BAB I

PENDAHULUAN

A. Latar Belakang

Game jika diartikan secara bahasa (berasal dari bahasa inggris) adalah permainan. Dengan bantuan teknologi, kini *game* dapat diartikan secara lebih luas lagi. Jadi, secara istilah *game* adalah permainan yang diprogram pada suatu perangkat yang dapat dijalankan secara offline maupun online. Game yang dibuat biasanya mempunyai peraturan, arena, user, dan lain lain. Seiring perkembangan nya zaman *game* mulai berkembang dan mempunyai banyak genre seperti mmorg, battle royal, moba, dan banyak lainnya lagi yang bertujuan untuk meningkatkan minat dan bakat pemain saat memainkan *gam*e.

Pada saat ini *game* bukan hanya dijadikan media hiburan saja ada juga beberapa orang yang menggunakan *game* sebagai media edukasi dan ada juga beberapa orang yang menjadikan *game* sebagai ladang pekerjaan bahkan beberapa *game* menjadi cabang olahraga dibeberapa pertandingan ajang internasional. Dengan beberapa alasan yang dikemukakan sebelumnya menyebabkan peningkatan jumlah orang yang memainkan *game*, sehingga pada saat ini *game* sudah dikenal oleh semua orang tampa mengenal batasan umur maupun genre.

Permainan tradisional merupakan suatu hasil budaya masyarakat yang telah ada sejak lama hingga sekarang, permainan tradisional menjadi salah satu permainan yang dapat bersaing dengan permainan yang lainnya. Adapun salah satu jenis permainan tradisonal yaitu sepeda ontel. Walaupun permainan tradisional ini seru

dimainkan akan tetapi anak zaman sekarang lebih menggemari permainan elektronik karena dalam bermain *game* lebih praktis dan dapat menjangkau ke seluruh bagian dunia. Sehingga generasi muda dapat mendapatkan kenangan atau kebahagiaan walaupun tanpa mengendarai sepeda onthel yang asli yang sekarang saja sudah sangat langka, pembuatan *game* bertema sepeda ontel memiliki tujuan untuk mengingat kembali sepeda ontel pada zaman dahulu, khususnya sepeda ontel jawa dengan bentuk lain yaitu dengan cara membuatkannya kedalam bentuk permainan untuk android.

Unity adalah salah satu tools yang dapat digunakan untuk membuat game tanpa harus menulis kode pemrograman, karena sebagian besar logika untuk game dapat dibuat menggunakan menu tools yang ada di unity.

Berdasarkan uraian tersebut, akan diangkat topik untuk tugas akhir dengan judul "**Aplikasi *Game Balap Sepeda Ontel Berbasis Android***".

B. Rumusan Masalah

Berdasarkan uraian latar belakang penulis dapat mengambil rumusan masalah yaitu bagaimana cara membuat game balap sepeda ontel dengan menggunakan unity.

C. Batasan Masalah

Pada penelitian ini terbatas pada batasan masalah berikut :

1. *Game* bersifat *singel player* dan *offline*.
2. *Game* ini dibuat dengan menggunakan *unity*.
3. *Game* dengan tampilan 2D.
4. *Game* ini yang terbuat dari 3 level.

D. Tujuan Penelitian

Berdasarkan rumusan masalah di atas, tujuan penulisan penelitian ini adalah untuk mengetahui bagaimana cara membuat sebuah *game* dengan menggunakan *unity.*

E. Manfaat Penelitian

1. Dapat memberi wawasan tentang sepeda ontel kepada pemain.
2. Menghibur para pemain.
3. Menjadi referensi untuk pengembang *game* pemula lainnya.

F. Sistematika Penulisan

Struktur penulisan skripsi ini adalah :

BAB I : PENDAHULUAN

Berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, metodologi penelitian, dan sistematika penulisan dari pembuatan skripsi ini.

BAB II : LANDASAN TEORI

Berisi tentang penelitian terdahulu dan dasar teori yang mendukung pembuatan skripsi ini.

BAB III: METODOLOGI PENELITIAN

Berisi tentang letak lokasi serta waktu, jenis penelitian, metode pengumpulan data, alat dan bahan penelitian, tahap penelitian, metode pengujian, dan analisa sistem.

BAB IV : HASIL DAN PERANCANGAN SISTEM

Berisi tentang gambaran umum, analisis, meliputi pendefinisian dan pemodelan sistem dalam bentuk *use case*, *Class* diagram, *Sequence* diagram serta disain sistem yang meliputi disain database, dan pegujian *system* dengan *whitebox* dan *blackbox*.

BAB V : PENUTUP

Berisi kesimpulan yang dapat diambil dari penulisan tugas akhir ini dan saran-saran pengembangannya.

DAFTAR PUSTAKA

LAMPIRAN

BAB II

TINJAUAN PUSTAKA

A. Kajian Teori

1. Algoritma dan Pemrograman

Algoritma adalah sekumpulan langkah rinci yang ditujukan untuk computer dalam menyelsaikan suatu masalah. Algotima dibuat pada tahapan peracangan program. Algoritma memiliki peranan penting untuk menghubungkan antara keluaran yang dikehendaki dan masukan-masukan yang tersedia. Adapun karakteristik algoritma yaitu:

1. Finiteness (Keterbatasan): Algoritma harus berakhir setelah melakukan sejumlah langkah proses.
2. Definiteness (Kepastian): Setiap langkah algoritma harus didefinisikan dengan tepat dan tidak menimbulkan makna ganda.
3. Input (Masukan): Sebuah algoritma memiliki nol atau lebih masukan (input) yang diberikan kepada algoritma sebelum dijalankan.
4. Output (Keluaran): Setiap algoritma memberikan satu atau beberapa hasil keluaran
5. Effectiveness (Efektivitas): Langkah-langkah algoritma dikerjakan dalam waktu yang “wajar”

Sebuah algoritma dapat dibangun dari tiga buah struktur dasar, yaitu struktur runtunan (sequence), struktur pemilihan (selection) dan struktur pengulangan (repetition). Runtunan terdiri satu atau lebih instruksi yang

dikerjakan secara berurutan dengan urutannya dalam algoritma. Pemilihanya itu instruksi yang dikerjakan berdasarkan kondisi tertentu atau syarat tertentu dimana suatu kondisi atau syarat tersebut dapat bernilai benar atau salah. Intruksi akan dilaksanakan manakala kondisi atau syarat bernilai benar, dan suatu instruksi tidak akan dikerjakan apabila konsidi atau syarat tidak terpenuhi. Pengulangan merupakan pengulangan sejumlah aksi yang sama sebanyak jumlah yang ditentukan atau kondisi yang diinginkan.

Sedangkan pemrogram adalah sekumpulan instruksi yang diwujudkan dalam bentuk bahasa, kode skema, ataupun bentuk lain, yang apabila digabungkan dengan media yang dapat dibaca dengan komputer akan mampu membuat komputer bekerja untuk melakukan fungsi-fungsi khusus, termasuk persiapan dalam merancang instruksi-instruksi tersebut. Pemrograman dapat juga diartikan sebagai proses menulis, menguji, dan memelihara kode yang memberi instruksi kepada komputer untuk melakukan tugas tertentu. Dalam pemrograman, kita menggunakan berbagai bahasa pemrograman yang memiliki sintaks dan aturan tertentu untuk menciptakan perangkat lunak yang dapat menyelesaikan masalah atau memenuhi kebutuhan tertentu. Dasar dari pemrograman melibatkan penggunaan variabel untuk menyimpan data, tipe data untuk menentukan jenis data yang disimpan, dan operasi logika atau matematis untuk memanipulasi data tersebut. Dalam pemrograman berorientasi objek, konsep objek dan kelas digunakan untuk mengorganisasi kode dalam entitas yang merepresentasikan benda nyata atau konsep abstrak, yang memiliki atribut dan perilaku. Penanganan kesalahan atau exception handling adalah komponen

penting lainnya, yang memastikan bahwa program dapat menangani kesalahan yang mungkin terjadi selama eksekusi tanpa berhenti tiba-tiba. Secara keseluruhan, pemrograman adalah keterampilan yang menggabungkan logika, kreativitas, dan ketelitian untuk menciptakan solusi digital yang efektif. (Hasyim, A. H. 2021)

2. *Game*

Game secara bahasa (berasal dari bahasa Inggris) adalah permainan. Dengan bantuan teknologi, kini *game* dapat diartikan secara lebih luas lagi. Jadi, secara istilah pengertian *game* adalah permainan yang diprogram pada suatu perangkat yang dapat dijalankan secara *offline* maupun *online*, Silvianita, Helva (2019).

Adapun pengertian *game* menurut para-ahli :

a. Clark C. Abt, *Game* adalah kegiatan yang melibatkan keputusan pemain, berupaya mencapai tujuan dengan "dibatasi oleh konteks tertentu" (misalnya, dibatasi oleh peraturan). (*Serious Games, New York : Viking Press, 1970*).
b. Dawang Muchtar 2005, *Game* atau permainan adalah sesuatu yang dapat dimainkan dengan aturan tertentu sehingga ada yang menang dan ada yang kalah, biasanya dalam konteks tidak serius atau dengan tujuan *refreshing*.
c. Katie Salen and Eric Zimmerman : 2003, Sebuah permainan adalah sebuah sistem di mana pemain terlibat dalam konflik buatan, ditentukan oleh aturan, yang menghasilkan hasil yang terukur.

3. Sepeda Ontel

Sepeda onthel merupakan sebuah sepeda kuno atau orang Jawa kebanyakan menyebutnya dengan sepeda Untho atau sepeda kebo karena bentuknya yang tinggi besar. Sepeda onthel ini sebenarnya merupakan sepeda buatan Belanda. Ada 2 produsen sepeda onthel asal Belanda yang terkenal hingga saat ini, yaitu: Rijiwiel dan Oudefiets. Sedangkan merk sepeda onthel yang paling terkenal adalah Fongers. Sepeda onthel ini merupakan satu-satunya moda transportasi rakyat yang populer saat penjajahan belanda dimana saat itu sepeda onthel banyak digunakan oleh para mandor, pegawai pemerintahan, dan lainnya.

Sepeda Onthel ini mulai banyak digunakan pada jaman Hindia Belanda. Kemudian pada tahun 1970-an keberadaan sepeda onthel mulai digeser oleh sepeda jengki yang berukuran lebih kompak baik dari ukuran tinggi maupun panjangnya dan tidak dibedakan desainnya untuk pengendara pria atau wanita. Waktu itu sepeda jengki yang cukup popular adalah merek Phoenix dari China. Selanjutnya, sepeda jengki pada tahun 1980-an juga mulai tergeser oleh sepeda MTB sampai sekarang.

Sepeda onthel kemudian pada tahun 1970-an lebih banyak digunakan oleh masyarakat pedesaan disbanding perkotaan. Namun pada akirnya karena usia dan kelangkaan, sepeda onthel telah berubah menjadi barang antik dan unik. Mulailah situasi berbalik, sepeda onthel yang dulunya terbuang, sekarang pada tahun 2000-an justru diburu kembali oleh semua kalangan mulai dari pelajar, mahasiswa sampai pejabat. Orang Jawa mengatakan inilah

“wolak-waliking jaman”. Keranjingan masyarakat terhadap sepeda onthel adalah tepat bersamaan dengan berkembangnya ancaman global warming.

Kini banyak klub-klub dan komunitas sepeda kuno dari berbagai daerah di Indonesia, tersebar dari Sabang hingga Merauke yang jumlahnya ratusan komunitas, itupun hanya yang sempat terpantau dan terdaftar, belum lagi masih banyak yang tidak terdaftar atau ikut organisasi dibawah naungan KOSTI (Komunitas Sepeda Tua Indonesia), Luqman, A. H. (2018).

4. *Unity*

Aplikasi *unity* 3D merupakan sebuah *software* pengolah gambar, grafik, suara, input, dan lain-lain yang ditujukan untuk membuat suatu *game*, meskipun tidak selamanya harus untuk *game*. Contohnya adalah seperti materi pembelajaran untuk simulasi membuat SIM.

Kelebihan dari *software* ini adalah bisa membuat *game* berbasis 3D maupun 2D, dan sangat mudah digunakan.

Unity merupakan *software* yang *ber-multiplatform*. *Unity* mampu di *publish* menjadi *Standalone* (.exe), berbasis *web*, *Android*, *IoS*, *Iphone*, XBOX, dan PS3. Walau bisa *dipublish* ke berbagai *platform*, *Unity* perlu lisensi untuk dapat dipublish ke platform tertentu. Untuk saat ini *unity* dapat dan sedang di kembangkan dalam penggunaan berbasis AR (*Augment Reality*).

Unity perlu adanya lisensi, sebagai contoh ketika ingin mengaktifkan *free user*, langkah pertama adalah mendownload software nya secara gratis pada *website* milik *unity*. Setelah selasai instalasi, maka *unity* meminta untuk

terhubung dengan internet untuk aktifasinya. Lalu *unity* akan automatis *run* ke program. Untuk langkah selanjutnya *unity* tidak perlu lagi memerlukan koneksi internet saat menjalankan aplikasinya. Reno, R. (2021).

a. *Interface Unity*



Gambar 2. 1 *Interface Unity*

1) *Toolbar* menyediakan akses ke akun *unity* dan layanan *cloud unity* anda. *Toolbar* juga berisi kontrol untuk mode *play*, batalkan riwayat, pencarian persatuan, menu visibilitas lapisan; dan menu tata letak editor.
2) Jendela *hierarchy* adalah representasi teks hierarkis dari setiap *gameobject* di adegan. Setiap item dalam *scene* memiliki entri dalam hierarki, sehingga kedua jendela secara terhubung. Hirarki mengungkapkan struktur bagaimana *gameobjects* menempel satu sama lain.
3) Tampilan *game* mensimulasikan seperti apa akhir pada *game* yang dirender melalui kamera pemandangan anda. Saat anda mengklik tombol *play* maka simulasi dimulai.

