelumnya dan panel selanjutnya. Antarmuka panale pengenalan rambu dapat dilihat pada Gambar 4.31 dan Gambar 4.32.



Gambar 4.36 Panel Pengenalan Rambu Arena Perkotaan

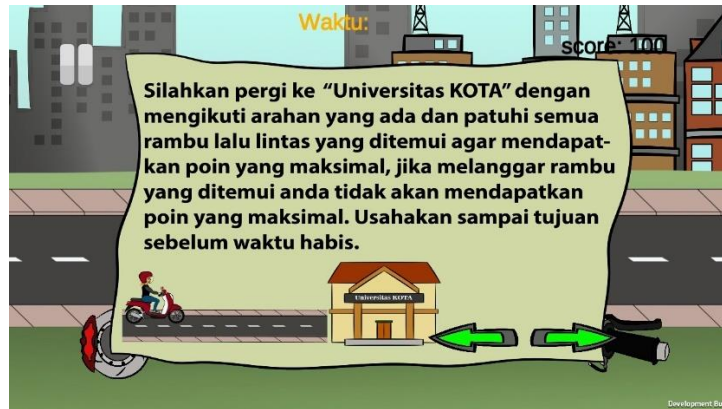


Gambar 4.37 Panel Pengenalan Lanjut Rambu Arena Perkotaan

2. Implementasi Antarmuka *gameplay* Arena Perkotaan

Panel *gameplay* arena perkotaan adalah panel yang menjelaskan peraturan dan alur pada arena perkotaan. Terdapat tombo panah kiri dan panah kanan yang memiliki

fungsi masing-masing. Tombol panah kiri untuk melihat panel sebelumnya dan tombol panah kanan digunakan untuk memulai arena pedesaan. Untuk lebih jelasnya dapat dilihat pada Gambar 4.33.



Gambar 4.38 Panel *Gameplay* Arena Perkotaan

3. Implementasi Antarmuka Awal Arena Perkotaan

Tampilan awal saat arena perkotaan mulai berjalan terdapat karakter utama yang tampak dari samping yang harus dikontrol *player* berjalan menuju tempat yang sudah ditentukan pada arena perkotaan. Terdapat tombol maju dan tombol rem yang masing-masing terletak pada kanan bawah dan kiri bawah seperti yang terlihat pada Gambar 4.34.



Gambar 4.39 Tampilan Awal Arena Perkotaan

4. Implementasi Rambu Polisi Tidur Arena Perkotaan

Rambu polisi tidur akan terlihat setelah *player* menjalankan karakter utama beberapa saat. Setelah rambu terlihat *player* akan melewati dua polisi tidur. Rambu ini

bertujuan untuk memperlambat gerak *player*, seperti yang ada pada Gambar 4.41. jika *player* tidak memperlambat jalan karakter utama, *player* mendapat kemungkinan menabrak pejalan kaki yang sedang menyebrang jalan. Seperti yang terlihat pada Gambar 4.35.



Gambar 4.40 Rambu Polisi Tidur

5. Implementasi rambu dilarang mendahului

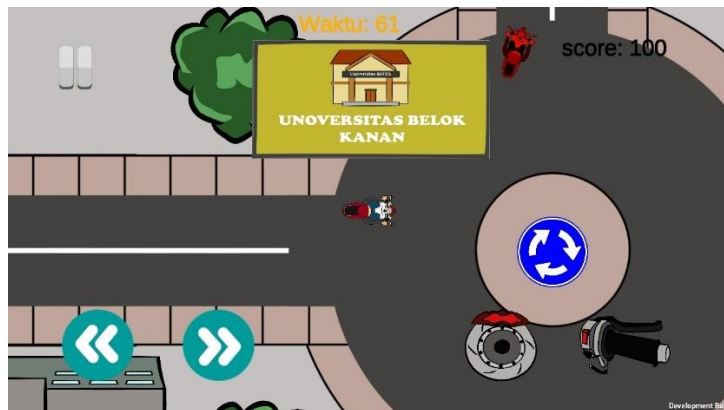
Tampilan rambu dilarang mendahului akan terlihat dari sudut atas. Jalur dilarang mendahului ditandai dengan adanya rambu dilarang mendahului pada samping jalan dan garis putih tanpa putus di tengah jalan. Jika *player* melewati garis putih lurus tanpa putus akan dikenakan sanksi pengurangan skor begitu juga jika *player* menabrak kendaraan di depannya. Untuk lebih jelasnya dapat dilihat pada Gambar 4.36.



Gambar 4.41 Rambu Dilarang Mendahului

6. Implementasi Antarmuka Bundaran

Di dalam *game* jalur bundaran akan terlihat dari atas. Terdapat rambu perintah pada tengah bundaran. Arti dari rambu tersebut adalah untuk memerintah arah *player* pada saat di bundaran. Pada halaman ini juga akan muncul panel yang menampilkan petunjuk arah untuk *player*. Implementasi Antarmuka Bundaran dapat dilihat pada Gambar 4.37.



