BAB I

PENDAHULUAN

A. Latar Belakang

Kabupaten Enrekang memiliki banyak wisata salah satunya Wisata Kampong Buttu Macca, Kecamatan Anggareja, Destinasi ini terkenal karena pemandangan alamnya yang memukau, terutama tebing-tebing yang menjulang tinggi dan sungai yang mengalir di antara bebatuan tersebut. Pengunjung dapat menikmati keindahan alam yang *spektakuler*, termasuk pemandangan tebing yang eksotis dan sungai yang jernih. Aktivitas yang *populer* di sini termasuk treking atau hiking melintasi jalur-jalur yang mengagumkan, fotografi alam, dan menikmati keindahan alam sekitar. Salah satu objek wisata Enrekang yang menjadikan gunung nona sebagai background andalan untuk berfoto. Letaknya yang berada di puncak gunung menjadikan letak objek ini semakin strategis karena tidak terhalang pohon untuk melihat keindahan deretan pegunungan. Selain itu terdapat banyak spot foto yang unik untuk diabadikan dengan hanya membayar Rp. 10.000 kita dapat menikmati indahnya objek ini. Untuk beberapa spot foto seperti ayunan dan sepeda kita akan dikenakan biaya tambahan.

Adapun juga Dante Pine menjadi tempat wisata yang cukup populer di kalangan masyarakat Enrekang, bahkan acap kali dikunjungi oleh turis mancanegara. Dante Pine berada di Kelurahan Tanete, Kecamatan Anggeraja. Lokasinya pun tak sulit dijangkau, sebab berada di samping Jl Poros penghubung Enrekang - Tana Toraja. Objek wisata ini menawarkan konsep hutan pinus yang

memukau, suhu udaranya yang sangat sejuk, sangat cocok untuk berwisata bersama teman sejawat maupun sanak keluaraga. Adapun fasilitas-fasilitasnya cukup lengkap, mulai tempat duduk, *Caffee,* hingga diiringi musik alam yang menjadikan pengunjung betah bersantai berlama-lama. dapun biaya tiket memasuki kawasan wisata ini, setiap pengunjung hanya dikenakan biaya sebesar Rp 10.000. Untuk diketahui, objek wisata Dante Pine yang dikelola pihak swasta ini mulai berdiri sejak tahun 2016.

Virtul Tour merupakan Simulasi dari suatu lingkungan nyata, biasanya terdiri dari sekumpulan foto-foto panorama, video maupun model 3 dimensi serta dapat menggunakan unsur-unsur multimedia lainnya seperti efek suara, music, narasi maupun tulisan, yang membuat pengguna seakan-akan pengguna berada pada lokasi tersebut. Peneliti berencana merancang sebuah aplikasi Pengenalan wisata Dante Pine kab. Enrekang memanfaatkan teknologi *Virtual tour* dan 3D (dimensi).

Oleh sebab itu peneliti mengusulkan untuk membuat aplikasi yang bisa memudahkan pengunjung dalam mengetahui letak-letak atau tempat-tempat yang di butuhkan oleh pengunjung saat berkunjung Wisata Dante Pine Kab. Enrekang. Dengan ini penulis mengangkat judul "**Aplikasi *Virtual tour* Wisata Dante Pine Kabupaten Enrekang Berbasis *Android***"

B. Rumusan Masalah

Dari penerapan materi diatas dapat dibuatkan rumusan masalah sebagai berikut:

1. Bagaimana merancang objek pada Wisata Dante Pine berbasis 3D ?
2. Bagaimana merancang *Virtual tour* Untuk Wisata Dante Pine kabupaten Enrekang Berbasis *Virtual Reality ?*

C. Batasan Masalah

Mengingat luas cakupan Wisata Dante Pine Kabupaten Enrekang, Maka penulis membatasi beberapa hal sebagai berikut :

1. Aplikasi ini dibuat dengan menggunakan *software Unity* yang berbasis pada android
2. User hanya dapat berkeliling di area Wisata Dante Pine
3. Aplikasi ini berjalan pada perangkat OS *Android*
4. Efek suara yang digunakan hanya berupa suara langkah kaki saat user bergerak di dalam aplikasi.

D. Tujuan Penelitian

Menghasilkan aplikasi visual wisata Dante Pine berbasis 3D dan Virtual Reality yang dapat memberikan informasi pada pengunjung dan dapat dijadikan teknologi pengenalan Wisata Dante Pine Kabupaten Enrekang kepada masyarakat luas.

E. Manfaat Penelitian

Beberapa manfaat Pembuatan Proposal Ini sebagai Berikut :

1. Untuk Penulis

Mengimplementasikan Ilmu yang diperoleh dibangku kuliah terutama mengenai aplikasi yang terlah dibuat serta meningkatkan kemampuan programing penulis sebagai bekal sebelum memasuki dunia kerja.

2. Untuk Akademik

   Memberika sumber acuan (referensi) dan bahan bagi upaya pengembangan aplikasi serta menambah literatur perpustakaan dan bahan pertimbangan yang berhubungan dengan tugas akhir ini

3. Untuk Instansi

   Memberikan Kemudahan Untuk Memperkenalkan Wisata Dante Pine Kabupaten Enrekang

4. Untuk Masyarakat

   Memperkenalkan Wisata Dante Pine Kabupaten Enrekang dengan Visualisasi 3D sehingga masyarakat mampu mengetahui dan mengenal Wisata Dante Kabupaten Enrekang.

F. Sistematika Penelitian

Dalam menyusun sistematika penulisan, penulis menguraikan ke dalam lima bab yaitu :

BAB I : PENDAHULUAN

Menguraikan tentang latar belakang penulisan, rumusan masalah, tujuan penelitian, manfaat penelitian, dan batasan masalah, dan juga sistematika penulisan.

BAB II : TINJAU PUSTAKA

Pada Bab ini berisi uraian - uraian tentang tinjauan penelitian sebelumnya dan teori - teori yang digunakan sebagai referensi dalam Aplikasi Virtual tour Wisata Dante Pine Kab. Enrekang.

BAB III : METODE PENELITIAN

Membahas tentang waktu dan tempat penelitian, metode penelitian, jenis penelitian, alat dan bahan yang digunakan dalam penelitian serta desain sistem.

BAB IV : HASIL DAN PERANCANGAN APLIKASI

Bab ini membahas tentang Analisis Sistem Dengan UML, Perancangan aplikasi,dan Implementasi.

BAB V : PENGUJIAN SISTEM

Bab ini membahas tentang metode pengujian menggunakan White box dan Black box.

BAB VI : PENUTUP

Membahas tentang kesimpulan dan saran

BAB II

TINJAUAN PUSTAKA

A. Kajian Teori

1. *Virtual tour*

Virtual tour merupakan teknologi yang menempatkan user di dalam gambar dan memungkinkan user untuk meningkatkan kesadaran situasional serta meningkatkan daya lihat, tangkap dan menganalisa data *virtual* secara signifikan. *Virtual tour* adalah simulasi dari sebuah lokasi yang sesungguhnya, umumnya terdiri oleh *sequence* video atau kumpulan foto. Virtual tour juga dapat menggunakan beberapa elemen multimedia lain, contohnya seperti *sound effect*, musik, narasi, dan teks, dan sering digunakan untuk menggambarkan varietas dari media yang berbasis video dan *fotografi.Panoramic tour* dan *Virtual tour* sering dianggap memiliki arti yang sama, yaitu sebuah *Virtual tour* yang dibuat dengan menggunakan bantuan kamera, terdiri dari sejumlah foto yang diambil dari satu posisi yang strategis dengan cara kamera dan lensa di putar mengelilingi satu lokasi yang sering disebut dengan istilah *no-parallax point* (titik tepat di belakang lensa dimana cahaya bertemu).

2. Virtual Reality

Virtual reality adalah teknologi yang membuat pengguna dapat berinteraksi dengan suatu lingkungan oleh komputer (*computer simulated enviroment).* Suatu lingkugan yang sebenarnya ditiru atau benar-benar suatu lingkungan hanya ada

didalam imajinasi (Sihite et al., 2013). Konsep *Virtual Reality* merupakan bidang objek tersebut dapat dijelajahi seperti pada dalam dunia aslinya (Kurnia, 2010). Jadi *Virtual Reality* merupakan teknologi yang membuat pengguna dapat berinteraksi dengan lingkungan sekitar merasakan interaksi didunia nyata.

Ada 4 elemen penting dalan *virtual reality* sebagai berikut :

a. *Virtual World,* sebuah konten yang menciptakan dunia virtual dalam bentuk screenplay maupun script.
b. *Immersion,* Sebuah sensasi yang membawa pengguna teknologi virtual reality merasakan ada didalam sebuah lingkungan yang nyata yang padahal fiktif. Immersion dibagi menjadi 3 jenis, yaitu :
    1) *Mental immersion* merupakan membuat mental penggunanya berasa seperti dilingkungan nyata.
    2) *Physical immersion* merupakan membuat fisik penggunanya merasaan suasana disekitar lingkungan yang diciptakan oleh dunia *virtual reality* tersebut.
    3) *Mentaly immersed,* memberikan sebuah sensasi kepada penggunanya berada larut dala lingkungan yang dihasilkan *virtual reality.*
c. *Sensor feedback* berfungsi untuk menyampaikan informasi dari virtual world ke indra penggunanya. Elemen ini merupakan *visual* (penglihatan), audio (Pendengaran) dan sentuhan.
d. *Interactivity* yang berfungsi untuk merespon aksi dari pengguna, sehingga pengguna dapat berinteraksi dalam medan fiktif atau *vitual world.*

3. Virtual Box 3D

Virtual Box 3D adalah semacam kecamaa berbentuk kotak yang digunakan untuk main game berformat 3D serta menikmati video atau film berformat 3D. Akan tetapi *Virtual Box* 3D khusus digunakan untuk perangkat smartphone. Berikut cara kerja *virtual Box* 3D :

a) Untuk lengkap pertama masukkan smartphone pada *Virtual Box* 3D dan tutup kembali *Virtual Box* 3D Tersebut.

b) Langkah Selanjutnya apabila smartphone sudah terpasang pada virtual Box 3D maka dapat langsung digunakan. Perlu diketahui bahwa *Virtual box* 3D terdapat tali yang berfungsi sebagai penyangga *Virtual box* 3D tersebut agar tidak jatuh ketika saat digunakan.



Gambar 2. 1 *Vr Box*

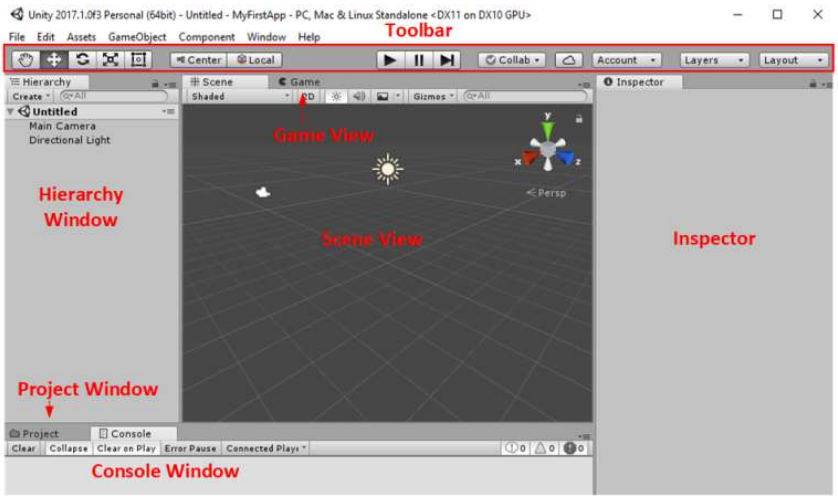
4. Unity

Unity adalah *game egine* buatan *Unity Technologies Inc. Unity* merupakan suatu aplikasi yang digunakan untuk mengembangkan *game multiplatform* yang didesain untuk mudah digunakan. *Unity* itu bagus dan penuh perpaduan dengan aplikasi yang profesional. *Editor* pada *Unity* dibuat degan *user* interface yang

sederhana. *Editor* ini dibuat setelah ribuan jam yang mana telah dihabiskan untuk membuatnya menjadi nomor satu dalam urutan rankking 10 teratas untuk *editor game*. Grafis pada *unity* dibuat dengan grafis tingkat tinggi untuk *Opengl* dan *directx. Unity* mendukung semua format *file*, terutamanya forma umum seperti semua format dari *artapplications. Unity* cocok dengan versi 64 bit dan dapat beroperasi pada Mac OS x dan windows dan dapat menghasilkan game untuk *Mac, Windows, Wii, iPhone, iPad* dan *Android.*

a. Pengenalan *Interface Unity*

Berikut ini adalah tampilan Interface pada *Unity,* diantaranya :



Gambar 2. 2 Bentuk *Interface Unity*

1) Toolbar

Toolbar digunakan untuk memanipulasi *Scene View* dan objek objek di dalamnya. Fungsinya beragam, yakni dari memindahkan posisi, rotasi, hingga skala.

2) Hierarchy Window

Hierarchy window digunakan untuk melihat daftar *Game Object* apa saja yang ada di dalam scene. Setiap objek yang ada di scene pasti masuk di *Hierarchy*

3) *Scene Window*

Scene Window digunakan untuk melihat secara keseluruhan objek yang digunakan di game. *Scenes Window* juga dapat mengedit objek, baik itu dalam hal posisi, rotasi maupun skala.

4) *Game View*

Game View digunakan untuk melihat hasil akhir dari game yang nantinya akan ditampilkan di layar *user*.

5) *Inspector Window*

Inspector Window digunakan untuk meng-edit semua properties pada objek yang aktif.

6) *Project Window*

Project Window digunakan untuk menampilkan daftar asset yang dapat digunakan dalam projek Anda. Ketika menambahkan asset baru, ia akan muncul di *Project Window.*

7) *Console Window*

Digunakan untuk mengetahui *log* dalam s*cript* serta informasi error dan peringatan yang da di dalam script.