4) Tampilan *scene* memungkinkan anda menavigasi dan mengedit *scene* anda secara visual. Tampilan pemandangan dapat menampilkan perspektif 3d atau 2d, tergantung pada jenis proyek yang sedang anda kerjakan.
5) *Overlay* berisi alat dasar untuk memanipulasi tampilan scene dan *gameobjects* di dalamnya. Anda juga dapat menambahkan *overlay* kustom untuk meningkatkan alur kerja anda.
6) Jendela *Inspector* memungkinkan anda untuk melihat dan mengedit semua properti *gameobject* yang saat ini dipilih. Karena berbagai jenis *gameobjects* memiliki set properti yang berbeda, tata letak dan isi inspektur jendela berubah setiap kali anda memilih *gameobject* yang berbeda.
7) Jendela *Proyek* menampilkan pustaka aset yang tersedia untuk digunakan dalam proyek anda. Saat anda mengimpor aset ke *project*, aset akan muncul di sini.
8) Status *bar* menyediakan notifikasi tentang berbagai proses *unity*, dan akses cepat ke alat dan setelan terkait.

b. Kelebihan *Unity*

Adapun kelebihan dari *unity* dibandingkan yang lainnya adalah:

1) Kemudahan pengembangan

Unity menawarkan antarmuka pengembangan yang intuitif dan mudah dipelajari, membuatnya cocok untuk pengembang pemula maupun yang berpengalaman. Fitur-fitur visual dan alur kerja yang efisien mempercepat proses pengembangan.

2) *Multiplatform*

Unity mendukung banyak platform, termasuk PC, konsol *game*, perangkat seluler, dan VR/AR. Ini memungkinkan pengembang untuk membuat aplikasi yang dapat dijalankan di berbagai perangkat, mencapai audiens yang lebih luas.

3) *Asset Store*

Unity memiliki *asset store* yang besar, di mana pengembang dapat membeli atau mengunduh sumber daya siap pakai seperti model 3D, tekstur, suara, dan *script*. Hal ini mempercepat pengembangan dengan menyediakan aset berkualitas tinggi yang dapat digunakan secara instan.

4) Komunitas dan Dukungan

Unity memiliki komunitas yang besar dan aktif, dengan forum pengguna yang ramah. Ini memungkinkan pengembang untuk berkolaborasi, berbagi pengetahuan, dan mendapatkan dukungan dari sesama pengembang.

5) Kinerja dan Grafis

Unity memiliki mesin game yang kuat dan dapat diandalkan. Dukungan *rendering* 3D yang canggih dan fitur-fitur grafis modern memungkinkan pengembang untuk menciptakan visual yang menarik dan realistis.

6) Skalabilitas

Unity dapat digunakan untuk membuat proyek yang berskala kecil maupun besar. Dengan dukungan untuk manajemen sumber daya dan pengoptimalan kinerja, *unity* dapat menangani proyek dengan kompleksitas yang tinggi.

7) Integrasi Eksternal

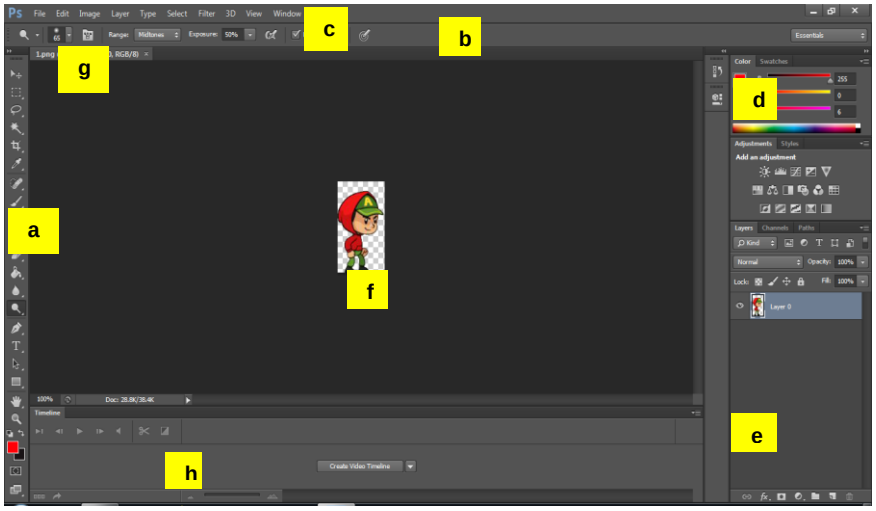
Unity mendukung integrasi dengan berbagai alat pihak ketiga dan layanan, seperti alat analitik, jejaring sosial, platform iklan, dan sistem keuangan. Ini memungkinkan pengembang untuk meningkatkan fungsionalitas aplikasi dan monetisasi.

5. *Adobe Photoshop C S6*

Photoshop CS6 merupakan aplikasi editor yang dikembangkan oleh *Adobe* dengan dukungan *Adobe Mercury Grapich Engine*, menghasilkan performa yang lebih cepat dibandingkan versi sebelumnya, (Madcoms Madiun, 2018).

a. Membuka *Photoshop CS6*

Langkah awal yang harus dilakukan untuk membuka *Photoshop CS6*, yaitu klik tombol **"***Start > All Program > Adobe CS6 > Adobe Photoshop CS6***"**. Tampilan *Photoshop CS6* saat pertama kali di jalankan, maka tampil jendela baru seperti gambar di bawah ini:



Gambar 2. 2 *Interface Photoshop CS6*

1) *Panel Tools*

Merupakan bagian yang memuat *tool-tool* untuk menggambar, menyeleksi, memodifikasi, dan memanipulasi objek foto atau gambar.

2) *Panel Control / Option Bar*

Merupakan bagian yang menampilkan daftar tombol perintah tambahan untuk memaksimalkan kinerja dari *piranti* terpilih. Tombol-tombol perintah yang ditampilkan pada bagian ini berubah-ubah menyesuaikan dengan tool yang dipilih pada bagian *panel Tools*.

3) *Menu Bar*

Merupakan bagian yang memuat menu-menu perintah yang pada dasarnya untuk menjalankan perintah-perintah manajemen file, perintah untuk mengolah layer dan dokumen,

serta memuat perintah lain untuk memberi efek filter dan mengatur tampilan *panel*.

4) *Workspace Switcher*

Merupakan bagian yang digunakan untuk mengubah tampilan lembar kerja sesuai dengan kebutuhan, seperti untuk kebutuhan *Essentials, Photography, Painting, dan Typography*.

5) *Dock Panel*

Merupakan bagian yang menampung panel-panel yang berisi serangkaian perintah dan parameter untuk mengolah dan memanipulasi objek gambar.

6) Jendela Dokumen

Merupakan lembar kerja utama yang menampilkan objek gambar atau foto yang akan dimanipulasi. Jendela dokumen ini memiliki elemen yang disebut tab dokumen yang terletak pada bagian atas dan berisi nama-nama dokumen yang aktif.

7) *Tab* Dokumen

Merupakan bagian yang menampung nama-nama dokumen yang sedang aktif.

8) Jendela *Timeline*

Merupakan bagian yang digunakan untuk membuat animasi dan mengatur jalannya animasi yang dikerjakan.

b. Fitur Terbaru dalam *Photoshop CS6*

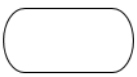
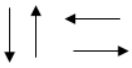
Adobe *Photoshop CS6* hadir dengan beberapa fitur yang menakjubkan, serta penyempurnaan performa dan beberapa fasilitas atau filter yang semakin pintar. Beberapa fitur terbaru yang diusung oleh *Adobe Photoshop CS6*, yaitu *Interface* baru dari *Adobe Photoshop CS6*, Penyempurnaan kinerja dari fasilitas *Content Aware, Crop tool* yang lebih presisi, *Camera RAW, Adactive Wide Angle*, *Color Range - Skin Tones, Filter Blur, Filter Oil Paint, Pick Filter Type, Color Lookup, Auto Recovery*, dan *Dash – Dot.*

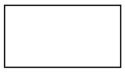
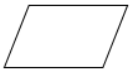
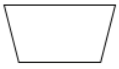
6. *Flowchart*

Flowchart adalah adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses (instruksi) dengan proses lainnya dalam suatu program, Hasyim, A. H. (2021).

Berikut ini adalah beberapa simbol yang digunakan dalam menggambar suatu *Flowchart* :

Tabel 2. 1 *Symbol* Flowchart

Simbol	Nama dan Fungsinya
	***Terminal Point Symbol / Simbol Titik****Terminal Menu*njukkan permulaan (start) atau akhir (stop) dari suatu proses.
	***Flow Direction Symbol* / Simbol Arus** adalah simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain (*connecting line*). Simbol ini juga berfungsi untuk *Menu*njukkan garis alir dari proses.

Simbol	Nama dan Fungsinya
	***Processing Symbol* / Simbol** Proses digunakan untuk *Menu*njukkan kegiatan yang dilakukan oleh komputer. Pada bidang industri (proses produksi barang), simbol ini menggambarkan kegiatan inspeksi atau yang biasa dikenal dengan simbol inspeksi
	***Decision Symbol* / Simbol** Keputusan merupakan simbol yang digunakan untuk memilih proses atau keputusan berdasarkan kondisi yang ada. Simbol ini biasanya ditemui pada *Flowchart* program.
	**I*nput-Output* / Simbol Keluar-**Masuk *Menu*njukkan proses *input-output* yang terjadi tanpa bergantung dari jenis peralatannya.
	***Predefined Process* / Simbol Proses** *Terdefinisi*merupakan simbol yang digunakan untuk *Menu*njukkan pelaksanaan suatu bagian prosedur (*sub*-proses).
	***Connector (On-page)*** Simbol ini fungsinya adalah untuk menyederhanakan hubungan antar simbol yang letaknya berjauhan atau rumit bila dihubungkan dengan garis dalam satu halaman
	***Connector (Off-page)*** Sama seperti *on-page connector*, hanya saya simbol ini digunakan untuk menghubungkan simbol dalam halaman berbeda. label dari simbol ini dapat menggunakan huruf atau angka
	***Preparation Symbol* / Simbol** Persiapan merupakan simbol yang digunakan untuk mempersiapkan penyimpanan di dalam storage.
	***Manual Input Symbol*** digunakan untuk *Menu*njukkan input data secara manual menggunakan *online* keyboard.
	***Manual Operation Symbol* / Simbol Kegiatan** Manual digunakan untuk *Menu*njukkan kegiatan/proses yang tidak dilakukan oleh komputer.

Simbol	Nama dan Fungsinya
	***Document Symbol*** Jika Anda menemukan simbol ini artinya input berasal dari dokumen dalam bentuk kertas, atau output yang perlu dicetak di atas kertas.
	***Multiple Documents*** sama seperti *document symbol* hanya saja dokumen yg digunakan lebih dari satu dalam simbol ini
	***Display Symbol*** adalah simbol yang menyatakan penggunaan peralatan output, seperti layar monitor, printer, plotter dan lain sebagainya
	***Delay Symbol*** sesuai dengan namanya digunakan untuk *Menu*njukkan proses delay (*Menu*nggu) yang perlu dilakukan. Seperti *Menu*nggu surat untuk diarsipkan

7. UML *(Unified Modelling Language)*

Unified Modelling Language (UML) adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, di mana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apa pun, serta ditulis dalam bahasa pemrograman apa pun .

Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa- bahasa berorientasi objek seperti *C++,Java, C#* atau *VB. NET*.

Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C.

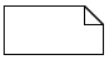
Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax*/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan.

Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya : *Grady Booch OOD(Object – Oriented Design*), Jim Rumbaugh OMT (*Object Modeling Technique*), dan Ivar Jacobson OOSE (*Object – Oriented Software Engineering*), (N, Sora. 2015).

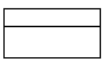
Adapun daftar simbol UML yaitu :

Tabel 2. 2 *Symbol Use Case Diagram*

No	GAMBAR	NAMA	KETERANGAN
1		*Actor*	Menypesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan *Use Case*.
2		*Dependency*	Hubungan di mana perubahan yang terjadi pada suatu elemen mandiri (*Independent*) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
3		*Generalization*	Hubungan di mana objek anak *(Descendent*) berbagai perilaku dan struktur data dari objek yang ada di atasnya objek induk (*Ancestor*).
4		*Include*	Menypesifikasikan bahwa *Use Case* sumber secara Eksplisit.

No	GAMBAR	NAMA	KETERANGAN
5		*Extend*	Menypesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan.
6		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		*System*	Menypesifikasikan paket yang menampilkan sistem secara terbatas.
8		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu *actor*
9		*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Tabel 2. 3 *Symbol Class Diagram*

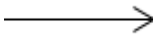
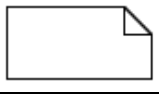
No.	GAMBAR	NAMA	KETERANGAN
1		*Generalization*	Hubungan di mana objek anak *(descendent)* berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
2		*Nary Association*	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		*Class*	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		*Collaboration*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu *actor*
5		*Realization*	Operasi yang benar-benar dilakukan oleh suatu objek.
6		*Dependency*	Hubungan di mana perubahan yang

No.	GAMBAR	NAMA	KETERANGAN
			terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya

Tabel 2. 4 *Symbol Sequence Diagram*

No	GAMBAR	NAMA	KETERANGAN
1		*LifeLine*	Objek *entity*, antarmuka yang saling berinteraksi.
No	**GAMBAR**	**NAMA**	**KETERANGAN**
2		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktivitas yang terjadi
3		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-nformasi tentang aktivitas yang terjadi

Tabel 2. 5 *Symbol State Chart Diagram*

No	GAMBAR	NAMA	KETERANGAN
1		*State*	Nilai atribut dan nilai *Link* pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
2		*Initial Pseudo State*	Bagaimana objek dibentuk atau diawali
3		*Final State*	Bagaimana objek dibentuk dan dihancurkan
4		*Transition*	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya
5		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6		*Node*	Elemen fisik yang eksis saat aplikasi dijalankan dan

No	GAMBAR	NAMA	KETERANGAN
			mencerminkan suatu sumber daya komputasi.