Gambar 4.42 Bundaran

7. Implementasi Antarmuka Marka Garis Diagonal pada Perempatan

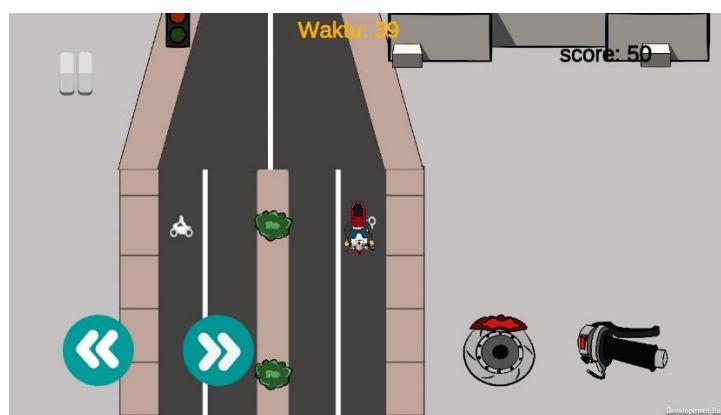
Lintasan ini akan dijumpai *player* ketika sudah melewati bundaran. Tampilan dari terlihat tampak atas. Marka garis diagonal digunakan pada simpang empat yang dilengkapi dengan lampu merah. Ketika *player* berhenti di lampu merah pada sebelah kanan jalan terlihat Universitas yang dituju, tetapi *player* tidak dapat belok kanan langsung karena jalur tersebut satu arah. Jika *player* melanggar akan terkena sanksi pengurangan skor. Untuk lebih jelasnya dapat dilihat pada Gambar 4.38.



Gambar 4.43 Implementasi Garis Diagonal

8. Implementasi Antermuka Marka Khusus Jalur Sepeda Motor

Marka khusus jalur sepeda motor ditampilkan dengan tampak atas pada *game*. Jalur ini akan dijumpai *player* setelah melewati lampu merah pada simpang empat. *Player* tidak boleh menggunakan jalur lain selain jalur sepeda motor. Jika *player* keluar dari jalur khusus sepeda motor maka akan mendapatkan sangsi. Bida dilihat pada Gambar 4.39.



Gambar 4.44 Implementasi Rambu Jalur Khusus Roda Dua

9. Implementasi Rambu Dilarang Putar Balik Dan Belok Kanan

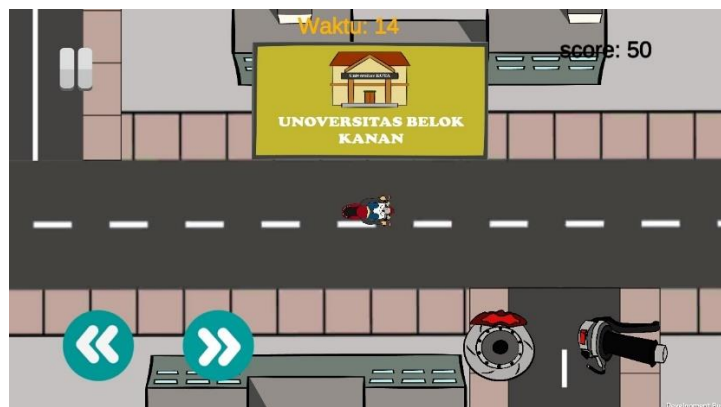
Rambu ini terletak pada bagian ujung jalan khusus sepeda motor. *Player* dilarang belok kanan selain ada rambu larang juga karna jalan tersebut merupakan jalan satu arah. Ketika *player* sampai di ujung persimpangan akan muncul panel yang memberikan petunjuk arah untuk menuju Universitas Kota. Untuk lebih jelasnya dapat kita lihat pada Gambar 4.40.



Gambar 4.45 Implementasi Rambu Dilarang Putar Balik Dan Belok Kanan

10. Implementasi Antarmuka Informasi Arah Jalan

Informasi arah jalan bertujuan untuk memberikan petunjuk ke mana arah yang harus dilewati oleh *player*. Jika tidak mengikuti arahan yang diberikan *player* mendapatkan sanksi pengurangan skor. Informasi arah jalan akan muncul ketika *player* mendekati sebuah persimpangan. Contoh implementasi informasi arah jalan dapat dilihat pada Gambar 4.41.



Gambar 4.46 Implementasi Informasi Arah Jalan

11. Implementasi Antarmuka *Player* Sampai Tujuan

Tampilan pada saat *player* sampai di tempat tujuan pada arena perkotaan terlihat gedung Universitas kota di kiri jalan. Jalan di depan Universitas dilengkapi dengan marka jalan dilarang parkir, seperti yang terlihat pada Gambar 4.42. Setelah *player* tepat di depan gedung Universitas akan muncul panel yang di dalamnya terdapat dua pilihan yaitu parkir di depan Universitas dan lanjut menuju parkir, seperti yang terlihat pada Gambar 4.43.



Gambar 4.47 Implementasi Antarmuka Depan Universitas



Gambar 4.48 Implementasi Panel Pilihan Parkir

4.1.1.3 Implementasi Suara Pada *Game*

Implementasi suara pada *game* adalah proses pemberian suara kepada objek pada *game* atau suatu kondisi yang terjadi ketika *game* dijalankan. Suara yang terdapat pada *game* adalah suara sepeda motor pemain dan suara ketika pemain melakukan pelanggaran rambu lalu lintas. Suara yang digunakan di dalam *game* didapat dengan cara mengunduh video youtube, yang kemudian dirubah menjadi format (mp3.).

