BAB II

TINJAUAN PUSTAKA

A. Kajian Teori

1. Rancang Bangun

"Rancang bangun" adalah pendekatan kerja yang menggabungkan dua tahap utama: merancang (menyusun konsep, spesifikasi, dan desain) lalu membangun (mengimplementasikan hingga siap digunakan). Tujuannya adalah menghasilkan solusi—perangkat lunak, perangkat keras, produk fisik, atau prosedur kerja—yang tepat guna, dapat dioperasikan, dan memenuhi kebutuhan pemangku kepentingan. Proses ini dimulai dari identifikasi masalah dan kebutuhan, dilanjutkan analisis kelayakan, perumusan kebutuhan fungsional/non-fungsional, penyusunan arsitektur/desain terperinci, pembuatan purwarupa (prototype), implementasi, pengujian, penerapan (deployment), hingga pemeliharaan dan peningkatan. Sifatnya iteratif: temuan di tahap uji dapat mengembalikan tim ke tahap desain untuk penyempurnaan.

Dalam tahap perancangan, fokusnya pada "bagaimana seharusnya solusi bekerja". Di sini ditentukan ruang lingkup, batasan, alur proses, model data, antarmuka, serta standar yang diacu. Artefak umum pada tahap ini misalnya diagram arsitektur, diagram UML, rancangan antarmuka (wireframe), spesifikasi teknis, dan rencana uji. Tahap pembangunan mewujudkan desain menjadi produk nyata melalui pemrograman atau fabrikasi, integrasi komponen, dan verifikasi. Kualitas dijaga lewat pengujian berlapis: uji unit, integrasi, sistem, kinerja,

keamanan, dan uji penerimaan pengguna (UAT). Hasil uji didokumentasikan untuk memastikan keterlacakan (traceability) dari kebutuhan ke implementasi.

Keberhasilan rancang bangun diukur dari kesesuaian fungsi dengan kebutuhan, keandalan, kemudahan pakai, kinerja, keamanan, maintainability, serta efisiensi biaya dan waktu. Manajemen proyek, pengendalian perubahan, dan mitigasi risiko berjalan paralel—misalnya dengan penjadwalan, versi kode, serta peninjauan desain/kode (review). Metodologi yang lazim dipakai dapat berbasis sekuensial (mis. waterfall) atau iteratif-inkremental (mis. Agile/Scrum), dipilih sesuai kompleksitas, ketidakpastian kebutuhan, dan tenggat.

Dalam konteks akademik (misalnya skripsi "Rancang Bangun …"), struktur umumnya memuat latar belakang dan rumusan masalah, tujuan dan batasan, tinjauan pustaka, metodologi rancang bangun, perancangan sistem/produk, implementasi, pengujian dan evaluasi, lalu kesimpulan dan saran. Evaluasi bisa memakai metrik yang relevan: akurasi/kinerja teknis, waktu respons, konsumsi sumber daya, atau instrumen usability seperti SUS untuk menilai pengalaman pengguna. Seluruh proses harus terdokumentasi rapi agar dapat direplikasi, diaudit, dan dikembangkan lebih lanjut.

Singkatnya, rancang bangun bukan sekadar "membuat sesuatu", melainkan siklus rekayasa menyeluruh yang memastikan solusi dibangun berdasarkan kebutuhan yang jelas, desain yang matang, implementasi yang terkontrol, pengujian yang ketat, dan dokumentasi yang dapat dipertanggungjawabkan.

2. *Virtual Reality*

Virtual Reality (VR) adalah sebuah teknologi yang memungkinkan pengguna merasakan pengalaman seolah-olah berada di dalam lingkungan buatan yang sepenuhnya diciptakan oleh komputer. Teknologi ini menghadirkan dunia virtual yang imersif dengan memanfaatkan perangkat khusus seperti head-mounted display (HMD) atau kacamata VR, sensor gerak, controller, hingga sistem audio 3D. Berbeda dengan media konvensional yang hanya bisa ditonton atau diakses melalui layar, VR memberikan pengalaman interaktif yang melibatkan indra penglihatan, pendengaran, bahkan dalam beberapa kasus sentuhan (haptic feedback). Dengan VR, pengguna dapat bergerak, berinteraksi, dan mengeksplorasi objek di dalam dunia digital secara real-time, sehingga tercipta ilusi seakan-akan benar-benar berada dalam ruang atau situasi tersebut.

Konsep dasar VR adalah menciptakan simulasi yang meniru dunia nyata maupun membangun dunia yang sepenuhnya imajinatif. Hal ini dilakukan dengan memproyeksikan gambar tiga dimensi (3D) yang responsif terhadap gerakan kepala dan tubuh pengguna, sehingga setiap perubahan sudut pandang akan diikuti dengan perubahan tampilan visual. Teknologi VR banyak dimanfaatkan di berbagai bidang, misalnya dalam hiburan dan game untuk memberikan pengalaman bermain yang lebih realistis, dalam pendidikan untuk simulasi pembelajaran interaktif, dalam kedokteran untuk pelatihan operasi atau terapi, hingga dalam industri untuk desain, pelatihan pekerja, dan visualisasi produk.

Keunggulan utama VR terletak pada sifatnya yang immersive (membawa pengguna larut dalam pengalaman) dan interactive (dapat merespons tindakan

pengguna). Dua aspek ini menjadikan VR sebagai salah satu inovasi teknologi yang mampu mengubah cara manusia belajar, bekerja, berkomunikasi, maupun menikmati hiburan. Namun, penggunaan VR juga memiliki tantangan, seperti harga perangkat yang relatif mahal, keterbatasan perangkat keras, potensi ketidaknyamanan fisik (motion sickness), dan kebutuhan akan konten berkualitas tinggi. Meskipun begitu, perkembangan VR terus berkembang pesat seiring meningkatnya kemampuan komputasi, grafis, serta integrasi dengan teknologi lain seperti Augmented Reality (AR), Mixed Reality (MR), dan Artificial Intelligence (AI). Dengan demikian, Virtual Reality dapat dipahami sebagai terobosan teknologi yang menghadirkan dunia digital layaknya nyata, membuka peluang besar untuk inovasi di berbagai bidang kehidupan.

3. *Augmented Reality*

Virtual Reality (VR) merupakan suatu teknologi yang memungkinkan pengguna untuk berinteraksi dengan lingkungan buatan yang disimulasikan oleh komputer, seolah-olah mereka berada di dalam dunia tersebut secara nyata. VR menciptakan pengalaman imersif melalui kombinasi tampilan visual tiga dimensi (3D), audio spasial, serta dalam beberapa kasus, umpan balik gerak atau haptic feedback. Teknologi ini bekerja dengan menampilkan citra visual melalui perangkat seperti head-mounted display (HMD), yang menutup seluruh bidang pandang pengguna, sehingga menciptakan ilusi bahwa mereka benar-benar berada di dalam lingkungan virtual. Menurut Milgram dan Kishino (1994), VR berada dalam spektrum reality-virtuality continuum, di mana VR merupakan sisi ekstrem dari lingkungan yang sepenuhnya virtual, berbeda dari augmented reality yang

masih mempertahankan elemen dunia nyata.

Dalam pengembangannya, VR memiliki tiga elemen utama, yaitu immersion, interaction, dan imagination. Immersion mengacu pada sejauh mana pengguna merasa "tenggelam" atau menjadi bagian dari dunia virtual. Interaction menunjukkan kemampuan sistem untuk merespons tindakan pengguna secara real-time, menciptakan kesan bahwa pengguna memiliki kontrol terhadap lingkungan tersebut. Sedangkan imagination adalah aspek yang mendorong pengguna membentuk persepsi dan pemahaman baru terhadap pengalaman virtual yang diberikan. Teknologi VR telah diterapkan dalam berbagai bidang seperti pendidikan, simulasi medis, pelatihan militer, hiburan, hingga pemasaran digital karena kemampuannya dalam menyampaikan informasi secara menarik dan kontekstual.

Secara teknis, pengembangan aplikasi VR melibatkan pemodelan objek 3D, desain lingkungan virtual, dan pemrograman logika interaksi. Software seperti Unity 3D, Unreal Engine, serta Blender digunakan untuk merancang elemen-elemen visual dan mengatur perilaku objek dalam dunia virtual. Dalam konteks pemasaran, VR menawarkan potensi besar untuk meningkatkan pengalaman pelanggan, karena memungkinkan mereka untuk mengeksplorasi produk atau layanan dalam bentuk simulasi yang realistis dan menarik tanpa harus hadir secara fisik. Oleh karena itu, VR bukan hanya menjadi inovasi teknologi, tetapi juga sebagai media transformasi cara manusia berinteraksi dengan informasi dan lingkungan digital Unity

Zaky, Ali (2016), Unity merupakan game engine yang ber-multiplatform.

Unity mampu di publish menjadi Standalone (.exe), berbasis web, Android, IoS Iphone, XBOX, dan PS3. Walaupun bisa dipublish ke berbagai platform. Unity perlu lisensi untuk dapat dipublish ke platform tertentu. Tetapi Unity menyediakan untuk free user dan bisa di publish dalam bentuk Standalone (.exe) dan web. Untuk saat ini Unity sedang di kembangkan berbasis AR (Augmented Reality). Fitur – fitur dalam unity 3D adalah :

a. Rendering

   Graphics engine yang digunakan adalah Direct3D (Windows, Xbox 360), OpenGL (Mac, Windows, Linux, PS3), OpenGL ES (Android, iOS), dan proprietary APIs (Wii). Ada pula kemampuan untuk bump mapping, reflection mapping, parallax mapping, screen space ambient occlusion (SSAO), dynamic shadows using shadow maps, render – to – texture and full – screen post – processing effects.

b. Scripting

   Script game engine dibuat dengan Mono 2.6, sebuah implementasi open – source dari .NET Framework. Programmer dapat menggunakan UnityScript (Bahasa Terkustomisasi yang terinspirasi dari sintax ECMAScript, dalam bentuk JavaScript), C#, atau Boo (terinsipirasi dari sintax Bahasa pemrograman phyton). Dimulai dengan dirilisnya versi 3.0, unity menyertakan versi MonoDevelop yang terkustomisasi untuk debug script.

c. Asset Tracking

   Unity juga menyertakan Server Unity Asset – sebuah solusi terkontrol untuk developer game asset dan script.

d. Platforms

Unity support pengembangan ke berbagai platform. Didalam project, developer memiliki kontrol untuk mengirim keperangkat mobile, web browser, desktop, dan console. Unity juga mengijinkan spesifikasi kompresi textur dan pengaturan resolusi di setiap platform yang didukung.

e. Asset Store

Diluncurkan Novmeber 2010, Unity Asset Store adalah sebuah resource yang hadir di Unity editor. Asset store terdiri dari koleksi lebih dari 4,400 asset packages, beserta 3D models, texture dan materials, sistem particle, musik dan efek suara, tutorial dan project, scripting package, editor extensions dan servis online.

f. Physics

Unity juga memiliki suport built – in untuk PhysX physics engine (sejak unity 3.0) dari Nvidia (sebelumnya Ageia) dengan penambahan kemampuan untuk simulasi real – time cloth pada arbitrary dan skinned meshes, thick ray cast, dan collision layers.