Tabel 2. 6 *Symbol Activity Diagram*

No.	GAMBAR	NAMA	KETERANGAN
1		*Actifity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		*Action*	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		*Initial Node*	Bagaimana objek dibentuk atau diawali.
No.	**GAMBAR**	**NAMA**	**KETERANGAN**
4		*Actifity Final Node*	Bagaimana objek dibentuk dan dihancurkan
5		*Fork Node*	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

8. *Black Box*

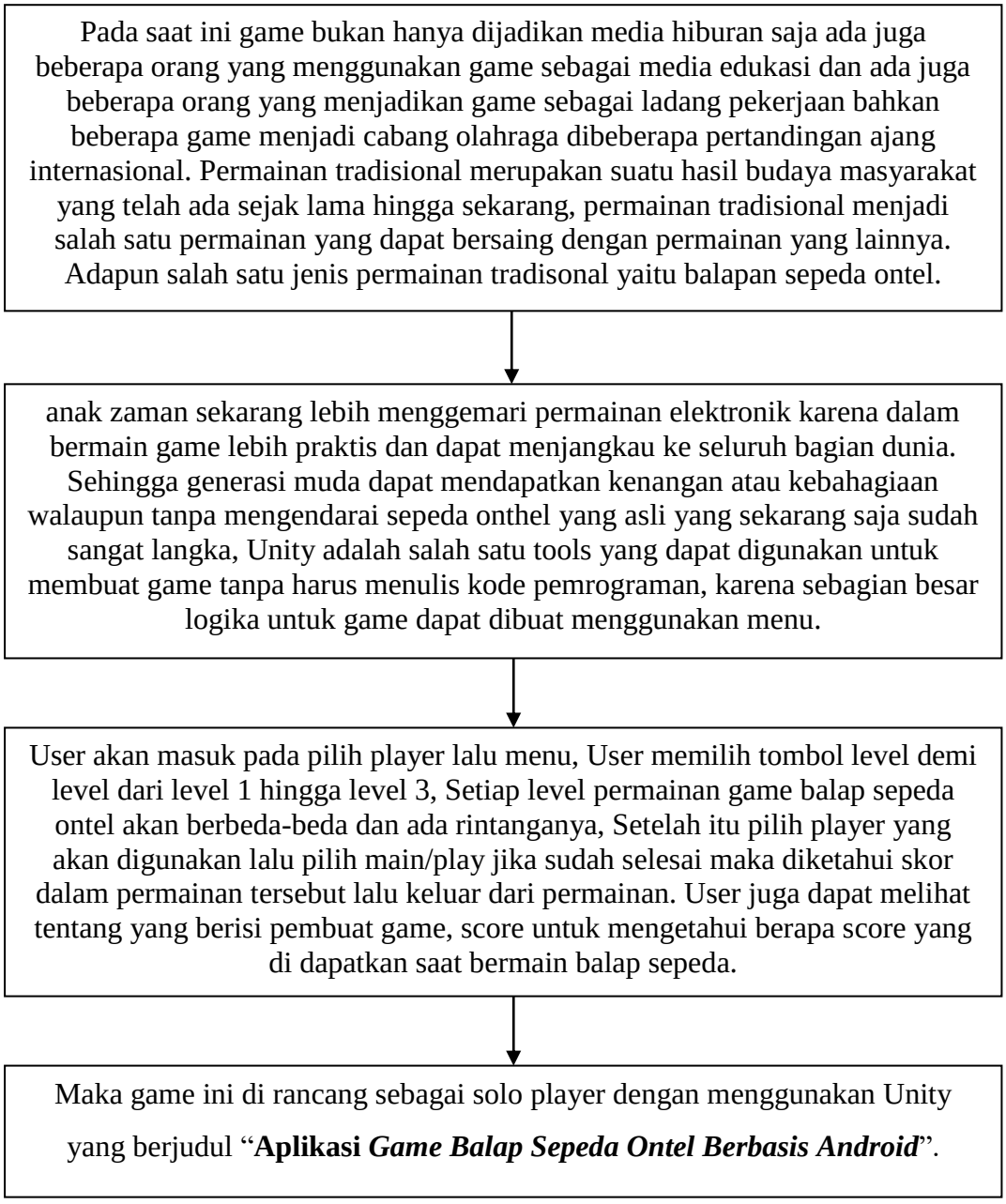
Pengujian *Black-Box* merupakan pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, penguji dapat mendefinisikan kumpulan kondisi *input* dan melakukan pengetesan pada spesifikasi fungsional program. Pengujian ini digunakan untuk melakukan *test fungsionalitas* serta untuk memeriksa validasi terhadap aplikasi yang dibangun. Berdasarkan pembahasan pada implementasi *user interface* sebelumnya akan diuji validasi sesuai dengan cara pengujian yang ada sehingga hasilnya sesuai dengan harapan. Pada pengujian *Black Box* ini fungsionalitas dari aplikasi dibagi menjadi dua bagian yaitu Fungsionalitas Menu dan Fungsionalitas Permainan.

9. *White Box*

White Box Testing adalah salah satu cara untuk menguji suatu aplikasi atau *software* dengan cara melihat modul untuk dapat meneliti dan menganalisa kode dari program yang di buat ada yang salah atau tidak. Secara umum, metode *white box testing* basis *path* didasarkan pada jalur dan struktur koneksi dari cabang tiap sistem beserta pengodean dan fungsi logikanya. Tujuan dari basis *path* adalah untuk menentukan jumlah jalur independen, sehingga jumlah kasus uji yang diperlukan dapat didefinisikan secara eksplisit untuk memaksimalkan cakupan pengujian.

10. Kerangka Pikir

Untuk lebih memperjelas kerangka pikir maka digambarkan dalam bentuk diagram sebagai berikut :



Gambar 2. 3 Kerangka Pikir

B. Kajian hasil penelitian terdahulu

1. Prasetyo, R. M. M. (2021). “Rancang Dan Bangun Game Edukasi Anak-Anak Berbasis Android Dengan Unity Menggunakan Metode Game Development Life Cycle”. Salah satu pemanfaatan teknologi dalam pembelajaran adalah dengan memanfaatkan fungsi *android*. Pemanfaatan *android* sebagai media belajar sangat efektif dan efisien dimana dapat membuat mereka berfikir bahwa belajar adalah suatu kegiatan yang menyenangkan karena mereka dapat bermain sekaligus belajar sehingga mereka tidak mudah bosan dan cepat memahami apa yang mereka pelajari. Dari hal inilah penulis ingin merancang dan membuat Game edukasi untuk anak- anak usia dini (3-8 tahun) yang bisa dipelajari kapanpun dan dimanapun dengan *smartphone* yang dilengkapin visualisasi yang menarik dengan memanfaatkan aplikasi *unity*.
2. Antara, I. G. E. (2019). “Pengembangan Game Tradisional Megala-gala Berbasis Android”. Sangat penting dikembangkan suatu game yang memadukan antara teknologi dan permainan tradisional dengan berlandaskan pada *lenses theory*. Salah satu permainan tradisional yang akan dikembangkan adalah megala-gala dengan berbasis sistem *android*. Aplikasi *mobile* yang dibuat dapat menjadi alternatif dan dapat meningkatkan pengetahuan tentang permainan tradisional yang mudah dipahami dan menarik. Diharapkan dari pengembangan game tradisional “megala-gala” berbasis android ini dapat menjadi sarana alternatif pengenalan permainan

tradisional serta nilai-nilai yang terkandung dalam permainan tradisional "megala-gala".

3. Saputra, S. Y. (2019). "Game Multiplayer "*Mini Car Circuit*" Berbasis *Android"*. pada penelitian ini akan diciptakan sebuah game balap *multiplayer* berbasis *android* dengan judul "*Mini Car Circuit*". *Game* berbasis *android* ini memiliki kelebihan yaitu *free* dan pemain dapat membuat *room* atau memasuki *room* yang sudah ada, di dalam *room* para pemain dapat memilih jenis mobil masing-masing dan memilih arena bermain. Game *multiplayer* ini menggunakan *accelerometer* untuk mengontrol pergerakan mobil untuk kekanan dan kekiri saat bermain dan *game* ini dibuat menggunakan *Unity* 3D. Adapun tujuan dari pembuatan *game* ini adalah Membangun aplikasi *game multiplayer* pada *android* yang mudah dimainkan, menantang dan menarik dan juga membangun aplikasi *game multiplayer* sebagai sarana hiburan oleh pengguna.

BAB III

METODOLOGI PENELITIAN

A. Jenis Penelitian

Jenis penelitian ini merupakan penelitian eksperimental yang bersifat aplikatif sehingga dari ruang lingkup masalah dapat dilakukan dengan metode studi pustaka *(library research)*, metode pengumpulan data lapangan *(field research).*

B. Waktu Penelitian

Waktu yang dibutuhkan dalam palaksanaan penelitian ini adalah ± 2 (dua) bulan di tahun 2023.

C. Alat dan Bahan Penelitian

Dalam melakukan penelitian, maka diperlukan alat dan bahan penelitian yang mendukung kegiatan penelitian tersebut. Alat dan bahan yang diperlukan antara lain:

1. Alat yang digunakan selama proses penelitian yaitu:

   Laptop ACER Aspire dengan spesifikasi :

Tabel 3. 1 Spesifikasi Laptop

No.	Nama	Spesifikasi
1	Processor	Intel(R) Celeron(R) CPU N4000
2	RAM	4 GB
3	Harddisk	1000 GB
4	Monitor	14 inci
5	Printer	Epson L350 Series

2. *Smartphone* yang di gunakan dalam pembuatan aplikasi yaitu:

Tabel 3. 2 *Smartphone* yang digunakan

No.	**Nama**	**Spesifikasi**
1	System Model	Xiaomi
2	Chipset	Qualcome MSM8197
3	RAM	4 GB
4	Storage	128 GB
5	OS	Android 9

3. *Software* yang digunakan dalam pembuatan aplikasi yaitu:

Tabel 3. 3 *Software* yang digunakan

No.	**Nama**	**Keterangan**
1	Sistem Operasi	windows 10 ultimate 64 bit
2	Aplikasi Program	*Unity*.

D. Prosedur standar penelitian

1. Persiapan Penelitian

Tahap persiapan adalah tahap yang dilakukan sebelum melakukan penelitian. Pada tahapan ini dimulai dengan mengkaji permasalahan yang telah ada kemudian melakukan studi literatur mengenai permasalahan yang sedang diteliti. Adapun permasalah yang penulis kaji yaitu kurangnya minat generasi muda pada budaya-budaya tradisional khususnya sepeda ontel.

2. Studi Literatur

Pada tahapan ini ada empat langkah yang harus dipenuhi guna untuk mencapai hasil maksimal dalam penelitian yaitu tahapan pengumpulan data dengan menggunakan beberapa metode pengumpulan data yang telah

dijelaskan pada poin di atas, pengolahan data, analisis data dan selanjutnya penafsiran hasil analisis.

Setelah kegiatan tersebut, proses selanjutnya yaitu dengan melakukan tugas lapangan dalam rangka mengumpulkan data, untuk kemudian akan diproses. Proses yang dimaksud meliputi penyuntingan, penerapan masalah dalam aplikasi program, serta analisis sebagai penarikan kesimpulan hasil akhir.

Seperti penjelasan yang diatas maka permasalahan yang sedang dikaji yaitu kurangnya minat generasi muda pada budaya tradisional khususnya sepeda ontel adapun penyelesaian dari permasalahan ini yaitu dengan membuat game balapan sepeda ontel berbasis android guna meningkatkan minat generasi muda tentang budaya tradisional khususnya sepeda ontel.

3. Perancangan

Peneliti kemudian merancang aplikasi yang ingin dibuat berdasarkan alternatif pemecahan masalah. Pada tahap ini peneliti membuat game balapan sepeda ontel berbasis android dengan menggunakan aplikasi unity dalam pembuatannya.

4. Pengujian

Setelah melakukan perancangan, peneliti kemudian menguji hasil perancangan yang telah dibuat. Jika hasil perancangan terdapat kekurangan atau kelemahan maka kembali ke tahap analisis. Pada tahap ini peneliti melakukan percobaan kepada game balapan sepeda ontel berbasis android yang telah dibuat jika dirasa sudah bagus maka lanjut ketahap selanjutnya.

5. Implementasi

   Setelah pada perancangan tidak terdapat kekurangan maka aplikasi game balapan sepeda ontel berbasis android siap untuk di gunakan oleh *user*.

6. Tahap Penyelesaian

   Tahap penyelesaian merupakan tahap akhir yang dilakukan pada penelitian yang dilakukan. Pada tahap ini penyusunan laporan penelitian disusun.

E. Teknik Pengumpulan Data

1. Secara tidak langsung (Studi Literatur)

   Metode tidak langsung ini maksudnya ialah mengumpulkan data-data maupun informasi yang terkait seperti mempelajari buku-buku *Unity* dan Sepeda ontel atau artikel terkait yang berasal dari media internet.

2. Secara Langsung

   Metode secara langsung yaitu mengumpulkan data-data atau informasi yang terkait dengan perancangan program aplikasi.

F. Metode Pengujian

Metode pengujian yang penulis gunakan yaitu :

1. *White box*

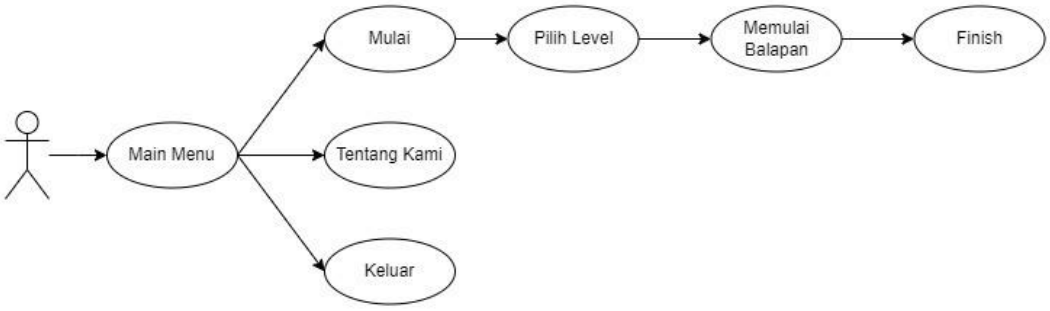
   White box testing adalah pengujian yang didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur kontrol dari desain program secara procedural untuk membagi pengujian ke dalam beberapa kasus pengujian. Secara sekilas dapat diambil kesimpulan *white box testing* merupakan petunjuk untuk mendapatkan program yang benar secara 100%.

2. *Black box*

Black box testing adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Jadi dianalogikan seperti kita melihat suatu kotak hitam, kita hanya bisa melihat penampilan luarnya saja, tanpa tahu ada apa dibalik bungkus hitamnya. Sama seperti pengujian *black box*, mengevaluasi hanya dari tampilan luarnya (*interface*-nya), fungsionalitasnya. tanpa mengetahui apa sesungguhnya yang terjadi dalam proses detailnya.