4.2 Kode Program Dalam *Game*

Untuk memberikan fungsi pada seluruh komponen yang digunakan di dalam *game* dibutuhkan kode program agar semua komponen dapat berfungsi. Pada pembuatan "*Game* Edukasi Berlalu Lintas Untuk Roda Dua Berbasis Android" kode program dibuat menggunakan bahasa pemrograman C#. Penulisan kode program menggunakan perangkat lunak Visual Studio 2017. Berikut beberapa kode program yang digunakan di dalam *game*:

- a. Gerak Motor

Kode ini digunakan untuk menggerakkan karakter utama yang menggunakan sudut dari samping. Kode program dapat dilihat pada Gambar 4.44.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class gerakMotor : MonoBehaviour
{
    public float speed = 500;
    public WheelJoint2D rodaBelakang;
    private float gerak = 0f;

    void Update() {
        if (kontrolMotor.tombolMaju == true) {
            gerak = -1 * speed;
        }
        else if(kontrolMotor.tombolRem == true) {
            gerak = 1 * speed;
        }
        else {
            gerak = 0;
        }
    }

    private void FixedUpdate() {
        if (gerak == 0f) {
            rodaBelakang.useMotor = false;
        }
        else {
            JointMotor2D motor = new JointMotor2D
            { motorSpeed = gerak, maxMotorTorque = 500 };
            rodaBelakang.motor = motor;
        }
    }
}
```

Gambar 4.49 Code Gerak Motro

b. *Timer Game*

Kode ini digunakan untuk mebuat waktu mundur pada *game*. Dapat dilihat pada Gambar 4.45.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.SceneManagement;

public class codeWaktu : MonoBehaviour
{
    [SerializeField] private Text WaktuText;
    [SerializeField] private float Waktu;

    public static float WaktuBerjalan;
```

```

private bool mulai = true;
public GameObject gamePlay, Tbl_clsPengenalan,
Tbl_balikR2, Tbl_balikR, Tbl_balikK;

// Start is called before the first frame update
void Start()
{
    WaktuBerjalan = Waktu;
    mulai = false;
}

// Update is called once per frame
void Update()
{
    if(WaktuBerjalan >= 0f && mulai)
    {
        WaktuBerjalan -= Time.deltaTime;
        WaktuText.text = "Waktu: " + WaktuBerjalan.ToString("0");
    }
    else if(WaktuBerjalan <= 0f)
    {
        mulai = false;
        WaktuText.text = "Waktu: 0";
        SceneManager.LoadScene("GameOver");
    }
}

public void Naktf_Pengenalan()
{
    WaktuBerjalan = Waktu;
    mulai = true;
    Tbl_clsPengenalan.SetActive(false);
    gamePlay.SetActive(false);
    Tbl_balikR2.SetActive(false);
    Tbl_balikR.SetActive(false);
    Tbl_balikK.SetActive(false);
}
}

```

Gambar 4.50 Contoh Kode Pemberian Waktu

c. Skor *Game*

Kode ini digunakan untuk menampilkan tulisan “Score” dan nilai skor pada *game*.

Kode program dapat dilihat pada Gambar 4.46.

```

using System.Collections;
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.SceneManagement;

public class ScoreScript : MonoBehaviour
{
    public static int nilaiscore = 100;
    Text score;
    public static int newNilai;

    // Start is called before the first frame update
    void Start()
    {
        newNilai = nilaiscore;
        newNilai = 100;
    }
}

```

```

        score = GetComponent<Text>();
    }

    // Update is called once per frame
    void Update()
    {
        score.text = "score: " + newNilai;
        if(newNilai <= 0)
        {
            SceneManager.LoadScene("GameOver");
        }
    }
}

```

Gambar 4.51 Contoh Kode Penampilan Skor

d. Pindah *Scene*

Kode ini digunakan untuk memuat *scene* yang ingin ditampilkan. Kode program yang digunakan dapat dilihat pada Gambar 4.47.

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class lampuMerah : MonoBehaviour
{
    void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.gameObject.tag.Equals("Player"))
        {
            SceneManager.LoadScene("Game1Atas");
            Debug.Log("SceneLampumerah");
        }
    }
}

```

Gambar 4.52 Contoh Kode Untuk Menampilkan *Scene* Lain

e. Lampu Merah

Kode ini digunakan untuk membuat mekanisme yang digunakan pada lampu merah.

Kode program yang digunakan dapat dilihat pada Gambar 4.48.

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Lampu_merah : MonoBehaviour
{
    public GameObject merah, kuning,
    hijau,merah1,kuning1,hijau1,merah2,kuning2,hijau2,merah3,kuning3,hijau3,
    colider;
    float currentTime = 0f;
    float startingTime = 48f;
    // Start is called before the first frame update
    void Start()
    {
        currentTime = startingTime;
        merah.SetActive(false);
        kuning.SetActive(false);
    }
}