5. Skecthup

Sketchup adalah perangkat lunak desain grafis tiga dimensi (3D) yang dikembangkan oleh perusahaan *Trimble Inc*. Perangkat lunak ini dirancang agar

mempermudah proses pembuatan model 3D dengan antarmuka pengguna yang mudah dipelajari. Keunggulan *skechtup* dibangingkan dengan aplikasi perangkat lunak 3D lainnya yaitu :

a. Antarmuka Pengguna *Intuitif* yang dirancang agar mudah dipelajari bagi pemula dan memberikan efesiensi bagi orang profesional.
b. Kecepatan dalam pembuatan *modeling* sebagai salah satu keunggulannya
c. Memiliki berbagai *open resouce plugin* yang memberikan kemudahan dalam pembuatan model 3D
d. Memiliki gudang model 3D atau *3D Warehouse* yang memiliki berbagai macam bentuk model 3D yang terorganisir.

Dengan Sketchup desain menjadi sangat edit *table*, memodifikasi komponen sangat *interaktif* memiliki buan komukasi grafis, yang tidak kalah menariknya adalah *Sketchup* tetap memiliki kemampuan mendesain secara akurat, bahkan mendekati permasalahan pada tingkatan abstrak yang menjadi target desain. Dengan kelebihan tersebut *Sketchup* sangat *efektif* dan efisien dalam mendesain terutama untuk diaplikasikan dalam dunia kerja yang sangat mengutamakan kecepatan proses mewujudkan gagasan kedalam presentasi dengan tanpa mengabaikan totalitas desainnya itu sendiri.

Adapun Tools Sketchup diantaranya sebagai berikut :

Tabel 2. 1 Penjelasan *Tools Sketchup* beserta fungsinya

Tools	Penjelasan
	Select yang berfungsi Sebagai memilih atau memodifikasi objek dan model ketika menggunakan alat-alat lain.

Lanjutan Tabel 2. 1

Tools	Penjelasan
	Make component digunakan untuk menggabungkan objek satu kesatuan.
	Paint bucket digunakan untuk mewarnai atau menyisipkan material pada objek.
	Line digunakan untuk menggambar garis lurus.
	Eraser berfungsi sebagai menghapus garis, gambar atau material.
	Freehand Untuk menggambar bebas.
	Rectangle digunakan sebagai membentuk objek kotak.
	Rotated Rectangle digunakan untuk menggambar persegi empat yang dapat dibuat kesegala arah penggambaran.
	Circel berfungsi membuat gambar lingkaran 2D dengan pengaturan segmen.
	Polygon digunakan sebagai membuat bidang persegi banyak dari segitiga sampai dengan lingkaran.
	Move digunakan untuk memindahkan suatu objek.

Lanjutan Tabel 2. 1

Tools	Penjelasan
	Push&Pull digunakan untuk mendorong atau mengubah objek menjadi 3 dimensi.
	Rotated digunakan untuk memutar objek.
	Follow Me berfunsi agar memudahkakan dalam membuat objek melengkukan atau menyerong
	Offset berfungsi sebagai menduplikasi garis objek yang disesuaikan.
	Scala berfungsi mengubah ukuran besar kecil objek yang di skala kan.
	Tape measure Tool digunakan untuk mengukur/Menggaris pajang atau lebar objek yang akan dibuat
	Orbit berfungsi Untuk memutar pandangan objek.
	Zoom Untuk memperbesar layar ke suatu objek.

6. Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis Linux yang mencakup sistem operasi, *middleware* dan aplikasi. *Android* menyediakan platform yang terbuka bagi para pengembang untuk menciptakan

aplikasi mereka. Awalnya, Google Inc. membeli *Android Inc.* yang merupakan pendatang baru yang membuat peranti lunak untuk *ponsel/smartphone*. Kemudian untuk mengembangkan *Android*, dibutuhkan *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk *Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia*.

a. Arsitektur *Android*

Secara garis besar arsitektur *Android* dapat dijelaskan dan digambarkan sebagai berikut:

1) *Applications* dan *Widgets* adalah *layer* di mana kita berhubungan dengan aplikasi saja, di mana biasanya kita *download* aplikasi kemudian kita lakukan instalasi dan jalankan aplikasi tersebut. Di layer terdapat aplikasi inti termasuk *klien email*, program sms, kalender, peta, *browser,* kontak, dan lain-lain. Semua aplikasi ditulis menggunakan bahasa pemrograman *java*.
2) *Applications Frameworks* adala h *layer* di mana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di sistem operasi *Android,* karena pada *layer* inilah aplikasi dapat dirancang dan dibuat, seperti *content-providers* yang berupa sms dan panggilan telepon.
3) *Libraries, libraries* ini adalah *layer* di mana fitur-fitur *Android* berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya.

4) *Android Run Time*, *layer* yang membuat aplikasi *Android* dapat dijalankan di mana prosesnya menggunakan Implementasi *Linux. Dalvik Virtual Machine (DVM)* merupakan mesin yang membentuk dasar kerangka aplikasi *Android*.
5) *Linux Karnel* adalah *layer* di mana inti dari operating sistem dari *Android* itu berada. Berisi file-file sistem yang mengatur sistem *processing, memory, resource, drivers,* dan sistem-sistem operasi *Android* lainnya.

b. Komponen *android*

Ada enam jenis komponen pada aplikasi *Android,* yaitu:

1) *Activities,* suatu *activities* akan menyajikan *user interface (UI)* kepada pengguna, sehingga pengguna dapat melakukan interaksi untuk menjalankan fungsi tertentu. Sebuah aplikasi *Android* bisa jadi hanya memiliki satu *activity*, tetapi umumnya aplikasi memiliki banyak *Activity* tergantung pada tujuan aplikasi dan desain dari aplikasi tersebut.
2) *Service, Service* tidak memiliki *Graphic User Interface (GUI),* tetapi *service* berjalan secara *background* untuk melakukan operasi-operasi yang longrunning (proses yang memakan waktu cukup lama) atau melakukan operasi untuk proses *remote.*
3) *Broadcast Reciever*, *broadcast reciever* berfungsi menerima dan bereaksi untuk menyiapkan notifikasi. *Broadcast Reciever* tidak memiliki *User Interface (UI)* tapi memiliki sebuah *Activity* untuk

merespon informasi yang mereka terima atau kepada pengguna. *Broadcast receiver* hanyalah pintu gerbang menuju komponen lain dan memang dirancang untuk hanya melakukan kerja seminimal mungkin.

4) *Content Provider*, *content provider* membuat kumpulan aplikasi data secara *spesifik* sehingga bisa digunakan oleh aplikasi lain. *Content providers* juga berguna untuk membaca dan menulis data yang berstatus *private* dan tidak dibagikan ke suatu aplikasi.

7. *Smartphone*

Williams dan Sawyer (2011) Menurutnya definisi *smartphone* adalah telepon selular dengan menggunakan berbagai layanan seperti, memori, layar, *mikroprosesor,* dan m*odem* bawaan. Sehingga fitur yang ada di *smartphone* ini terasa lebih lengkap di bandingkan dengan fitur *henphone* lainnya. Ponsel cerdas (bahasa inggris: *smartphone*) adalah telepon genggam yang mempunyai kemampuan dengan penggunaan dan fungsi yang menyerupai komputer. Belum ada standar pabrik yang menentukan arti ponsel cerdas. Bagi beberapa orang, ponsel cerdas merupakan telepon yang bekerja menggunakan seluruh perangkat lunak sistem operasi yang menyediakan hubungan standar dan mendasar bagi pengembang aplikasi. Bagi yang lainnya, ponsel cerdas hanyalah merupakan sebuah telepon yang menyajikan fitur canggih seperti surel (surat *elektronik), internet* dan kemampuan membaca buku *electronic (e-book*) atau terdapat papan ketik (baik sebagaimana jadi maupun dihubung keluar) dan penyambung *VGA.* Dengan kata lain, ponsel cerdas merupakan komputer kecil yang mempunyai kemampuan sebuah telepon.

Singkatnya, *smartphone* adalah perangkat yang memungkinkan anda melakukan panggilan telepon sekaligus memiliki fitur yang dimasa lalu hanya bisa ditemukan pada perangkat *mobile* yang sesuai dengan jaman sekarang atau komputer seperti kemampuan untuk mengirim dan menerima e-mail dan edit dokumen.

Ponsel terutama digunakan untuk menelpon, sementara perangkat *mobile* yang sesuai dengan jaman sekarang digunakan sebagai semacam asistensi digital pribadi atau digital organizer. Perangkat dapat menyimpan info kontak dan agenda harian serta bisa disinkronisasi dengan komputer. Perkembangan ini lantas mendorong terciptanya apa yang kini dikenal sebagai *smartphone.*

8. JDK (Java Development Kit)

Java Development Kit merupakan perangkat lunak yang digunakan untuk melakukan proses komplikasi dari kode java menjadi bytecode yang dapat dimengerti dan dapat dijalankan oleh *Java Runtime Envirotment*.

Java Development Kit wajib terinstall pada computer yang akan melakukan proses pembuatan aplikasi berbasis java. Namun *Java Development Kit* tidak wajib terinstall dikomputer yang akan menjalankan aplikasi yang dibangun menggunakan *java*.

JDK adalah semacam suatu kotak peralatan (*kit*) yang digunakan untuk *Development. JDK* ini berguna saat anda menulis kode program. Seperti halnya *JRE, JDK* juga memiliki *JVM* didalamnya.

a. *Java* merupakan program berfungsi meng-kompile kode sumbar.

b. *Java* merupakan program yang berfungsi meluncurkan suatu aplikasi.

9. Bahasa Pemrograman C#

Menurut handoyo (2011), C# (*C sharp*) adalah" sebuah bahasa merupakan salah satu tiang utama teknologi. *Net. Anders* mengetahui berbagai macam kekurangan pada bahasa C++, *Delphi, Java,* dan *Smaltalk,* karena itu Anders menciptakan bahasa C# yang lebih tangguh. Hal ini juga menjelaskan mengapa C# memiliki kemiripan dengan beberapa bahasa tersebut.

C# bergantung pada *CLR* yang juga merupakan sumber library bagi program. Net lain. Semua program C# memerlukan *CLR* (berarti juga memerlukan. *Net Framework)* untuk dapat dijalankan. Sama halnya dengan Visual Basic 6 yang 12 memerlukan *runtime library* tertentu untuk dapat dijalankan. Bahasa C# dapat digunakan untuk menciptakan aplikasi *windows, console, clas*s yang dapat digunakan kembali, dan aplikasi *web.*

Pemrograman berbasis objek yang didukung oleh *Microsoft .NET framework*". *Microsoft. NET Framework* adalah perantara agar aplikasi dengan bahasa pemrograman yang didukung dapat berkomunikasi dengan sistem operasi yang digunakan oleh komputer kebanyakan orang. C# merupakan sebuah bahasa pemrograman yang sangat menjanjikan.

C# didasarkan pada bahasa pemrograman C++, C# juga memiliki kemiripan dengan beberapa bahasa pemrograman seperti *Visual Basic, Java, Delphi,* dan tentu saja C++. C# memiliki kemudahan syntax (cara penulisan) seperti *Visual Basic* dan tentu saja ketangguhan seperti Java dan C++. Kemiripan - kemiripan ini tentunya memudahkan *programmer* dari berbagai latar belakang bahasa pemrogrman tidak

perlu waktu yang lama untuk menguasainya, karena bagaimanapun juga C# lebih sederhana dibandingkan bahasa - bahasa pemrograman seperti C++ dan Java.

C# didesain oleh program designer dari *Microsoft* , *Anders Hajlsberg*. Sebelum bekerja pada *Microsoft , Anders* bekerja di *Borland,* tempat dia menulis *Pascal compiler*. Sebelum mengembangkan C# *Anders* pernah mengembangkan J++ untuk *Microsoft* . Setelah itu Anders mengembangkan C# dan *Common Language Runtime* yang merupakan mesin Virtual dan Runtime library.

Ada beberapa alasan kuat yang mendasari pemilihan bahasa C# untuk mengembangkan aplikasi - aplikasi yaitu:

a. C# benar - benar beriorantasi objek

   C# adalah bahasa yang benar-benar *objek oriented*. Ini dapat dilihat dari kemampuan C# dalam membentuk objek, *class*, melakukan *encapsulation, inheritance* dan *poymorphsm* dengan mudah.

b. C# sangat sederhana

   Bahasa C# bersifat sederhana karena didasarkan pada bahasa C dan C++ bahkan bahasa Java. Tetapi C# lebih sederhana dari bahasa-bahasa tersebut karena C# dibuat dengan menghilangkan kelemahan-kelemahan dari bahasa-bahasa yang mendasarinya.

c. Mampu membuat berbagai aplikasi

   Dengan C# kita dapat membuat berbagai macam aplikasi, mulai dari aplikasi console, pengolahan kata*, form web,* dan lainnya.

d. *Efisien*

Bahasa C# merupakan bahasa yang mengandalkan *library* yang sangat lengkap, karena itu bahasa C# hanya memiliki sedikit *keywords*. Jadi para pengembang dapat mengingatnya dan memahami kegunaanya dengan baik.

e. C# bersifat modular

Bahasa C# sangat modular, tiap *class* disimpan dalam namespace yang dapat dimanfaatkan kembali oleh program lain yang membutuhkannya.

10. Camera insta360 X4

Kamera Insta360 adalah jenis kamera digital yang menggunakan dua atau lebih lensa ultra-*wide* untuk menangkap gambar dan video dengan sudut pandang 360 derajat. Setiap lensa merekam area yang luas, kemudian sistem perangkat lunak menggabungkan (*stitching)* hasil rekaman tersebut menjadi satu kesatuan tampilan imersif. Teknologi ini memungkinkan pengguna melihat hasil tangkapan dari segala arah tanpa batas sudut pandang. *Kamera Insta360* umumnya dilengkapi fitur perekaman resolusi tinggi, stabilisasi berbasis *giroskop,* serta dukungan pengeditan otomatis melalui aplikasi resmi, sehingga banyak digunakan dalam pembuatan konten virtual tour, virtual reality (VR), dan dokumentasi kegiatan outdoor atau olahraga ekstrem**.**

11. UML (*Unified modelling language*)

Kroenke et al (2018), *Unified modelling language (UML*) adalah seperangkat diagram, struktur, dan teknik untuk memodelkan dan merancang program dan aplikasi berorientasi objek. UML menawarkan sebuah standar untuk merancang model sebuah sistem. UML adalah sekumpulan alat yang digunakan untuk melakukan abstraksi terhadap sebuah sistem atau perangkat lunak berbasis

objek. UML merupakan singkatan dari *Unified Modeling Language*. UML juga menjadi salah satu cara untuk mempermudah pengembangan aplikasi yang berkelanjutan.