G. Desain Sistem

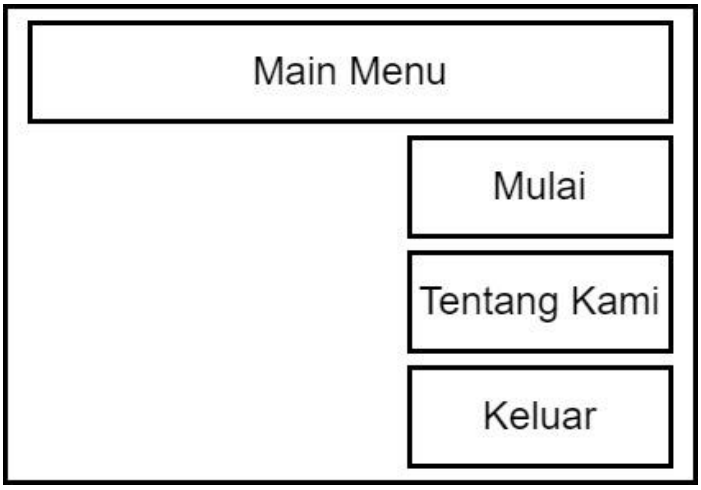
1. Desain sistem yang diusulkan



Gambar 3. 1 *Use case* sistem yang diusulkan

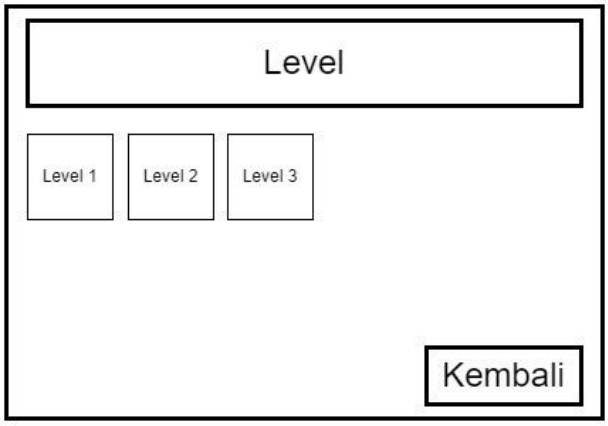
User akan masuk ke tampilan main menu, User dapat memilih tombol mulai untuk memainkan game dan memilih level 1 sampai 2, User dapat memilih Tentang kami untuk melihat biodata pembuat, level demi level dari level 1 hingga level 2, Setiap level permainan game balap sepeda ontel akan berbeda-beda dan ada rintanganya.

H. Interface



Gambar 3. 2 Tampilan menu utama

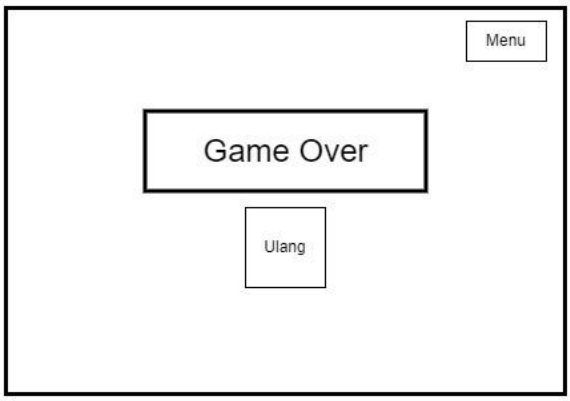
Tampilan main menu user akan disajikan tampilan dari menu utama game balap sepeda ontel, dalam menu utama tersebut terdapat tombol mulai, tentang kami dan keluar. Tombol tersebut memiliki masing – masing fungsi, tombol mulai untuk menuju pilih level, tombol tentang kami untuk menampilkan biodata pembuat, tombol keluar untuk keluar dari aplikasi.



Gambar 3. 3 Tampilan pilih *level*

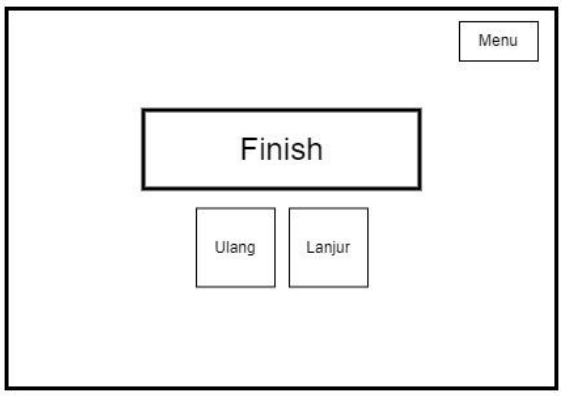
Pada tampilan menu game ini apabila user memilih level game maka sistem

akan menuju untuk memilih level sesuai yang user inginkan.



Gambar 3. 4 Tampilan *game over*

Tampilan ini adalah ketika player tidak berhasil melewati rintangan, tampilan game over ini terdiri dari tombil restar untuk memulai ulang pemainan, dan tombol menu untuk kembali ke munu utama.



Gambar 3. 5 Tampilan *finish*

Tampilan ini akan keluar ketika user mencapai finis, pada tampilan ini terdapat tombol menu dan memilih level selanjutnya.

BAB IV

HASIL DAN PEMBAHASAN

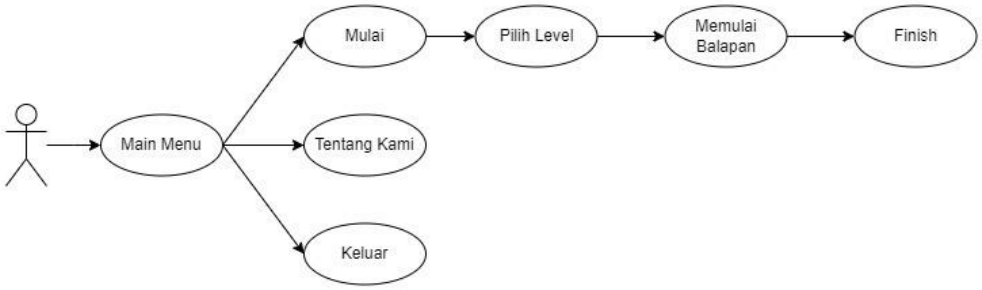
A. Analisis Aliran Data UML

Analisis aliran data bertujuan mengetahui alur proses game. Dalam analisis sistem ini, penulis menggunakan pengembangan orientasi objek sehingga menggunakan *Use Case Diagram, Activity Diagram, Sequence Diagram* dan *Class Diagram.*

1. Use Case Diagram

Use Case Diagram berfungsi untuk menjalankan manfaat sistem jika dilihat menurut pandangan orang yang berada diluar sistem.

a. *Use Case Diagram User*.



Gambar 4. 1 *Use Case Diagram User*

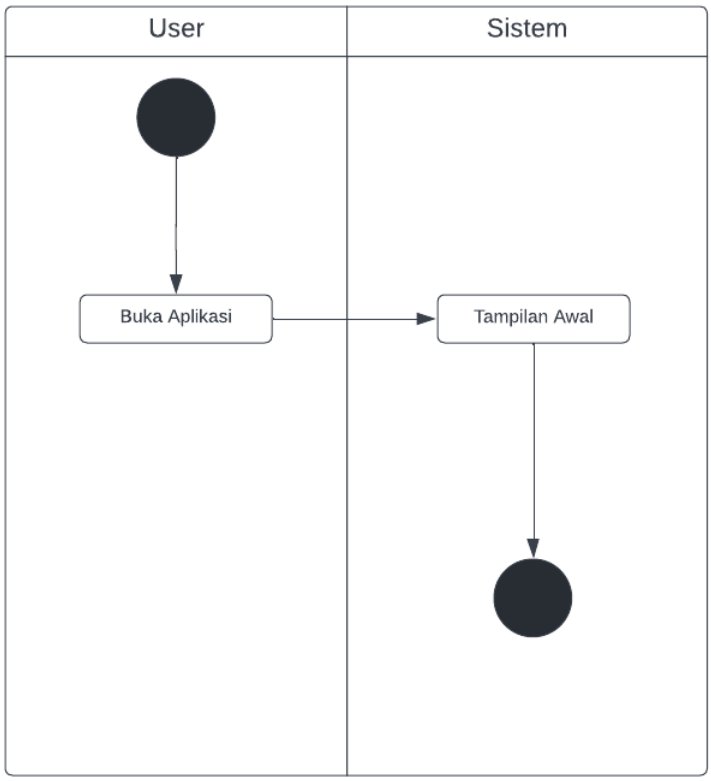
Nama *Use Case*	**Deskripsi *Use Case***
Main Menu	Merupakan tampilan awal dari *game* yaitu mulai, tentang kami, dan keluar.
Mulai	Merupakan tombol untuk menuju layout pilih level.

Nama *Use Case*	Deskripsi *Use Case*
Tentang Kami	Merupakan tampilan biodata pembuat.
Pilih Level	Merupakan layout pemilihan level untuk menuju layout sesuai level agar dapat memainkan *game*.
Mumulai Balapan	Merupakan layout untuk memainkan *game* sesuai level yang dipilih.
Finis	Merupakan layout yang muncul pada saat level selesai
Keluar	Merupakan pop up untuk keluar *game*

2. Activity Diagram

Activity diagram adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, mengspesifikasikan dan membangun perangkat lunak. *Activity* diagram ini menjelaskan tentang aktivitas-aktivitas yang terjadi dalam sebuah aliran proses pada Aplikasi *Game* Balap Sepeda Ontel Berbasis Android.

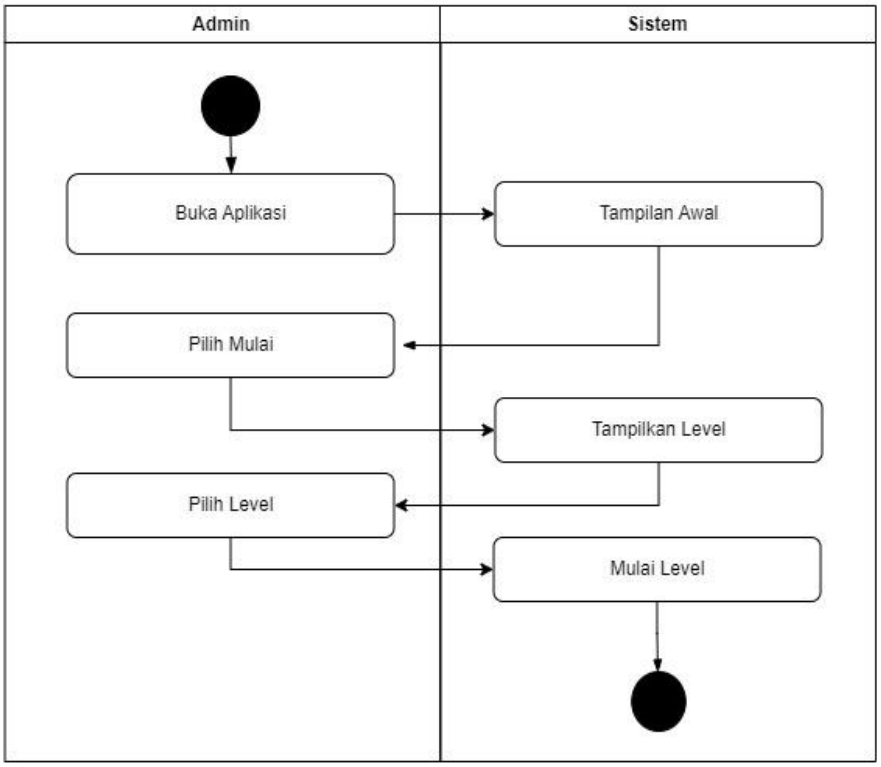
a. *Actifity diagram* Tampilan awal



Gambar 4. 2 *Activity diagram* Tampilan awal

Pada gambar 4.2 menjelaskan pada saat user membuka aplikasi. Pertama yang dilakukan user adalah membuka aplikasi kemudian sistem akan menampilkan tampilan awal.

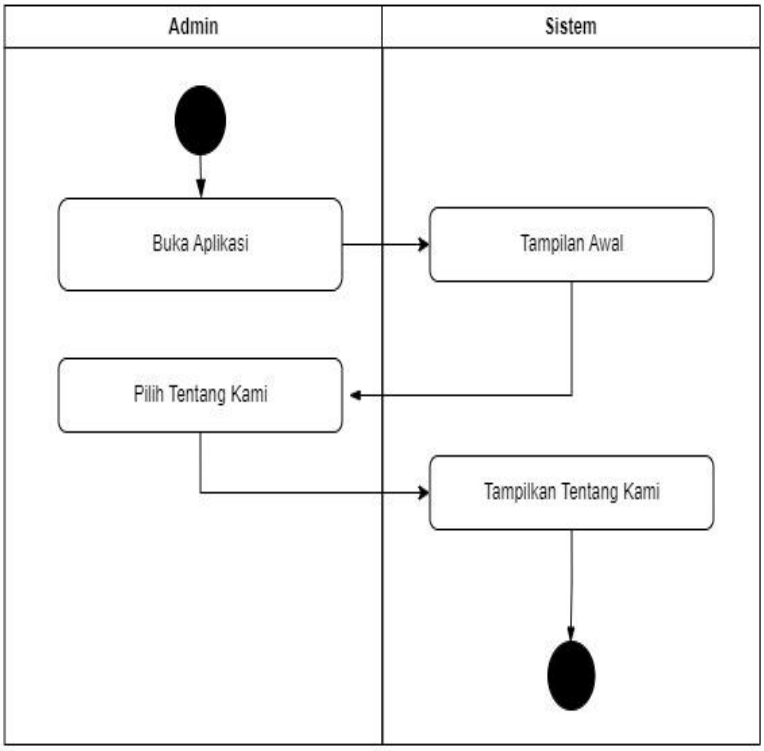
b. *Actifity diagram* Plih level



Gambar 4. 3 *Activity diagram* Pilih level

Pada gambar 4.3 menjelaskan bagaimana proses memilih level. Langkah yang dilakukan pertama yaitu user membuka aplikasi, selanjutnya system menampilkan tampilan awal. Lalu user memilih tombol pilih mulai lalu system menampilkan tampilan level lalu untuk pilih level maka level siap dimainkan.

c. *Actifity diagram* tentang kami



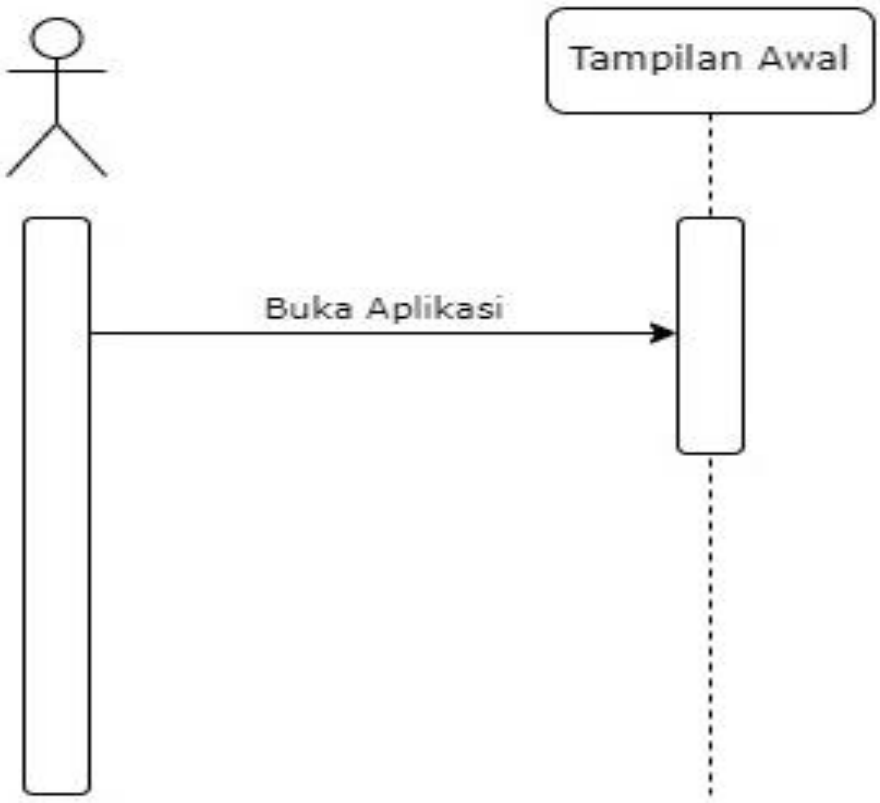
Gambar 4. 4 *Activity diagram* Tentang Kami

Pada gambar 4.4 merupakan proses untuk melihat tentang kami. Dimana jika setelah sistem menampilkan tampilan Tentang kami, selanjutnya muncul tampilan biodata pembuat.