```

```
        hijau.SetActive(false);
        merah1.SetActive(false);
        kuning1.SetActive(false);
        hijau1.SetActive(false);
        merah2.SetActive(false);
        kuning2.SetActive(false);
        hijau2.SetActive(false);
        merah3.SetActive(false);
        kuning3.SetActive(false);
        hijau3.SetActive(false);
        colider.SetActive(true);
    }

    // Update is called once per frame
    void Update()
    {
        currentTime -= 1 * Time.deltaTime;
        for (int i = 0; i<5; i++)
        {

            if (currentTime >= 38)
            {
                hijau.SetActive(false);
                kuning.SetActive(false);
                merah.SetActive(true);
                merah1.SetActive(false);
                kuning1.SetActive(false);
                hijau1.SetActive(true);
                merah2.SetActive(true);
                kuning2.SetActive(false);
                hijau2.SetActive(false);
                merah3.SetActive(true);
                kuning3.SetActive(false);
                hijau3.SetActive(false);
            }

            else if (currentTime <38 && currentTime >36)
            {
                hijau.SetActive(false);
                merah.SetActive(true);
                kuning.SetActive(false);
                merah1.SetActive(false);
                kuning1.SetActive(true);
                hijau1.SetActive(false);
                merah2.SetActive(false);
                kuning2.SetActive(true);
                hijau2.SetActive(false);
                merah3.SetActive(true);
                kuning3.SetActive(false);
                hijau3.SetActive(false);
            }

            else if (currentTime <=36 && currentTime>26)
            {
                merah.SetActive(true);
                kuning.SetActive(false);
                hijau.SetActive(false);
                merah1.SetActive(true);
                kuning1.SetActive(false);
                hijau1.SetActive(false);
                merah2.SetActive(false);
                kuning2.SetActive(false);
                hijau2.SetActive(true);
                merah3.SetActive(true);
                kuning3.SetActive(false);
            }
        }
    }
}
```

```
        hijau3.SetActive(false);
    }
    else if(currentTime<=26 && currentTime > 24)
    {
        merah.SetActive(true);
        kuning.SetActive(false);
        hijau.SetActive(false);
        merah1.SetActive(true);
        kuning1.SetActive(false);
        hijau1.SetActive(false);
        merah2.SetActive(false);
        kuning2.SetActive(true);
        hijau2.SetActive(false);
        merah3.SetActive(false);
        kuning3.SetActive(true);
        hijau3.SetActive(false);
    }
    else if(currentTime <= 24 && currentTime >14)
    {
        merah.SetActive(true);
        kuning.SetActive(false);
        hijau.SetActive(false);
        merah1.SetActive(true);
        kuning1.SetActive(false);
        hijau1.SetActive(false);
        merah2.SetActive(true);
        kuning2.SetActive(false);
        hijau2.SetActive(false);
        merah3.SetActive(false);
        kuning3.SetActive(false);
        hijau3.SetActive(true);
    }
    else if(currentTime<=14 && currentTime > 12)
    {
        merah.SetActive(false);
        kuning.SetActive(true);
        hijau.SetActive(false);
        merah1.SetActive(true);
        kuning1.SetActive(false);
        hijau1.SetActive(false);
        merah2.SetActive(true);
        kuning2.SetActive(false);
        hijau2.SetActive(false);
        merah3.SetActive(false);
        kuning3.SetActive(true);
        hijau3.SetActive(false);
    }
    else if(currentTime<12 && currentTime > 2)
    {
        merah.SetActive(false);
        kuning.SetActive(false);
        hijau.SetActive(true);
        merah1.SetActive(true);
        kuning1.SetActive(false);
        hijau1.SetActive(false);
        merah2.SetActive(true);
        kuning2.SetActive(false);
        hijau2.SetActive(false);
        merah3.SetActive(true);
        kuning3.SetActive(false);
        hijau3.SetActive(false);
        colider.SetActive(false);
    }
    else if (currentTime<=2 && currentTime > 0)
    {
        merah.SetActive(false);
```

```

        kuning.SetActive(true);
        hijau.SetActive(false);
        merah1.SetActive(false);
        kuning1.SetActive(true);
        hijau1.SetActive(false);
        merah2.SetActive(true);
        kuning2.SetActive(false);
        hijau2.SetActive(false);
        merah3.SetActive(true);
        kuning3.SetActive(false);
        hijau3.SetActive(false);
        colider.SetActive(true);
    }
    else if(currentTime <=0)
    {
        hijau.SetActive(false);
        currentTime = 48;
    }
}
}
}

```

Gambar 4.53 Kode Program Lampu Merah

f. Pengurangan Skor

Kode ini digunakan untuk mengurangi skor yang ada pada *game*. Skor akan berkurang jika player melangar atau menabrak objek yang ada pada *game*. Kode program yang digunakan dapat dilihat pada Gambar 4.49.

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class scoreLampumerah : MonoBehaviour
{
    public AudioSource SuaraSalah;
    void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.gameObject.tag.Equals("Player"))
        {
            SuaraSalah.Play();
            scoreGameAtas.NewnilaiAwalGameAtas -= 10;
            Debug.Log("nabrak collider");
        }
    }
}

```

Gambar 4.54 Contoh Kode Untuk Pengurangan Skor

g. Kode Penampila Panel

Kode ini digunakan untuk menampilkan panel yang digunakan di dalam *game* seperti panel pengenalan rambu, panel petunjuk arah, dan yang lain-lain. Contoh kode program yang digunakan dapat dilihat pada Gambar 4.50.

```

using System.Collections;
using System.Collections.Generic;

```

```

using UnityEngine;

public class pengenal_awal : MonoBehaviour
{
    public GameObject Kontrol, Rambu, Rambu2, gamePlay, Tbl_Lanjut, Tbl_Rambu2,
    Tbl_gamePlay, Tbl_clsPengenalan, Tbl_balikR2, Tbl_balikR, Tbl_balikK;
    float waktuAwal = 2f;
    float waktuBerjalan = 0f;
    // Start is called before the first frame update
    void Start()
    {
        waktuBerjalan = waktuAwal;
        Kontrol.SetActive(false);
        Rambu.SetActive(false);
        Rambu2.SetActive(false);
        gamePlay.SetActive(false);
        Tbl_Lanjut.SetActive(false);
        Tbl_Rambu2.SetActive(false);
        Tbl_gamePlay.SetActive(false);
        Tbl_clsPengenalan.SetActive(false);
        Tbl_balikK.SetActive(false);
        Tbl_balikR.SetActive(false);
        Tbl_balikR2.SetActive(false);
    }

