

Rancang Bangun Aplikasi *Drone Simulator* Berbasis Android Menggunakan Game Engine Unity

Muhammad Iskandar Java¹, Ricky Eka Putra²

^{1,2} Program Studi S1 Teknik Informatika, Fakultas Teknik, Universitas Negeri Surabaya

¹muhammadjava@mhs.unesa.ac.id

²rickyveka@unesa.ac.id

Abstrak— *Drone* merupakan sebuah teknologi yang sedang populer pada saat ini. Akan tetapi untuk membeli sebuah *drone* tidaklah murah dan beresiko akan terjadi kerusakan pada saat proses belajar mengendalikannya. Namun dengan adanya teknologi digital, seseorang dapat dengan mudah belajar menggunakan teknologi manual dalam bentuk *virtual* atau biasa yang disebut dengan simulasi atau *simulator*. Keuntungan yang diberikan oleh *simulator* sendiri yaitu biayanya yang relatif lebih murah dibandingkan biaya untuk membeli peralatan yang sesungguhnya seperti *drone*. Agar memudahkan seseorang dalam belajar menggunakan *drone*, dibuatlah aplikasi bernama *Drone Simulator*. Aplikasi tersebut memberikan fitur dimana pengguna dapat menerbangkan *drone* dengan kontrol yang mirip dengan *drone* asli. Aplikasi tersebut berjalan pada platform android sehingga dapat digunakan dimana saja. Dalam pembangunan aplikasi tersebut, metode pengembangan sistem yang digunakan adalah metode SDLC (*Software Development Life Cycle*) dengan model *waterfall* dengan alur analisis kebutuhan sistem, perancangan, implementasi, testing dan pemasaran. Dari hasil pengujian yang dilakukan, aplikasi berjalan dengan baik tanpa adanya error pada pengujian *black box testing* dan didapatkan nilai yang cukup baik dari hasil uji kuisioner yaitu sebesar 73% dari 45 responden.

Kata Kunci— *drone*, simulasi, *simulator*, digital, *Drone Simulator*, *waterfall*, testing, *black box testing*.

I. PENDAHULUAN

Dewasa ini banyak teknologi yang dimanfaatkan untuk membantu berbagai pekerjaan manusia baik itu secara nyata ataupun digital. Salah satu contoh teknologi yang sedang populer saat ini adalah *drone*. *Drone* merupakan pesawat tanpa awak yang dapat dikendalikan oleh manusia pada jarak jauh. Dengan adanya teknologi *drone*, berbagai pekerjaan yang sulit dilakukan oleh manusia dapat diatasi seperti mengambil gambar atau video dari udara, memantau wilayah dengan medan yang berbahaya, dan dapat digunakan untuk patroli pada bidang militer [6]. Terdapat berbagai jenis *drone* yang dikembangkan pada saat ini seperti jenis *Fixed Wing Drone* yang menggunakan sayap untuk terbang dan *Rotary Wing Drone* yang menggunakan baling-baling untuk terbang. Untuk jenis *rotary wing* terdapat berbagai tipe sesuai dengan jumlah baling-baling yang dimilikinya yaitu *single copter* dengan satu baling-baling, *double copter* dengan dua baling-baling, *tricopter* dengan tiga baling-baling dan *quadcopter* dengan empat baling-baling [7].

Untuk mengendalikan sebuah *drone* agar dapat melakukan tugas yang diinginkan tidaklah mudah, seorang pilot *drone* perlu berlatih menguasai kontrol pada *drone* agar *drone* yang

dikendalikan tidak menabrak atau terjatuh yang kemudian akan mengakibatkan rusaknya komponen pada *drone*. Umumnya *drone* memiliki kontroller stik kiri dan stik kanan. Stik kiri berfungsi untuk mengendalikan putaran *drone* dan menggerakkan *drone* naik dan turun. Sedangkan stik kanan berfungsi untuk menggerakkan *drone* ke samping atau maju dan mundur [9]. Melihat harga *drone* yang masih terbilang mahal di pasaran maka hal tersebut membuat seseorang yang masih awam dengan kontrol pada *drone* akan berpikir dua kali untuk membelinya karena resiko *drone* akan terjatuh dan rusak pada saat belajar.

Saat ini hampir setiap pekerjaan tradisional dapat diimplementasikan ke dalam sebuah sistem digital, seperti jam analog yang menjadi jam digital, berjualan di pasar sekarang dapat dilakukan secara online, permainan tradisional menjadi permainan yang ada pada ponsel dan masih banyak lagi. Teknologi digital memungkinkan penggunaannya mendapatkan banyak keuntungan, yaitu meminimalisir harga, perangkat yang digunakan relatif kecil, cepat dalam berkomunikasi, dapat digunakan hampir dimana saja, dan sebagainya [4].

Pada penelitian yang berjudul Rancang Bangun Permainan Ujian Simulasi Berkendara 3D, sistem yang dibangun bertujuan untuk mengenali cara mengemudi yang baik sebelum mendapatkan SIM, seperti uji parkir, uji melewati jalan sempit, dan juga tanjakan. Aplikasi yang dibangun pada penelitian tersebut menggunakan *software Unity* dengan bahasa pemrograman C#. Pada pengujiannya, game tersebut berjalan dengan baik dan dapat memberikan gambaran dalam hal menyetir mobil.