3. Sequence Diagram

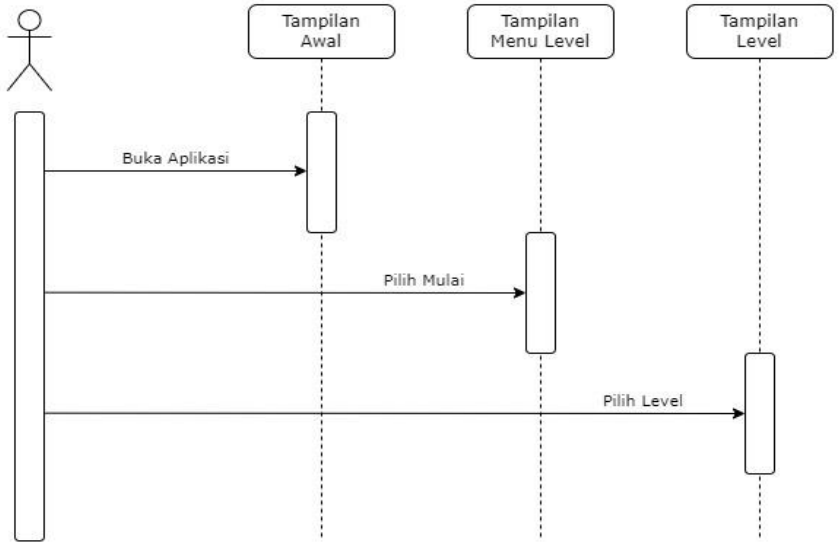
Sequence Diagram merupakan salah satu diagram Interaction yang menjelaskan bagaimana suatu operasi itu dilakukan message (pesan) apa yang dikirim dan kapan pelaksanaannya.

a. *Sequence diagram* Tampilan Awal



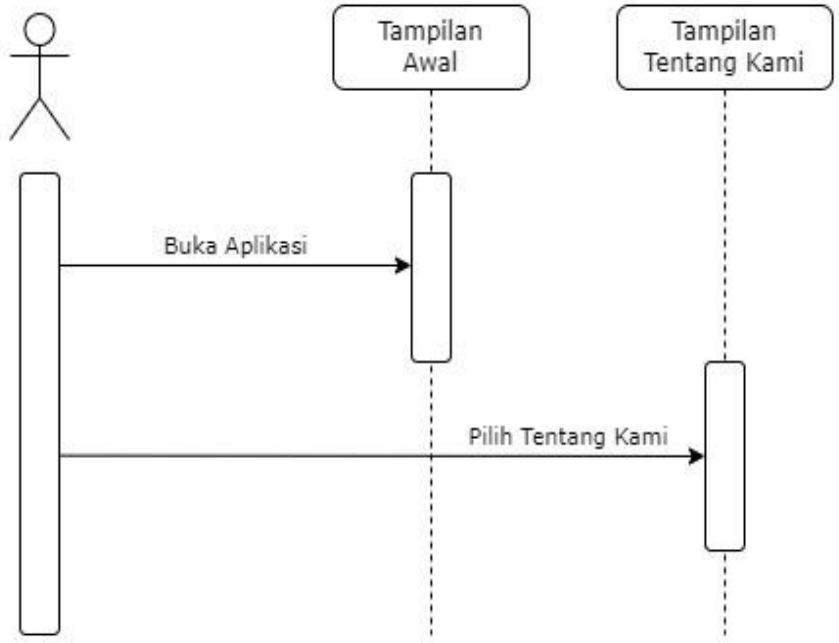
Gambar 4. 5 Sequence diagram Tampilan Awal

b. *Sequence diagram* Pilih Level



Gambar 4. 6 *Sequence diagram* Pilih Level

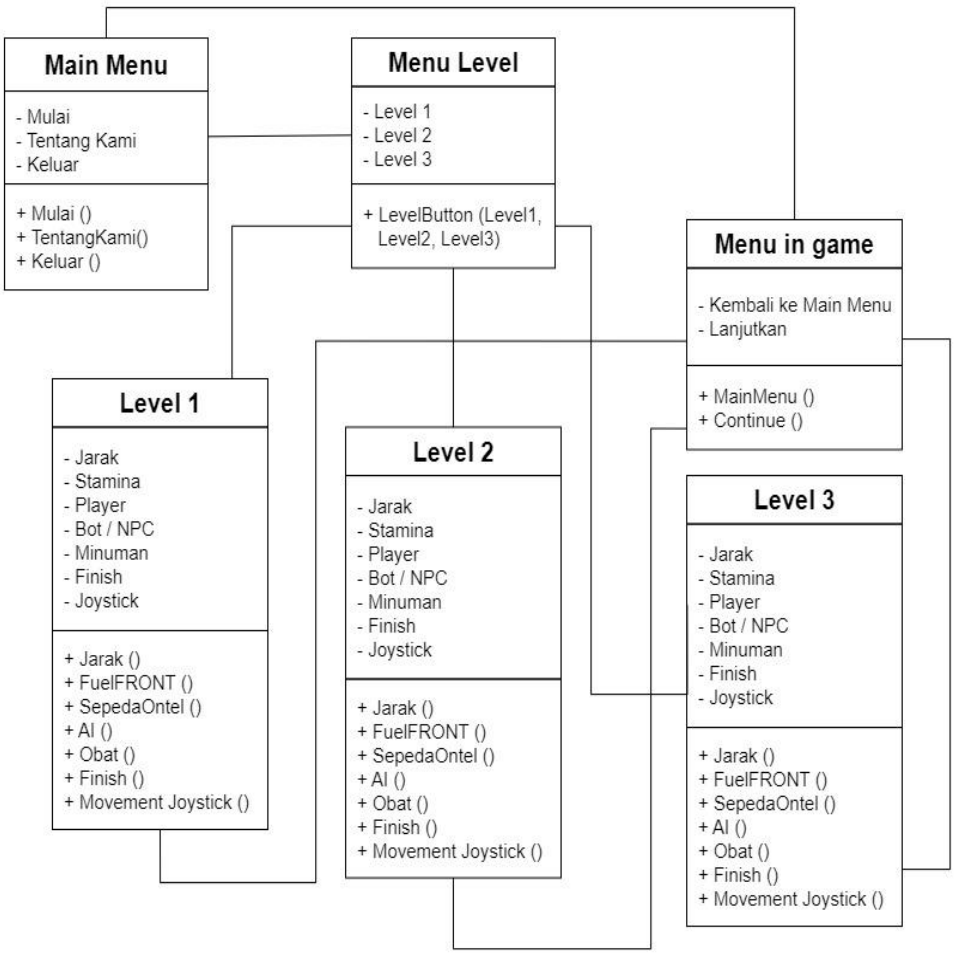
c. *Sequence diagram* Tentang Kami



Gambar 4. 7 Sequence diagram Tentang Kami

4. Class Diagram

Class diagram adalah jenis diagram struktur statis dalam UML yang menggambarkan struktur sistem dengan menunjukkan sistem class, atributnya, metode, dan hubungan antar objek.



Gambar 4. 8 Struktur *Class Diagram*

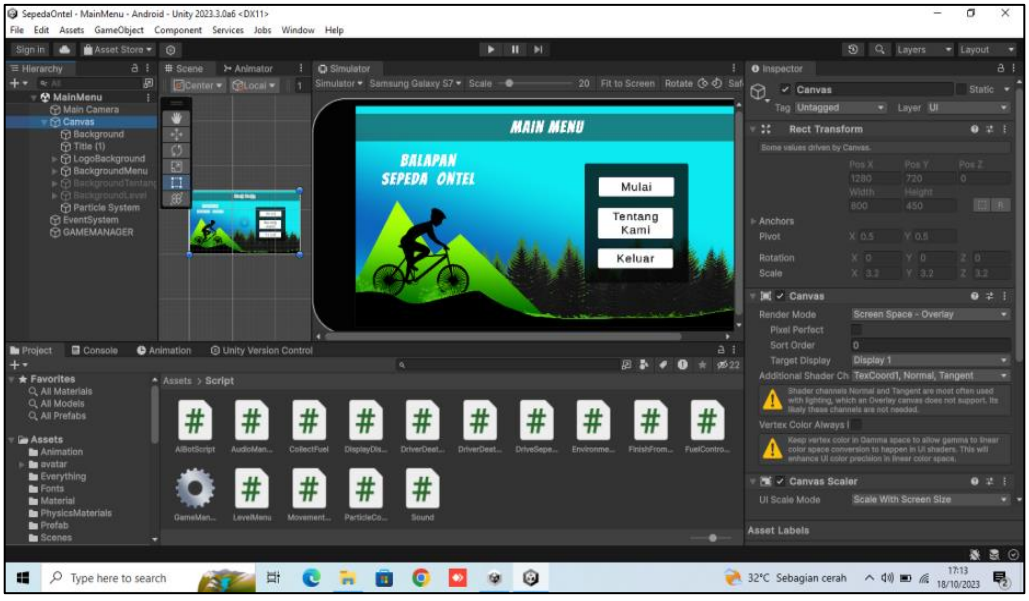
B. Detail Sistem

1. Tampilan awal

Merupakan halaman awal bagi user untuk memulai permainan pada halaman ini merupakan halaman awal pada saat membuka game.



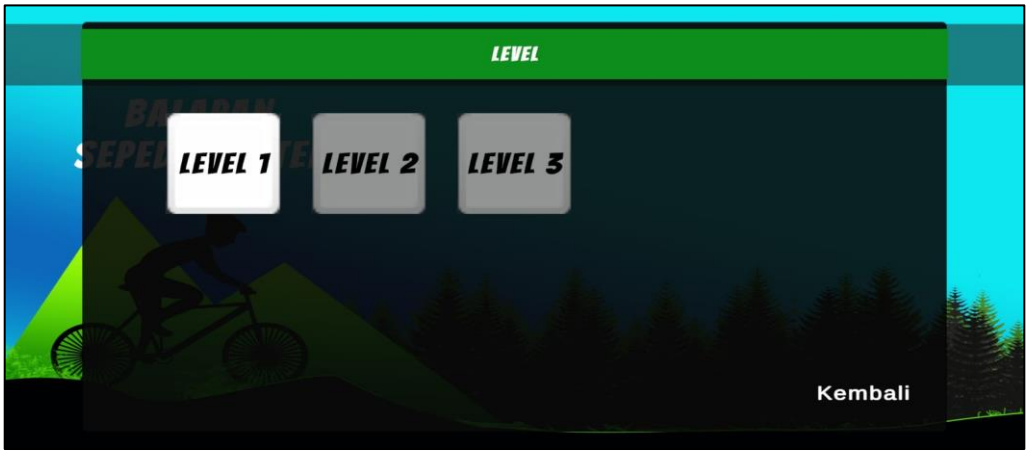
Gambar 4. 9 *Tampilan Awal*



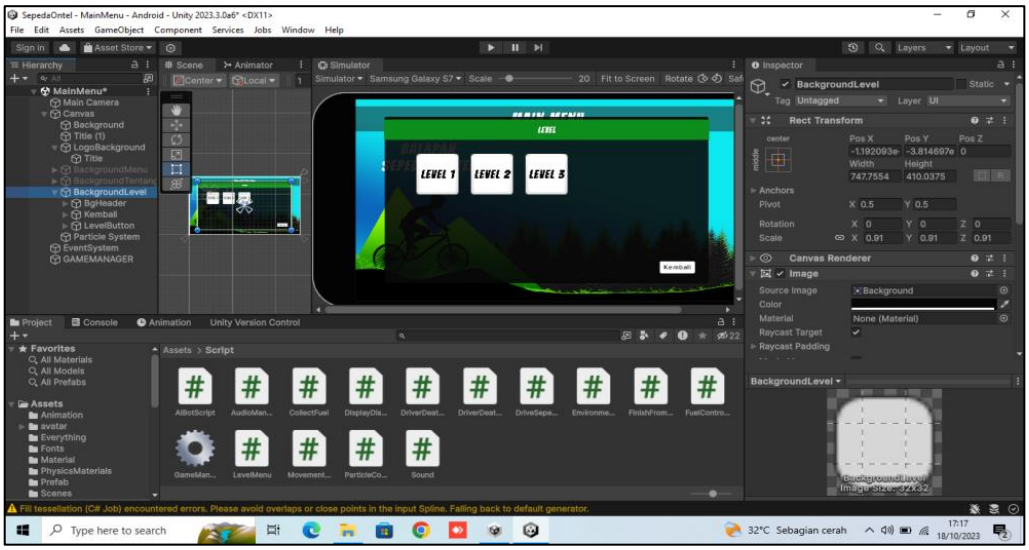
Gambar 4. 10 *Tampilan Awal dalam unity*

2. Halaman Tombol mulai

Merupakan halaman yang berfungsi untuk menampilkan Level-level yang ada dalam game.