    // Update is called once per frame
    void Update()
    {
        waktuBerjalan -= 1 * Time.deltaTime;
        if (waktuBerjalan <=1 && waktuBerjalan >0 )
        {
            Kontrol.SetActive(true);
            Tbl_Lanjut.SetActive(true);
        }
    }

    public void Aktf_Rambu()
    {
        Rambu.SetActive(true);
        Kontrol.SetActive(false);
        Tbl_Lanjut.SetActive(false);
        Tbl_Rambu2.SetActive(true);
        Tbl_gamePlay.SetActive(false);
        Tbl_balikK.SetActive(true);
    }

    public void BalikKontrol()
    {
        Rambu.SetActive(false);
        Kontrol.SetActive(true);
        Tbl_balikK.SetActive(false);
        Tbl_Rambu2.SetActive(false);
        Tbl_Lanjut.SetActive(true);
    }

    public void Aktf_Rambu2()
    {
        Rambu.SetActive(false);
        Rambu2.SetActive(true);
        Tbl_Rambu2.SetActive(false);
        Tbl_gamePlay.SetActive(true);
        Kontrol.SetActive(false);
        Tbl_Lanjut.SetActive(false);
        Tbl_balikR.SetActive(true);
    }
}

```

```

}

public void BalikR1()
{
    Rambu.SetActive(true);
    Rambu2.SetActive(false);
    Tbl_Rambu2.SetActive(true);
    Tbl_balikR.SetActive(false);
    Tbl_Lanjut.SetActive(false);
    Tbl_gamePlay.SetActive(false);
    Tbl_balikK.SetActive(true);
    Kontrol.SetActive(false);
}

public void Aktf_gamePlay()
{
    Rambu2.SetActive(false);
    gamePlay.SetActive(true);
    Tbl_gamePlay.SetActive(false);
    Tbl_clsPengenalan.SetActive(true);
    Tbl_balikR2.SetActive(true);
    Kontrol.SetActive(false);
    Tbl_Lanjut.SetActive(false);
}

public void BalikR2()
{
    gamePlay.SetActive(false);
    Rambu2.SetActive(true);
    Tbl_gamePlay.SetActive(true);
    Tbl_clsPengenalan.SetActive(false);
    Tbl_balikR.SetActive(true);
    Tbl_balikR2.SetActive(false);
    Kontrol.SetActive(false);
    Tbl_Lanjut.SetActive(false);
}
}

```

Gambar 4.55 Contoh Kode Untuk Menampilkan Panel

h. Kode penggerak *Non Playable Karakter* (NPC)

Kode ini digunakan untuk menggerakkan objek kendaraan roda dua selain karakter utama secara otomatis. Kode yang digunakan dapat dilihat pada Gambar 4.51.


```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class AI_Patrol : MonoBehaviour
{
    public GameObject Co1, Co2, Co3, Co11;
    float currentTime = 0f;
    float startingTime = 48f;
    public float speed;

    private void Start()
    {
        currentTime = startingTime;
    }

    // Update is called once per frame
    void Update()
    {
        currentTime -= 1 * Time.deltaTime;
        if (currentTime < 48)
        {
            Co1.transform.Translate(0, 2 * Time.deltaTime * speed, 0);
            Co11.transform.Translate(0, 2 * Time.deltaTime * speed, 0);
        }