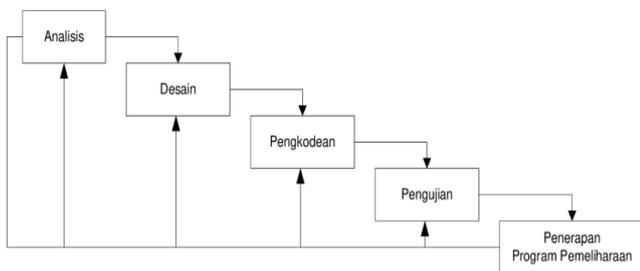
Dari permasalahan tersebut, ditawarkan sebuah aplikasi simulasi menerbangkan *drone* yang berjalan pada perangkat mobile android sehingga dapat digunakan dimana saja. Aplikasi ini dilengkapi fitur kontrol *drone* yang mirip seperti *drone* asli sehingga para pemain dapat belajar menerbangkan *drone* tanpa harus takut *drone* akan rusak. Dengan beberapa track lintasan yang disediakan pada aplikasi ini, diharapkan pengguna dapat berlatih mengendalikan *drone* pada medan yang sulit seperti pada ruang yang sempit.

Dalam pengembangannya, aplikasi ini dibangun menggunakan *engine Unity* dengan bahasa pemrograman C# dan aplikasi Blender3D untuk pembuatan model 3D nya. Unity merupakan salah satu game engine yang dapat mengolah gambar, grafik, suara, input, dan lain-lain untuk keperluan pengembangan game [5]. Unity dibuat oleh Unity Technology dan dapat digunakan secara gratis. Sedangkan Blender 3D merupakan aplikasi grafik komputer yang berfungsi untuk membuat gambar atau animasi berkualitas tinggi dengan

menggunakan geometri tiga dimensi [8]. Dengan fitur-fitur yang disediakan oleh Unity dan Blender 3D, proses pembuatan aplikasi ini menjadi lebih mudah dan cepat.

II. METODOLOGI

Penelitian ini menggunakan metode SDLC (*System Development Life Cycle*) dengan model *waterfall* dalam pengembangan aplikasinya. Seperti namanya yaitu *waterfall* yang berarti air terjun, metode *waterfall* merupakan salah satu metode pengembangan sistem dimana metode tersebut menggambarkan pengembangan sistem yang dilakukan secara sequensial dan juga berurutan [1]. Alasan mengapa model *waterfall* dipilih untuk penelitian ini karena dalam proses implementasinya *waterfall* dikerjakan secara urut dan tidak saling tumpang tindih sehingga hasil akhir dari penelitian yang dilakukan dapat sesuai dengan tujuan awal penelitian. Gbr 1 menunjukkan alur dari model *waterfall*.



Gbr 1. Alur metode *waterfall*

Terdapat lima langkah pada model *waterfall* yaitu:

A. Analisis Kebutuhan

tahapan pertama dalam mengembangkan aplikasi adalah dengan menganalisa kebutuhan sistem aplikasi *drone simulator* melalui studi literatur. Dengan demikian, keinginan user terhadap aplikasi simulasi penerbangan *drone* dapat diperoleh.

B. Desain atau Perancangan

Tahap perancangan pada sistem yang akan digunakan yaitu menggunakan *class diagram*, *use case diagram* dan perancangan tampilan dan *environment* pada aplikasi. *Class diagram* digunakan untuk mendeskripsikan arsitektur dan struktur sistem dengan mendefinisikan *class*, atribut, operasi, dan hubungan antar objek dalam sebuah diagram [12]. Sedangkan *use case diagram* digunakan untuk mendeskripsikan interaksi yang terjadi antara pengguna dengan sistem dimana terdapat kumpulan aksi, layanan dan fungsi yang dibutuhkan sistem untuk berjalan [3].

C. Implementasi

Tahap implementasi dilakukan setelah tahap perancangan selesai. Implementasi merupakan tahapan mengembangkan sistem yang sesungguhnya yaitu aplikasi dari hasil rancangan yang telah dibuat. Tahap ini terdiri dari *3D modelling*, dan pembuatan aplikasi *Drone Simulator*.

D. Testing

Pada tahap testing, seluruh fitur pada aplikasi yang telah berhasil dibuat akan dilakukan pengujian konstan untuk

mencari kesalahan. Pengujian dilakukan terus menerus hingga tidak ada kesalahan pada aplikasi [11]. Metode yang digunakan untuk testing pada penelitian ini adalah metode *blackbox* dan kuisisioner.

E. Penerapan dan Pemeliharaan Program

Aplikasi yang telah teruji dan berjalan baik akan disebarkan pada pengguna atau dipasarkan dan dilakukan pembaharuan untuk memodifikasi sistem atau performa. Pembaharuan dilakukan atas dasar permintaan dari pengguna atau karena kerusakan program yang ditemukan selama masa pemakaian [11].