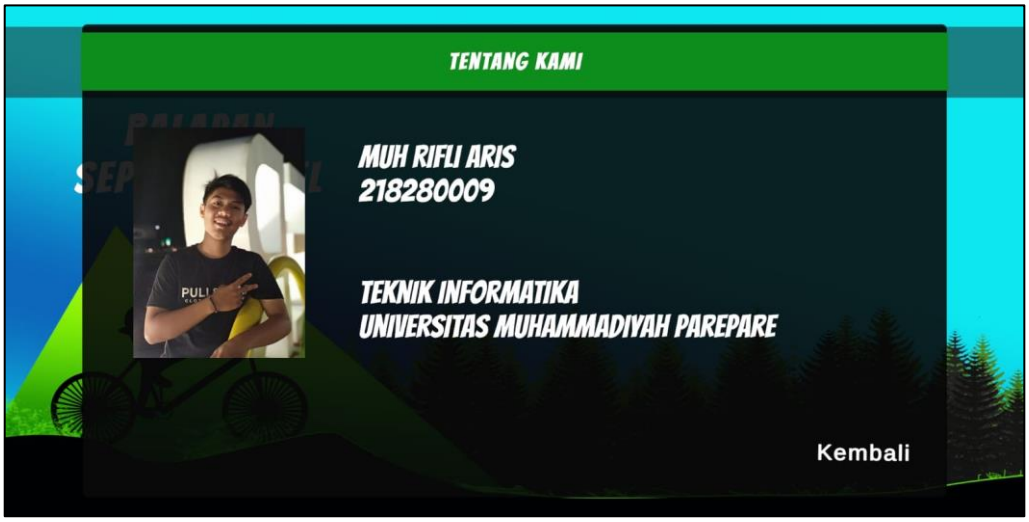
Gambar 4. 11 *Halaman Level*



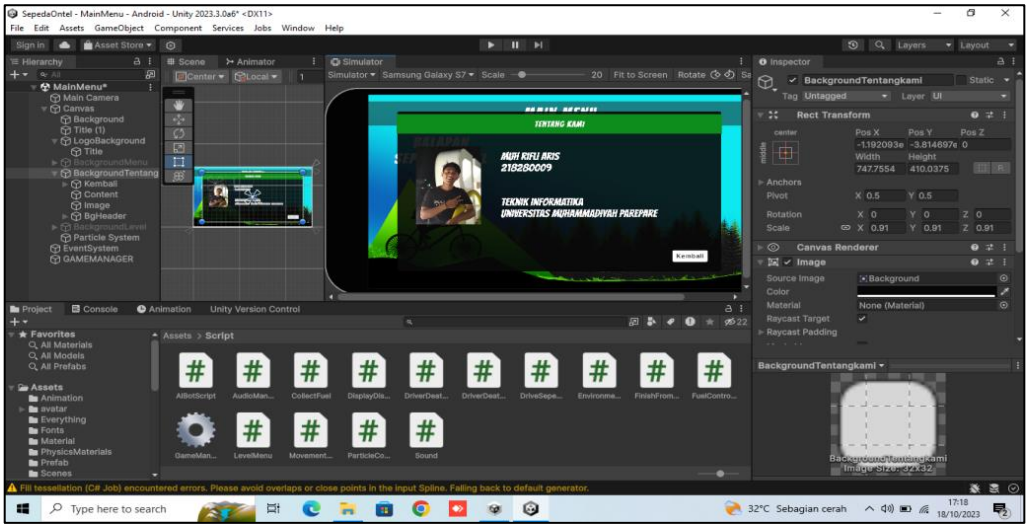
Gambar 4. 12 *Tampilan level dalam unity*

3. Halaman Tentang kami

Merupakan halaman yang berfungsi untuk menampilkan halaman biodata pembuat.



Gambar 4. 13 *Halaman Tentang Kami*



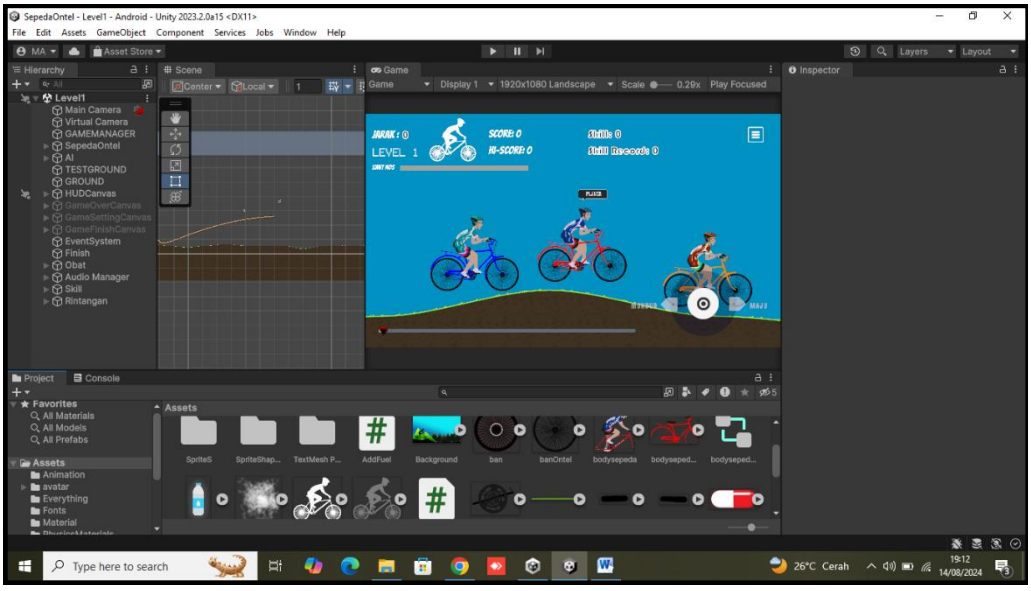
Gambar 4. 14 *Tampilan Tentang kami dalam unity*

4. Tampilan Dalam Game

Merupakan halaman dalam game dan menjadi halaman dimana pengguna memainkan game balapan sepeda ontel.



Gambar 4. 15 *Halaman Dalam Game*



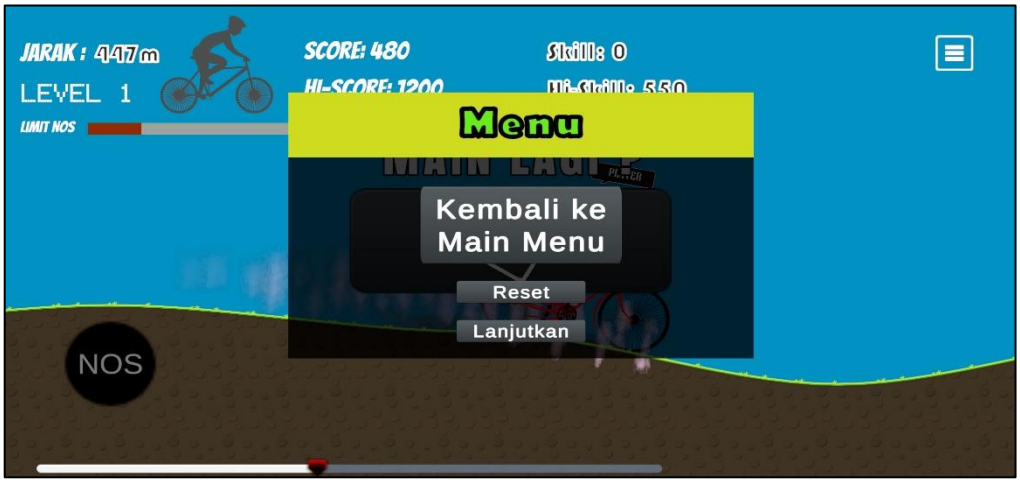
Gambar 4. 16 *Halaman Dalam Game dalam unity*

Seperti yang dapat dilihat di gambar diatas pada halaman ini terdapat beberapa codingan yang berguna untuk menjalankan fitur-fitur di game antara lain codingan dalam pergerakan Non-Playable Character (NPC) yang mejadi

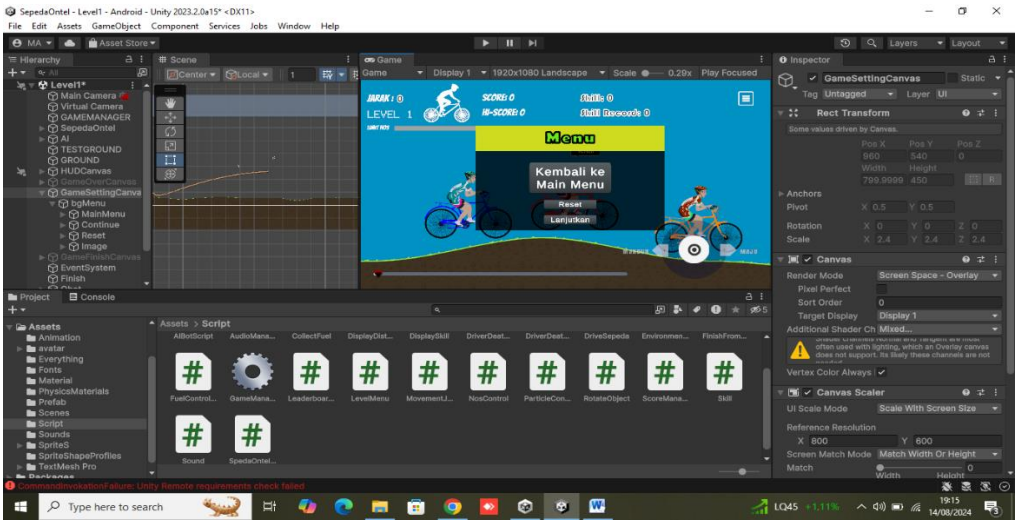
saingan pada game ini, codingan dalam pergerakan karakter yang dimainkan, codingan stamina dan memiliki skill "*Nitrous Oxide System*" (NOS) sebagai tantangan yang ada dalam game ini dan masih banyak lagi.

5. Halaman Menu dalam Game

Merupakan halaman yang berfungsi menampilkan menu dalam game yang berguna sebagai fitur yang dapat digunakan untuk kembali pada halaman main menu atau dapat juga digunakan sebagai fitur *pause* (jeda) *game* sementara jika sewaktu-waktu pengguna memerlukannya.



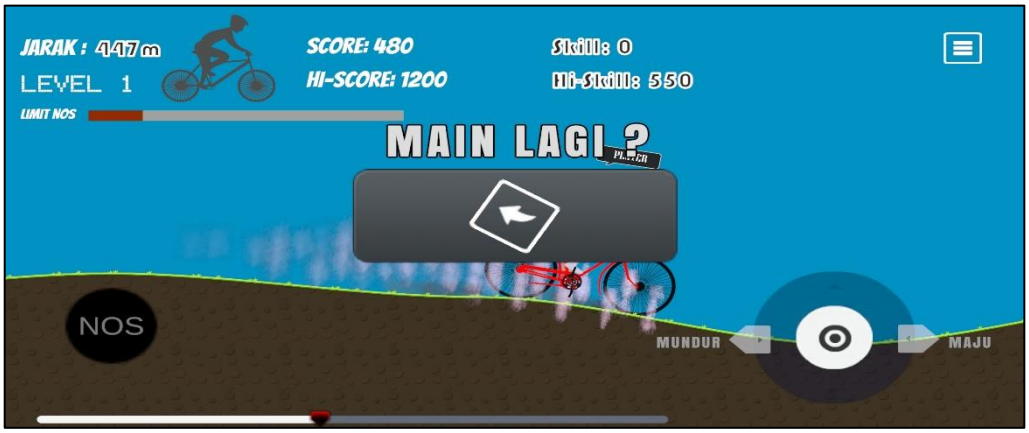
Gambar 4. 17 *Halaman Menu Dalam Game*



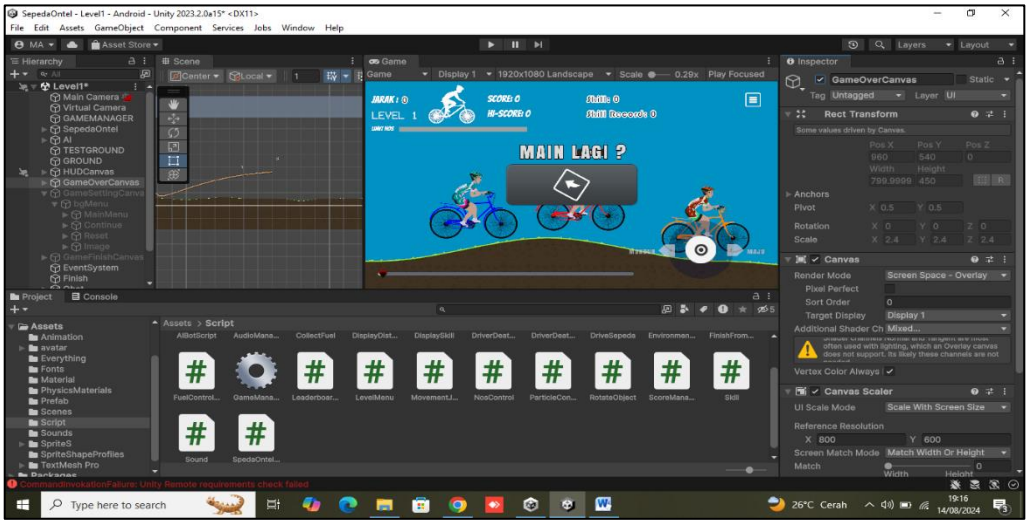
Gambar 4. 18 *Halaman Menu Dalam Game dalam unity*

6. Halaman Tampilan Game Over

Merupakan halaman yang tampil ketika pemain gagal menyelesaikan level yang dimainkan dalam game, Adapun hal-hal yang dapat menggagalkan pemain dalam menyelesaikan game yaitu kehabisan stamina atau kepala karakter yang dimainkan menyentuh tanah.



Gambar 4. 19 *Halaman Game Over*

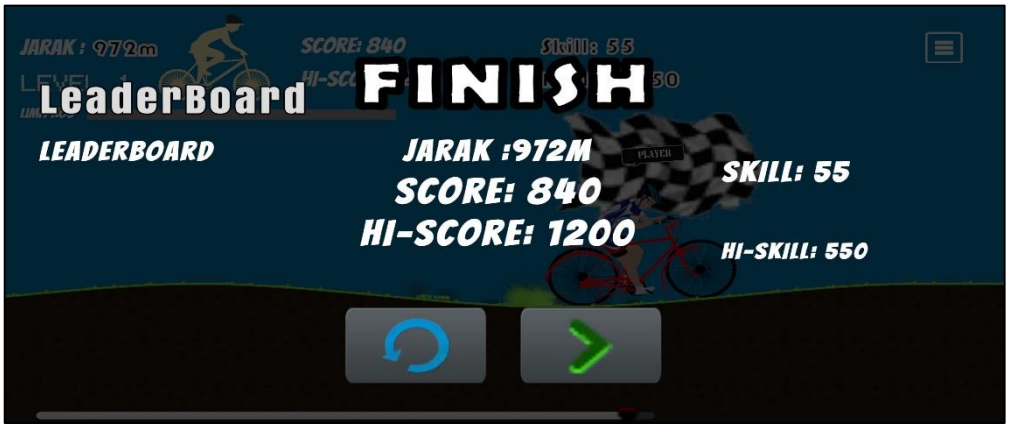


Gambar 4. 20 *Halaman Game Over Dalam unity*

Seperti yang dapat dilihat di gambar diatas pada halaman ini terdapat codingan yang berguna untuk menjalankan syarat yang dapat memunculkan halaman game over ini.