        if(currentTime < 36)
        {
            Co2.transform.Translate(0, 2 * Time.deltaTime * speed, 0);
        }
        if (currentTime < 24)
        {
            Co3.transform.Translate(0, 2 * Time.deltaTime * speed, 0);
        }
    }
}

```

Gambar 4.56 Kode Penggerak NPC

4.3 Hasil Pengujian

Pengujian terdiri dari pengujian usabilitas dan pengujian materi. Pengujian usabilitas dilakukan untuk mencari tahu tingkat kualitas pengalaman pengguna dengan antarmuka “Gamae Edukasi Belalu Lintas Untuk Roda Dua Berbasis Android”. Pengujian materi dilakakun untuk mengetahui kesesuaian materi pada *game* dan kesesuaian *game* dengan rumusan masalah yang telah dibuat sebelumnya.

4.3.1 Hasil Pengujian Usabilitas

Hasil pengujian usabilitas diambil dari hasil jawaban soal kuesioner yang diberikan responden setelah memainkan “*Game* edukasi Berlalu Lintas Untuk Roda Dua Berbasis Android”. Hasil jawaban kuesioner dapat dilihat pada Tabel 4.1.

Tabel 4.1 Hasil Jawaban Kuesioner Pengujian Usabilitas

| No | Pertanyaan | 1 | 2 | 3 | 4 | 5 |
|----|--|---|---|---|---|---|
| 1. | Saya dapat memahami apa yang harus dilakukan pada setiap tampilan pada <i>game</i> . | | 2 | 4 | 8 | 6 |
| 2. | Tampilan tombol pada <i>game</i> sudah sesuai dengan fungsinya. | | | 5 | 7 | 8 |
| 3. | Desain tombol kontrol pada game mudah dipahami. | | 2 | 7 | 5 | 6 |
| 4. | Informasi yang diberikan memudahkan anda dalam bermain <i>game</i> ini. | 1 | 4 | 5 | 5 | 5 |
| 5. | Saya dapat memahami peraturan dalam bermain <i>game</i> ini. | 2 | 3 | 4 | 9 | 2 |

Dari kuesioner yang dibuat memiliki lima pilihan jawaban yang mempunyai gradasi dari sangat setuju hingga sangat tidak setuju. Berikutnya adalah pemberian nilai terhadap jawaban responden. Nilai yang dibuat dapat dilihat pada Tabel 4.2.

Tabel 4.2 Nilai Jawaban

| Skala Jawaban | Nilai |
|---------------------|-------|
| Sangat Tidak Setuju | 1 |
| Kurang Setuju | 2 |
| Cukup Setuju | 3 |
| Setuju | 4 |
| Sangat Setuju | 5 |

Setelah penentuan nilai dari masing-masing jawaban. Dibutuhkan skor ideal untuk mendapatkan nilai skor maksimum. Rumus yang digunakan dapat dilihat pada persamaan (4.1).

$$\text{Skor Ideal} = \text{Nilai skala} \cdot \text{Jumlah Responden} \quad (4.1)$$

Hasil perhitungan dengan rumus di atas dapat dilihat pada Tabel 4.3.

Tabel 4.3 Skor Ideal

| Skor ideal | Skala |
|--------------------|-------|
| $1 \cdot 20 = 20$ | 1 |
| $2 \cdot 20 = 40$ | 2 |
| $3 \cdot 20 = 60$ | 3 |
| $4 \cdot 20 = 80$ | 4 |
| $5 \cdot 20 = 100$ | 5 |

Nilai maksimum yang didapat adalah 100. Nilai maksimum digunakan untuk mencari indeks persentase dari semua jawaban yang didapat dari responden. Sebelum mencari indeks persentase terlebih dulu kita mencari total skor pada kuisioner yang telah diisi. Untuk mencari total skor dapat dilihat pada Tabel 4.4.

Tabel 4.4 Total Skor Soal Nomor Satu

| Skala Jawaban | <i>Skala jawaban · Nilai skala</i> | Hasil |
|---------------|------------------------------------|-----------|
| 1 | $0 \cdot 1$ | 0 |
| 2 | $2 \cdot 2$ | 4 |
| 3 | $4 \cdot 3$ | 12 |
| 4 | $8 \cdot 4$ | 32 |
| 5 | $6 \cdot 5$ | 30 |
| Total | | 78 |

Setelah total skor didapat selanjutnya mencari nilai indeks persentase menggunakan persamaan (4.2).

$$p = \frac{f}{n} \cdot 100\% \quad (4.2)$$

P: Persentase

F: Total skor

N: nilai maksimum

Dengan menggunakan persamaan di atas didapat nilai persentase sebesar 78% untuk soal nomor satu. Setelah didapat indeks persentase, langkah berikutnya adalah menyesuaikan indeks interval dan skala . Indeks interval yang digunakan dapat dilihat pad Tabel 4.5.

Tabel 4.5 Indeks Interval

| Indeks | Skala |
|---------------|---------------------|
| 0% - 19,99% | Sangat Tidak Setuju |
| 20% -39,9% | Tidak Setuju |
| 40% - 59,9% | Ragu-ragu |
| 60% - 79,9% | Setuju |
| 80% - 100% | Sangat Setuju |

Berdasarkan Tabel 4.5. indeks Interval dengan nilai 78%, maka dapat disimpulkan bahwa responden “Setuju” untuk jawaban soal nomor satu. Hal yang sama dilakukan untuk nomor berikutnya sehingga akan menghasilkan seperti Tabel 4.6.

Tabel 4.6 Indeks Persentase Seluruh Pertanyaan

| Nomor Soal | Indeks Persentase |
|------------|-------------------|
| 1 | 78% |
| 2 | 83% |
| 3 | 75% |
| 4 | 69% |
| 5 | 66% |

Dari Tabel 4.6. rata-rata indeks persentase yang didapat sebesar 74,2%, maka dapat disimpulkan kualitas pengalaman pengguna dengan antarmuka “*Game Edukasi Berlalu Lintas Untuk Roda Dua Berbasis Android*” 74,2% menyatakan setuju antarmuka yang digunakan berkualitas. Namun dilihat dari hasil indeks Persentase pada setiap soal menunjukkan hanya tiga pertanyaan yang mendapatkan nilai cukup baik. Terdapat dua pertanyaan yang mendapatkan hasil yang kurang baik yaitu soal nomor empat dan nomor