Aplikasi atau sistem yang tidak terdokumentasi biasanya dapat menghambat pengembangan karena *developer* harus melakukan penelusuran dan mempelajari kodeprogram UML juga dapat menjadi alat bantu untuk *transfer* ilmu tentang sistem atau aplikasi yang akan dikembangkan dari satu *developer* ke *developer* lainya. Tidak hanya antar *developer* terhadap orang bisnis dan siapapun dapat memahami sebuah sistem dengan adanya UML.

UML diciptakan oleh *Object Management Group* yang diawali dengan versi 1.0 pada Januari 1997. Dalam pengembangan berorientasi objek ada beberapa prinsip yang harus dikenal: *Object, Class, Abstraction, Encapsulation, Inheritance* dan *Polymorphism.*

a. *Use case*

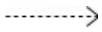
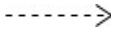
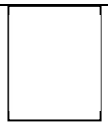
Diagram *use case* menyajikan intraksi antara use case dan aktor, dimana aktor dapat berupa orang, peralatan, atau sistem lain yang berintraksi dengan sistem yang sedang di bangun (*Sholiq.2006*).

Adapun simbol-simbol *use case* Diagram antara lain :

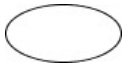
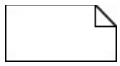
Tabel 2. 2 *Use Case*

NO	GAMBAR	NAMA	KETERANGAN
1		*Actor*	Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.

Lanjutan Tabel 2. 2

NO	GAMBAR	NAMA	KETERANGAN
2		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent*) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*).
3		*Generalization*	Hubungan dimana objek anak *(descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk *(ancestor).*
4		*Include*	Menspesifikasikan bahwa use case sumber secara *eksplisit.*
5		*Extend*	Menspesifikasikan 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		*Sistem*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.

Lanjutan Tabel 2. 2

NO	GAMBAR	NAMA	KETERANGAN
8		Use Case	*Deskripsi* dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
9		*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

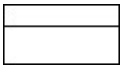
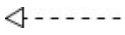
b. *Class* diagram

Class diagram adalah *visual* dari struktur sistem program pada jenis-jenis yang di bentuk. Class diagram merupakan alur jalannya database pada sebuah sistem. Class diagram merupakan penjelasan proses *database* dalam suatu program. Dalam sebuah laporan sistem maka class diagram ini wajib ada.

Tabel 2. 3 Simbol *Class* Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		*Generalization*	Hubungan dimana objek anak *(descendent*) berbagi perilaku dan

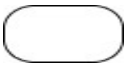
Lanjutan Tabel 2. 3

NO	GAMBAR	NAMA	KETERANGAN
			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-banar dilakukan oleh suatu *object.*
6		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
7		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.

c. Activity diagram

Activity diagram adalah sesuatu yang menjelaskan tentang alir kegiatan dalam program yang sedang dirancang, bagaimana proses alir berawal, keputusan yang mungkin terjadi, dan bagaimana sistem akan berakhir.

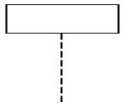
Tabel 2. 4 Simbol *Activity* Diagram

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.
4		*Actifity Final Node*	Bagaimana objek dibentuk dan dihancurkan
5		*Fork Node*	Satu aliran yang berubah menjadi beberapa aliran.

d. *Sequence* Diagram

Sequence Diagram adalah suatu diagram yang menjelaskan interaksi objek dan menunjukkan (memberi tanda atau petunjuk) komunikasi diantara objek-objek tersebut.

Tabel 2. 5 Simbol *Sequence* Diagram

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

12. *Flowchart*

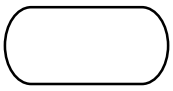
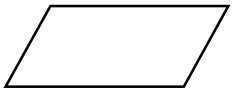
Menurut Hastuty, A. (2021), Flowchart adalah bagan yang menampilkan langkah-langkah prosedur program dalam bentuk simbol-simbol standar. Tujuannya agar alur proses mudah dipahami baik oleh programmer maupun orang awam.

Flowchart (diagram alir) dapat digunakan sebagai alternatif untuk menyajikan algoritma. *Flowchart* adalah bentuk penyajian grafis yang menggambarkan solusi langkah demi langkah terhadap suatu masalah.

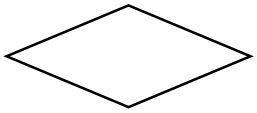
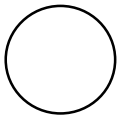
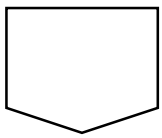
Ada lima macam bagan alir, di antaranya:

a. Bagan Alir Sistem *(system flowchart)* merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem.
b. Bagan Alir Dokumen *(document flowchart)* disebut juga bagan alir formulir *(form flowchart)* merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusannya.
c. Bagan Alir Skematik *(schematic flowchart)* merupakan bagan alir yang menggambarkan prosedur di dalam sistem dengan menggunakan simbol-simbol bagan alir sistem dan gambar-gambar komputer serta peralatan lainnya yang digunakan oleh sistem.
d. Bagan Alir Program *(program flowchart)* merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program.
e. Bagan Alir Proses *(process flowchart)* merupakan bagan alir yang banyak digunakan di teknik industri untuk menggambarkan proses dalam suatu prosedur.

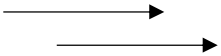
Tabel 2. 6 Tabel *Actor* Pengguna

NO	GAMBAR	NAMA	KETERANGAN
1		Terminal	Menyatakan permulaan atau akhir suatu program
2		*Input/Output*	Menyatakan proses input atau output tanpa tergantung jenis peralatanya

Lanjutan Tabel 2.6

NO	GAMBAR	NAMA	KETERANGAN
3		*Process*	Menyatakan suatu Tindakan (Proses) yang dilakukan oleh computer
4		*Decision*	Menunjukan suatu kondisi tertentu yang akan menghasilkan dua kemunkinan jawaban:ya / tidak
5		*Connector*	Menyatakan sambungan dari proses ke proses lainya dalam halaman yang sama
6		*Offline Connector*	Menyatakan sambungan dari proses ke proses lainya dalam halaman yang berbeda
7		*Prefefned Process*	Menyatakan penyediaan tempat penyimpanan suatu pengolaan untuk memberi harga awal

Lanjutan Tabel 2.6

NO	GAMBAR	NAMA	KETERANGAN
8		*Punched card*	Menyatakan input berasal dari kartu atau output ditulis ke kartu
9		*Document*	Mencetak keluaran dalam bentuk dokumen (Melalui printer)
10		*Flow*	Menyatakan jalannya arus suatu proses

13. *Flowgraph*

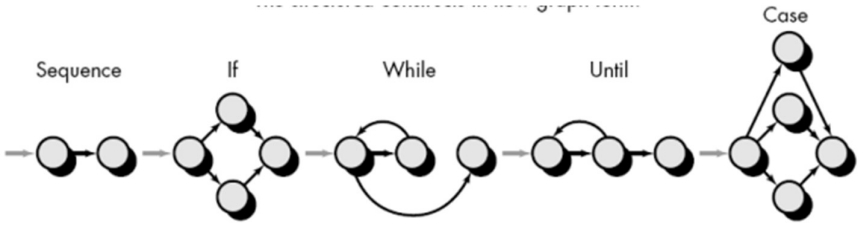
Menurut Hastuty (2021: 22), flowgraph adalah grafik yang digunakan untuk menggambarkan alur kontrol dari sebuah program. *Flowgraph* berbeda dengan f*lowchart*:

a. *Flowchart* dipakai pada tahap perancangan untuk menggambarkan logika penyelesaian masalah secara detail.

b. *Flowgraph* digunakan pada tahap pengujian program (khususnya *white box testing),* karena *fokus* pada aliran kontrol program, bukan *detail proses*.

Jenis notasi pada *flowchart* digambarkan secara berbeda (*diamond,* persegi panjang, jajar genjang, dst) untuk menggambarkan proses yang berbeda, sedangkan notasi pada *flowgraph* hanya diwakili oleh sebuah notasi lingkaran. Dari penggunaannya, *flowchart* digunakan pada tahapan perancangan untuk

menggambarkan logika dari program sedangkan flow graph digunakan pada tahapan pengujian yang berfokus pada penggambaran aliran kontrol sebuah program. Berikut ini adalah notasi struktur kontrol pada flow graph untuk menggambarkan sekuensial, seleksi, maupun perulangan:



Gambar 2. 3 Notasi struktur kontrol pada *flowgraph*

Notasi lingkaran disebut sebagai f*lowgraph* node yang digunakan untuk menggambarkan statementstatement berikut: 1. Satu atau lebih statement secara sekuensial yang dikelompokkan 2. Percabangan seleksi dari satu statement kedua pilihan statement (seleksi) 3. Penggabungan dua statement yang dilanjutkan pada satu statement yang sama (merge) Sedangkan notasi garis panah disebut sebagai edge atau link, menggambarkan aliran kontrol. Setiap *edge* harus dihubungkan dari/kesebuah node, meskipun node tersebut tidak mewakili sebuah statement khusus. Area yang dibatasi oleh node dan edge disebut sebagai region. Secara sederhana*, flow graph* dapat dibuat dari grafik *flowchart* ataupun dari *pseudocode/program design language/source code* yang telah dibuat sebelumnya.

A. Metode Pengujian

Beberapa *test-case* harus dilaksanakan dengan beberapa perbedaan *strategi, query,* atau jalur navigasi yang mewakili penggunaan sistem yang tipical, kritis atau abnormal. Isu kunci pada pengembangan sistem adalah pemilihan sekelompok *test-*

case yang cocok, sekecil dan secepat mungkin, untuk meyakinkan perilaku sistem secara detail. Pengujian harus mencakup unit *testing*, yang mengecek validasi dari prosedur dan fungsi-fungsi secara independen dari komponen sistem yang lain. Kemudian *modul testing* harus menyusul dilakukan untuk mengetahui apakah penggabungan beberapa unit dalam satu modul sudah berjalan dengan baik, termasuk eksekusi dari beberapa modul yang saling berelasi, apakah sudah berjalan sesuai karakteristik sistem yang diinginkan.

Jika struktur kendali antar modul sudah terbukti bagus, maka pengujian yang tak kalah pentingnya adalah pengujian unit. Pengujian unit digunakan untuk menguji setiap modul untuk menjamin setiap modul menjalankan fungsinya dengan baik. Ada 2 metode untuk melakukan unit testing, yaitu:

a. White box Testing

White box Testing merupakan salah satu cara untuk menguji suatu aplikasi atau *software* dengan cara melihat modul untuk dapat meneliti dan menganalisa kode dari program yang dibuat ada yang salah atau tidak. Apabila modul ini dan telah diproduksi dalam output yang tidak memenuhi persyaratan, kode akan dikompilasi ulang dan diperiksa lagi sampai mencapai apa yang diharapkan. Singkatnya *White box Testing* ini menguji dengan cara melihat *pure code* dari suatu aplikasi atau *software* yang diuji tanpa memperdulikan tampilan atau UI dari aplikasi tersebut.

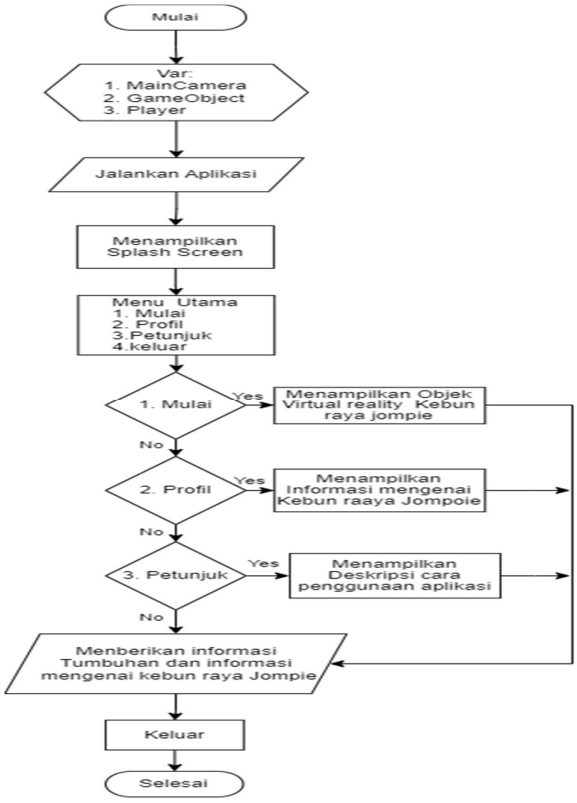
Untuk membuat *white box testing*, maka tester harus memiliki kompetensi dalam pemrograman, untuk memahami kode sumber yang ia pelajari, juga harus memiliki pandangan global tentang fungsi aplikasi, elemen-elemen yang dibuatnya,

dan tentu saja, kode sumbernya. Tidak seperti dalam pengujian “kotak hitam”, tester memiliki profil “pengembang”, bukan profil “pengguna”.

Dengan membuat tes *white box testing*, tester dapat melihat baris kode mana yang dipanggil untuk setiap fungsi. Hal ini memungkinkan untuk menguji aliran data, dan penanganan pengecualian dan kesalahan. Ketergantungan sumber daya, serta logika dan kebenaran internal kode juga dipelajari. Itulah sebabnya tes ini terutama berguna selama pengembangan aplikasi, bahkan jika itu dapat dilakukan selama beberapa fase kehidupan proyek. Metode *white box testing* dapat diterapkan untuk (terutama) tes unit, tes integrasi, dan tes sistem.