4. *Blander*

Blender adalah perangkat lunak open-source yang digunakan secara luas untuk keperluan modeling 3D, animasi, sculpting, texturing, hingga rendering. Dalam pengembangan game Virtual Reality (VR) seperti *House of Horror*, Blender memiliki peran penting dalam pembuatan aset-aset 3D yang akan digunakan di dalam game. Langkah pertama adalah memahami antarmuka Blender, termasuk cara bernavigasi di viewport menggunakan orbit, pan, dan zoom, serta mengenal

shortcut dasar seperti G (grab), R (rotate), dan S (scale) untuk memanipulasi objek.

Selanjutnya, pengguna akan belajar membuat objek dasar seperti kubus, bidang datar, silinder, dan bola. Penggunaan *edit mode* untuk mengubah bentuk objek sangat penting, di mana fitur seperti extrude, inset, bevel, dan loop cut digunakan untuk membentuk model yang lebih kompleks. Dalam konteks game horor, ini dapat diterapkan untuk membuat desain interior rumah, furnitur seperti kursi dan meja, serta elemen-elemen menyeramkan seperti tengkorak, patung tua, atau pintu rusak. Untuk menjaga performa game, aset dibuat dalam bentuk low-poly dan dioptimalkan untuk menghindari penggunaan vertex yang berlebihan.

Proses berikutnya adalah memberi tekstur dan material pada objek. Ini melibatkan UV Mapping, yaitu proses membuka permukaan objek 3D ke dalam bentuk 2D agar bisa diberi gambar tekstur. Material dapat diatur menggunakan node seperti base color, roughness, dan metallic, ditambah penggunaan bump map atau normal map untuk menciptakan kesan detail pada permukaan tanpa menambah polygon. Misalnya, tekstur tembok retak, kayu lapuk, dan noda darah bisa diaplikasikan untuk memperkuat suasana horor.

Untuk pencahayaan, Blender menyediakan berbagai jenis lampu seperti point, sun, area, dan spot light. Dalam game horor, pencahayaan menjadi elemen penting untuk menciptakan ketegangan, seperti dengan menggunakan lampu redup atau efek kedipan. Pengaturan kamera juga penting untuk presentasi adegan, terutama jika ingin merender preview scene yang sudah dibuat.

Materi juga mencakup rigging dan animasi sederhana, seperti membuka pintu atau lampu yang berkedip. Hal ini dilakukan dengan memberi tulang (armature) dan

menggunakan keyframe animasi di timeline. Setelah model dan animasi siap, file diekspor ke format yang sesuai untuk game engine seperti Unity, misalnya .fbx atau .glb. Penting untuk mengatur skala dan posisi objek dengan benar sebelum ekspor agar tidak terjadi kesalahan saat digunakan dalam pengembangan game.

Secara keseluruhan, materi Blender ini disusun untuk mempersiapkan mahasiswa atau pengembang pemula dalam menciptakan aset 3D berkualitas untuk game VR, khususnya dalam genre horor. Penekanan pada efisiensi modeling, tekstur realistis, dan pencahayaan dramatis akan sangat mendukung terciptanya suasana mencekam dalam permainan *House of Horror*.

3. Bahasa Pemrograman C#

Filus, Teo (2017), C# atau yang dibaca C sharp adalah bahasa pemrograman sederhana yang digunakan untuk tujuan umum, dalam artian bahasa pemrograman ini dapat digunakan untuk berbagai fungsi misalnya untuk pemrograman server-side pada website, membangun aplikasi desktop ataupun mobile, pemrograman game dan sebagainya. Selain itu C# juga bahasa pemrograman yang berorientasi objek, jadi C# juga mengusung konsep objek seperti inheritance, class, polymorphism dan encapsulation. Dalam prakteknya C# sangat bergantung dengan framework yang disebut .NET Framework, framework inilah yang nanti digunakan untuk mengcompile dan menjalankan kode C#.

C# dikembangkan oleh Microsoft dengan merekrut Anders Helsberg. Tujuan dibangunnya C# adalah sebagai bahasa pemrograman utama dalam lingkungan .NET Framework (lihat C#). Banyak pihak juga yang menganggap bahwa Java dengan C# saling bersaing, bahkan ada juga yang menyatakan jika pernah belajar

Java maka belajar C# akan sangat mudah dan begitu juga sebaliknya. Anggapan tersebut sebenarnya tidak salah karena perlu diketahui sebelum adanya C# Microsoft mengembangkan J++ dengan maksud mencoba membuat Java agar berjalan pada platform Windows, karena adanya masalah dari pihak luar maka Microsoft menghentikan proyek J++ dan beralih untuk mengembangkan bahasa baru yaitu C#.

Dalam mengembangkan aplikasi ataupun fungsi-fungsi lain yang berbasis C#, kita perlu IDE (Integrated Development Environment) untuk membantu pekerjaan kita. Di Windows kita dapat menggunakan Visual Studio yang memiliki fitur yang sangat lengkap. Akan tetapi tidak semua dari kita memakai OS (Operating System) Windows dan memiliki kapasitas harddisk yang memadai karena perlu diketahui Visual Studio ini sangat memakan tempat di harddisk kita. Karena itu ada IDE alternatif Visual Studio yaitu Xamarin Studio yang bisa dipakai lintas platform yang fiturnya menurut saya tidak kalah hebatnya dengan Visual Studio dan lebih dari cukup untuk membangun aplikasi C#. Selain itu ada juga tool seperti SharpDevelop namun sepertinya hanya bisa untuk platform Windows. Oh iya bagi pembaca yang memakai OS selain windows dapat memakai Mono untuk pengganti .NET Framework. Dalam pemrograman C# (mungkin juga berlaku untuk beberapa pemrograman lainnya) memiliki 5 struktur dasar yang harus diingat yaitu:

a. Resource atau library

   Struktur pertama ini merupakan pendefinisian library apa yang harus ada pada program kita atau library apa yang kita impor.

b. Namespace

Struktur kedua ini adalah nama dari project kita.

c. Nama Class

Struktur ketiga ini berbicara tentang apa nama dari Class yang kita buat dan bisa juga langsung diberi penanda seperti Main Class yang menandakan bahwa Class tersebut Class utama.

d. Deklarasi Method

Struktur keempat ini merupakan pendeklarasian method sebagai awalan untuk menjalankan method atau perintah yang ada di dalamnya, jika didefinisikan dengan "Main" maka method tersebut yang dijalankan pertama kali oleh compiler.

e. Method atau Command

Struktur kelima adalah method atau perintah yang kita berikan untuk di eksekusi oleh compiler.

4. *Android SDK (Soft Development Kit)*

Androi SDK adalah *tools* API *(Aplikation Programming Interface)* yang diperlukan untuk mulai mengembangkan aplikasi ada *Platform* Android menggunakan bahasa pemograman Java. Android merupakan *subset* perangkat lunak untuk ponsel yang meliputi system operasi, *middleware* dan aplikasi kunci yang di – *releace* Google. Saaat ini disediakan Android SDK (*Software Development Kit*) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemograman Java. Sebagai Platform aplikasi yang dibutuhkan yang bukan merupakan aplikasi bawaan *Handphone / Smarthphone.* Beberapa fitur Android yang paling penting adalah :

1. Framework Aplikasi yang mendukung penggantian komponen dan reusable.
2. Mesin virtual Dalvik dioptimalkan untuk perangkat mobile
3. Integrated browser berdasarkan engine open sourse WebKit.
4. Grafis yang dioptimalkan da didukung oleh library garafis 2D, grafis 3D berdasarkan spesifikasi opengl ES 1,0 (Opsional akselerasi hardware)
5. SQLite untuk penyimpanan data
6. Media support yang mendukung audio, video, dan gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF) GSM Telephony (tergantung hardware)
7. Bluetoot, EDGE, #G, dan WiFi (tergantung hardware)
8. Lingkungan Development yang lengkap dan kaya termasuk perangkat emulator, tools untuk debugging, profil dan kinerja memori dan plugin untuk IDE Eclips. (Nazruddin Safaat H.2012.5)

5. *ADT (Android Development Tools)*

Android Development Tools (ADT) adalah Plugin yang didesain untuk IDE Eclips yang memeberikan kita kemudahan dalam mengembangkan aplikasi dengan menggunakan IDE Eclips. Dengan menggunakan ADT untuk eclips akan memudahkan kita dalam pembuatan aplikasi project android, membuat GUI aplikasi, dan menambahkan komponen – komponen yang lainnya, begitu juga kita dapat melakukan running aplikasi menggunakan Android SDK melalui eclipse. Dengan ADT juga kita dapat melakukan pembuatan package android (apk) yang digunakan untuk distribusi aplikasi android yang kita rancang. (Nazruddin Safaat H.2012.6)

6. *Android*

Android merupakan system operasi untuk mobile device yang awalnya dikembangkan oleh Android Inc. Perusahaan ini kemudian dibeli oleh Google pada tahun 2005. Kemudian untuk mengembangkan Android dibentuklah Open Handset Alliance yang merupakan gabungan dari 34 perusahaan peranti keras, lunak, dan telekomunikasi termasuk Google, HTC, Intel, Motorola, Qualcomm, TMobile, dan NVidia. (Ayu Yudha.2012:1)

Berikut tahapan peluncuran sistem operasi Android yang diciptakan Google :

1. Android versi 1.1

   Pada 9 Maret 2009, Google merilis Android versi 1.1. Android versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, voice search (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan email.