III. HASIL DAN PEMBAHASAN

A. Analisis Kebutuhan

Dari studi literatur yang dilakukan, didapatkan beberapa kebutuhan yang harus dipenuhi baik kebutuhan yang berupa fungsional maupun kebutuhan nonfungsional agar aplikasi yang dibangun dapat menyelesaikan permasalahan-permasalahan yang telah dirumuskan. Kebutuhan-kebutuhan tersebut yaitu:

1) Kebutuhan Fungsional

- Pengguna dapat memilih arena atau *environment* sesuai dengan keinginan mereka.
- Aplikasi mampu memberikan *physics* pada *drone* seperti *drone* pada dunia nyata.
- Aplikasi menyediakan kontrol untuk mengendalikan *drone* digital yang mirip dengan kontrol pada *drone* yang sebenarnya.
- Pengguna dapat mengendalikan *drone* hingga bergerak sesuai dengan arah yang dituju.

2) Kebutuhan Nonfungsional

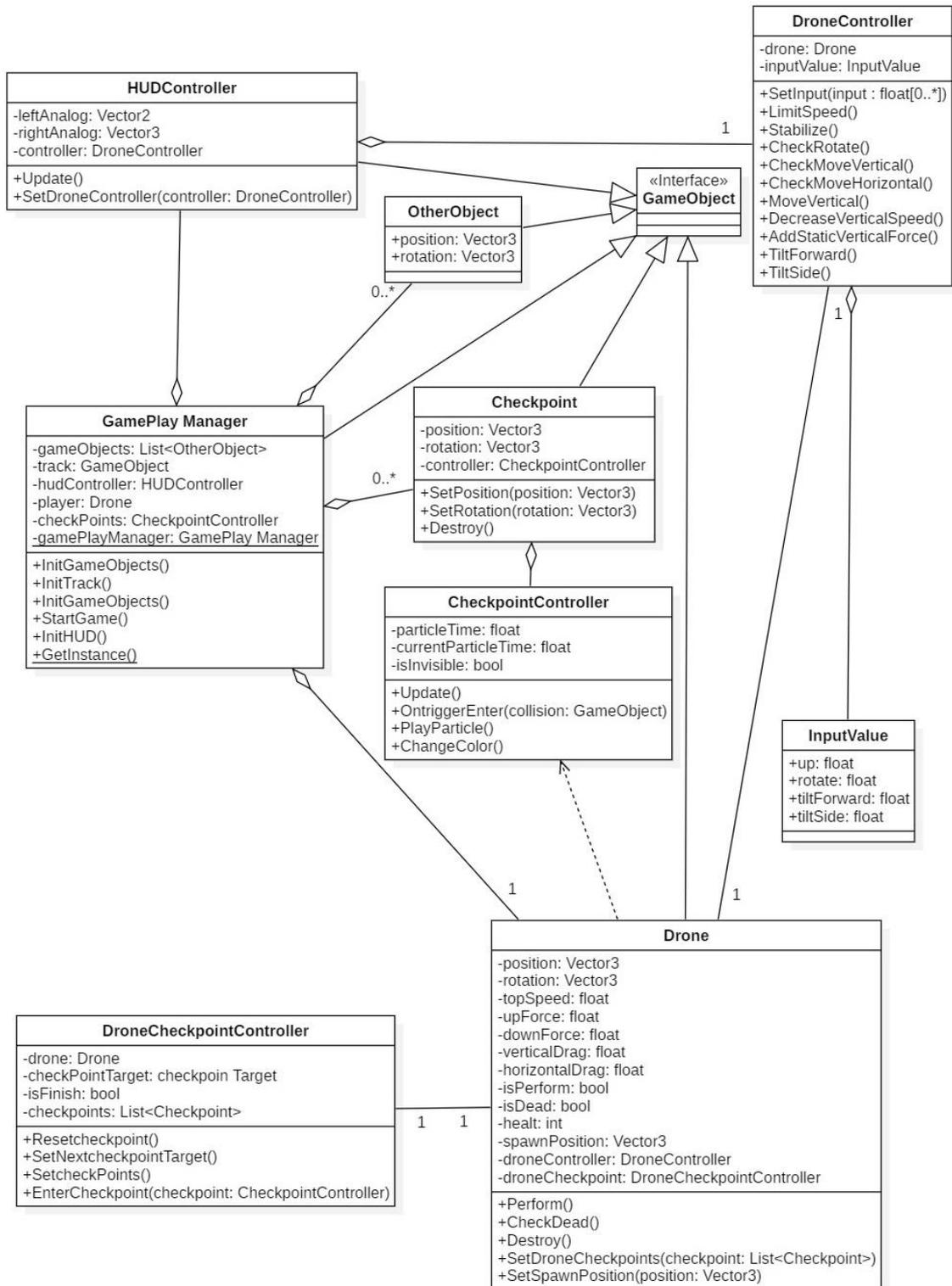
- Kebutuhan untuk Pengembangan
 - OS Windows 10
 - Processor Intel Core i5 2.7 GHz.
 - Intel HD Graphics, Nvidia GT940.
 - Memory 8 GB DDR4.
 - Harddisk 500 GB
 - Aplikasi Star UML
 - Aplikasi Corel Draw X7
 - Game Engine Unity
 - Aplikasi Blender
 - IDE Visual Studio 2017
- Kebutuhan untuk Penggunaan
 - OS Android Oreo
 - Chipset Snapdragon 636
 - CPU Kryo 260 1.8 GHz
 - RAM 2GB
 - Penyimpanan 32 GB