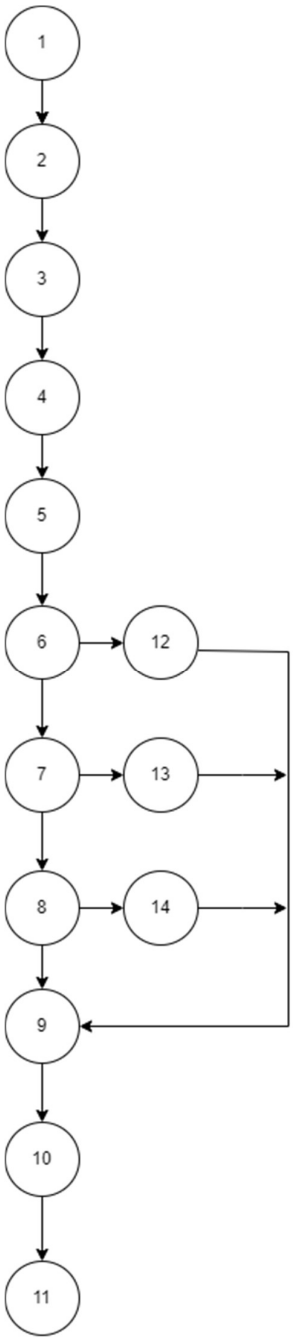
Contoh pengujian *Whitebox*

a. *Flowchart*



Gambar 2. 4 Contoh *Flowchart*

b. Flowgraph



Gambar 2. 5 Contoh *Flowgraph*

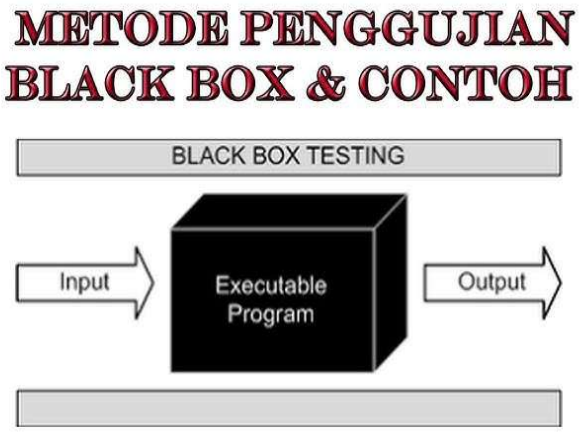
c. Grafik Matriks

Tabel 2. 7 *Grafik Matriks*

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	E–1
1		1													1–1=0
2			1												1–1=0
3				1											1–1=0
4					1										1–1=0
5						1									1–1=0
6							1				1				2–1=1
7								1					1		2–1=1
8									1					1	2–1=1
9										1					1–1=0
10											1				1–1=0
11															0
12									1						1–1=0
13											1				1–1=0
14									1						1–1=0
SUM (E+1)															3 + 1 = 4

b. Black box Testing

Berdasarkan hasil uji coba yang dilakukan, seluruh navigasi dan tombol fasilitas program lainnya serta proses yang di jalankan tidak terjadi kesalahan, tetapi aplikasi mempunyai aturan-aturan yang sudah di tetapkan dan harus di ikuti karena apabila di hiraukan maka sistem akan menolak perintah yang tidak sesuai seperti kesalahan ketika *user* belum menginput data yang harusnya di input sesuai ketentuan sistem yang di jalankan dan sistem memberikan informasi kepada *user* karena data yang ingin diproses belum lengkap atau tidak memenuhi ketentuan untuk proses selanjutnya. Pengujian terhadap cara kerja perangkat lunak itu sendiri yaitu prosedur programnya *(basis path)* atau proses *looping* (pengulangan) yang berfokus pada efektifitas aplikasi yang dirancang.

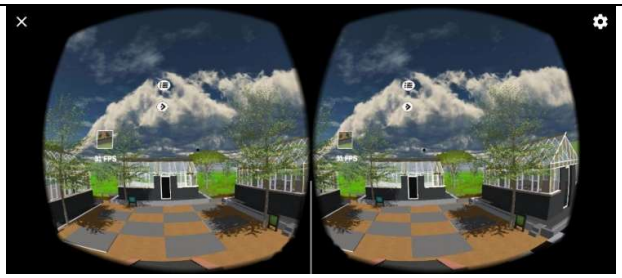


Gambar 2. 6 *BlackBox testing*

Tabel 2. 8 Contoh Pengujian *Blackbox*

Test Faktor	Hasil	Keterangan
Tampilan *Vitual reality*	✓	Berhasil menampilkan halaman Rumah kaca pada kebun raya jompie yang mana pengguna seolah-olah berada dalam kebun raya jompie.

Lanjutan Tabel 2. 8

Screen Shoot


B. Kerangka Pikir

Dante Pine berada di Kelurahan Tanete, Kecamatan Anggeraja. Lokasinya pun tak sulit dijangkau, sebab berada di samping Jl Poros penghubung Enrekang - Tana Toraja. Dante Pine menjadi tempat wisata yang cukup populer di kalangan masyarakat Enrekang, bahkan acap kali dikunjungi oleh turis mancanegara. Objek wisata ini menawarkan konsep hutan pinus yang memukau, suhu udaranya yang sangat sejuk, sangat cocok untuk berwisata bersama teman sejawat maupun sanak keluaraga.

↓

Aplikasi teknologi *Virtual Reality* mensimulasi objek atau seluruh lingkungan yang memungkinkan pengguna merasakan berada pada dunia nyata terhadap objek dan informasi *Virtual* tiga dimensi (3D).

↓

Teknologi *Virtual tour* Wisata Dante Pine Kabupaten Enrekang Dibuat menggunakan Unity sebagai platfrom Pembuatan *Virtual tour* dan Aplikasi *Blender* dan *sketchup* sebagai platform pembuatan model 3D.

↓

sebuah Aplikasi *Virtual tour* Wisata Dante Pine Kab. Enrekang sebagai ajang promosi salah satu wisata di Kab. Enrekang

C. Tinjauan Penelitian Terdahulu

1. Hasil Penelitian yang pernah dilakukan oleh **Eri Satria, Ayu Latifah, 1. Muhamad ParojiInstitut Teknologi Garut, dengan judul Penelitian " Rancang Bangun Aplikasi Katalog Wisata di Garut Menggunakan Teknologi Virtual Reality".** Metode Penelitian yang di gunakan dalam kebutuhan penelitian ini adalah Multimedia Development life cycle (MDLC). Adalah Kumpulan foto dan Video panorama lokasi pendakian dan teks informasi dan Aplikasi Virtual Reality Video dan Foto 230 di mana setiap titik lokasi saling terhubung. Aplikasi ini juga berisi teks dan tombol untuk menampilkan informasi dan berpindah dari suatu titik lokasi. Adapun persamaannya yaitu informasi yang di tampilkan secara visual kepada pengguna agar bisa berintraksi dan seakan-akan pengguna berada di tempat tersebut.
2. **Alfarabi, Hastuty, Yunus (2024)" " Aplikasi Virtual Tour Interaktif Kebun Raya Jompie Berbasis VR"** berfokus pada pengembangan aplikasi virtual tour interaktif di Kebun Raya Jompie, yang bertujuan untuk memberikan sarana edukasi dan eksplorasi lingkungan kebun raya secara virtual. Penelitian tersebut menekankan aspek interaktivitas pengguna dalam mengenal koleksi tanaman dan kawasan Kebun Raya Jompie melalui teknologi Virtual Reality (VR). Sementara itu, penelitian Anda mengembangkan aplikasi virtual tour wisata Dante Pine Kabupaten Enrekang berbasis Android dengan tujuan utama sebagai media promosi pariwisata yang dapat memperkenalkan potensi wisata daerah kepada masyarakat luas maupun wisatawan. Dari segi metode, penelitian Alfarabi dkk. lebih menekankan pada media pembelajaran interaktif berbasis

VR, sedangkan penelitian Anda menggunakan pendekatan Research and Development (R&D) dengan model waterfall yang dilengkapi pengujian white box dan black box untuk memastikan kualitas aplikasi. Dengan demikian, perbedaan mendasar terletak pada objek penelitian (Kebun Raya Jompie vs wisata Dante Pine), tujuan penelitian (edukasi lingkungan vs promosi pariwisata), serta kontribusi (penyediaan media edukasi berbasis VR vs penyediaan media promosi digital pariwisata berbasis Android).

3. **Ade Hastuty, (2025)“ Hacking the Future of College Administration: Process Transformation through AR/VR Immersive Technology”.** Penelitian ini menitikberatkan pada pemanfaatan teknologi **AR/VR** untuk **transformasi administrasi perguruan tinggi**, dengan tujuan meningkatkan efisiensi dan modernisasi layanan akademik. Sementara itu, penelitian saya hanya berfokus pada **pengembangan aplikasi Virtual Tour wisata Dante Pine berbasis Android** menggunakan teknologi **360° dan VR** untuk mendukung **promosi pariwisata daerah**. Dari segi metode, Ade Hastuty menggunakan pendekatan analisis transformasi proses bisnis pendidikan, sedangkan saya menerapkan metode **Research and Development (R&D)** dengan model waterfall. Perbedaan utama terletak pada **bidang penerapan** (administrasi pendidikan vs pariwisata), **tujuan** (efisiensi layanan kampus vs promosi wisata), serta **kontribusi** (inovasi manajemen kampus vs media promosi digital destinasi wisata).

BAB III

METODE 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 Dan Tempat

1. Lokasi

Lokasi penelitian dilakukan pada Wisata Dante Pine Kab. Enrekang. Lokasi ini dipilih dengan pertimbangan kemudahan dalam mendapatkan data dan informasi yang akurat.

2. Waktu Penelitian

Waktu yang dibutuhkan dalam palaksanaan penelitian ini adalah ± 5 bulan.

C. Metode 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 pustaka atau *artikel* yang berasasl dari media *internet.*

2. Secara *Langsung (Observasi)*

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

D. Alat Dan Bahan Penelitian

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

1. Alat

Alat yang digunakan selama proses penelitian yaitu :

a) Laptop Hp dengan Spesifikasi :

Tabel 3. 1 Spesifikasi Laptop Hp

Spesifikasi	
Processor	Intel Core i3-1115G4
RAM	4 GB
Solid State Drive	512 GB PCIe NVMe M.2
Monitor	14.0 inci
Sistem Operasi	Windows 11 64-bit

b) Kamera Instax360 X4 dengan Spesifikasi :

Tabel 3. 2 Spesifikasi Kamera Intax360 X4

Spesifikasi	
Kamera	Dual-lens 1/2″ CMOS, aperture f/1.9, sudut pandang 170°
Foto	72 MP (360°), 18 MP (single lens)
Video	8K@30fps,5.7K@60fps, 4K@100fps (360°); 4K@60fps, 2.7K@120fps (single lens)
Stabilisasi	FlowState Stabilization, 360° Horizon Lock
Layar	Sentuh 2,5″ Gorilla Glass
Baterai	2.290 mAh

c) *Smartphone Samsung* a12 dengan spesifikasi:

Tabel 3. 3 Spesifikasi *Smartpone*

Spesifikasi	
Processor	*MediaTek Helio P35*
RAM	*4 GB*
Layar	*PLS TFT LCD*
Baterai	5000 mAh
Kamera Depan	8 MP
Kamera Belakang	48 MP
Android	10 MIUI

2. Bahan

Software yang digunakan dalam pembuatan aplikasi yaitu :

Tabel 3. 4 Bahan

Bahan
Windows 10
Unity
Blender
Sketchup

E. Tahapan Penelitian

Tahapan penelitian merupakan bagian dari proses yang mendeskripsikan suatu analisa pada proses penelitian. Berikut merupakan proses tahapan penelitian.

1. Identifikasi Masalah

Tahapan awal dari suatu penelitian yaitu mengidentifikasi sebuah masalah seperti bagaimana membuat aplikasi virtual tour dengan tampilan virtual reality pada Wisata Dante Pine Kabupaten Enrekang agar menarik minat wisatawan untuk berkunjung ke Wisata Dante Pine Kabupaten Enrekang.

2. Pengumpulan Data dan Studi Kepustakaan

Mencari sebuah data dan informasi dan mengumpulkannya data tersebut melalui jurnal, artikel, *internet* dan penelitiam secara lansung.

3. Perancangan

Tahap perncangan aplikasi yang dibangun, dengan menganalisa kebutuhan berdasarkan skenario kebutuhan pengguna terhadap aplikasi dan bagaimana itu semua dapat merespon interaksi dari pengguna.

4. Desain Bangunan menggunakan aplikasi pendukung

Desain objek bangunan ataupun tumbuhan merupakan proses pembutan objek 3D (Tiga Dimensi) yang terdapat dalam Wisata Dante Pine Kabupaten Enrekang.

5. Pembangunan Aplikasi Menggunakan *Unity*

a. Penyusunan Objek

Pada tahap ini objek 3D disusun sedemikian rupa sehingga menyerupai Wisata Dante Pine Kabupaten Enrekang.

b. Coding

Tahap ini memberikan *Scrip* atau coding pada sebuah objek yang telah dimasukkan kedalam *unity* dengan menggunakan *MS Visual Studio* atau menggunakan *Notepade++*.

c. Render

Pada Tahap ini *Render* merupakan Proses terakhir dalam pembangunan aplikasi, saat merender semua data-data mulai dari objek. Model, *texture*, pencahayaan dan parameter akan dikonfersikan menjadi sebuah output atau aplikasi.

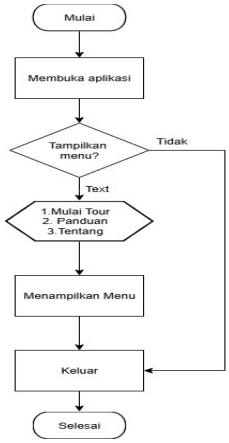
6. Pengujian

Tahap pengujian dibuat agar aplikasi dapat berjalan dengan semestinya. Jika saat tahap pengujian aplikasi mengalami masalah atau kendala, maka aplikasi akan kembali ke tahap perancangan. Untuk menganalisa masalah yang terjadi pada proses tersebut.