lima. Masing-masing hanya mendapatkan nilai 69% dan 66%. Soal nomor empat dan nomor lima adalah soal yang saling berkaitan. Beberapa responden memberikan nilai kurang baik terkait dengan informasi yang diberikan di dalam *game*. Sehingga dapat disimpulkan bahwa informasi yang disediakan di dalam *game* masih belum sepenuhnya memberikan kemudahan kepada responden dalam memainkan *game*.

4.3.2 Hasil Pengujian Materi

Hasil pengujian materi didapat melalui jawaban kuesioner yang telah dibagikan. Terdapat dua kuesioner yang digunakan dalam pengujian materi yaitu kuesioner pengujian kesesuaian materi di dalam *game*, pengujian materi di dalam *game* dan pengujian *satisfaction*. Kuesioner pengujian materi pada *game* diberikan kepada ahli yang mengerti tentang rambu lalu lintas. pengujian *satisfaction* kepada beberapa responden.

Pengujian Materi Pada *Game*

Pengujian ini menggunakan kuesioner yang ditujukan kepada orang yang benar-benar memahami tentang rambu lalu lintas. Kuesioner ini diisi oleh satu Polisi Lalu Lintas yang sedang bertugas. Hasil dari kuesioner yang diberikan dapat dilihat pada Tabel 4.7.

Tabel 4.7 Hasil Kuesione Pengujian Materi

| No | Pertanyaan | 1 | 2 | 3 | 4 | 5 |
|----|--|---|---|---|---|---|
| 1. | Apakah rambu yang ada pada <i>game</i> sering dilanggar? | | | | 1 | |
| 2. | Apakah fungsi rambu yang ada pada <i>game</i> sudah sesuai dengan kegunaannya? | | | | 1 | |
| 3. | Apakah penempatan marka dan rambu pada <i>game</i> sudah sesuai? | | | | 1 | |
| 4. | Bagaimana kesesuaian rintangan di dalam <i>game</i> dengan kehidupan nyata? | | | | 1 | |

Dari kuesioner yang dibuat memiliki lima pilihan jawaban yang mempunyai gradasi dari sangat setuju hingga sangat tidak setuju. Berikutnya adalah pemberian nilai terhadap jawaban responden. Nilai yang dibuat dapat dilihat pada Tabel 4.8.

Tabel 4.8 Nilai Jawaban

| Skala Jawaban | Nilai |
|---------------------|-------|
| Sangat Tidak Setuju | 1 |
| Kurang Setuju | 2 |
| Cukup Setuju | 3 |
| Setuju | 4 |
| Sangat Setuju | 5 |

Setelah penentuan nilai dari masing-masing jawaban. Dibutuhkan skor ideal untuk mendapatkan nilai skor maksimum. Rumus yang digunakan dapat dilihat pada persamaan (4.3).

$$\text{Skor Ideal} = \text{Nilai skala} \cdot \text{Jumlah Responden} \quad (4.3)$$

Hasil perhitungan dengan rumus di atas dapat dilihat pada Tabel 4.9.

Tabel 4.9 Skor Ideal

| Skor ideal | Skala |
|-----------------|-------|
| $1 \cdot 1 = 1$ | 1 |
| $2 \cdot 1 = 2$ | 2 |
| $3 \cdot 1 = 3$ | 3 |
| $4 \cdot 1 = 4$ | 4 |
| $5 \cdot 1 = 5$ | 5 |

Nilai maksimum yang didapat adalah 5. Nilai maksimum digunakan untuk mencari indeks persentase dari semua jawaban yang didapat dari responden. Sebelum mencari indeks persentase terlebih dulu kita mencari total skor pada kuisisioner yang telah diisi. Untuk mencari total skor dapat dilihat pada Tabel 4.10.

Tabel 4.10 Total Skor Soal Nomor Satu

| Skala Jawaban | Skala jawaban · Nilai skala | Hasil |
|----------------------|------------------------------------|--------------|
| 1 | $0 \cdot 1 = 1$ | 0 |
| 2 | $0 \cdot 2 = 1$ | 0 |
| 3 | $0 \cdot 3 = 1$ | 0 |
| 4 | $1 \cdot 4 = 1$ | 4 |
| 5 | $0 \cdot 5 = 1$ | 0 |
| Total | | 4 |

Setelah total skor didapat selanjutnya mencari nilai indeks persentase menggunakan persamaan (4.4).

$$p = \frac{f}{n} \cdot 100\% \quad (4.4)$$

P: Persentase

F: Total skor

N: nilai maksimum

Dengan menggunakan persamaan diatas didapat nilai persentase sebesar 80% untuk soal nomor satu. Setelah didapat indeks persentase, langkah berikutnya adalah melihat pada interval penilaian angka 80% masuk dalam indeks sangat setuju. Indeks interval yang digunakan dapat dilihat pad Tabel 4.11.

Tabel 4.11 Indeks Interval

| Indeks | Skala |
|---------------|---------------------|
| 0% - 19,99% | Sangat Tidak Setuju |
| 20% -39,9% | Tidak Setuju |
| 40% - 59,9% | Ragu-ragu |
| 60% - 79,9% | Setuju |
| 80% - 100% | Sangat Setuju |

Berdasarkan Tabel 4.6, indeks Interval dengan nilai 80%, maka dapat disimpulkan bahwa responden “Sangat Setuju” untuk jawaban soal nomor satu. Hal yang sama dilakukan untuk nomor berikutnya sehingga akan menghasilkan seperti Tabel 4.2.

Tabel 4.12 Indeks Persentase Seluruh Pertanyaan

| Nomor Soal | Indeks Persentase |
|------------|-------------------|
| 1 | 80% |
| 2 | 80% |
| 3 | 80% |
| 4 | 80% |

Dari Tabel 4.7, rata-rata indeks persentase yang didapat sebesar 80%, maka dapat disimpulkan responden sangat setuju dengan materi yang ada pada “*Game* Edukasi Berlalu Lintas Untuk Roda Dua Berbasis Android”.

Pengujian Manfaat *Game*