B. Desain atau Perancangan

Perancangan sistem untuk pembuatan aplikasi *Drone Simulator* menggunakan *Class Diagram*, *activity diagram* dan *Use Case diagram*. Perancangan sistem dibuat agar struktur, penggunaan, dan apa saja yang disediakan program untuk pengguna dapat dengan mudah untuk dimengerti. *Class diagram*, *activity diagram* dan *use case* pada penelitian ini dibuat menggunakan *software* bernama Star UML. Alasan

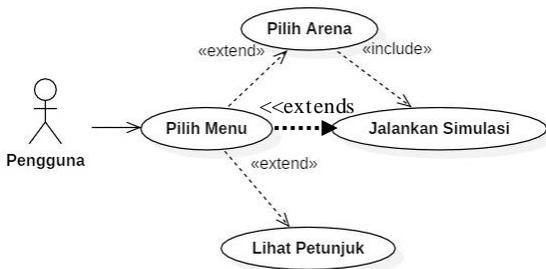
menggunakan *software* tersebut karena fitur yang diberikan cukup lengkap dan tampilannya cukup sederhana sehingga mudah untuk digunakan. Untuk perancangan desain antar muka pada aplikasi *Drone Simulator* dibuat menggunakan *software* bernama *Corel Draw*. Alasan menggunakan *software* tersebut

karena *Corel Draw* memiliki banyak fitur yang mendukung untuk menggambar objek 2D dari yang sederhana hingga yang kompleks. Hasil perancangan yang dibuat untuk pengembangan aplikasi *drone simulator* adalah sebagai berikut:



Gbr 2 Class diagram yang digunakan pada sistem

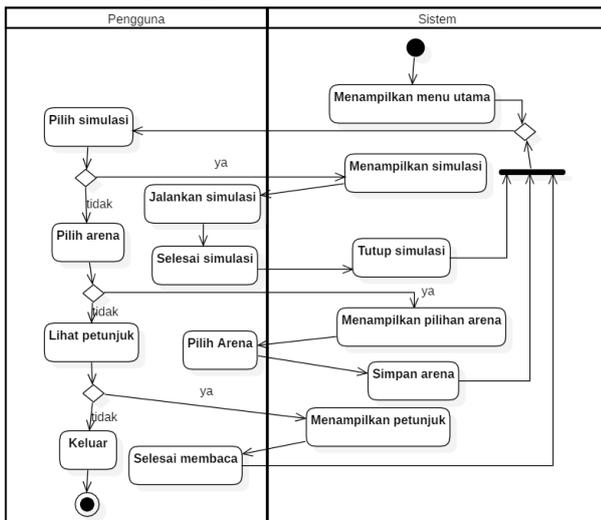
1) Perancangan Sistem:
 a. Use Case Diagram



Gbr 3 Use Case Diagram yang digunakan oleh sistem

Gbr 4. merupakan Use Case Diagram untuk aplikasi Drone Simulator yang akan digunakan oleh pengguna. Actor pada aplikasi ini hanya satu yaitu pengguna. Terdapat empat aksi yang dapat dilakukan. Saat pengguna menggunakan aplikasi, hal pertama yang akan ditampilkan adalah menu utama. Menu utama yang tersedia pada aplikasi Drone Simulator ada tiga pilihan yaitu pilih arena untuk memilih arena yang ingin dicoba, mulai simulasi untuk langsung menuju simulasi dan lihat petunjuk untuk menampilkan cara menggunakan aplikasi.

b. Activity Diagram



Gbr 4 Activity diagram yang digunakan oleh sistem

Activity diagram menggambarkan aktivitas apa saja yang terjadi pada aplikasi drone simulator dengan hubungan antara entitas satu dengan entitas yang lain. Dengan adanya entitas, maka dapat diketahui pelaku dari aktivitas-aktivitas yang ada.

Pada aplikasi ini, hanya terdapat dua entitas, yaitu pengguna dan sistem. Ketika aplikasi dibuka, maka sistem akan menampilkan beberapa pilihan untuk pengguna, yaitu mulai simulasi, pilih arena, lihat petunjuk dan keluar. Saat pengguna memilih mulai simulasi, maka sistem akan langsung menampilkan

simulasi untuk dijalankan oleh pengguna. Ketika pengguna selesai menjalankan simulasi, sistem akan menutup simulasi dan kembali pada menu awal aplikasi. Ketika pengguna memilih menu pilih arena, maka sistem akan menampilkan daftar arena yang tersedia untuk dipilih oleh pengguna dan akan disimpan. Jika pengguna memilih lihat petunjuk, maka sistem akan menampilkan tata cara penggunaan simulasi dan jika pengguna memilih keluar maka sistem akan keluar dari aplikasi. Activity diagram pada aplikasi ini ditunjukkan pada Gbr 3.