F. Desain sistem

1. Sistem yang diusulkan

Adapun yang diusulkan yaitu :



Gambar 3. 1 Desain Sistem yang diusulkan

Sistem yang diusulkan user membuka aplikasi kemudian pada saat aplikasi berjalan maka akan menampilkan menu utama, pada menu utama akan menampilkan 4 button yaitu 1) Mulai untuk memulai *virtual tour* 3D, 2) Panduan

untuk menampilkan informasi mengenai panduan aplikasi virtual tour, 3) Tentang untuk menampilkan informasi pengembang aplikasi, 4) Keluar,apabila user telah selesai menggunakan aplikasi maka memilih tombol keluar.

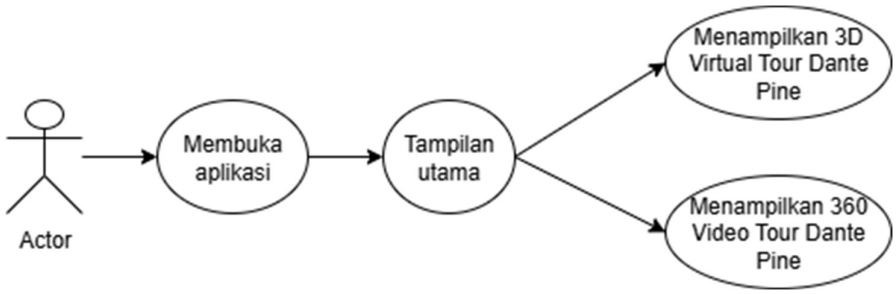
BAB IV

HASIL DAN PEMBAHASAN

A. Rancangan Sistem

Sistem yang penulis buat adalah dengan *metode* pengunjung mengambil alat berupa smartphone berbasis *android* untuk selanjutnya menggunakannya sebagai Alat untuk memunculkan furniture dalam bentuk 3D dan animasi.

1. *Use Case* Diagram



Gambar 4. 1 *Use Case* Diagram

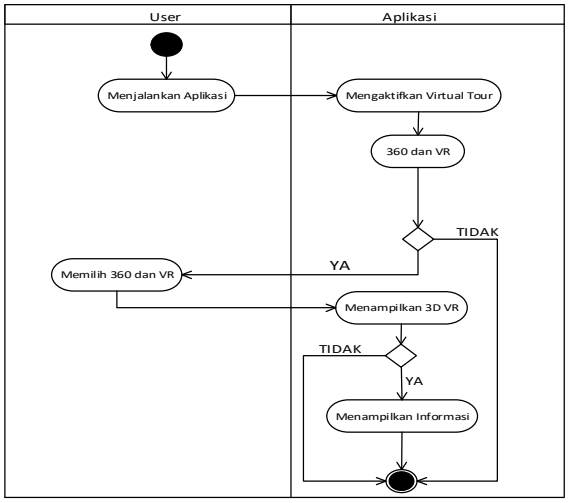
Penjelasan *Use Case* Diagram di atas :

a. *Actor* Pengguna

Gambar 4. 1 yaitu Pengguna membuka aplikasi untuk memulai *Virtual Tour* wisata dante pine, pada tampilan utama terdapat menu Mulai, panduan, Tentang, dan Keluar. Ada pun Menampilkan 3D Virtual Tour Dante Pine yaitu untuk melihat tampilan 3D dari wisata dante pine sedangkan Menampilkan 360 Video *Virtual Tour* dante pine yaitu untuk menampilkan video sekeliling dante pine yang berupa video 360.

2. *Activity* Diagram

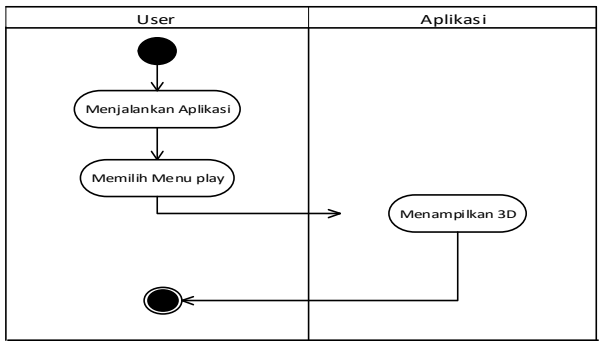
a. *Activity* Diagram Aplikasi



Gambar 4. 2 Diagram *Activity* Aplikasi

Activity Diagram di atas menjelaskan bahwa ketika *user* menjalankan Aplikasi ini akan langsung di arahkan untuk memilih pilihan menu yang di dalamnya juga sudah tersedia menu aktifkan VR dan tombol *Menu VR*, apabila *user* memilih menu akan menampilkan 3D Tour Wisata Dante dalam bentuk 3D, jika terkendala koneksi Aplikasi akan keluar, dan jika Virtual tour tidak terdeteksi dengan baik, informasi yang di inginkan tidak akan tampil.

b. *Actifity* Diagram Menu *play*

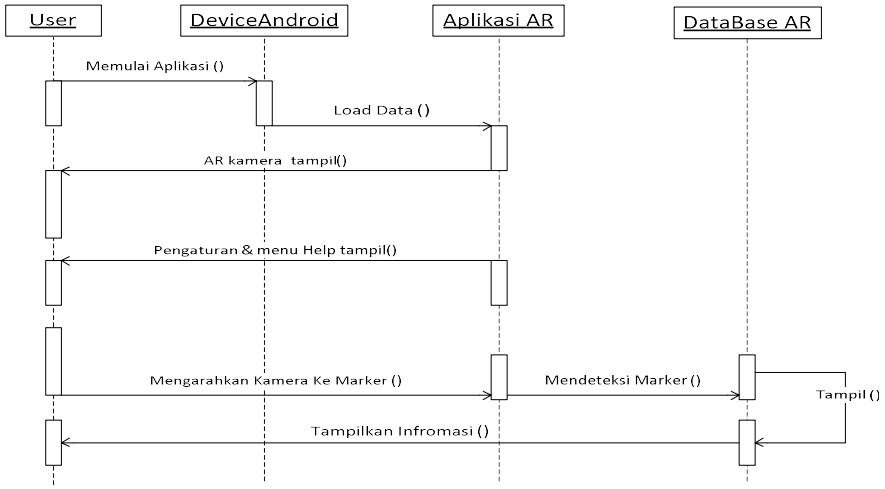


Gambar 4. 3 Diagram *Activity Menu play*

Aktifity Diagram di atas menjelaskan bahwa jika *user* menjalankan Aplikasi ini di bagian sudut atas aplikasi terdapat tombol play yang apabila di tekan akan tampil sebuah *PopUp* yang berisikan proses Memilih kamera yang diinginkan.

3. *Sequence* Diagram

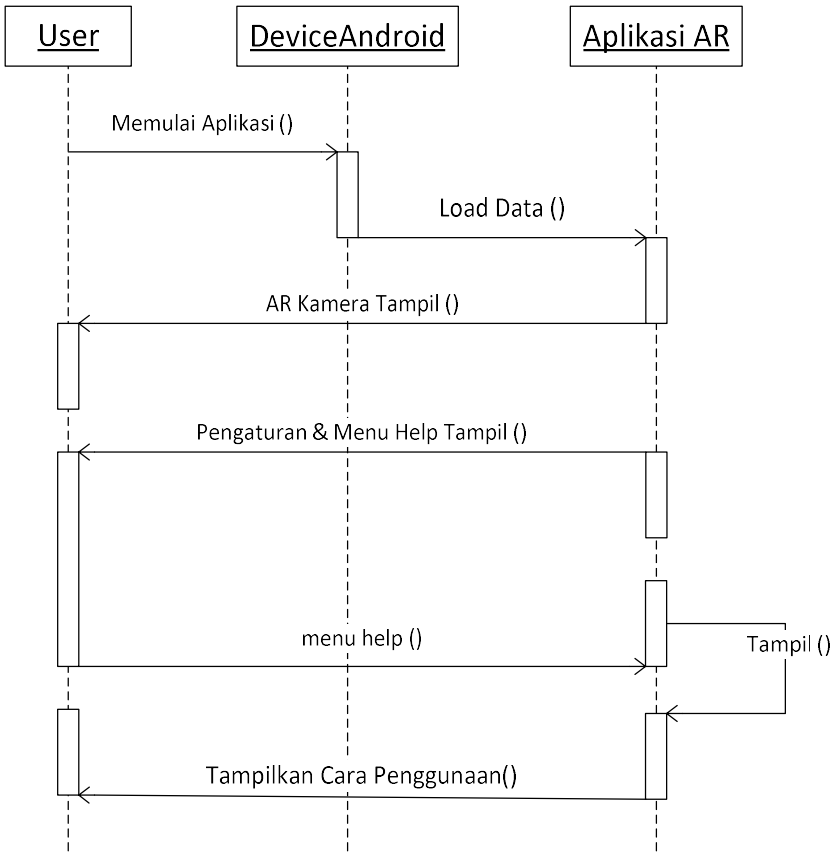
a. *Sequence* Diagram Aplikasi



Gambar 4. 4 Diagram *Sequence*

Diagram *Sequence* di atas menjelaskan bahwa ketika *user* memulai Aplikasi maka device Android yang di gunakan akan meload data untuk selanjutnya memulai aplikasi ini, setelah itu akan tampil Virtual tour dan Menu pengaturan serta tombol, setelah Virtual tour tampil *user* akan mengarahkan device android tersebut ke *Virtual tour* yang telah di sediakan yang kemudian aplikasi melakukan proses pendeteksian *Virtual tour* untuk selanjutnya melakukan pencocokan dengan database jika sesuai dengan data base informasi dari objek akan tampil wisata dalam bentuk 3D.

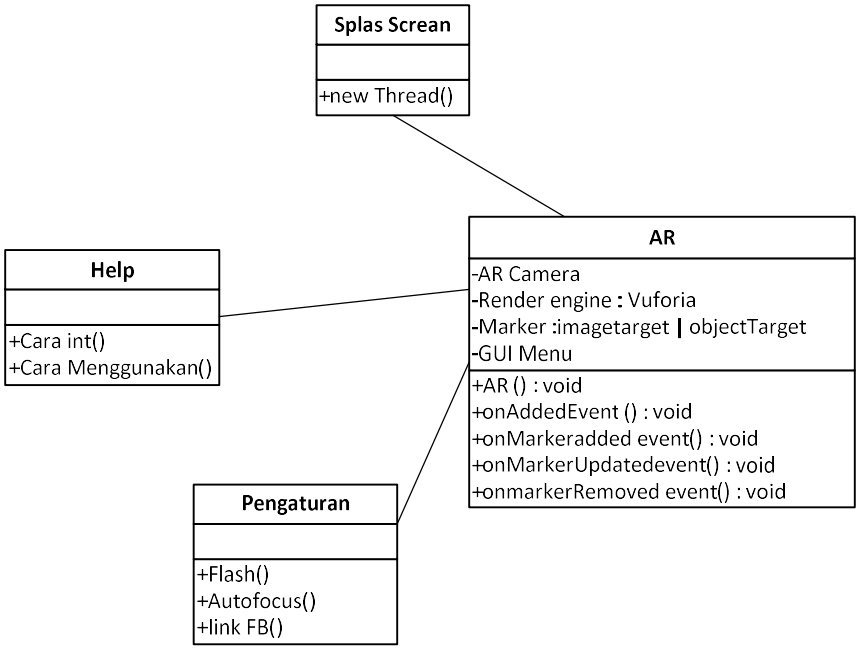
b. *Sequence* Diagram Menu *Help*



Gambar 4. 5 Diagram Menu *Help*

Diagram *Sequence* di atas menjelaskan bahwa ketika *user* menjalankan Aplikasi akan tampil Virtual tour di sertai dengan menu Panduan dan tombol *Tentang*, ketika *user* menekan tombol *tentang* Aplikasi Akan menampilkan *PopUp* berupa Cara penggunaan dari Aplikasi ini.

4. *Class* Diagram

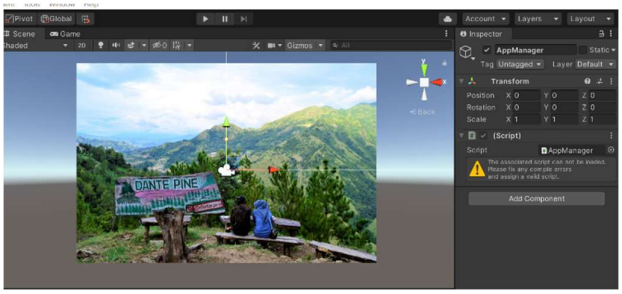


Gambar 4. 6 Diagram *Class*

Class Diagram di atas menjelaskan keterkaitan antara setiap *Form* dan fungsi-fungsi yang ada di dalamnya, seperti ketika Aplikasi di jalankan pertama kali maka yang akan tampil pertama kali adalah munculnya sebuah splashcren sebelum masuk ke *Form* VR yang merupakan inti dari aplikasi ini di dalamnya proses menampilkan wisata dalam bentuk 3d terjadi, adapun jika kita menekan tombol *Help* maka akan di arahkan ke *Form PopUp* yang berisi Cara penggunaan aplikasi ini secara umum.