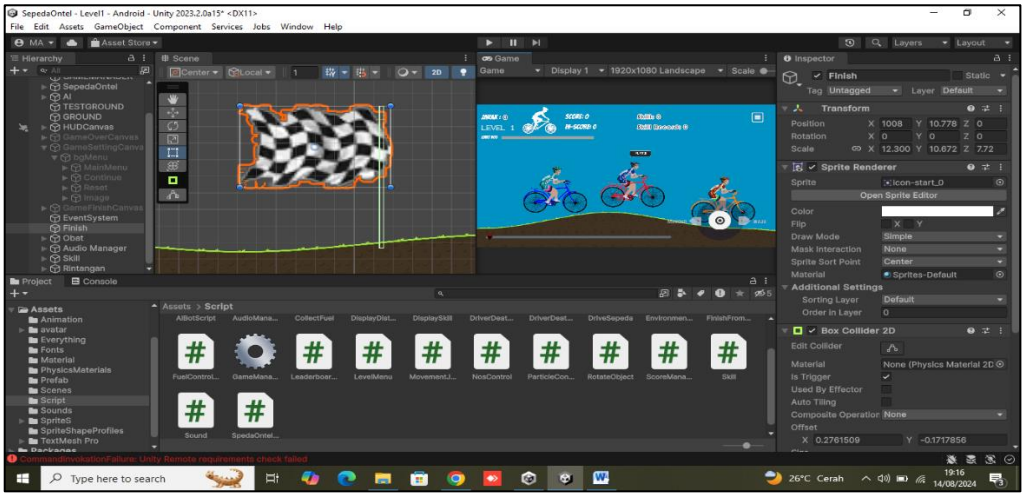
7. Halaman Tampilan Finish

Merupakan halaman yang tampil ketika pemain berhasil menyelesaikan level yang dimainkan dalam game, Adapun syarat yang harus dilakukan pemain dalam menyelesaikan game yaitu mencapai garis finish tanpa kehabisan stamina dan kepala karakter yang dimainkan tidak menyentuh tanah.



Gambar 4. 21 *Tampilan Finish*

Seperti yang dapat dilihat di gambar dibawah pada halaman ini terdapat codingan yang berguna untuk menjalankan syarat yang dapat memunculkan halaman finish ini.



Gambar 4. 22 *Tampilan Finish dalam unity*

C. Pengujian Sistem

Pengujian sistem pada penelitian ini menggunakan metode Black Box Testing dan juga White Box Testing. Berikut di bawah ini hasil pengujian menggunakan metode tersebut.

1. Black Box

   Pengujian sistem dilakukan dengan cara pengujian black box. Black box adalah metode pengujian perangkat lunak yang menguji fungsionalitas aplikasi tanpa mengintip ke dalam struktur atau cara kerja internalnya. Metode pengujian ini dapat diterapkan secara virtual ke setiap tingkat pengujian perangkat lunak: unit, integrasi, sistem, dan penerimaan.

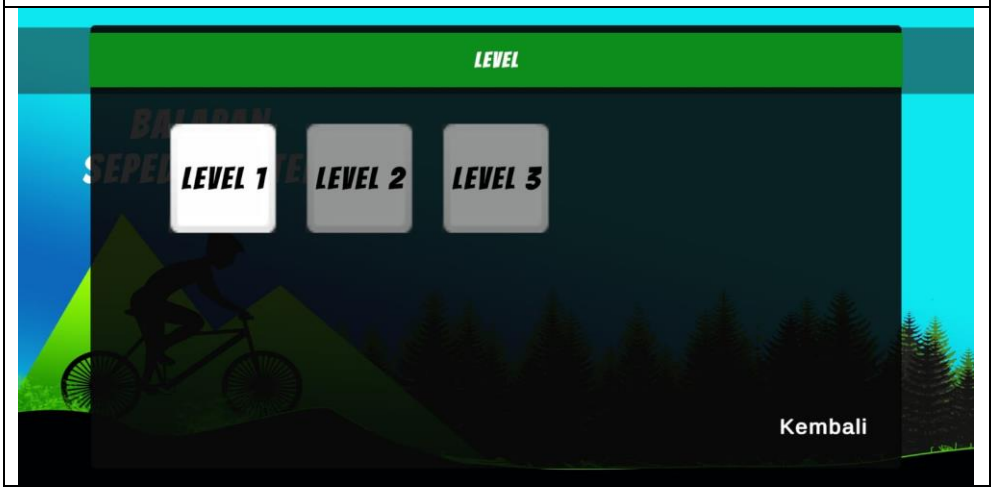
   a. Black Box Tampilan awal

Tabel 4. 1 *Black Box Tampilan awal*

Tes Faktor	Hasil	Keterangan
Menu utama atau tampilan awal game	✓	Informasi bahwa tampilan menu utama
Screenshot		
		

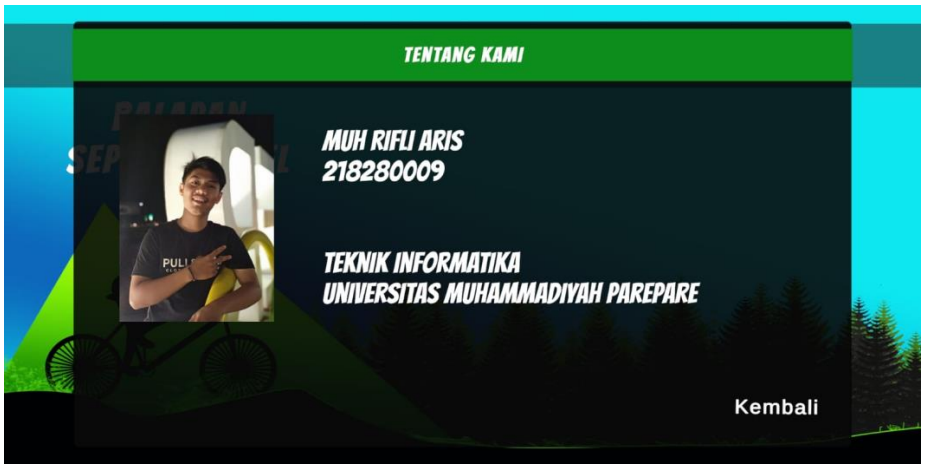
b. Black Box Tombol mulai

Tabel 4. 2 *Black Box Tombol mulai*

Tes Faktor	Hasil	Keterangan
Tombol mulai apabila telah ditekan	✓	Informasi tampilan layout pilih level
Screenshot		
		

c. Black Box Tentang Kami

Tabel 4. 3 *Black Box Pilih menu*

Tes Faktor	Hasil	Keterangan
Gambar menu yang menuju tentang kami apabila ditekan	✓	Informasi tampilan biodata pembuat
Screenshot		
		

d. Black Box Dalam Game

Tabel 4. 4 *Black Box Dalam Game*

Tes Faktor	Hasil	Keterangan
Memilih level	✓	Proses pemilihan level
Screenshot		
		

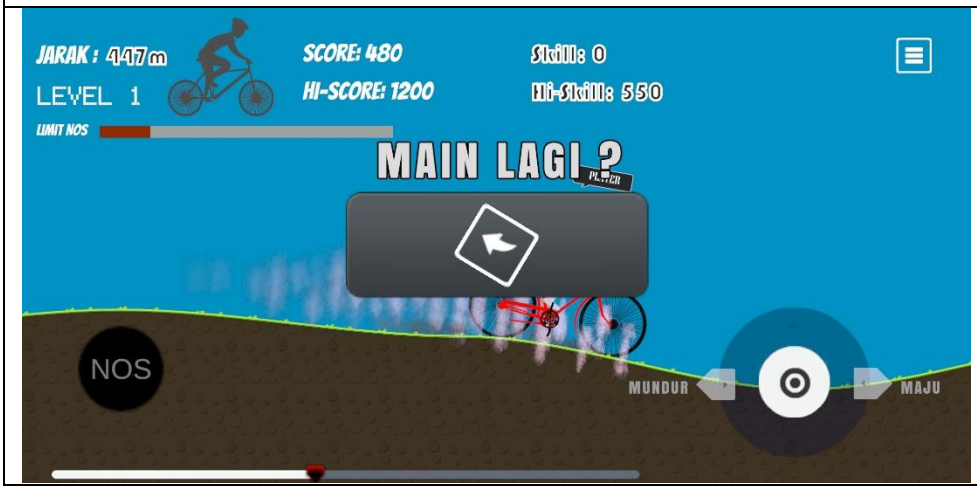
e. Black Box Menu Dalam Game

Tabel 4. 5 *Black Box Menu Dalam Game*

Tes Faktor	Hasil	Keterangan
Memilih Menu Dalam Game	✓	Memilih kembali menu utama atau lanjut
Screenshot		
		

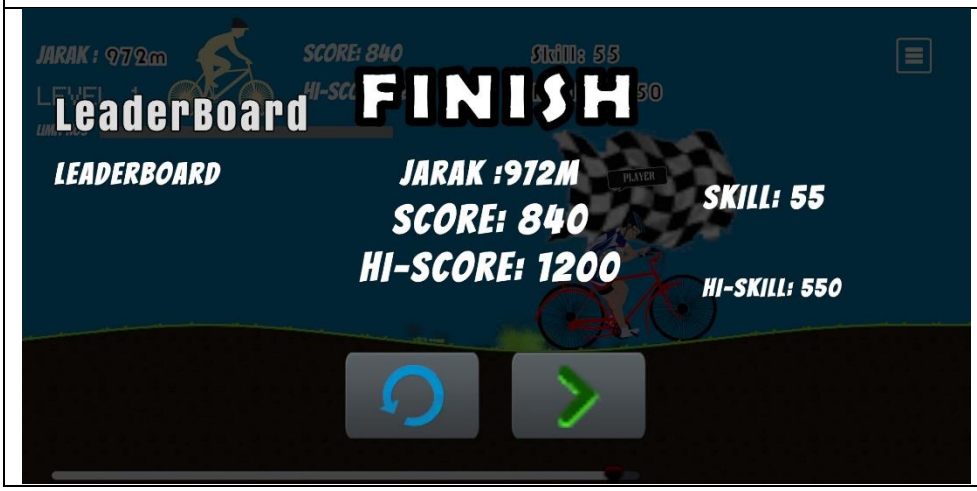
f. Black Box Game Over

Tabel 4. 6 *Black Box Game Over*

Tes Faktor	Hasil	Keterangan
Tampilan game over	✓	Menampilkan tombol mengulang level
Screenshot		
		

g. Black Box Finish

Tabel 4. 7 *Black Box Finish*

Tes Faktor	Hasil	Keterangan
Tampilan finish	✓	Menampilkan tombol mengulang atau melanjutkan
Screenshot		
		

2. White Box

Pengujian aplikasi dilakukan dengan cara pengujian white box. White box adalah metode pengujian perangkat lunak yang menguji struktur internal atau cara kerja aplikasi, yang bertentangan dengan fungsinya. Dalam pengujian kotak putih, perspektif internal sistem digunakan untuk merancang kasus uji.

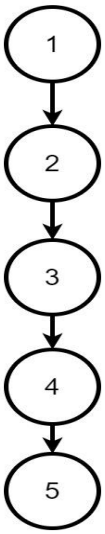
a. White *Box Tampilan awal*

1) *Flowchart*



Gambar 4. 23 *Flowchart Tampilan awal*

2) *Flowgraph*



Gambar 4. 24 *Flowgraph Tampilan awal*

Berdasarkan Gambar 4. 23 di atas dapat dilakukan perhitungan:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus : V(G) = E – N + 2

E (*edge*) = 4

N (*node*) = 5

P (Predikat *node*) = 0

Penyelesaian :

V(G) = E – N + 2

= 4 – 5 + 2

= 1

Predikat (P) = P + 1

= 0 + 1

= 1

(2) Berdasarkan perhitungan *Cyclomatic Complexcity* dari *flowgraph* di atas memiliki Region = 1

(3) *Independent path* pada *flowgraph* tersebut yakni:

Path 1 = 1 – 2 – 3 – 4 – 5

(4) Grafik matriks halaman tampilan awal

Tabel 4. 8 Grafik matriks halaman tampilan awal

	1	2	3	4	5	E-1
1		1				1 – 1 = 0
2			1			1 – 1 = 0
3				1		1 – 1 = 0
4					1	1 – 1 = 0
5						0
SUM (E + 1)						0 + 1 = 1

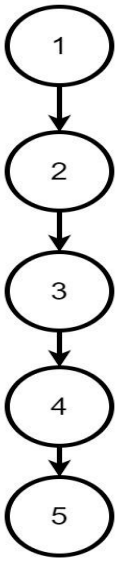
b. *White Box Tentang Kami*

1) Flowchart



Gambar 4. 25 *Flowchart Tentang Kami*

2) Flowgraph



Gambar 4. 26 *Flowgraph Tentang Kami*

Berdasarkan Gambar 4. 25 di atas dapat dilakukan perhitungan:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus : V(G) = E – N + 2

E (*edge*) = 4

N (*node*) = 5

P (Predikat *node*) = 0

Penyelesaian :

V(G) = E – N + 2

= 4 – 5 + 2

= 1

Predikat (P) = P + 1

= 0 + 1

= 1

(2) Berdasarkan perhitungan *Cyclomatic Complexcity* dari *flowgraph* di atas memiliki Region = 1

(3) *Independent path* pada *flowgraph* tersebut yakni:

Path 1 = 1 – 2 – 3 – 4 – 5

(4) Grafik matriks halaman tentang kami

Tabel 4. 9 Grafik matriks halaman tentang kami

	1	2	3	4	5	E-1
1		1				1 – 1 = 0
2			1			1 – 1 = 0
3				1		1 – 1 = 0
4					1	1 – 1 = 0
5						0
SUM (E + 1)						0 + 1 = 1