2. Android Versi 1.5 (Cupcake)

   Pada pertengahan Mei 2009, Google kembali merilis telepon seluler dengan menggunakan Android dan SDK (Software Development Kit).

3. Android Versi 1.6 (Donut)

   Donut (versi 1.6) dirilis pada September 2009

4. Android Versi 2.0/2.1 (Eclair)

   Pada 3 Desember 2009 kembali diluncurkan ponsel Android dengan versi 2.0/2.1 (Eclair),

5. Android Versi 2.2 (Froyo: Frozen Yoghurt)

   Pada 20 Mei 2010, Android versi 2.2 (Froyo) diluncurkan. Perubahan-

perubahan umumnya dukungan Adobe Flash 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat.

6. Android Versi 2.3 (Gingerbread)

   Pada 6 Desember 2010, Android versi 2.3 (Gingerbread) diluncurkan. Perubahannya yaitu peningkatan kemampuan permainan (gaming), peningkatan fungsi copy paste.

7. Android Versi 3.0 (Honeycomb)

   Dirilis Februari 2011 sebagai android 3.0 revisi 1 serta android versi 3.0 revision 2 telah dirilis pada juli 2011.

8. Android versi 4.0 (ICS :Ice Cream Sandwich)

   Diumumkan pada tanggal 19 Oktober 2011, membawa fitur Honeycomb untuk smartphone.

9. Android versi 4.1 (Jelly Bean)

   Dirilis tahun 2012 ini. Ada sejumlah keunggulan yang ditawarkan Android 4.1 sebagai berikut :

   b. Full chrome browser.
   c. Dual boot.
   d. File Manager yang lebih baik.
   e. Keyboard Virtual
   f. Sistem operasi lebih ringan.
   g. Hemat Baterai. (Nazruddin Safaat H.2012.11)

10. Android vesi 4.4 (KitKat)

    Google mengumumkan Android 4.4 KitKat (dinamai dengan izin

dari Nesle dan Hershey) pada 3 September 2013, dengan tanggal rilis 31 Oktober 2013. Sebelumnya, rilis berikutnya setelah Jelly Bean diperkirakan akan diberi nomor 5.0 dan dinamai Key Lime Pie. Ada sejumlah fitur yang ditawarkan Android 4.4 sebagai berikut

a. Pembaruan antarmuka dengan bar status dan navigasi transparan pada layar depan
b. Optimasi kinerja pada perangkat dengan spesifikasi yang lebih rendah
c. NFC Host Card Emulation sebagai emulator kartu pintar
d. Perluasan fungsionalitas bagi layanan pendengar notifikasi
e. API umum untuk mengembangkan dan mengelola klien pesan teks, kemampuan untuk menentukan aplikasi SMS standar

7. *UML (Unified Modelling Language)*

Sora (2018), Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalamindustri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UMLmenawarkan sebuah standar untuk merancang model sebuah sistem.

UML adalah sekumpulan alat yang digunakan untuk melakukan abstraksiterhadap 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 kode program *UML* juga dapat menjadi alat bantu untuk transfer ilmu tentang

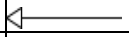
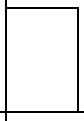
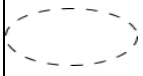
sistem atau aplikasi yang akan dikembangkan dari satu *developer* ke *developer* lainya. Tidak hanya antar *developer* terhadap orang bisnis dan siapapun dapat memahami sebuah sistem dengan adanya *UML*. UML diciptakan oleh Object Management Group yang diawali dengan versi 1.0 pada Januari 1997. Dalam pengembangan berorientasi objek ada beberapa prinsip yang harus dikenal:*Object, Class, Abstraction, Encapsulation, Inheritance dan Polymorphism*. Dalam *UML* sendiri terdapat beberapa diagram yaitu :

a. Use Case Diagram

Use Case diagram menggambarkan *fungsionalitas* yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah *Use Case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use Case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-create sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah *entitas* manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. Adapun simbol-simbol *Use Case Diagram* antara lain :

Tabel 2. 1. Simbol *Use Case* Diagram

No	Gambar	Nama	Keterangan
1		*Actor*	Menspesifikasikan himpuan peran yang pengguna *Main*kan ketika berinteraksi dengan *Use Case*.

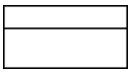
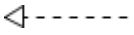
2		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*).
3		*Generalization*	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang 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		*System*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
9		*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan 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 jenis diagram struktur dalam Unified Modeling Language (UML) yang secara visual menggambarkan struktur statis suatu sistem perangkat lunak .

Adapun simbol-simbol *Class Diagram* antara lain :

Tabel 2. 2. Simbol *Class* Diagram

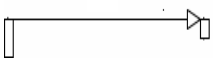
No	Gambar	Nama	Keterangan
1		*Generalization*	Hubungan dimana objek anak *(descendent)* berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
2		*Nary Association*	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		*Class*	Himpunan dari objek-objek yang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

Sequence Diagram atau diagram urutan adalah model visual yang menampilkan interaksi objek objek dalam suatu sistem secara rinci,menggambarkan urutan pesan yang dikirim antar objek dan waktu eksekusinya.

Adapun simbol-simbol *Sequence* Diagram antara lain :

Tabel 2. 3. Simbol *Sequence* Diagram

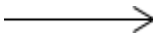
No	Gambar	Nama	Keterangan
1		*LifeLine*	Objek *entity*, antarmuka yang saling berinteraksi.
2		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

d. StateChart Diagram

Statechart Diagram adalah diagram dalam UML yang digunakan untuk memodelkan perilaku dinamis suatu objek atau sistem dengan menggambarkan urutan keadan (state) yang dialami suatu objek dan transisi dari suatu keadan yang dipicu oleh peristiwa (event).

Adapun simbol-simbol *StateChart Diagram* antara lain :

Tabel 2. 4. Simbol *StateChart* Diagram

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

e. Activity Diagram

Activity Diagram adalah diagram dalam UML yang menampilkan alur kerja ,proses atau aliran aktivitas dalam suatu sistem,yang mirip dengan flowchart tetapi lebih rinci,karena dapat menggambarkan aktivitas parallel,bercabang,atau keputusan dalam alur proses. Adapun simbol-simbol Activity Diagram antara lain :

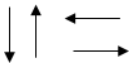
Tabel 2. 5. Simbol *Activity* Diagram

No	Gambar	Nama	Keterangan
1		*Activity*	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		*Activity Final Node*	Bagaimana objek dibentuk dan dihancurkan
5		*Fork Node*	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

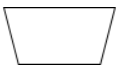
8. *Flowchart*

Rahmad Arifianto, (2017). *Flowchart* adalah adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses (instruksi) dengan proses lainnya dalam suatu program. Berikut ini adalah beberapa simbol yang digunakan dalam menggambar suatu *Flowchart*:

Tabel 2. 6. Simbol *Flowchart*

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

Lanjutan Tabel 2.6

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

Lanjutan Tabel 2.6

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

B. Penelitian Terdahulu

Melihat dari sejumlah judul dan tema yang berkaitan dengan penelitian yang dilakukan, diperoleh persamaan dan perbedaan dari penelitian yang akan dilakukan. Tujuannya membuktikan bahwa penulisan penelitian asli dan bukan duplikasi dari penelitian lain, seperti berikut ini :

1. Nadira, Sujaini dan Pratiwi (2016) dalam penelitianya yang berjudul "Implementasi Augmented Reality Pada Brosur Teknik Informatika Universitas Tanjungpura Menggunakan Metode Marker " saat ini brosur Teknik Informatika Universitas Tanjungpura masih berisi gambar-gambar dari kampus itu sendiri, tetapi gambar-gambar tersebut masih kurang memberikan informasi yang jelas sehingga calon mahasiswa sulit untuk membayangkan gambaran dari kampus Teknik Informatika Universitas Tanjungpura dan

akhirnya calon mahasiswa tidak terlalu tertarik untuk memilih Teknik Informatika Universitas Tanjungpura. Manfaat dari penerapan Augmented Reality pada brosur Teknik Informatika Universitas Tanjungpura yaitu dapat memberikan gambaran yang jelas karena dilengkapi dengan gambar bangunan kampus dalam bentuk 3 dimensi. Dalam penelitian ini menggunakan metode Marker Based Tracking. Hasil yang didapat dari penelitian ini adalah peneliti menguji aplikasi yang dibuat dan hasilnya berjalan lancar pada handphone android dan pengguna menyatakan setuju bahwa aplikasi Augmented Relaity brosur dapat memperjelas informasi pada brosur Teknik Informatika Universitas Tanjungpura.

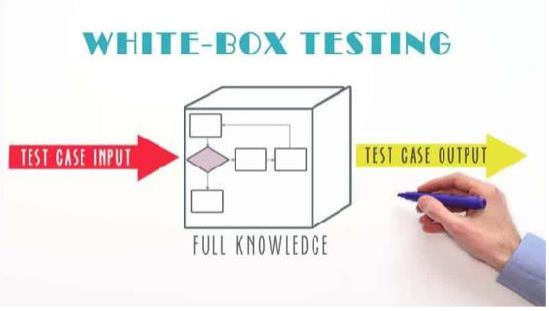
2. Hermawan dan Hariadi (2015) dalam penelitianya yang berjudul "Pemanfaatan Augmented Reality Sebagai Media Informasi Kampus Menggunakan Brosur" Manfaat dari penerapan Augmented Reality pada brosur kampus yaitu pembaca dapat melihat bagian brosur Augmented Relaity dengan detail serta lingkungan pada brosur juga akan terasa lebih hidup dengan adanya animasi pendukung seperti gedung, pohon, dan lain sebagainya. Metode penelitian yang akan digunakan dalam pembuatan brosur berbasis Augmented Reality yaitu terdiri dari empat proses yaitu perencanaan, pengumpulan data, analisis dan dokumentasi hasil yang telah dibuat. Hasil yang didapat dari penelitian ini adalah aplikasi berjalan lancar dan telah diuji menggunakan komputer dan dapat menampilkan objek 3 dimensi pada brosur yang dapat memberikan informasi yang belum ada pada brosur sebelumnya.