B. Pembuatan Aplikasi

1. *AppManager*



Gambar 4. 7 AppManager

Script AppManager berfungsi sebagai pengendali utama untuk sistem menu dan navigasi dalam aplikasi Unity. Script ini dirancang khusus untuk mengelola antarmuka pengguna yang kompleks dengan beberapa panel dan transisi antar scene yang berbeda. Pada saat aplikasi dimulai, script ini secara otomatis mengaktifkan panel utama sambil menonaktifkan panel tutorial dan about, kemudian memulai serangkaian animasi UI yang menarik untuk memberikan pengalaman pengguna yang lebih engaging.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
using DG.Tweening;
public class AppManager : MonoBehaviour
{
    public GameObject MainPanel, TutorialPanel, AboutPanel;
    public List<GameObject> UIAnimation = new List<GameObject>();
    public float fadeTime = 1f;
he first frame update
    void Start()
```

```
    // Start is called before the first frame update
    void Start()
    {
        MainPanel.SetActive(true);
        AboutPanel.SetActive(false);
        TutorialPanel.SetActive(false);

        StartCoroutine(ListAnimation());
    }
    IEnumerator ListAnimation()
    {
        foreach (var item in UIAnimation)
        {
            item.transform.localScale = Vector3.zero;
        }
        foreach (var item in UIAnimation)
        {
            item.transform.DOScale(1f,
fadeTime).SetEase(Ease.OutBounce);
            yield return new WaitForSeconds(0.25f);
        }
    }
    public void MenuPilihan()
    {
        SceneManager.LoadScene("PilihanMenu");
    }

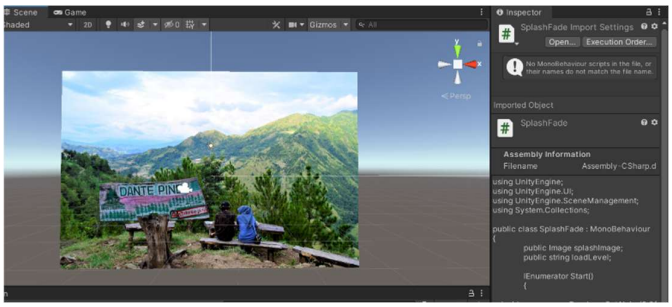
    public void Menu3dVirtualTour()
    {
        SceneManager.LoadScene("3D Virtual Tour");
    }

    public void Menu360VideoTour()
    {
        SceneManager.LoadScene("360 Video Tour");
    }
    public void MainMenu()
    {
        SceneManager.LoadScene("MainMenu");
    }
    public void MainMenuPanel()
    {
        MainPanel.SetActive(true);
        AboutPanel.SetActive(false);
        TutorialPanel.SetActive(false);
    }
    public void PetunjukMenuPanel()
    {
        MainPanel.SetActive(false);
        TutorialPanel.SetActive(true);
    }
```

```
 MainPanel.SetActive(false);
        AboutPanel.SetActive(true);
    }

    public void QuitApps()
    {
        Application.Quit();
    }
}
```

2. *SplashFade*



Gambar 4. 8 *SplashFade*

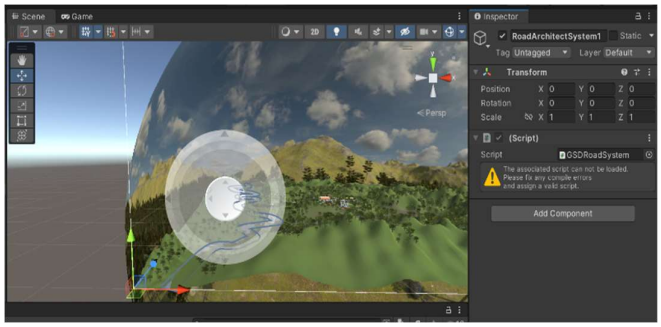
SplashFade merupakan Menampilkan gambar splash dengan efek fade masuk dan keluar Otomatis berpindah ke scene berikutnya setelah animasi selesai.

```
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.SceneManagement;
using System.Collections;
public class SplashFade : MonoBehaviour
{
      public Image splashImage;
      public string loadLevel;

      IEnumerator Start()
      {
splashImage.canvasRenderer.SetAlpha(0.0f);
            FadeIn();
            yield return new WaitForSeconds(3f);
```

```
FadeOut();
            yield return new WaitForSeconds(2f);
            SceneManager.LoadScene(loadLevel);
      }
      void FadeIn()
      {
            splashImage.CrossFadeAlpha(1.0f, 1.5f, false);
      }
      void FadeOut()
      {
            splashImage.CrossFadeAlpha(0.0f, 2.5f, false);
      }
}
```

3. *GSDRoadSystem*



Gambar 4. 9 *GSDRoadSystem*

GSDRoadSystem berfungsi sebagai sistem manajemen utama untuk plugin Road System Generator di Unity, yang dirancang khusus untuk membuat dan mengelola sistem jalan procedural dalam lingkungan pengembangan game atau simulasi.

```
using UnityEngine;
#if UNITY_EDITOR
using System.Collections;
#endif
public class GSDRoadSystem : MonoBehaviour{
    #if UNITY_EDITOR

public bool opt_bMultithreading = true;
    public bool opt_bSaveMeshes = false;
        public bool opt_bAllowRoadUpdates = true;
```

```
    public GameObject AddRoad(bool bForceSelect = false){
        Object[] tObj =
GameObject.FindObjectsOfType(typeof(GSDRoad));
        int NewRoadNumber = (tObj.Length+1);

        //Road:
        GameObject tRoadObj = new GameObject("Road" +
NewRoadNumber.ToString());
        UnityEditor.Undo.RegisterCreatedObjectUndo(tRoadObj,
"Created road");
        tRoadObj.transform.parent = transform;
        GSDRoad tRoad = tRoadObj.AddComponent<GSDRoad>();

        //Spline:
        GameObject tSplineObj = new GameObject("Spline");
        tSplineObj.transform.parent = tRoad.transform;
        tRoad.GSDSpline = tSplineObj.AddComponent<GSDSplineC>();
        tRoad.GSDSpline.mSplineRoot = tSplineObj;
        tRoad.GSDSpline.tRoad = tRoad;
        tRoad.GSDSplineObj = tSplineObj;
        tRoad.GSDRS = this;
        tRoad.SetupUniqueIdentifier();

        tRoad.ConstructRoad_ResetTerrainHistory();

        if(bForceSelect){
            UnityEditor.Selection.activeGameObject = tRoadObj;
        }

        return tRoadObj;
    }
    public Camera EditorPlayCamera = null;
    public void EditorCameraSetSingle(){
        if(EditorPlayCamera == null){
            Camera[] EditorCams =
(Camera[])GameObject.FindObjectsOfType(typeof(Camera));
            if(EditorCams != null && EditorCams.Length == 1){
                EditorPlayCamera = EditorCams[0];
            }
        }
    }
    public void UpdateAllRoads(){
        GSDRoad[] tRoadObjs =
(GSDRoad[])GameObject.FindObjectsOfType(typeof(GSDRoad));
//      int i=0;

        int RoadCount = tRoadObjs.Length;

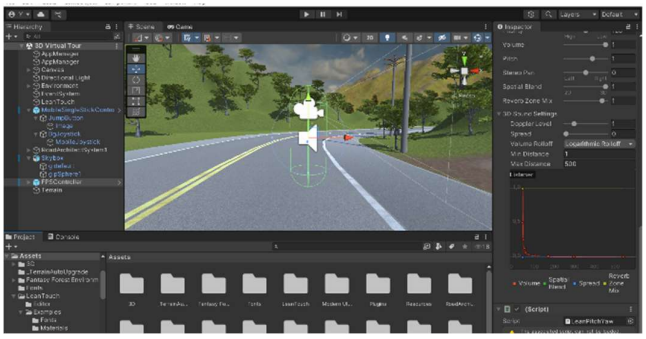
       GSDRoad tRoad = null;
        GSDSplineC[] tPiggys = null;
        if(RoadCount > 1){
```

```
            tPiggys = new GSDSplineC[RoadCount];
            for (int h = 0; h < RoadCount; h++) {
                tRoad = tRoadObjs[h];
                tPiggys[h] = tRoad.GSDSpline;
            }
        }

        tRoad=tRoadObjs[0];
        if(tPiggys != null && tPiggys.Length > 0){
            tRoad.PiggyBacks = tPiggys;
        }
        tRoad.UpdateRoad();
    }

    //Workaround for submission rules:
    public void UpdateAllRoads_MultiThreadOptions(){
        GSDRoad[] tRoadObjs =
(GSDRoad[])GameObject.FindObjectsOfType(typeof(GSDRoad));
        int RoadCount = tRoadObjs.Length;
        GSDRoad tRoad = null;
        for(int h=0;h<RoadCount;h++){
            tRoad=tRoadObjs[h];
            if(tRoad != null){
                tRoad.opt_bMultithreading = opt_bMultithreading;
            }
        }
    }
    //Workaround for submission rules:
    public void UpdateAllRoads_SaveMeshesAsAssetsOptions(){
        GSDRoad[] tRoadObjs =
(GSDRoad[])GameObject.FindObjectsOfType(typeof(GSDRoad));
        int RoadCount = tRoadObjs.Length;
        GSDRoad tRoad = null;
        for(int h=0;h<RoadCount;h++){
            tRoad=tRoadObjs[h];
            if(tRoad != null){
                tRoad.opt_bSaveMeshes = opt_bSaveMeshes;
            }
        }
    }
    #endif
}
```

4. *Lean Pitch Yaw*



Gambar 4. 10 *Lean Pitch Yaw*

LeanPitchYaw merupakan bagian dari Lean Touch plugin yang berfungsi sebagai sistem kontrol rotasi touch-based untuk objek dalam Unity, khususnya dirancang untuk mengontrol rotasi kamera atau objek lain dengan gerakan drag finger pada layar sentuh. Script ini mengimplementasikan sistem pitch dan yaw control yang memungkinkan pengguna untuk melakukan rotasi vertikal (pitch) dan horizontal (yaw) pada objek dengan cara yang intuitif dan responsif. Sistem ini sangat cocok untuk aplikasi mobile yang memerlukan kontrol kamera first-person atau third-person, aplikasi AR/VR, atau any interactive 3D experience yang membutuhkan navigasi view yang smooth.

```
using UnityEngine;
namespace Lean.Touch
{
	// This script allows you to tilt & pan the current
GameObject (e.g. camera) by dragging your finger(s)
	[ExecuteInEditMode]
	public class LeanPitchYaw : MonoBehaviour
	{
		[Tooltip("Ignore fingers with StartedOverGui?")]
		public bool IgnoreGuiFingers = true;

		[Tooltip("Ignore fingers if the finger count doesn't
match? (0 = any)")]
```

```
      [Tooltip("If you want the rotation to be scaled by the camera
FOV, then set that here")]
            public Camera Camera;

            [Tooltip("Pitch of the rotation in degrees")]
            [Space(10.0f)]
            public float Pitch;

            [Tooltip("The strength of the pitch changes with
vertical finger movement")]
            public float PitchSensitivity = 0.25f;

            [Tooltip("Limit the pitch to min/max?")]
            public bool PitchClamp = true;

            [Tooltip("The minimum pitch angle in degrees")]
            public float PitchMin = -90.0f;

            [Tooltip("The maximum pitch angle in degrees")]
            public float PitchMax = 90.0f;

            [Tooltip("Yaw of the rotation in degrees")]
            [Space(10.0f)]
            public float Yaw;

            [Tooltip("The strength of the yaw changes with
horizontal finger movement")]
            public float YawSensitivity = 0.25f;

            [Tooltip("Limit the yaw to min/max?")]
            public bool YawClamp;

            [Tooltip("The minimum yaw angle in degrees")]
            public float YawMin = -45.0f;

            [Tooltip("The maximum yaw angle in degrees")]
            public float YawMax = 45.0f;
#if UNITY_EDITOR
            protected virtual void Reset()
            {
                  if (Camera == null)
                  {
                        Camera = GetComponent<Camera>();
                  }
            }
#endif

            protected virtual void LateUpdate()
            {
                  // Get the fingers we want to use
            var fingers = LeanTouch.GetFingers(IgnoreGuiFingers,
```

```
// Get the scaled average movement vector of these fingers
                  var drag = LeanGesture.GetScaledDelta(fingers);

                  // Get base sensitivity
                  var sensitivity = GetSensitivity();

                  // Adjust pitch
                  Pitch += drag.y * PitchSensitivity * sensitivity;

                  if (PitchClamp == true)
                  {
                        Pitch = Mathf.Clamp(Pitch, PitchMin,
PitchMax);
                  }

                  // Adjust yaw
                  Yaw -= drag.x * YawSensitivity * sensitivity;

                  if (YawClamp == true)
                  {
                        Yaw = Mathf.Clamp(Yaw, YawMin, YawMax);
                  }

                  // Rotate to pitch and yaw values
                  transform.localRotation = Quaternion.Euler(Pitch,
Yaw, 0.0f);
            }

            private float GetSensitivity()
            {
                  // Has a camera been set?
                  if (Camera != null)
                  {
                        // Adjust sensitivity by FOV?
                        if (Camera.orthographic == false)
                        {
                              return Camera.fieldOfView / 90.0f;
                        }
                  }

                  return 1.0f;
            }
      }
}
```

5. *GSDTerrain*



Gambar 4. 11 *GSDTerrain*

GSDTerrain berfungsi sebagai komponen manajemen terrain yang terintegrasi dengan sistem GSD Road System, dirancang khusus untuk mengelola modifikasi dan rendering terrain dalam konteks pembangunan jalan procedural. Script ini merupakan editor-only component yang hanya aktif dalam Unity Editor environment, seperti yang ditunjukkan dengan preprocessing directives dan logic yang menonaktifkan script dalam build runtime. Tujuan utama script ini adalah untuk menyediakan interface dan data management untuk terrain objects yang akan dimodifikasi oleh road system, termasuk splat mapping untuk tekstur jalan dan manajemen unique identifier untuk setiap terrain instance.