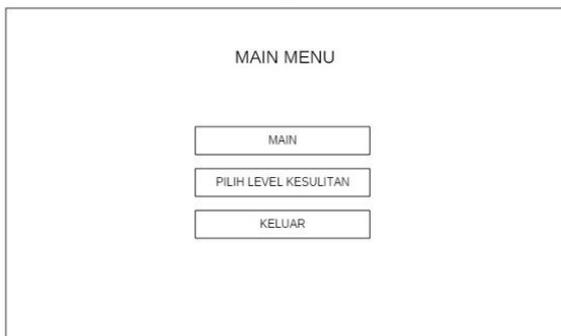
c. Class Diagram

Gbr 2 merupakan class diagram dimana menggambarkan struktur program yang dibuat dan menggambarkan objek secara keseluruhan yang ada pada aplikasi drone simulator. Class diagram tersebut juga menggambarkan hubungan antara objek satu dengan objek yang lain pada saat aplikasi dijalankan.

Pada aplikasi ini, terdapat sembilan kelas utama yang berelasi ketika simulasi dijalankan yaitu HUDController, DroneController, Drone, DroneCheckpointcontroller, GamePlayManager, Checkpoint, Checkpointcontroller, OtherObject, dan InputValue. GamePlayManager merupakan class yang bertugas mengatur semua objek dan skenario yang ada pada saat simulasi, sehingga semua class objek seperti Drone, Checkpoint, OtherObject dan HUDController merupakan childclass dari GamePlayManager. Class-class objek yang ada pada simulasi memiliki childclass yang disebut dengan istilah komponen. Class Drone memiliki dua childclass dengan nama DroneCheckpointcontroller dan DroneController yang berfungsi masing-masing untuk mengontrol objek drone dan berinteraksi dengan objek cekpoin. Pada class Checkpoin terdapat childclass dengan nama Checkpointcontroller yang berfungsi untuk mengecek apakah objek drone berinteraksi dengan objek cekpoin.

2) Perancangan Antar Muka: Antar muka pada aplikasi ini memiliki beberapa komponen yaitu tombol, teks dan gambar. Tombol digunakan untuk menjalankan aksi yang diinginkan, teks digunakan untuk memberikan keterangan yang berkaitan pada aktivitas yang sedang dilakukan dan gambar digunakan untuk memberikan gambaran dari keterangan yang ada. Berikut adalah rancangan antar muka pada aplikasi Drone Simulator:

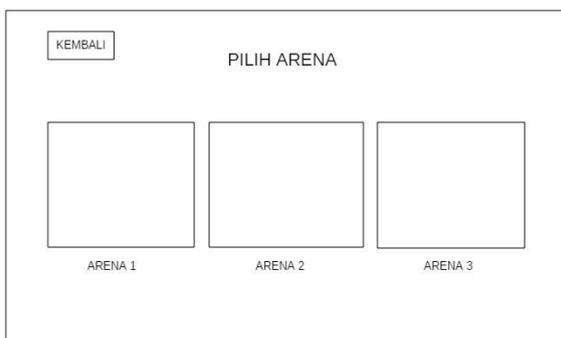
a. Antar Muka Menu Utama



Gbr 5 Tampilan main menu atau menu utama

Antarmuka pada menu utama terdapat tiga pilihan yaitu main untuk menjalankan simulasi, pilih arena untuk memilih arena yang akan dibuat simulasi, dan keluar untuk keluar aplikasi. Desain tampilan “Main Menu” dapat dilihat pada Gbr 5.

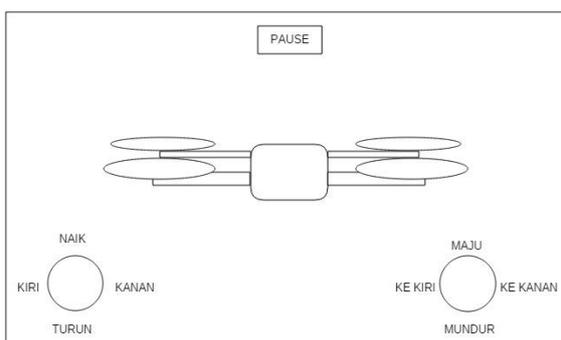
b. Antar Muka Pilih Arena



Gbr 6 Tampilan menu pilih arena

Pada menu pilih arena ditampilkan gambar beberapa arena yang dapat dipilih seperti yang ditunjukkan pada Gbr 6.

c. Antar Muka Simulasi



Gbr 7 Tampilan saat simulasi

Antar muka simulasi menampilkan drone yang dikendalikan dengan mode *third person* atau orang ketiga dengan kontrol analog yang ada pada kanan

bawah dan kiri bawah layar seperti yang ditunjukkan pada Gbr 7.

C. Implementasi

Aplikasi *Drone Simulator* ini berjalan pada platform Android dan berjalan secara *offline* sehingga dapat digunakan dimanapun dan kapan pun. Simulasi *drone* pada aplikasi ini berbasis 3D sehingga memberikan sensasi seperti menjalankan *drone* yang asli. Hasil aplikasi yang dibangun merupakan hasil dari rancangan yang dibuat sebelumnya, berikut adalah hasil dan pembahasan dari aplikasi *Drone Simulator*:

1) *Model 3D*: Model 3D pada aplikasi *Drone simulator* ini dibuat dengan menggunakan aplikasi Blender 3D. Hasil dari objek yang telah dibuat dari aplikasi Blender kemudian diimport ke dalam projek *Unity* dan dijadikan sebagai *gameobject* pada *Unity*. Setelah model 3D menjadi *gameobject*, maka objek tersebut dapat diberikan kode berupa *script* agar *gameobject* dapat berfungsi pada aplikasi.