3.

C. Metode Pengujian

a. *White Box*

White box testing adalah suatu pengujian oleh *software* atau aplikasi dengan melihat modul untuk menganalisis kode program ada yang salah atau tidak. Bila suatu modul yang diproduksi tidak memenuhi syarat, kode akan dikompilasi ulang dan periksa lagi hingga mencapai apa yang diharapkan dalam arti bahwa *white box* testing menguji dengan cara melihat pure code dari suatu perangkat lunak atau aplikasi yang diuji tanpa mempedulikan tampilan atau UI dari aplikasi tersebut.



Gambar 2. 1 *White Box Testing*

1) Teknik

Pada white box testing ada beberapa teknik yang digunakan. Berikut akan kami sajikan :

a) *Basis path testing* adalah metode yang memungkinkan perancang tast case untuk membuat pengukuran kompleksitas logika dari rancangan prosedural dan menggunakan pengukuran untuk mendefinisikan himpunan basis dari jalur eksekusi.

b) *Flow graph* merupakan notasi sederhana guna mempresentasi control flow.

c) *Cyclomatic complexity* yang digunakan guna mengetahui jumlah path yang

perlu dicari. Di mana *cyclomatic complexity* merupakan metric perangkat lunak yang menyediakan ukuran kuantitatif dari kompleksitas logika program.

d) *Branch coverage* merupakan pengujian yang didesain agar masing-masing branch code yang paling tidak diuji satu kali.

e) *Loop testing* merupakan testing pada berbagai looping/perulangan yang terdapat pada program, seperti *for, while, dan do-while*. Dengan pengujian ini Anda bisa mengecek kondisi perulangan, sudahkah berjalan dengan benar ataupun tidak.

f) *Condition coverage* merupakan teknik pengujian yang bertujuan guna melakukan uji semua kode untuk menghasilkan value FALSE atau *TRUE. Teste*r bisa memastikan software dalam beroperasi dan menghasilkan output sesuai *input user*.

g) *Multiple condition coverage* merupakan teknik yang digunakan guna menguji semua kombinasi beragam kode yang digunakan dalam berbagi kondisi.

h) *Statement coverage* merupakan teknik yang minimal 1 kali dilakukan guna menguji masing-masing statement pada *software*. Anda bisa mengetahui beberapa code yang error dengan pengujian ini sehingga bisa segera memperbaikinya.

2) Trik Pengujian

White box testing merupakan pengujian dengan dasar pada pengecekan di dalam detail desain dari suatu rancangan pemrograman untuk membagi

pengujian. Dalam membuat pengujian white box maka tester perlu memiliki pemahaman terkait pemrograman. Sebab untuk memahami kode sumber yang dipelajari dan harus memiliki pandangan global tentang fungsi aplikasi serta elemen yang dirancangnya. Metode white test ini dapat diterapkan untuk tes integrasi, tes unit dan tes sistem.

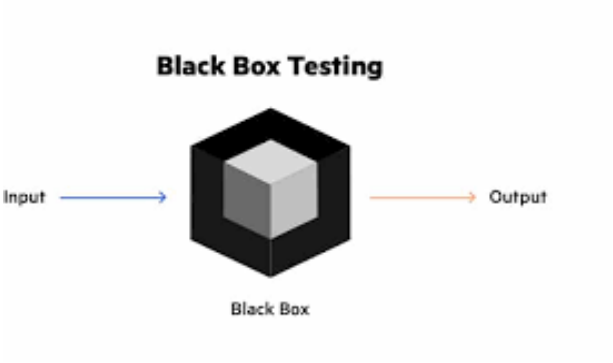
3) Contoh *White Box*

Contoh *white box* testing menggunakan teknik *cyclomatic*, di mana pada dasarnya dengan cara menghitung jumlah siklomatik yaitu menghitung jumlah conditional I predikat dan tambahkan 1. Misalnya terdapat lima *conditional* maka *cyclomaticnya* adalah 6. Berikut cara menghitungnya :

a) 1-2-3-4-5-10 yaitu suatu properti yang dimiliki orang lain, tidak mempunyai uang untuk sewa
b) 1-2-3-4-6-10 suatu properti yang dimiliki orang lain dan membayar sewa
c) 1-2-3-10 yaitu properti yang dimiliki oleh pemain
d) 1-2-7-10 yaitu properti yang tersedia dan tidak memiliki cukup uang
e) 1-2-7-8-10 yaitu properti yang tersedia, memiliki uang dan tidak ingin membelinya
f) 1-2-7-8-9-10 yaitu properti yang tersedia, memiliki uang dan membelinya.

(Sumber : *https://itbox.id/blog/white-box-testing-adalah/*)

b. *Black Box*



Gambar 2. 2 *Black Box Testing*

Pengujian *black box* adalah pengujian yang dilakukan semata-mata dengan memantau hasil eksekusi melalui data pengujian dan memvalidasi fungsionalitas perangkat lunak. Jadi, mirip dengan melihat kotak hitam, kita hanya bisa melihat tampilan luarnya saja dan tidak tahu apa yang ada di balik bungkus hitamnya. Mirip dengan pengujian *black box*, hanya mengevaluasi tampilan luar (antarmuka) dan fungsionalitasnya. Tanpa mengetahui apa yang terjadi dalam proses detail (hanya mengetahui input dan output).

1) Jenis Pengujian

Pada suatu black box testing terdapat tiga jenis pengujian, berikut akan kami jelaskan :

a) *Functional testing*

Merupakan suatu proses pengujian pada fitur spesifik atau fungsi pada suatu software. Pengujian ini bisa dilakukan guna memastikan bahwa user tidak dapat masuk pada aplikasi yang diuji tanpa informasi sehingga program tersebut terjaga keamanannya.

b) *Non functional testing*

Memiliki fungsi yang tidak kalah dari functional testing, di mana Non functional testing merupakan suatu pengujian pada beberapa aspek tambahan non fungsional. Bila functional testing menguji software menjalankan suatu tugas atau peringkat.

c) *Regression testing*

Jenis ini merupakan pengujian yang dilakukan pada software untuk mengamati apakah terjadinya kemunduran atau regresi ketika software diperbaharui. Dalam regression test dilakukan pengecekan aspek non fungsional maupun fungsional.

2) Teknik Menguji *Black Box* Dengan *Software*

Tulisan di atas sudah menjelaskan apa itu black box testing dan contohnya. Namun *black box* testing memiliki beberapa teknik dalam menguji suatu software. Berikut akan kami jelaskan :

a) *Equivalence partitioning* merupakan cara kerja dengan melakukan pembagian atau partition menjadi beberapa partisi dari input data.

b) *Fuzzing merupakan* teknik untuk mencari gangguan atau bug dari software dengan cara injeksi data yang terbilang sesi semi otomatis atau cacat.

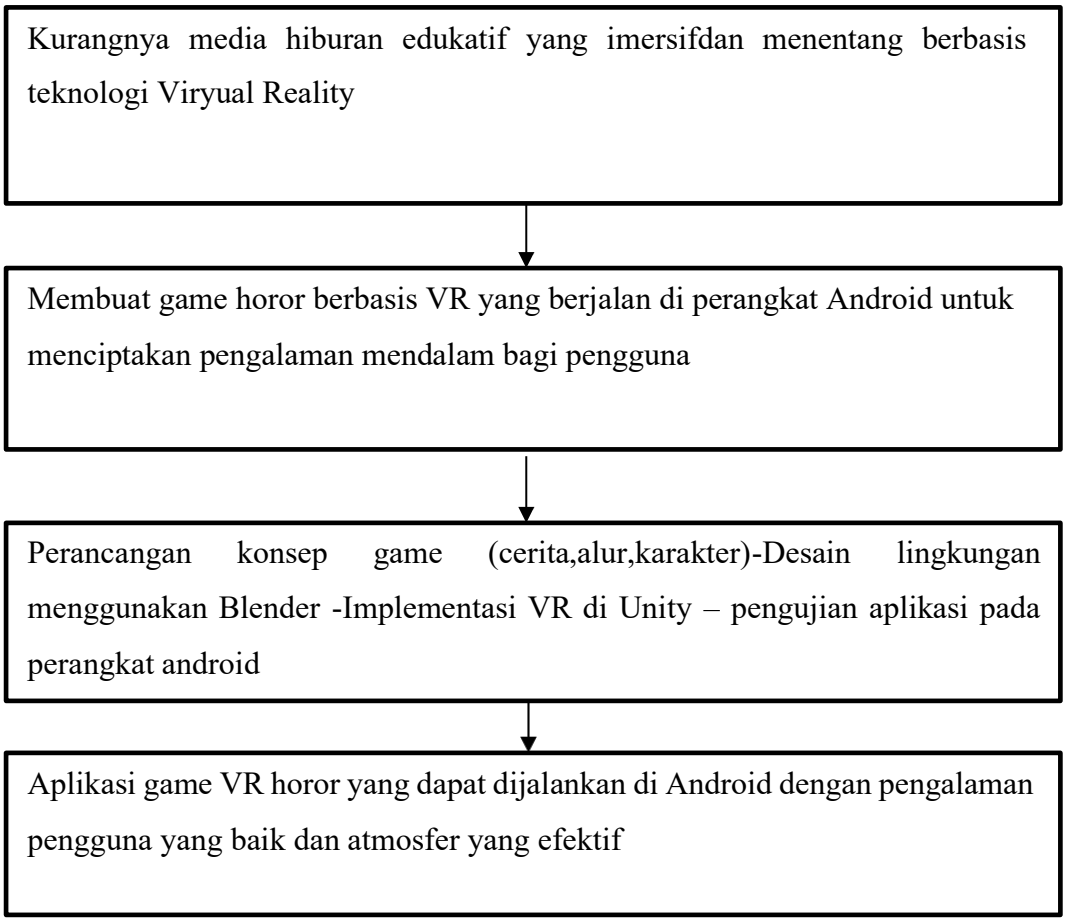
c) *Boundary value analysis* merupakan teknik yang lebih fokus pada *boundary* di mana jika ada error dari sisi atau luar baik minimum maupun maksimum pada pada software yang didapat.

d) *Orthogonal array testing* merupakan teknik yang digunakan jika input domain yang relatif ukurannya kecil tetapi cukup berat jika digunakan dalam skala besar.

e) *Cause effect graph* merupakan teknik yang digunakan di mana graphic sebagai pacuannya. Dalam teknik ini di mana graphic menggambarkan relasi antara penyebab dan efek dari error tersebut.

f) *State transition* merupakan testing yang digunakan untuk mengecek apakah kondisi dari navigasi dan mesin dalam bentuk grafik. (Sumber : *https://itbox.id/blog/black-box-testing-adalah/)*.