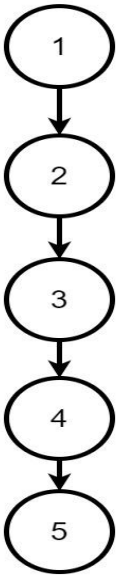
c. *White Box Menu Dalam Level*

1) *Flowchart*



Gambar 4. 27 *Flowchart Menu Dalam Level*

2) *Flowgraph*



Gambar 4. 28 *Flowgraph Menu Dalam Level*

Berdasarkan Gambar 4. 27 di atas dapat dilakukan perhitungan

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus : V(G) = E – N + 2

E (*edge*) = 4

N (*node*) = 5

P (Predikat *node*) = 0

Penyelesaian :

V(G) = E – N + 2

= 4 – 5 + 2

= 1

Predikat (P) = P + 1

= 0 + 1

= 1

(2) Berdasarkan perhitungan *Cyclomatic Complexcity* dari *flowgraph* di atas memiliki Region = 1

(3) *Independent path* pada *flowgraph* tersebut yakni:

Path 1 = 1 – 2 – 3 – 4 – 5

(4) Grafik matriks halaman menu dalam level

Tabel 4. 10 Grafik matriks halaman menu dalam level

	1	2	3	4	5	E-1
1		1				1 – 1 = 0
2			1			1 – 1 = 0
3				1		1 – 1 = 0
4					1	1 – 1 = 0
5						0
SUM (E + 1)						0 + 1 = 1

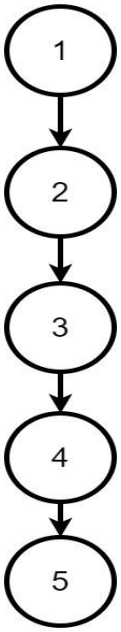
d. *White Box Game Over*

1) Flowchart



Gambar 4. 29 *Flowchart Game Over*

2) Flowgraph



Gambar 4. 30 *Flowgraph Game Over*

Berdasarkan Gambar 4. 29 di atas dapat dilakukan perhitungan:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus : V(G) = E – N + 2

E (*edge*) = 4

N (*node*) = 5

P (Predikat *node*) = 0

Penyelesaian :

V(G) = E – N + 2

= 4 – 5 + 2

= 1

Predikat (P) = P + 1

= 0 + 1

= 1

(2) Berdasarkan perhitungan *Cyclomatic Complexcity* dari *flowgraph* di atas memiliki Region = 1

(3) *Independent path* pada *flowgraph* tersebut yakni:

Path 1 = 1 – 2 – 3 – 4 – 5

(4) Grafik matriks halaman game over

Tabel 4. 11 Grafik matriks halaman game over

	1	2	3	4	5	E-1
1		1				1 – 1 = 0
2			1			1 – 1 = 0
3				1		1 – 1 = 0
4					1	1 – 1 = 0
5						0
SUM (E + 1)						0 + 1 = 1

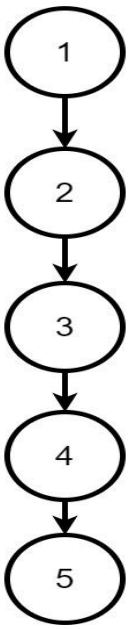
e. *White Box Finish*

1) Flowchart



Gambar 4. 31 *Flowchart Finish*

2) Flowgraph



Gambar 4. 32 *Flowgraph Finish*

Berdasarkan Gambar 4. 31 di atas dapat dilakukan perhitungan:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus : V(G) = E – N + 2

E (*edge*) = 4

N (*node*) = 5

P (Predikat *node*) = 0

Penyelesaian :

V(G) = E – N + 2

= 4 – 5 + 2

= 1

Predikat (P) = P + 1

= 0 + 1

= 1

(2) Berdasarkan perhitungan *Cyclomatic Complexcity* dari *flowgraph* di atas memiliki Region = 1

(3) *Independent path* pada *flowgraph* tersebut yakni:

Path 1 = 1 – 2 – 3 – 4 – 5

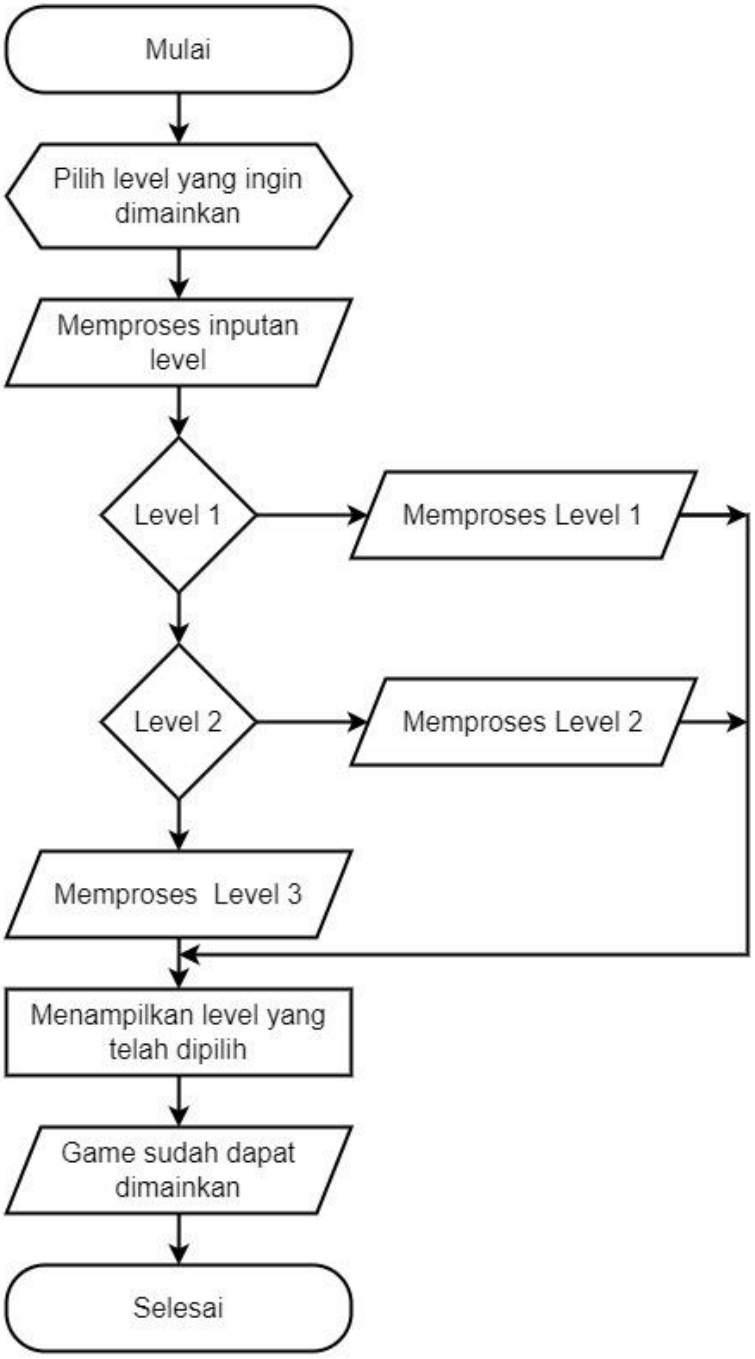
(4) Grafik matriks halaman finish

Tabel 4. 12 Grafik matriks halaman finish

	1	2	3	4	5	E-1
1		1				1 – 1 = 0
2			1			1 – 1 = 0
3				1		1 – 1 = 0
4					1	1 – 1 = 0
5						0
SUM (E + 1)						0 + 1 = 1

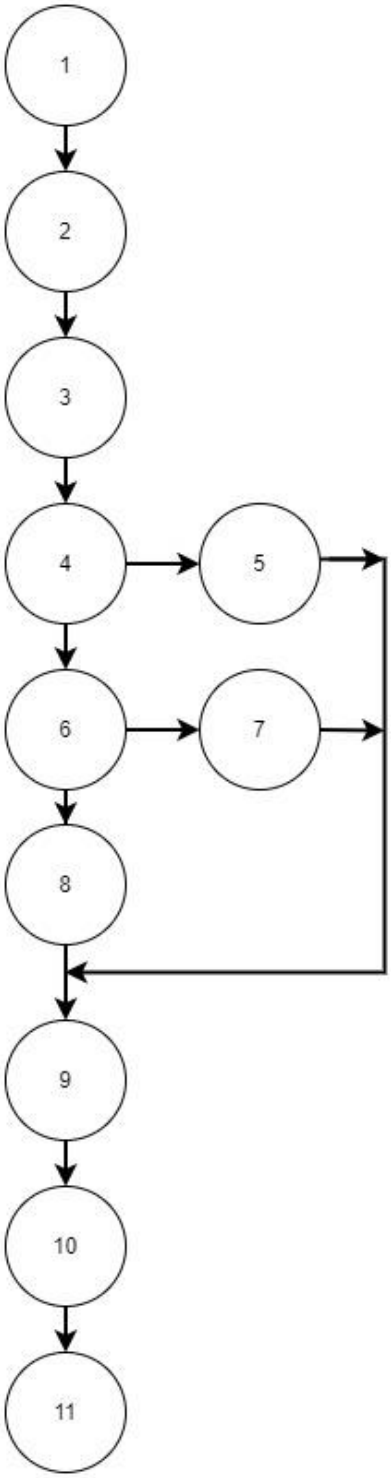
f. *White Box* mainkan game

1) *Flowchart*



Gambar 4. 33 *Flowchart mainkan game*

2) *Flowgraph*



Gambar 4. 34 *Flowgraph mainkan game*

Berdasarkan Gambar 4. 33 di atas dapat dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus : V(G) = E – N + 2

E (*edge*) = 12

N (*node*) = 11

P (Predikat *node*) = 2

Penyelesaian :

V(G) = E – N + 2

= 12 – 11 + 2

= 3

Predikat (P) = P + 1

= 2 + 1

= 1

(2) Berdasarkan perhitungan *Cyclomatic Complexcity* dari *flowgraph* di atas memiliki Region = 3

(3) *Independent path* pada *flowgraph* tersebut yakni:

Path 1 = 1 – 2 – 3 – 4 – 5 – 9 – 10 – 11

Path 2 = 1 – 2 – 3 – 4 – 6 – 7 – 9 – 10 – 11

Path 3 = 1 – 2 – 3 – 4 – 6 – 8 – 9 – 10 – 11

(4) Grafik matriks halaman mainkan game

Tabel 4. 13 Grafik matriks halaman mainkan game

	1	2	3	4	5	6	7	8	9	10	11	E-1
1		1										1 – 1 = 0
2			1									1 – 1 = 0
3				1								1 – 1 = 0
4					1	1						2 – 1 = 1
5									1			1 – 1 = 0
6							1	1				2 – 1 = 1
7									1			1 – 1 = 0
8									1			1 – 1 = 0
9										1		1 – 1 = 0
10											1	1 – 1 = 0
11												0
SUM (E + 1)												2 + 1 = 3

BAB V

PENUTUP

A. Kesimpulan

Berdasarkan penelitian yang telah dilakukan maka dapat ditarik kesimpulan bahwa pada pembuatan game balap sepeda ontel ini menggunakan aplikasi unity dengan berbagai fitur yaitu adanya tampilan main menu terdiri dari beberapa menu yaitu menu mulai, menu tentang kami dan menu keluar. Dalam menu mulai terdiri dari pilihan level 1 sampai level 3, adapun tingkat kesulitan ditentukan oleh tingkat kesulitan jalur dan jarak antara stamina yang semakin sulit ketika level semakin tinggi. Adapun menu tentang kami menjelaskan tentang bioadata pembuat game ini. Jadi dapat disimpulkan bahwa pada penelitian yang telah dilakukan penulis telah berhasil dalam merancang dan membuat aplikasi *game* balap sepeda ontel menggunakan unity berbasis *Android*.

B. Saran

Pada penelitian ini penulis menyadari bahwa masih ada beberapa kekurangan yang sangat di perlu perbaikan dan pengembangan di penelitian selanjutnya. Maka, penulis memiliki beberapa saran untuk pengembangan selanjutnya, sebagai berikut:

1. Penelitian selanjutnya perlu menambahkan mode multiplayer agar menarik untuk dimainkan.
2. *Game* yang penulis buat masih sederhana, terutama penampilan grafis dan animasi, sehinnga untuk pengembangan berikutnya dapat diperbaharui dengan tampilan yang lebih menarik dan lebih interaktif.

DAFTAR PUSTAKA

Antara, I. G. E., Putrama, I. M., & Sindu, I. G. P. (2019). *Pengembangan Game Tradisional Megala-gala Berbasis Android*. KARMAPATI (Kumpulan Artikel Mahasiswa Pendidikan Teknik Informatika), 8(2), 285-297.

Hasyim, A. H. (2021). Dasar Pemrograman. Makassar: CV. Bangun Bumitama. ISBN 978-602-51846-4-2.

Luqman, A. H. (2018). *Sistem Sosial Komunitas Sepeda Onthel dalam Mempertahankan Eksistensi Komunitas* (Studi Kasus: Bintaro Onthel Solidarity (BOS), Tangerang Selatan) (Bachelor's thesis, Jakarta: Fakultas Ilmu Sosial dan Ilmu Politik UIN Syarif Hidayatullah).

Madiun, Madcoms. 2018. *Kupas Tuntas Adobe Photoshop CS6*. Yogyakarta : Andi.

N, Sora. 2015. Pengertian UML dan Jenis - jenisnya. From http://www.pengertianku.net/2015/09/pengertian-uml-dan-jenis-jenisnya-serta-contoh-diagramnya.html. (Diakses pada 25 Mei 2022).

Prasetyo, R. M. M., Syaputra, H., Cholil, W., & Sauda, S. (2021). *Rancang Dan Bangun Game Edukasi Anak-Anak Berbasis Android Dengan Unity Menggunakan Metode Game Development Life Cycle*. Jurnal Nasional Ilmu Komputer, 2(2), 103-111.

Reno, R. (2021). *Pembuatan Game Balap Kelinci dengan Unity Berbasis Android. Jurnal Ilmiah Rekayasa dan Manajemen Sistem Informasi*, 7(1), 19-26.

Saputra, S. Y., & Subari, S. (2019). *Game Multiplayer "Mini Car Circuit" Berbasis Android*. J-INTECH (Journal of Information and Technology), 7(01), 58-66.

Silalahi, Decson Roberto. 2017. *Game "Catch Me" Menggunakan Game Engine Contruct 2 Berbasis Diagram UML dan Storyboard*. Skripsi. Batam :TEKNIK MULTIMEDIA JARINGAN POLITEKNIK NEGERI.

Silvianita, Helva. 2019. Pengertian Game.https://www.nesabamedia.com/pengertian-game.html. (Diakses pada 24 Mei 2022).