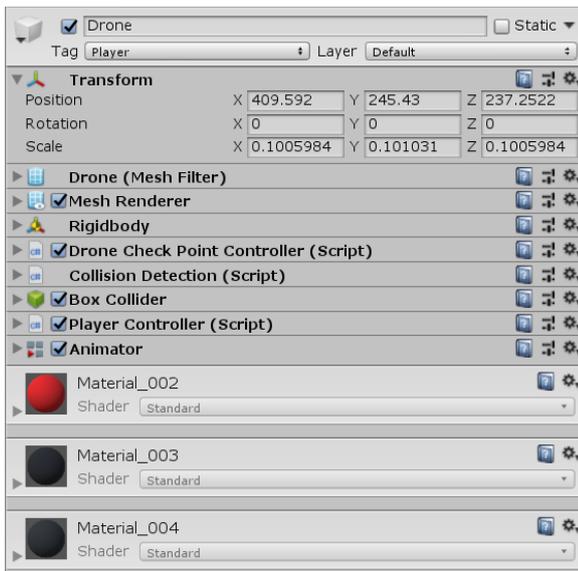
a. Model 3D *Drone*

Model *drone* pada aplikasi ini merupakan *drone* dengan jenis *Quadcopter*, yaitu dengan empat baling-baling yang ada di depan dan belakang. Model *drone* ini diimport ke dalam projek *Unity* dan dijadikan *gameobject*. Agar *drone* terlihat lebih realistis, maka ditambahkan animasi pada keempat baling-balingnya. Fungsi dari objek *drone* ini adalah untuk merepresentasikan *drone* yang dikendalikan oleh pengguna aplikasi seperti yang ditunjukkan pada Gbr 8.



Gbr 8 Model 3D *drone*

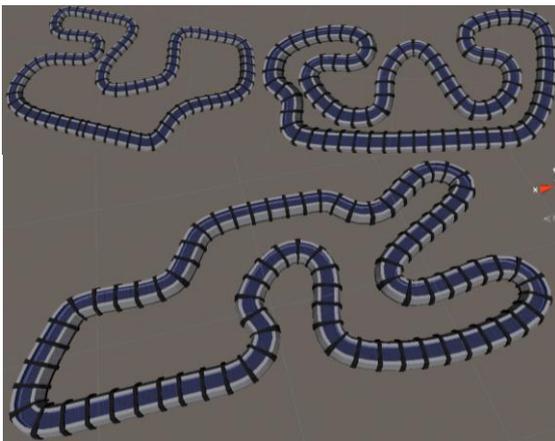
Model *Drone* yang menjadi sebuah *gameobject* pada *Unity* memiliki beberapa komponen agar dapat dijalankan oleh pengguna yaitu *Transform*, *MeshFilter*, *Rigidbody*, *DroneCheckpointcontroller*, *CollisionDetection*, *BoxCollider*, *DroneController*, dan *Animator*. Komponen-komponen pada *gameobject drone* dapat dilihat pada Gbr 9.



Gbr 9 Komponen-komponen pada gamobject drone

b. Model 3D Arena Lintasan

Model arena lintasan pada penelitian ini dibuat sebanyak tiga macam dan dibuat menggunakan aplikasi *Blender* 3D. Arena lintasan berfungsi agar pengguna dapat berlatih menjalankan *Drone* pada medan yang berliku-liku. Beberapa model lintasan yang dibuat dapat dilihat pada Gbr 10.



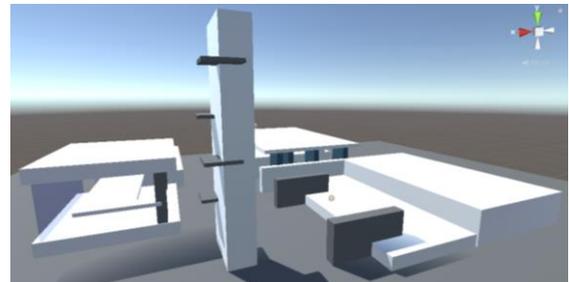
Gbr 10 Model 3D arena lintasan

Gameobject lintasan yang dibuat memiliki cekpoin yang tersebar di sepanjang lintasan yang ditunjukkan garis-garis dengan warna hitam. Fungsi dari cekpoin tersebut adalah untuk mengukur jarak dan titik tempat *respawn* atau tempat muncul *Drone* ketika terjadi tabrakan dan jatuh.

c. Model 3D Arena Bebas

Pada model 3D arena bebas, model ini dibuat langsung memanfaatkan fitur pada *Unity* yaitu berupa pembuatan objek balok. Model arena bebas tidak menggunakan aplikasi *Blender* dalam

pembuatannya karena objek yang diperlukan sederhana, yaitu berupa kumpulan dari balok yang disusun. Walaupun dibuat dari objek sederhana, namun model arena bebas tetap memiliki rintangan yang telah disesuaikan dengan tempat seperti pada umumnya yaitu *outdoor* maupun *indoor*. Model arena bebas pada aplikasi *Drone Simulator* dapat dilihat pada Gbr 11.