D. Kerangka Pikir



BAB III

METODE PENELITIAN

A. Waktu Penelitian

Proses penelitian Waktu yang dipergunakan untuk penelitian ini 2 (dua) bulan dimulai Juni sampai dengan juli.

Tabel 3. 1. Jadwal Pelaksanaan Penelitian

No.	Uraian Kegiatan	Bulan 2025		
		Juni	Juli	Agustus
1	Studi Literatur			
2	Pembuatan Aplikasi			
3	Pengujian Aplikasi			
4	Hasil Pengujian			

B. Jenis Penelitian

Jenis penelitian yang digunakan dalam skripsi ini adalah penelitian rekayasa perangkat lunak (software engineering research) yang bersifat pengembangan (development research). Penelitian ini difokuskan pada proses perancangan dan pembangunan aplikasi game berbasis Virtual Reality dengan tema *House of Horror* yang berjalan pada perangkat Android. Pendekatan ini digunakan untuk menghasilkan produk berupa aplikasi game yang bersifat interaktif dan imersif, dengan menerapkan teknologi VR dan perangkat mobile. Dalam pelaksanaannya, penelitian ini juga memanfaatkan metode rekayasa perangkat lunak untuk

merancang sistem, membuat aset 3D, melakukan implementasi di dalam game engine (seperti Unity), serta melakukan pengujian terhadap fungsionalitas dan pengalaman pengguna. Dengan demikian, jenis penelitian ini bertujuan tidak hanya untuk mengembangkan sebuah aplikasi, tetapi juga untuk mengevaluasi kualitas dari produk tersebut berdasarkan aspek teknis dan pengalaman pengguna.

C. Metode Pengumpulan Data

Pengumpulan data dilakukan untuk mengumpulkan seluruh informasi yang terkait dan mendukung pelaksanaan penelitian penelitian adalah kajian pustaka, metode pengumpulan data dengan cara mencari buku yang ada di perpustakaan atau mencari di internet untuk mendapatkan referensi yang serupa dengan penelitian.

D. Alat Dan Bahan Penelitian

1. **Laptop/PC**
   a) Spesifikasi minimum:
      1) Prosesor: Intel Core i5 atau setara
      2) RAM: 8 GB
      3) GPU: NVIDIA GeForce GTX atau setara
      4) Storage: SSD 256 GB (disarankan)
2. **Smartphone** Android
   a) Dukungan giroskop (gyroscope)
   b) Versi Android minimal 9.0 (Pie)

c) RAM minimal 4 GB

3. VR Headset Mobile

a) Google Cardboard atauheadsetVR mobile lainnya yang kompatibel dengan Android

5. Bahan Penelitian (Perangkat Lunak dan Aset):

a. **Blender**

Untuk modeling aset 3D (ruangan, objek horor, karakter)

b. **Unity 3D**

Game engine utama untuk pengembangan aplikasi VR

c. **Android Studio + Android SDK**

Untuk kompilasi dan pembuatan file APK Android

d. **GIMP / Adobe Photoshop**

Untuk membuat dan mengedit tekstur objek

e. **Audacity / Freesound Audio Pack**

Untuk editing dan konversi efek suara horor (suara pintu, jeritan, angin, dll.)

f. **VR SDK (Google VR SDK atau Unity XR Toolkit)**

Untuk mendukung fungsi Virtual Reality dalam Unity

g. **Aset Pendukung**

1) Tekstur horor (tembok retak, darah, kayu lapuk)

2) Efek suara dan musik latar horor

3) Font dan ikon bertema menyeramkan

E. Tahap Penelitian

1. Identifikasi Masalah dan Studi Literatur
   a) Mengidentifikasi kurangnya media hiburan interaktif berbasis teknologi VR yang menarik dan mudah diakses.
   b) Melakukan studi pustaka terkait teknologi Virtual Reality, game development, Unity, Android, dan pengalaman pengguna.
2. Perumusan Tujuan dan Spesifikasi Sistem
   a) Menentukan tujuan dan ruang lingkup aplikasi game VR horor yang akan dikembangkan.
   b) Merancang kebutuhan fungsional dan non-fungsional sistem.
3. Perancangan Game
   a) Membuat alur cerita (storyline), konsep gameplay, dan rancangan antarmuka pengguna.
   b) Mendesain environment game, karakter, dan objek dalam Blender.
   c) Menentukan efek suara, musik latar, dan pencahayaan untuk menciptakan atmosfer horor.
4. Pengembangan Aplikasi Game
   a) Mengimpor aset 3D ke dalam Unity.
   b) Menerapkan sistem kontrol VR berbasis head-tracking menggunakan Google VR SDK atau Unity XR Toolkit.
   c) Menyusun skenario permainan dan integrasi logika interaksi.

5. Uji Coba Aplikasi
   a) Melakukan pengujian fungsionalitas aplikasi pada perangkat Android dan VR headset mobile.
   b) Menguji kenyamanan pengguna (user experience) melalui observasi dan kuesioner, seperti System Usability Scale (SUS).
6. Evaluasi dan Analisis Hasil
   a) Menganalisis hasil pengujian untuk menilai performa aplikasi dan kepuasan pengguna.
   b) Mengidentifikasi kekurangan dan area yang perlu dikembangkan.
7. Penyusunan Laporan Skripsi
   a) Menyusun hasil penelitian ke dalam bentuk dokumen skripsi yang sistematis dan sesuai format akademik.
   b) Menyertakan dokumentasi proses, desain, pengujian, serta hasil akhir aplikasi.

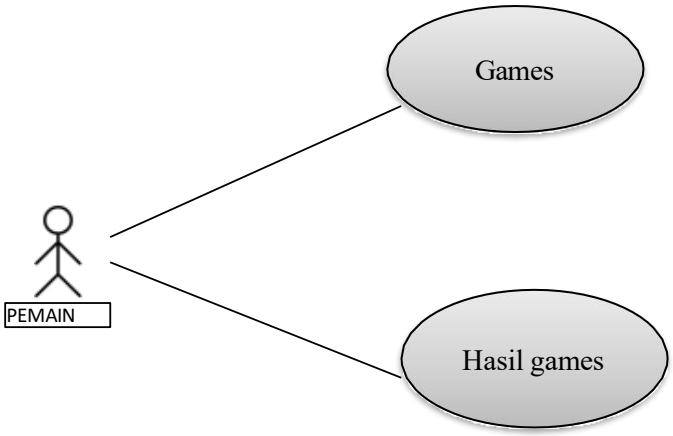
F. Metode Pengujian

Untuk melakukan pengujian dari sistem yang dibangun yaitu penerapan menggunakan Android Studio dengan bahasa pemrograman C# dibuat sebuah mekanisme pengujian yang tercantum dalam rencana pengujian yang di golongkan oleh jenis pengguna. Pengujian dilakukan dengan menguji setiap proses yang ada pada sistem. Pengujian ini dilakukan secara *WhiteBox*, yaitu pengujian terhadap cara kerja perangkat lunak itu sendiri yaitu prosedur programnya (*basis path*) atau proses *looping* (pengulangan). Dan juga *BlackBox*, yaitu pengujian yang dilakukan

hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak.

G. Desain Sistem

1. Sistem Yang Berjalan



Gambar 3. 1. Sistem Yang Berjalan

a. Aktor (User)

1) Digambarkan dengan ikon manusia di sebelah kiri.
2) Dalam konteks ini, user adalah pemain yang menggunakan aplikasi Virtual Reality Games House of Horror berbasis Android.
3) User berperan sebagai pihak yang berinteraksi langsung dengan sistem untuk menjalankan permainan dan melihat hasilnya.

b. Use Case "Games"

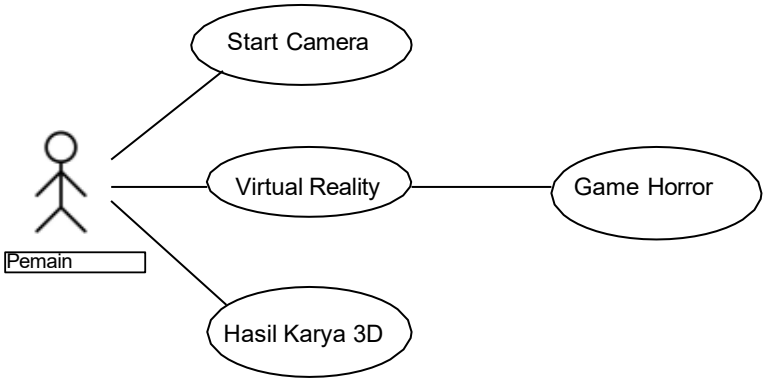
1) Obyek berbentuk oval dengan label *games*.