Pengujian ini bertujuan untuk mengetahui apakah *game* yang dibangun sudah sesuai dengan rumusan masalah yang telah dibuat sebelumnya. Pengujian dilakukan dengan menyebarkan kuesioner dengan pertanyaan yang meliputi apakah “*Game* Edukasi Berlalu Lintas Untuk Roda Dua Berbasis Android” dapat mengedukasi dan bagaimana penilaian tampilan pada *game* ini. Hasil dari kuesioner pengujian dapat dilihat pada Tabel 4.13. Pengujian dilakukan terhadap dua puluh responden.

Tabel 4.13 Hasil Kuesioner Pengujian Manfaat *Game*

| No | Pertanyaan | 1 | 2 | 3 | 4 | 5 |
|----|---|---|---|---|----|---|
| 1. | Apakah <i>game</i> ini dapat memberikan pengetahuan kepada anda tentang rambu lalu lintas khususnya untuk roda dua? | | | 5 | 12 | 3 |
| 2. | Apakah <i>game</i> ini menambah pengetahuan kepada anda tentang fungsi rambu lalu lintas yang ada? | | | 6 | 9 | 5 |

| | | | | | | |
|----|---|--|--|---|----|---|
| 3. | Apakah <i>game</i> ini memberikan pengetahuan tentang rambu lalu lintas yang belum anda ketahui sebelumnya? | | | 7 | 6 | 7 |
| 4. | Apakah tampilan <i>game</i> ini menarik untuk dimainkan? | | | 9 | 8 | 3 |
| 5. | Apakah tampilan <i>gameplay</i> pada <i>game</i> ini sudah cukup baik? | | | 2 | 16 | 2 |
| 6. | Bagaimana kemudahan dalam memainkan <i>game</i> ini? | | | 5 | 11 | 4 |
| 7. | Apakah informasi petunjuk permainan membantu dalam menyelesaikan <i>game</i> ini? | | | 1 | 14 | 5 |

Dari hasil data yang sudah didapat akan dilakukan pengukuran Indeks persentase dari setiap soal menggunakan metode yang sama seperti sebelumnya. Hasil dari perhitungan menggunakan metode skala likert dapat dilihat pada Tabel 4.14.

Tabel 4.14 Indeks Persentase Pengukuran Kuesione Pengujian Manfaat

| Nomor Soal | Indeks Persentase |
|------------|-------------------|
| 1 | 78% |
| 2 | 79% |
| 3 | 80% |
| 4 | 74% |
| 5 | 80% |
| 6 | 79% |
| 7 | 84% |

Rata-rata indeks persentase pada setiap soal kuesioner yang didapat adalah 79,1% dilihat dari indeks interval, hasil seluruh jawaban yang diberikan responden menyatakan setuju.

Dari hasil pengujian yang telah dilakukan mulai pengujian kompatibilitas sampai dengan pengujian manfaat *game*, maka hasil analisis adalah sebagai berikut:

a. Menambah Pengetahuan Pengguna

Dari hasil jawaban responden mengenai nilai edukasi yang terdapat pada “*Game Edukasi Berlalu Lintas Untuk Roda Dua Berbasis Android*” menghasilkan nilai rata-rata indeks Persentasi dari soal nomor satu sampai dengan nomor tiga adalah 79% dapat diartikan sebagian besar responden setuju bahwa “*Game Edukasi Berlalu Lintas Untuk Roda Dua Berbasis Android*” dapat memberikan edukasi kepada penggunanya.

b. Tingkat Kepuasan Tampilan

Dari hasil jawaban soal nomor empat dan nomor lima, menghasilkan nilai rata-rata indeks persentase yang adalah 77%. Dengan begitu “*Game Edukasi Berlalu Lintas Untuk Roda Dua Berbasis Android*” memiliki tampilan yang menarik.

c. Kemudahan Penggunaan Aplikasi

Dari hasil perhitungan skala likert, indeks persentase untuk soal yang menanyakan tingkat kemudahan dalam memainkan “*Game Edukasi Berlalu Lintas Untuk Roda Dua Berbasis Android*” mendapat nilai 81,5%. Dengan begitu “*Game Edukasi Berlalu Lintas Untuk Roda Dua Berbasis Android*” merupakan *game* yang mudah dimainkan.

4.4 Analisis Kelebihan Dan Kekurangan Sistem

Dalam pengembangan sistem analisis kelebihan dan kekurangan sistem diperlukan sebagai bahan evaluasi setelah selesai melakukan pengujian. Kekurangan yang didapat akan dijadikan sebagai bahan pertimbangan untuk mengembangkan penelitian untuk kedepannya agar lebih sempurna lagi. Kelebihan dan kekurangan dari penelitian “*Game Edukasi Berlalu Lintas Untuk Roda Dua Berbasis Android*” ini adalah:

4.4.1 Kelebihan Sistem

Kelebihan dari “*Game Edukasi Berlalu Lintas Untuk Roda Dua Berbasis Android*” adalah sebagai berikut:

- a. “*Game Edukasi Berlalu Lintas Untuk Roda Dua Berbasis Android*” dapat memberikan tambahan pengetahuan tentang rambu lalu lintas.

- b. “*Game* Edukasi Berlalu Lintas Untuk Roda Dua Berbasis Android” dapat dimainkan dengan mudah bagi penggunanya.
- c. “*Game* Edukasi Berlalu Lintas Untuk Roda Dua Berbasis Android” memiliki tampilan yang menarik.

4.4.2 Kekurangan

Kekurang dari “*Game* Edukasi Berlalu Lintas Untuk Roda Dua Berbasis Android” adalah sebagai berikut:

- a. Belum ada pilihan karakter utama.
- b. *Non Playable Karakter* (NPC) pada *game* kurang bervariasi.
- c. Lintasan yang belum bervariasi.
- d. Masih ada beberapa pengguna yang belum mengerti mengenai peraturan yang ada pada *game*.
- e. Informasi untuk peraturan bermain belum cukup jelas sehingga beberapa *player* masih belum paham cara memainkan *game* ini.