Gbr 11 Model 3D arena bebas

d. Model 3D Cekpoin

Cekpoin pada aplikasi *Drone Simulator* dibuat dengan aplikasi *Blender*. Cekpoin memiliki fungsi yaitu sebagai titik dimana *drone* akan muncul setelah *drone* terjatuh. Titik muncul bergantung pada cekpoin terakhir yang telah dilewati. Cekpoin memiliki lubang pada tengahnya dimana lubang tersebut harus dilewati oleh *Drone* agar mendapatkan poin. Model 3D dari cekpoin dapat dilihat pada Gbr 12.



Gbr 12 Model 3D cekpoin

Agar *gameobject* cekpoint pada aplikasi ini dapat berfungsi, *gameobject* cekpoin memiliki beberapa komponen yaitu *Checkpointcontroller*, *MeshRenderer* dan *MeshCollider*.

2) *Antar Muka Aplikasi:*

a. Antar Muka Main Menu

Pada antar muka Main menu, terdapat empat pilihan yaitu pilihan untuk mulai simulasi dan pilihan untuk menuju ke memilih arena, pilihan untuk menampilkan petunjuk dan pilihan untuk keluar dari aplikasi. Antar muka untuk Main Menu dapat dilihat pada Gbr 13.



Gbr 13 Tampilan menu utama

b. Antar Muka Pilih Arena

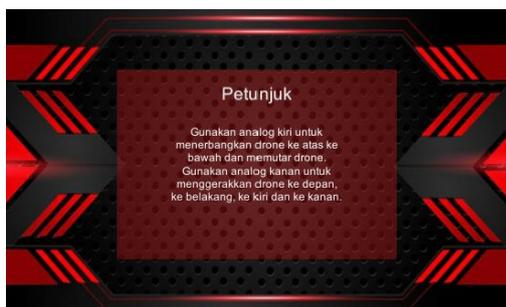
Pada antar muka pilih arena, terdapat tampilan dari setiap arena yang tersedia untuk disimulasikan, mulai dari arena lintasan pertama hingga arena yang bebas. Disini pengguna dapat memilih arena mana yang nantinya akan menjadi tempat untuk *Drone* diterbangkan pada saat proses simulasi seperti yang ditunjukkan pada Gbr 14.



Gbr 14 Tampilan menu pilih arena

c. Antar muka petunjuk

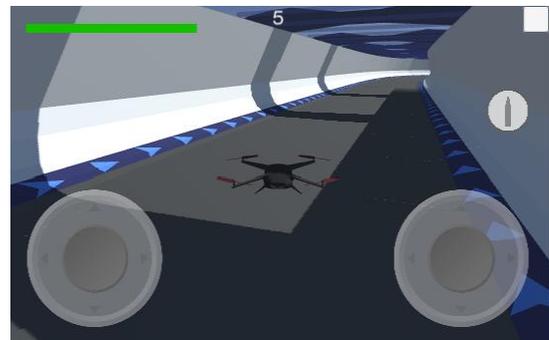
Antar muka petunjuk berfungsi untuk menampilkan cara dalam pengoperasian *Drone* agar pengguna baru dapat mengerti bagaimana cara mengontrol sebuah *drone*. Antar muka petunjuk ditunjukkan pada Gbr 15.



Gbr 15 Tampilan petunjuk

d. Simulasi

Simulasi merupakan hal yang utama dalam aplikasi ini. Simulasi pada aplikasi ini, pengguna akan ditampilkan keadaan *Drone* dengan posisi orang ketiga atau *third person*. Untuk mengendalikan *Drone*, pengguna harus menggunakan kedua analog yang telah disediakan di kanan bawah dan kiri bawah. Analog kiri bawah berfungsi untuk menerbangkan *drone* ke atas, ke bawah, putar kanan dan putar kiri. Sedangkan analog kiri berfungsi untuk menggerakkan *Drone* ke depan, ke belakang, ke kiri dan ke kanan. Pada saat simulasi, pengguna diberikan sebuah tantangan yaitu mencapai semua cekpoin yang ada pada arena, baik itu arena lintasan maupun arena bebas. Jika *Drone* yang dijalankan menabrak dan kemudian terjatuh, maka *drone* akan muncul kembali pada posisi cekpoin yang dicapai terakhir kali. Tampilan pada saat simulasi ditunjukkan pada Gbr 16.



Gbr 16 Tampilan simulasi

Testing

Pada penelitian ini, testing dilakukan dengan metode *black box* dan kuisisioner. *Black box* testing adalah menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan [10]. *Black box* testing akan dilakukan terus-menerus hingga tidak ada error yang ditemukan. Hasil dari testing menggunakan metode *black box* testing ditunjukkan pada Tabel I. Sedangkan hasil dari pengujian menggunakan metode kuisisioner yang diisi oleh 45 orang, diperoleh nilai sangat baik berjumlah 30 poin, baik berjumlah 74 poin, kurang baik berjumlah 32 poin dan sangat kurang berjumlah 4 poin.