2) Menunjukkan bahwa user dapat memainkan game VR House of Horror yang sudah dirancang.
3) Dalam implementasinya, aktivitas ini mencakup menjalankan aplikasi, masuk ke dalam lingkungan virtual, dan menjelajahi permainan horor berbasis VR.

c. Use Case "Hasil Games"
1) Obyek berbentuk oval dengan label *Hasil games*.
2) Menunjukkan bahwa setelah memainkan permainan, user dapat melihat atau memperoleh hasil dari permainan.
3) Hasil ini bisa berupa skor, level yang dicapai, maupun feedback dari performa permainan user di dalam aplikasi.

d. Hubungan (Relasi)
1) Garis dengan panah dari aktor ke masing-masing use case menandakan bahwa user adalah pihak yang memicu eksekusi use case.
2) Artinya, tanpa adanya user, kedua proses ini (memainkan game dan melihat hasil game) tidak akan berjalan.

2. Sistem Yang Diusulkan



Gambar 3. 2. Sistem Yang Diusulkan

Aktor	Use Case / Fungsi	Deskripsi Kegiatan
User	Membuka Aplikasi	User menjalankan/membuka aplikasi *Game House Horror* pada perangkatnya.
User	Menu Mulai	User mengakses menu utama untuk memulai permainan.
User	Menampilkan Game House Horror	Aplikasi menampilkan tampilan awal *Game House Horror* setelah menu mulai dipilih.
User	Mengeksplorasi 3D Game House	User dapat menjelajahi rumah horor dalam bentuk 3D.
User	Menampilkan Informasi Game House Horror	Sistem menampilkan informasi detail mengenai *Game House Horror* seperti ruangan, karakter, dan fitur.

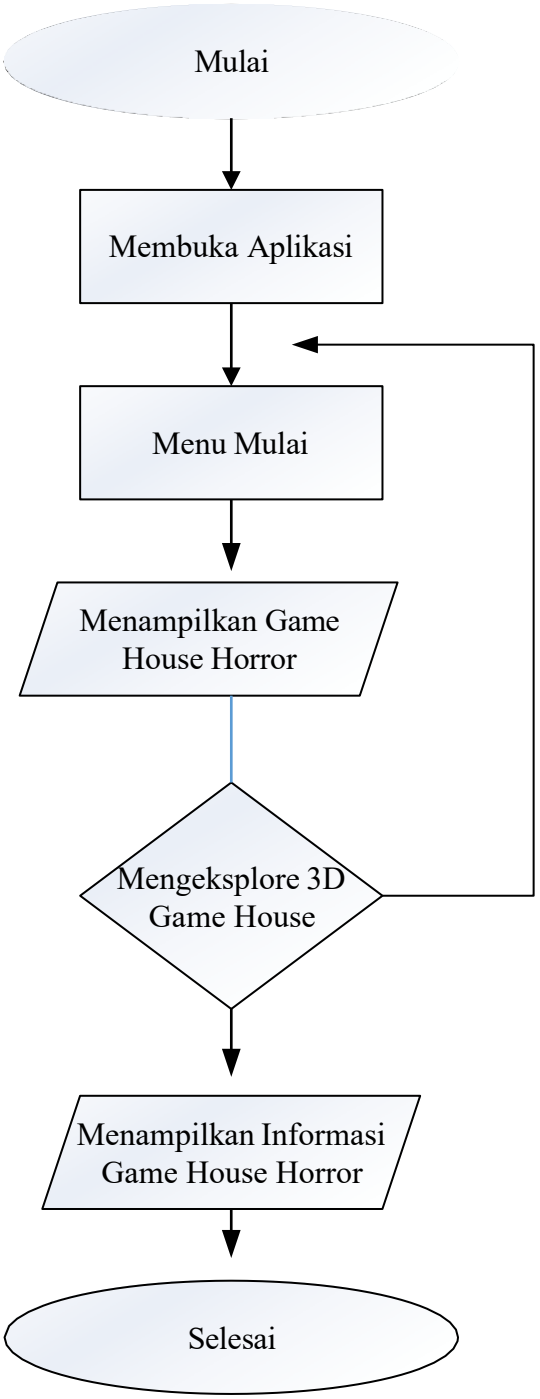
Use case diagram yang ditampilkan menggambarkan interaksi antara user (pengguna) dengan sistem Virtual Reality Games *House of Horror* berbasis Android. Dalam diagram ini, user merupakan aktor utama yang berinteraksi langsung dengan beberapa fitur utama dari sistem.

Pertama, user dapat melakukan aksi "Start Camera" yang berfungsi sebagai awal untuk mengaktifkan sistem VR, dimana kamera mungkin digunakan untuk navigasi atau sekadar membuka tampilan VR. Aksi ini kemudian mengarah ke proses "Virtual Reality", yang menunjukkan bahwa setelah kamera aktif,

sistem akan masuk ke dalam mode realitas virtual.

Selanjutnya, user juga dapat langsung mengakses fitur "Game Horror", yang terhubung erat dengan modul Virtual Reality. Hal ini menunjukkan bahwa game horror berjalan dalam lingkungan VR, menciptakan pengalaman bermain yang imersif. Selain itu, dari interaksi dengan game horror dan virtual reality tersebut, sistem menghasilkan "Hasil Karya 3D", yang merujuk pada visualisasi atau simulasi objek dan lingkungan dalam format tiga dimensi sebagai bagian dari pengalaman bermain. User juga dapat berinteraksi langsung dengan hasil karya 3D, yang kemungkinan melibatkan visualisasi karakter, ruangan menyeramkan, atau objek-objek interaktif di dalam game. Semua use case ini saling terhubung dan menggambarkan alur fungsionalitas utama dalam game, mulai dari inisialisasi (start camera), menjalankan VR, memainkan game horror, hingga menikmati hasil visualisasi 3D yang dihasilkan oleh sistem. Secara keseluruhan, diagram ini menunjukkan alur sederhana namun terstruktur mengenai bagaimana pengguna menggunakan fitur-fitur utama dari aplikasi VR horror tersebut, dengan fokus pada pengalaman visual dan interaktif yang menjadi inti dari game berbasis Virtual Reality.

H. Flowchart

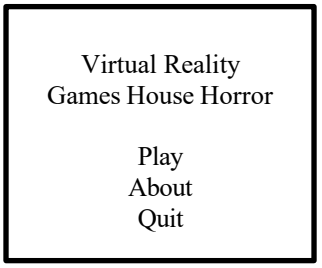
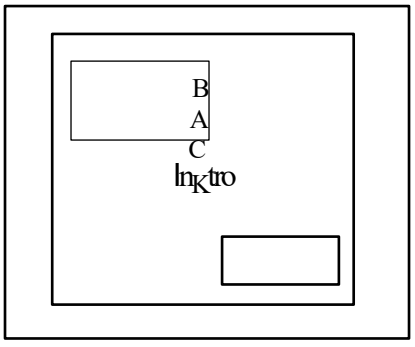


Flowchart di atas menggambarkan alur kerja aplikasi Game House Horror.

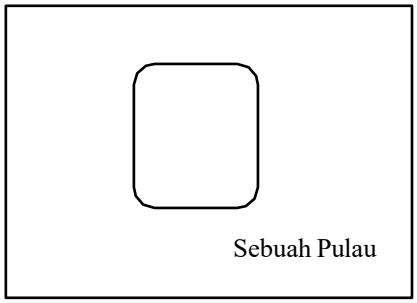
Proses dimulai ketika pengguna membuka aplikasi, kemudian diarahkan ke menu

utama atau menu mulai untuk memulai permainan. Setelah memilih menu mulai, sistem menampilkan tampilan awal Game House Horror yang siap dieksplorasi. Pada tahap berikutnya, pengguna dapat menjelajahi rumah horor dalam bentuk 3D untuk memperoleh pengalaman bermain yang interaktif. Terdapat alur balik ke menu utama yang memungkinkan pengguna kembali ke halaman awal jika ingin memulai ulang atau mengganti pilihan. Setelah eksplorasi dilakukan, sistem menampilkan informasi terkait Game House Horror seperti detail ruangan, karakter, atau fitur-fitur lain yang tersedia. Seluruh rangkaian proses diakhiri dengan tahap selesai sebagai tanda bahwa pengguna telah melalui semua tahapan dalam aplikasi ini.

I. Rancangan Interface

Scene	Keterangan
Scene 01 	**Menu Utama** Ketika pemain menjalankan game maka akan tampil menu utama. Dalam menu utama terdapat dua pilihan yaitu Play, Quit. Klik tombol Play untuk masuk ke dalam intro game. Klik tombol About untuk untuk menerima informasi. Klik tombol Quit untuk keluar dari aplikasi.
Scene 02 	**Menu Intro** Pada *scene* ini pengguna akan membaca isi di didalam menu intro awal mulai permainan Petualangan dimulai. Klik tombol back untuk kembali ke menu utama dan klik tombol skip untuk memulai permainan Petualangan.