```
using UnityEngine;
#if UNITY_EDITOR
using System.Collections.Generic;
#endif
[ExecuteInEditMode]
public class GSDTerrain : MonoBehaviour{

```

```
#if UNITY_EDITOR
      [SerializeField][HideInInspector]
      private int mGSDID = -1;

    public int GSDID{
        get{return mGSDID;}
        set {
                  //Do nothing.
            }
    }
      [HideInInspector]
      public Terrain tTerrain;

      //Splat map:
      public int SplatResoWidth = 1024;
      public int SplatResoHeight = 1024;
      public Color SplatBackground = new Color(0f,0f,0f,1f);
      public Color SplatForeground = new Color(1f,1f,1f,1f);
      public float SplatWidth = 30f;
      public bool SplatSkipBridges = false;
      public bool SplatSkipTunnels = false;
      public bool SplatSingleRoad = false;
      public int SplatSingleChoiceIndex = 0;
      public string RoadSingleChoiceUID = "";

      void OnEnable(){
            CheckID();
            if(!tTerrain){ tTerrain =
transform.gameObject.GetComponent<Terrain>(); }
      }

      public void CheckID(){
            if(Application.isEditor){
                  if(mGSDID < 0){
                        mGSDID = GetNewID();
                  }
if(!tTerrain){ tTerrain =
transform.gameObject.GetComponent<Terrain>(); }
            }
      }
```

```
private int GetNewID(){
            Object[] tTerrainObjs =
GameObject.FindObjectsOfType(typeof(GSDTerrain));
            List<int> AllIDS = new List<int>();
            foreach(GSDTerrain TID in tTerrainObjs){
                  if(TID.GSDID > 0){
                        AllIDS.Add (TID.GSDID);
                  }
            }
            bool bNotDone = true;
            int SpamChecker = 0;
            int SpamCheckerMax = AllIDS.Count + 64;
            int tRand;
            while(bNotDone){
                  if(SpamChecker > SpamCheckerMax){
                        Debug.LogError("Failed to generate
GSDTerrainID");
                        break;
                  }
                  tRand = Random.Range(1,2000000000);
                  if(!AllIDS.Contains(tRand)){
                        bNotDone = false;
                        return tRand;
                  }
                  SpamChecker+=1;
            }
            return -1;
      }
      #endif
      void Start (){
            #if UNITY_EDITOR
            this.enabled = true;
            CheckID();
            if(!tTerrain){ tTerrain =
transform.gameObject.GetComponent<Terrain>(); }
            #else
            this.enabled = false;
            #endif
      }
}
```

6. UIGradient



Gambar 4. 12 *UIGradient*

UIGradient berfungsi sebagai mesh effect untuk memberikan efek gradien pada elemen UI Unity. Script ini mewarisi dari BaseMeshEffect yang memungkinkannya memodifikasi mesh vertices dari UI elements secara real-time untuk menciptakan berbagai jenis gradien visual. Komponen ini memberikan fleksibilitas tinggi dalam styling UI elements dengan empat tipe gradien berbeda yaitu Horizontal, Vertical, Radial, dan Diamond, masing-masing dengan karakteristik visual yang unik dan dapat dikustomisasi sesuai kebutuhan desain

```
using System.Collections.Generic;
namespace UnityEngine.UI
{
    namespace Michsky.UI.ModernUIPack
    {
        public class UIGradient : BaseMeshEffect
        {
            [Header("GRADIENT")]
            [SerializeField]
            UnityEngine.Gradient _effectGradient = new
UnityEngine.Gradient() { colorKeys = new GradientColorKey[] { new
GradientColorKey(Color.black, 0), new
GradientColorKey(Color.white, 1) } };

            [Header("SETTINGS")]
            [SerializeField]
            Type _gradientType;
```

```
            [SerializeField]
            Blend _blendMode = Blend.Multiply;
            [SerializeField]
            [Range(-1, 1)]

            float _offset = 0f;
            #region Properties
            public Blend BlendMode
            {
                get { return _blendMode; }
                set
                {
                    _blendMode = value;
                    graphic.SetVerticesDirty();
                }
            }
            public UnityEngine.Gradient EffectGradient
            {
                get { return _effectGradient; }
                set
                {
                    _effectGradient = value;
                    graphic.SetVerticesDirty();
                }
            }
            public Type GradientType
            {
                get { return _gradientType; }
                set
                {
                    _gradientType = value;
                    graphic.SetVerticesDirty();
                }
            }

            public float Offset
            {
                get { return _offset; }
                set
                {
                    _offset = value;
                    graphic.SetVerticesDirty();
                }
            }
            #endregion

            public override void ModifyMesh(VertexHelper helper)
            {
                if (!IsActive() || helper.currentVertCount == 0)

                    return;
```

```
                List<UIVertex> _vertexList = new List<UIVertex>();

                helper.GetUIVertexStream(_vertexList);

                int nCount = _vertexList.Count;
                switch (GradientType)
                {
                    case Type.Horizontal:
                        {
                            float left = _vertexList[0].position.x;
                            float right = _vertexList[0].position.x;
                            float x = 0f;

                            for (int i = nCount - 1; i >= 1; --i)
                            {
                                x = _vertexList[i].position.x;

                                if (x > right) right = x;
                                else if (x < left) left = x;
                            }

                            float width = 1f / (right - left);
                            UIVertex vertex = new UIVertex();

                            for (int i = 0; i <
helper.currentVertCount; i++)
                            {
                                helper.PopulateUIVertex(ref vertex,
i);

                                vertex.color =
BlendColor(vertex.color, EffectGradient.Evaluate((vertex.position.x
- left) * width - Offset));

                                helper.SetUIVertex(vertex, i);
                            }
                        }
                        break;

                    case Type.Vertical:
                        {
                            float bottom =
_vertexList[0].position.y;
                            float top = _vertexList[0].position.y;
                            float y = 0f;
                            for (int i = nCount - 1; i >= 1; --i)
                            {
                                y = _vertexList[i].position.y;

                                if (y > top) top = y;
                                else if (y < bottom) bottom = y;
                            }
```

```
                            float height = 1f / (top - bottom);
                            UIVertex vertex = new UIVertex();

                            for (int i = 0; i <
helper.currentVertCount; i++)
                            {
                                helper.PopulateUIVertex(ref vertex,
i);

                                vertex.color =
BlendColor(vertex.color, EffectGradient.Evaluate((vertex.position.y
- bottom) * height - Offset));

                                helper.SetUIVertex(vertex, i);
                            }
                        }
                        break;

                    case Type.Diamond:
                        {

                            float bottom =
_vertexList[0].position.y;
                            float top = _vertexList[0].position.y;
                            float y = 0f;

                            for (int i = nCount - 1; i >= 1; --i)
                            {
                                y = _vertexList[i].position.y;

                                if (y > top) top = y;
                                else if (y < bottom) bottom = y;
                            }

                            float height = 1f / (top - bottom);

                            helper.Clear();
                            for (int i = 0; i < nCount; i++)
helper.AddVert(_vertexList[i]);

                            float center = (bottom + top) / 2f;
                            UIVertex centralVertex = new UIVertex();
                            centralVertex.position = (Vector3.right
+ Vector3.up) * center + Vector3.forward *
_vertexList[0].position.z;
                            centralVertex.normal =
_vertexList[0].normal;
                            centralVertex.color = Color.white;
                            helper.AddVert(centralVertex);
```

```
                            for (int i = 1; i < nCount; i++)
helper.AddTriangle(i - 1, i, nCount);
                            helper.AddTriangle(0, nCount - 1,
nCount);

                            UIVertex vertex = new UIVertex();

                            for (int i = 0; i <
helper.currentVertCount; i++)
                            {
                                helper.PopulateUIVertex(ref vertex,
i);

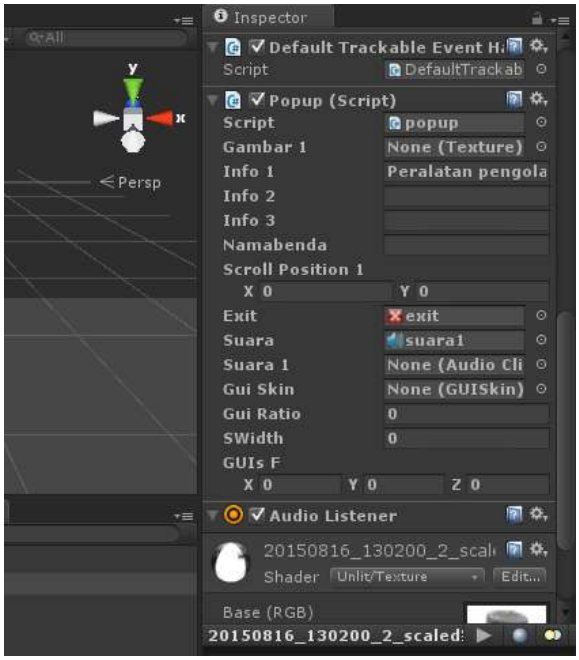
                                vertex.color =
BlendColor(vertex.color, EffectGradient.Evaluate(

Vector3.Distance(vertex.position, centralVertex.position) * heig...