TABEL I
HASIL BLACK BOX TESTING

No	Skenario	Test Case	Harapan	Hasil
1	Masuk menu arena	Klik tombol “Pilih Arena” pada <i>Main Menu</i>	Menuju ke menu pilih arena	Berhasil
2	Mulai simulasi	Klik tombol “Mulai Simulasi” pada <i>Main Menu</i>	Menuju ke tampilan simulasi	Berhasil
3	Kontrol <i>drone</i> ke atas	Drag analog kiri ke atas	<i>Drone</i> terbang ke atas	Berhasil
4	Kontrol <i>drone</i> ke bawah	Drag analog kiri ke bawah	<i>Drone</i> terbang ke bawah	Berhasil
5	Putar <i>drone</i> ke kiri	Drag analog kiri ke kiri	<i>Drone</i> berputar ke kiri	Berhasil
6	Putar <i>drone</i> ke kanan	Drag analog kiri ke kanan	<i>Drone</i> berputar ke kanan	Berhasil
7	Kontrol <i>drone</i> maju	Drag analog kanan ke atas	<i>Drone</i> bergerak maju	Berhasil
8	Kontrol <i>drone</i> mundur	Drag analog kanan ke bawah	<i>Drone</i> bergerak mundur	Berhasil
9	Kontrol <i>drone</i> ke kiri	Drag analog kanan ke kiri	<i>Drone</i> bergerak ke kiri	Berhasil
10	Kontrol <i>drone</i> ke kanan	Drag analog kanan ke kanan	<i>Drone</i> bergerak ke kanan	Berhasil

IV. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, kesimpulan yang dapat diperoleh dari perancangan aplikasi *Drone Simulator* Berbasis Android Menggunakan *Game Engine Unity* adalah, dalam proses perancangan aplikasi, diperlukan sebuah metode pengembangan sistem sesuai dengan kebutuhan sistem. Metode pengembangan sistem yang digunakan pada penelitian ini adalah metode SDLC (*System Development Life Cycle*) dengan model *waterfall* dengan alur analisis kebutuhan, kemudian dilanjutkan dengan desain dan perancangan sistem dengan membuat arsitektur program dalam bentuk diagram dan membuat desain tampilan, kemudian dilanjutkan implementasi atau pembuatan sistem, kemudian testing untuk mendapatkan error dan setelah itu barulah sistem siap untuk dipasarkan. Dari hasil kuisioner yang dilakukan pada 45 responden, menyatakan aplikasi yang telah dibangun dapat diterima dengan baik yaitu sebesar 73%.

REFERENSI

- [1] Dini. (2015, Oktober 15). *Kelebihan dan Kekurangan Metode Waterfall dalam Pengembangan Sistem*. Retrieved from dosenit.com: <https://dosenit.com/kuliah-it/teknologi-informasi/kelebihan-dan-kekurangan-metode-waterfall>
- [2] Dewantoro, MuryFajar. 2015. “Rancang Bangun Permainan Ujian Simulasi Berkendara 3d”. *Tesis*. Sarjana thesis, Universitas Brawijaya.
- [3] Gemino, A., & Parker, D. (2009). Use Case Diagrams in Support of Use Case Modeling: Deriving Understanding from the Picture. *Journal of Database Management*, 1-24.
- [4] Goodman, P. (2019, Maret 4). *16 Advantages of Digital Technology*. Retrieved from turbofuture.com: <https://turbofuture.com/computers/Advantages-of-Digital-Technology>
- [5] Komputer, W. (2014). *Mudah Membuat Game 3Dimensi menggunakan Unity 3D*. Yogyakarta: Penerbit ANDI.
- [6] Purnomo, L. (2017, November 26). *10 Manfaat dan Kegunaan Drone Dalam Berbagai Bidang*. Retrieved from liupurnomo.com: <https://www.liupurnomo.com/10-manfaat-dan-kegunaan-drone-dalam-berbagai-bidang/>
- [7] Purnomo, L. (2018, Januari 28). *Mengenal Jenis – Jenis Drone*. Retrieved from liupurnomo.com: <https://www.liupurnomo.com/mengenal-jenis-jenis-drone/>
- [8] Purwanto, E. (2014, Mei 12). *Cerita di Balik Software 3D Blender*. Retrieved from koinfo.go.id: <https://bpptik.koinfo.go.id/2014/05/12/419/cerita-di-balik-software-3d-blender/>
- [9] Rahayu, M. (2018, Juni 30). *Cara Menerbangkan Drone untuk Pemula!* Retrieved from plazakamera.com: <https://www.plazakamera.com/cara-menerbangkan-drone-untuk-pemula/>
- [10] Rosa AS, m. S. (2011). *Modul Pembelajaran Rekayasa Perangkat Lunak : Terstruktur dan Beorientasi Objek*. Bandung: Modula.
- [11] Sharma, L. (2016, April 17). *What is WaterFall Model?* Retrieved from toolsqa.com: <https://www.toolsqa.com/software-testing/waterfall-model/>
- [12] Sparks, G. (2011, September 8). *Database Modelling in UML*. Retrieved from methodsandtools.com: <http://www.methodsandtools.com/archive/archive.php?id=9>