Scene 03



Sedang didalam game

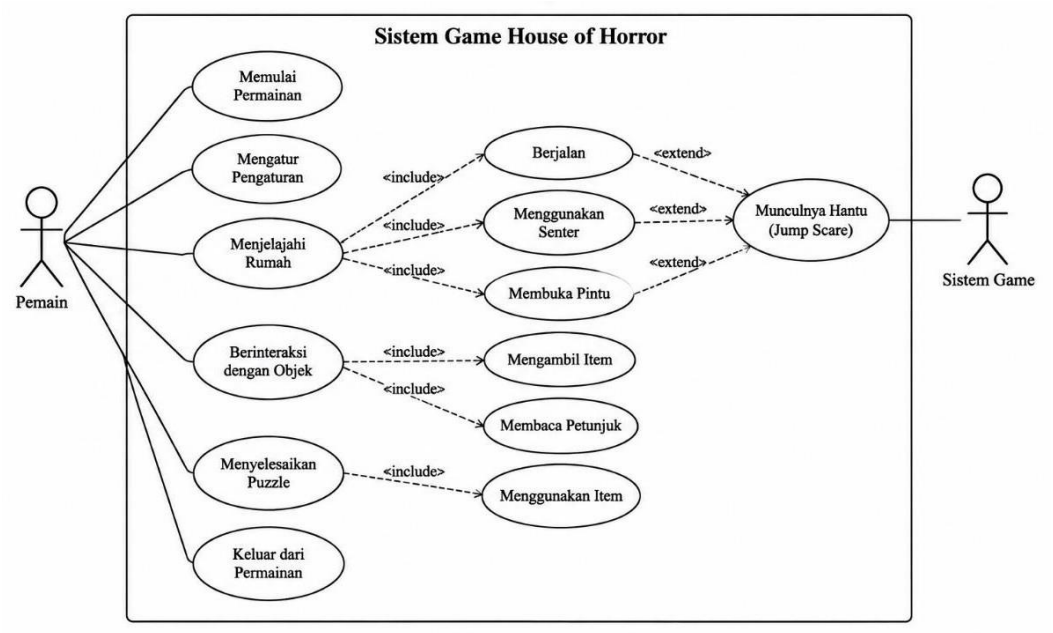
Pada *scene* ini pengguna dapat melakukan kegiatan Petualangan di sebuah tempat horror yang tersedia seperti membunuh monster, hewan untuk agar dapat bertahan hidup sampai waktu permainan Petualangan horror berakhir

BAB IV

HASIL DAN PEMBAHASAN

A. Rancangan Sistem

1. *Usecase Diagram*



Gambar 4.1. *Usecase Digram*

Use Case Diagram pada *Game House of Horror* menggambarkan interaksi antara aktor utama, yaitu Pemain, dengan sistem permainan. Pemain berperan sebagai pengguna yang menjalankan seluruh aktivitas dalam game, sedangkan sistem game bertugas menyediakan lingkungan permainan, mekanisme interaksi, serta kejadian-kejadian yang muncul selama permainan berlangsung.

Pada awal permainan, pemain dapat melakukan Memulai Permainan untuk masuk ke dalam dunia game dan Mengatur Pengaturan sesuai kebutuhan, seperti pengaturan suara atau kontrol permainan. Setelah permainan dimulai, pemain melakukan aktivitas utama berupa Menjelajahi Rumah, yaitu mengeksplorasi area rumah horor untuk menemukan petunjuk dan menyelesaikan tantangan. Aktivitas menjelajahi rumah memiliki beberapa proses pendukung (include), yaitu Berjalan untuk berpindah tempat, Menggunakan Senter untuk membantu melihat area gelap, serta Membuka Pintu untuk mengakses ruangan lain.

Selain melakukan eksplorasi, pemain juga dapat melakukan Berinteraksi dengan Objek yang tersedia di dalam lingkungan permainan. Interaksi tersebut mencakup aktivitas seperti Mengambil Item dan Membaca Petunjuk yang berfungsi membantu pemain memahami alur cerita serta menyelesaikan misi permainan. Pemain juga memiliki aktivitas Menyelesaikan Puzzle, yang membutuhkan penggunaan item tertentu (include) untuk membuka akses atau melanjutkan permainan ke tahap berikutnya.

Selama proses eksplorasi, sistem game dapat memunculkan kejadian Munculnya Hantu (Jump Scare) sebagai fitur tambahan (extend) yang terjadi ketika pemain berada pada kondisi atau area tertentu. Kejadian ini memberikan elemen kejutan dan meningkatkan suasana horor dalam permainan. Setelah menyelesaikan permainan atau ketika ingin berhenti, pemain dapat melakukan aktivitas Keluar dari Permainan untuk mengakhiri sesi permainan.

Secara keseluruhan, use case diagram tersebut menunjukkan bahwa sistem Game House of Horror berfokus pada pengalaman eksplorasi pemain melalui

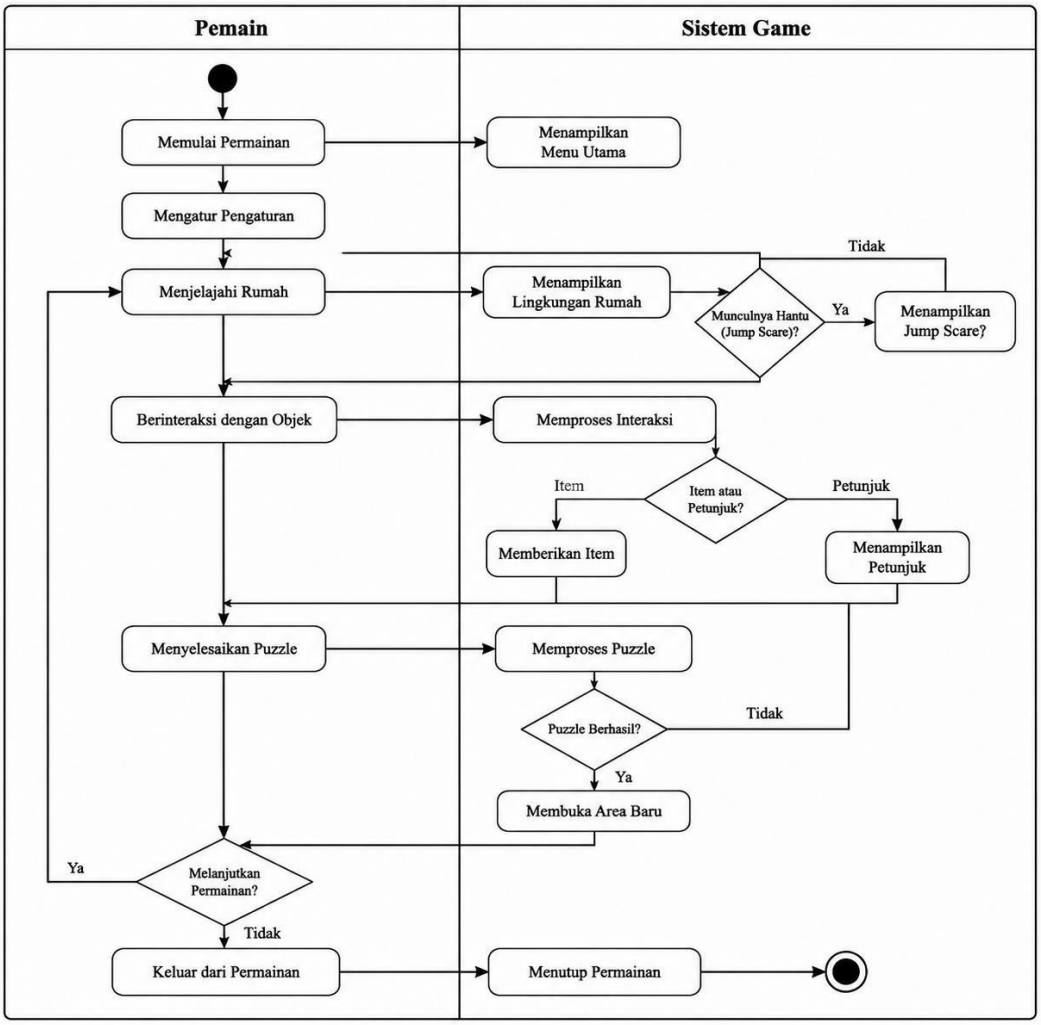
mekanisme berjalan, pencarian objek, pemecahan puzzle, serta kejadian horor berupa jump scare yang dikendalikan oleh sistem permainan.

2. Tabel *Usecase*

Tabel 4.1. Tabel *Usecase*

No	Nama Use Case	Aktor	Deskripsi
1	Memulai Permainan	Pemain	Pemain menjalankan game untuk masuk ke dalam sistem permainan dan memulai sesi bermain.
2	Mengatur Pengaturan	Pemain	Pemain melakukan pengaturan awal seperti suara, kontrol, atau konfigurasi permainan sebelum bermain.
3	Menjelajahi Rumah	Pemain	Pemain melakukan eksplorasi lingkungan rumah horor untuk mencari petunjuk, objek, dan menyelesaikan tujuan permainan.
4	Berjalan	Pemain	Pemain bergerak dari satu area ke area lain untuk menjelajahi lingkungan permainan.
5	Menggunakan Senter	Pemain	Pemain menggunakan senter sebagai alat bantu untuk melihat area yang gelap.
6	Membuka Pintu	Pemain	Pemain membuka pintu untuk memasuki ruangan baru atau melanjutkan eksplorasi.
7	Berinteraksi dengan Objek	Pemain	Pemain melakukan interaksi terhadap objek yang ditemukan selama permainan.
8	Mengambil Item	Pemain	Pemain mengambil benda atau item tertentu yang dapat digunakan untuk membantu menyelesaikan permainan.
9	Membaca Petunjuk	Pemain	Pemain membaca informasi atau petunjuk yang ditemukan untuk memahami alur cerita dan menyelesaikan tantangan.
10	Menyelesaikan Puzzle	Pemain	Pemain menyelesaikan teka-teki atau tantangan dalam permainan agar dapat melanjutkan ke tahap berikutnya.
11	Menggunakan Item	Pemain	Pemain menggunakan item yang telah diperoleh untuk menyelesaikan puzzle atau membuka akses tertentu.
12	Munculnya Hantu (Jump Scare)	Sistem Game	Sistem memunculkan karakter hantu secara tiba-tiba sebagai efek kejutan berdasarkan kondisi tertentu dalam permainan.
13	Keluar dari Permainan	Pemain	Pemain mengakhiri permainan dan keluar dari sistem game.

3. *Activity Digram*



Gambar 4.2. *Activity Diagram*

Activity Diagram pada *Game House of Horror* menggambarkan alur aktivitas antara Pemain sebagai pengguna dan Sistem Game sebagai pengelola proses permainan. Proses dimulai ketika pemain menjalankan permainan melalui aktivitas Memulai Permainan, kemudian sistem akan merespons dengan menampilkan menu utama. Setelah itu, pemain dapat melakukan Mengatur Pengaturan sebelum masuk

ke dalam permainan utama, seperti menyesuaikan kontrol atau pengaturan suara agar sesuai dengan kebutuhan pemain.