<...etc...>
```

C. Rancangan Input Dan Output

1. Rancangan Input



Gambar 4. 13 Input Data Aplikasi

D. Detail Aplikasi

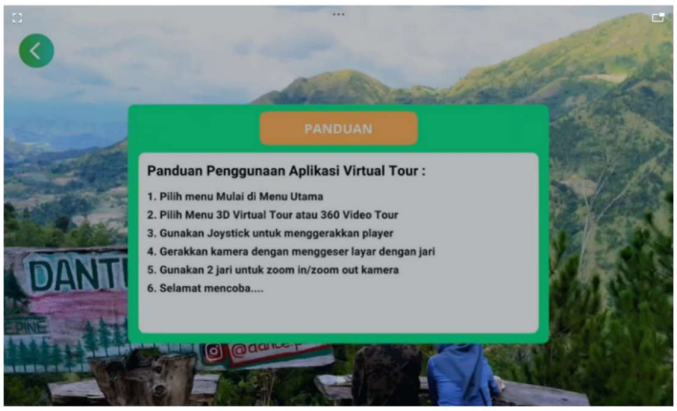
1. *Form* Virtual tour



Gambar 4. 14 *Form* Virtual tour

Gambar di atas adalah tampilan *Virtual tour* ketika *user* telah mengaktifkan Aplikasi ini. nantinya *user* sudah dapat memilih mulai Tour maka aplikasi ini akan menampilkan aktif kamera untuk mendapatkan Informasi renang di dalam Virtual tour ini juga di sertakan tombol play dan *Form home* dimana di dalamnya terdapat beberapa tombol yang memiliki fungsi-fungsi tertentu.

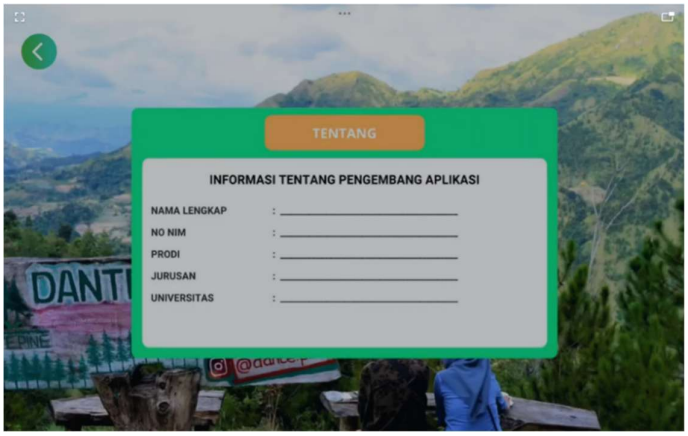
2. *Form* panduan



Gambar 4. 15 Menu panduan

Gambar tampilan di atas adalah sebuah *PopUp* informasi yang tampil apabila *user* menekan tombol panduan pada Menu utama, isinya kurang lebih cara penggunaan Aplikasi secara umum.

3. *Form* Tentang



Gambar 4. 16 Menu tentang

Gambar tampilan di atas adalah sebuah *PopUp* informasi yang tampil apabila *user* menekan tombol tentang pada Menu utama, isinya kurang lebih cara penggunaan Aplikasi secara umum.

4. *Form* Mulai Tour



Gambar 4. 17 Menu Mulai tour

Gambar tampilan di atas adalah sebuah *PopUp* informasi yang tampil apabila *user* menekan tombol Mulai tour pada Menu utama, isinya kurang lebih cara penggunaan Aplikasi secara umum.

5. *Form* 3D Tour



Gambar 4. 18 Menu 3D Tour

Gambar tampilan di atas adalah sebuah *PopUp* informasi yang tampil apabila *user* menekan 3D Tour maka akan muncul 3D wisata Dante pada Menu utama, isinya kurang lebih cara penggunaan Aplikasi secara umum.

6. *Form* 360



Gambar 4. 19 Menu 360

Gambar tampilan di atas adalah sebuah *PopUp* informasi yang tampil apabila *user* menekan 360 Tour maka akan muncul 360 wisata Dante pada Menu utama, isinya kurang lebih cara penggunaan Aplikasi secara umum.

E. Pengujuan Sistem

1. Implementasi

Implementasi sistem merupakan tahap penerapan dari suatu teknologi yang didesain untuk siap dioperasikan. Pada tahap ini design sistem yang telah dirancang diterapkan ke dalam bahasa pemrograman yang kemudian di lakukan pengujian sistem.

Adapun bahasa pemrograman yang digunakan untuk membuat aplikasi ini adalah bahasa pemrograman C# (Sharp).

a. Kebutuhan *Hardware*

Spesifikasi minimum *hardware* pengguna aplikasi ini adalah sebagai berikut :

Tabel 4. 1 Spesifikasi *Hardware Smartphone*

Jenis	Spesifikasi
Smartphone	Semua Merk
Processor	Intel® pentium™ 1.60 Ghz
RAM	4 GB
Memori	6 GB
Camera	5 MP (Autofokus : Ya || Flash : Ya)
Display	Touchscreen 4.0 inches

b. Kebutuhan *Software*

Spesifikasi minimum *software* pengguna aplikasi ini yaitu sebagai berikut:

Tabel 4. 2 Spesifikasi *Software*

Jenis	Spesifikasi
Sistem Operasi	Android Ver. 11

2. Pengujian aplikasi

a. Metode Pengujian

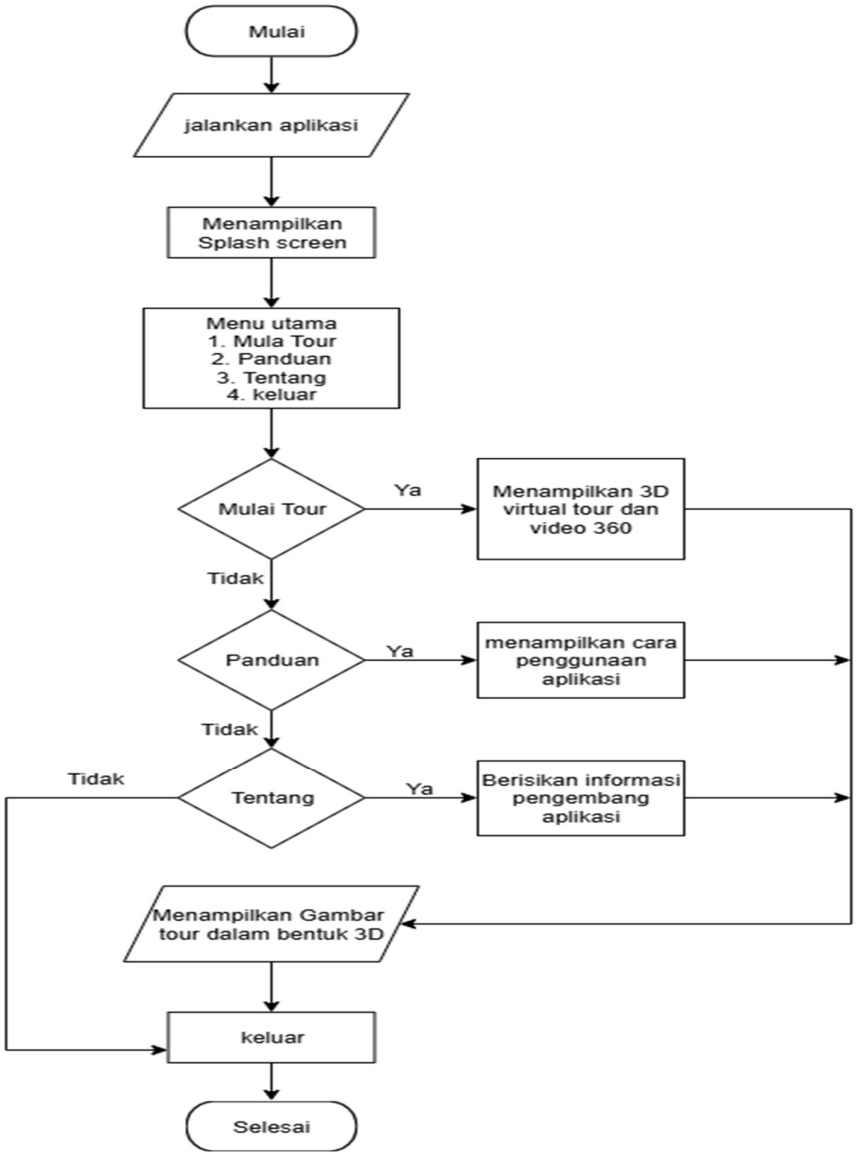
Pengujian aplikasi dilakukan dengan menggunakan dua metode pengujian yaitu Pengujian *Black box (White box testing)* dan Pengujian *White box (Black box testing).*

b. Teknik Pengujian

Pengujian *black-box* berkaitan dengan pengujian yang dilakukan pada *interface* perangkat lunak. Meskipun didesain untuk mengungkapkan kesalahan, pengujian *black-box* digunakan untuk memperlihatkan bahwa fungsi-fungsi perangkat lunak adalah operasional, bahwa input diterima dengan baik dan output dihasilkan dengan tepat, dan integritas informasi external (seperti file data) dipelihara. Pengujian blackbox menguji beberapa aspek dasar suatu aplikasi dengan sedikit memperhatikan struktur logika internal perangkat lunak tersebut. sedangkan Pengujian *Whitebox* berfokus pada efektifitas aplikasi yang dirancang

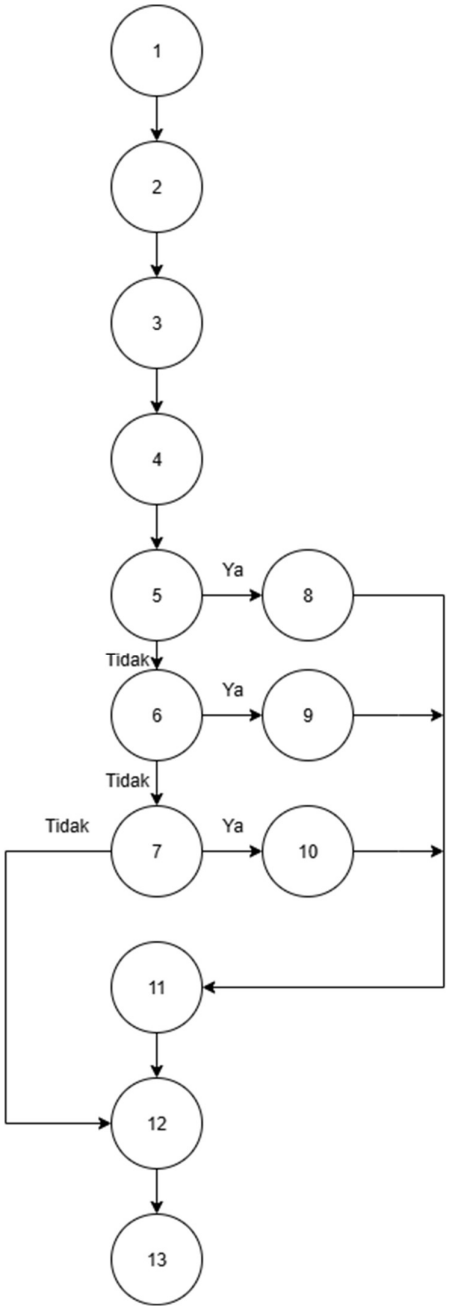
3. Pengujian *Whitebox*

a. *Flowchart* Aplikasi



Gambar 4. 20 *Flowchart* Aplikasi

b. *Flowgraph* Aplikasi



Gambar 4. 21 *Flowgraph* Aplikasi

Dari gambar *Flowgraph* diatas dapat dilakukan proses perhitungan sebagai berikut :

Diketahui :

N = 13 E = 15 R = 4

penyelesaian :

V(G)=(15−13)+2

=2+2

=4

Path 1= 1-2-3-4-5-6-7-11-12-13

Path 2= 1-2-3-4-5-8-11-12-13

Path 3= 1-2-3-4-5-6-9-11-12-13

Path 4= 1-2-3-4-5-6-7-10-11-12-13

c. *Grafik Matriks*

Tabel 4. 3 *Grafik Matriks*

	1	2	3	4	5	6	7	8	9	10	11	12	13	E-1
1		1												1-1=0
2			1											1-1=0
3				1										1-1=0
4					1									1-1=0
5						1		1						2-1=1
6							1		1					2-1=1
7										1	1			2-1=1
8											1			1-1=0
9											1			1-1=0
10											1			1-1=0
11												1		1-1=0
12													1	1-1=0
13														0
Sum(E+1)														3+1=4

4. Pengujian *Blackbox*

Tabel 4. 4 *Blackbox* 3d Virtual tour

Test Factor	Hasil	Keterangan
Jika mendeteksi Virtual tour/objek dengan stabil	✓	Berhasil, Karena tampil 3d furniture
ScreenShoot		



5. Pengujian Jarak

Berdasarkan gambar di atas Aplikasi dapat mengenali Virtual tour 3D ini dengan baik sama seperti Virtual tour 3D sebelumnya harus dengan bantuan pencahayaan yang bagus dan kualitas 3D pada saat di arahkan.

Adapun Hasil Uji Pengukuran dari Pendeteksian Virtual tour 3D di atas adalah Sebagai berikut :

Tabel 4. 5 Hasil Pengukuran Virtual tour

Jarak	**Virtual tour 3D**	**KET**
20 cm	✗	✓=terlihat ✗= tidak terlihat
25 cm	✓	
30 cm	✓	
35 cm	✓	
40 cm	✓	
45 cm	✓	
50 cm	✓	
Derajat Kemiringan	**Virtual tour 3D**	
30^0	✓	
35^0	✓	
40^0	✓	
45^0	✓	
50^0	✓	
55^0	✓	
60^0	✓	
65^0	✓	
70^0	✓	
75^0	✓	

Lanjutan Tabel 4. 5

Derajat Kemiringan	Virtual tour 3D	Derajat Kemiringan
80^0	✓	✓=terlihat ×= tidak terlihat
85^0	✓	
90^0	✓	

BAB V

PENUTUP

A. Kesimpulan

Berdasarkan hasil penelitian dan pengembangan aplikasi virtual tour wisata Dante Pine Kabupaten Enrekang berbasis Android, dapat disimpulkan bahwa aplikasi virtual tour berhasil dikembangkan sesuai dengan tujuan penelitian untuk memberikan solusi digital dalam mempromosikan wisata Dante Pine melalui teknologi virtual reality berbasis Android. Aplikasi ini mampu memberikan pengalaman virtual yang mendekati pengalaman nyata mengunjungi lokasi wisata dengan menghadirkan fitur-fitur utama yang komprehensif.

Semua fitur yang direncanakan telah berhasil diimplementasikan dengan baik, meliputi panorama 360° yang memberikan pengalaman *visual immersive*, peta interaktif dengan GPS tracking untuk navigasi, informasi lengkap tentang objek wisata dan fasilitas yang tersedia, galeri multimedia berupa foto dan video berkualitas tinggi, serta panduan perjalanan yang membantu wisatawan merencanakan kunjungan mereka. Hasil pengujian menunjukkan bahwa aplikasi memiliki performa yang baik dengan tingkat *responsivitas* tinggi dan *interface* yang *user-friendly*, dimana aplikasi dapat berjalan dengan lancar pada berbagai *device Android* dengan spesifikasi minimum yang telah ditentukan.

B. Saran

Saran yang dapat diajukan oleh peneliti mengenai pengembangan adalah sebagai berikut:

Untuk implementasi dan pemasaran, disarankan melakukan publikasi aplikasi di Google Play Store, menjalin kerjasama dengan Dinas Pariwisata Kabupaten Enrekang untuk promosi resmi, serta berkolaborasi dengan stakeholder seperti komunitas fotografi, tour operator lokal, dan influencer travel untuk meningkatkan awareness dan mengintegrasikan layanan booking. Penelitian lanjutan dapat dilakukan untuk menganalisis dampak aplikasi terhadap peningkatan kunjungan wisatawan, studi komparatif efektivitas virtual tour, dan pengembangan aplikasi serupa untuk destinasi wisata lain di Sulawesi Selatan. Pemerintah daerah juga disarankan untuk membuat regulasi pendukung pariwisata digital, mengalokasikan anggaran khusus untuk digitalisasi destinasi wisata, serta meningkatkan infrastruktur telekomunikasi di area wisata.

DAFTAR PUSTAKA

Aditya Titus,K.(2019). Rancang Bangun Aplikasi Inventaris Berbasis Website pada Kelurahan Bantengan.Universitas PGRI Madiun.

Alfarabi, S. Z., Hastuty, A., & Yunus, M. Aplikasi Virtual tour Interaktif Kebun Raya Jompie Berbasis VR.

Asri, M. A., Hastuty, A., & Yunus, M. Aplikasi Pemilihan Frame Kacamata Berbasis Augmented Reality.

Farhan Muhammad,L., (2021).Implementasi White box Testing Dengan Teknisi Basis Path Pada Pengujian Form Login.Universitas Siliwangi Tasikmalaya.Vol.7.No.2,2021.

Ferdy, F., & Wahyuddin, W. (2024). Aplikasi Game Edukasi Mitigasi Bencana Alam (Gempa Bumi Dan Tsunami) Mengunakan Metode Waterfall Berbasis Android. *Jurnal Sintaks Logika*, *4*(1), 1-6.

Fidel A.E, Dkk.(2020).Virtual reality pendakian gunung Klabat di Sulawesi Utara.Jurnal Pendidikan dan konseling.4(5).

Givan, M. (2024). *APLIKASI GAME CHECKERS DENGAN MENGGUNAKAN METODE ALPHA BETA DAN MINIMAX* (Doctoral dissertation, UNIVERSITAS MUHAMMADIYAH PAREPARE).

Hastuty, A. (2021). Dasar Pemrograman. *Cetakan*, *2*, 18-23.

Hastuty, A. (2025). Hacking the Future of College Administration: Process Transformation through AR/VR Immersive Technology. *EDUKASIA Jurnal Pendidikan dan Pembelajaran*, *6*(1), 95-106.

Juansyah Andi.(2015).Pembangunan Aplikasi Child Tracker Berbasis Assisted-Global Positioning System (A-GPS) Dengan Platform Android.Universitas Komputer Indonesia.

Lengkong,O., Dkk (2017) ”Perancangan Aplikasi Virtual Reality Pengenalan Tempat Wisata di Sulawesi Utara Berbasis Android”

Muslimin, A. F., Wahyuddin, W., & Nugraha, W. A. (2024). Membuat Game Virtual Reality Kota Parepare. *Jurnal Sintaks Logika*, *4*(3), 1-12.

Paridawati Ita.(2021). Persepsi Orangtua Terhadap Penggunaan Smartphone pada Anak Usia Dini di Desa Indrasakti Kecamatan Tapung Kabupaten Kampar. Universitas Pahlawan Tuanku Tambusai

Rismayani. 2021. Aplikasi Virtual tour Fakultas Teknik Berbasis Andorid Mobile .Kota Parepare. Universitas Muhammadiyah Parepare.

Satria, E., Latifah,A., ParojiInstitut,M., Rancang Bangun Aplikasi Katalog Wisata di Garut Menggunakan Teknologi Virtual Reality. Teknologi Garut

Stefano Lourent M.(2018).Rancang Bangun Game Adventure Of Unsrat Menggunakan Game Unity.Universitas Sam Ratulangi Manado.

Wahyuddin, W., & Hasnawati, H. (2023). Aplikasi Media Pembelajaran Pengenalan Pancasila Menggunakan Augmented Reality. *Jurnal Sintaks Logika*, *3*(3), 8-15

Wira Putra.D.T.(2019). Pembayaran Restitusi SPPD.Insititut Teknologi Pada Unified modelling language (UML) dalam Perancangan Sistem Informasi Permohonan ng.

https://www.researchgate.net/publication/330288127_A_survey_on_image_and_video_stitching/link/5cab101492851c64bd57c146/download?_tp=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI6InB1YmxpY2F0aW9uIiwicGFnZSI6InB1YmxpY2F0aW9uIn19