Setelah permainan dimulai, pemain melakukan aktivitas Menjelajahi Rumah. Pada tahap ini, sistem akan menampilkan lingkungan rumah horor yang menjadi area eksplorasi pemain. Selama proses eksplorasi, pemain dapat bergerak, menggunakan senter, membuka pintu, dan mencari berbagai objek yang tersedia di dalam rumah. Sistem akan melakukan pengecekan kondisi tertentu, salah satunya adalah kemungkinan munculnya hantu (jump scare). Apabila kondisi terpenuhi, sistem akan menampilkan kejadian jump scare untuk memberikan efek kejutan kepada pemain, sedangkan apabila kondisi tidak terpenuhi maka pemain dapat melanjutkan eksplorasi.

Selanjutnya, pemain dapat melakukan Berinteraksi dengan Objek yang ditemukan selama perjalanan. Sistem akan memproses interaksi tersebut dan menentukan jenis objek yang ditemukan, apakah berupa item atau petunjuk. Jika pemain menemukan item, sistem akan memberikan item tersebut agar dapat digunakan dalam permainan. Jika yang ditemukan berupa petunjuk, sistem akan menampilkan informasi yang membantu pemain memahami alur cerita atau menyelesaikan tantangan.

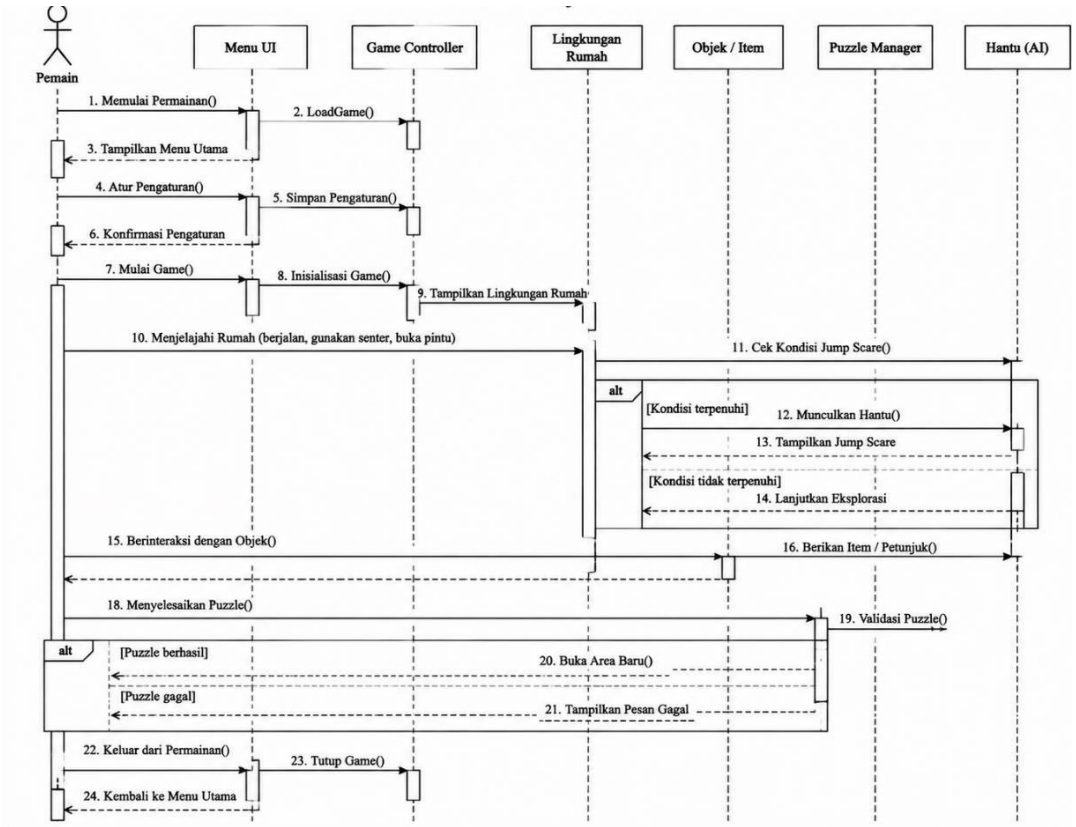
Setelah memperoleh informasi atau item yang diperlukan, pemain dapat melakukan aktivitas Menyelesaikan Puzzle. Sistem akan memproses puzzle dan melakukan pengecekan apakah puzzle berhasil diselesaikan atau belum. Jika puzzle belum berhasil, pemain harus kembali melakukan pencarian atau interaksi untuk memperoleh informasi tambahan. Jika puzzle berhasil diselesaikan, sistem akan

membuka area baru sehingga pemain dapat melanjutkan perjalanan ke tahap berikutnya.

Pada akhir proses, sistem akan memberikan pilihan kepada pemain apakah ingin melanjutkan permainan atau mengakhiri permainan. Jika pemain memilih melanjutkan, alur permainan kembali ke tahap eksplorasi rumah. Namun, jika pemain memilih berhenti, pemain melakukan aktivitas Keluar dari Permainan dan sistem akan menjalankan proses Menutup Permainan hingga sesi permainan selesai.

Secara keseluruhan, activity diagram tersebut menunjukkan bagaimana interaksi pemain dan sistem berjalan secara berurutan dalam permainan *House of Horror*, mulai dari memulai permainan, eksplorasi lingkungan, interaksi objek, penyelesaian puzzle, munculnya elemen horor, hingga proses keluar dari permainan. Diagram ini menggambarkan mekanisme utama permainan yang berfokus pada eksplorasi, pemecahan masalah, dan pengalaman horor berbasis kejutan.

4. *Sequence Diagram*



Gambar 4.3. *Sequence Diagram*

Sequence Diagram pada *Game House of Horror* menggambarkan urutan komunikasi dan interaksi antara Pemain, Menu UI, Game Controller, Lingkungan Rumah, Objek/Item, Puzzle Manager, dan Hantu (AI) selama permainan berlangsung. Diagram ini menunjukkan bagaimana setiap komponen sistem saling berkomunikasi mulai dari permainan dimulai hingga pemain keluar dari game.

Proses dimulai ketika Pemain menjalankan permainan melalui aktivitas Memulai Permainan(). Permintaan tersebut diterima oleh Menu UI, kemudian diteruskan ke Game Controller melalui proses LoadGame() untuk mempersiapkan sistem permainan. Setelah sistem siap, Menu UI akan menampilkan Menu Utama

kepada pemain. Pemain kemudian dapat melakukan pengaturan awal melalui fungsi Atur Pengaturan(), seperti pengaturan suara atau kontrol permainan. Pengaturan tersebut disimpan oleh Game Controller melalui proses Simpan Pengaturan(), kemudian sistem memberikan konfirmasi bahwa pengaturan berhasil diterapkan.

Setelah konfigurasi selesai, pemain memilih untuk memulai permainan melalui fungsi Mulai Game(). Game Controller kemudian melakukan proses Inisialisasi Game() dan meminta sistem untuk menampilkan Lingkungan Rumah sebagai area permainan. Setelah lingkungan berhasil ditampilkan, pemain mulai melakukan eksplorasi rumah dengan aktivitas seperti berjalan, menggunakan senter, membuka pintu, dan mencari objek tertentu.

Pada saat pemain menjelajahi rumah, sistem melakukan pengecekan kondisi tertentu melalui proses Cek Kondisi Jump Scare() yang dikendalikan oleh sistem Hantu (AI). Jika kondisi terpenuhi, sistem akan menjalankan proses Munculkan Hantu() dan menampilkan efek Jump Scare kepada pemain. Namun, jika kondisi tidak terpenuhi, sistem akan mengizinkan pemain untuk melanjutkan eksplorasi tanpa gangguan.

Selama eksplorasi, pemain dapat melakukan Interaksi dengan Objek(). Sistem kemudian memproses objek yang ditemukan dan memberikan respons berupa Item atau Petunjuk. Item yang diperoleh dapat digunakan untuk membantu pemain menyelesaikan tantangan, sedangkan petunjuk digunakan untuk memahami alur cerita atau menemukan solusi permainan.

Setelah mendapatkan informasi atau item yang diperlukan, pemain melakukan aktivitas Menyelesaikan Puzzle(). Permintaan tersebut diproses oleh

Puzzle Manager melalui fungsi Validasi Puzzle() untuk menentukan apakah puzzle berhasil diselesaikan atau tidak. Jika puzzle berhasil, sistem akan menjalankan proses Buka Area Baru() sehingga pemain dapat melanjutkan ke area berikutnya. Jika puzzle gagal, sistem akan menampilkan pesan kegagalan dan pemain perlu mencoba kembali.

Pada akhir permainan, pemain dapat memilih Keluar dari Permainan(). Permintaan tersebut diteruskan ke Menu UI dan Game Controller untuk menjalankan proses Tutup Game(). Setelah permainan ditutup, sistem mengembalikan pemain ke Menu Utama.

Secara keseluruhan, sequence diagram ini menunjukkan bahwa alur permainan *House of Horror* bergantung pada komunikasi berurutan antara pemain dan berbagai komponen sistem. Pemain menjadi pengendali utama aktivitas permainan, sedangkan sistem bertugas mengatur tampilan, lingkungan permainan, interaksi objek, mekanisme puzzle, serta kejadian horor seperti jump scare agar pengalaman bermain berjalan secara dinamis.

B. Rancangan *Interface*

1. Tampilan Utama Aplikasi



Gambar 4.4 Tampilan Utama

Berikut penjelasan item-item pada “Tampilan Utama” (main menu) pada game **House of Horror** berikut :

a. **MULAI** (ikon pintu): Tombol untuk memulai permainan. Pemain akan masuk ke gameplay utama, yaitu menjelajahi rumah horor.
b. **PANDUAN** (ikon buku terbuka): Menampilkan petunjuk cara bermain, seperti kontrol pergerakan, cara berinteraksi dengan objek, dan tujuan permainan.
c. **TENTANG** (ikon huruf "i" dalam lingkaran): Berisi informasi mengenai game, misalnya versi aplikasi, dan pengembang.
d. **KELUAR** (ikon pintu dengan panah keluar): Tombol untuk menutup aplikasi dan keluar dari game.

2. Tampilan Tentang

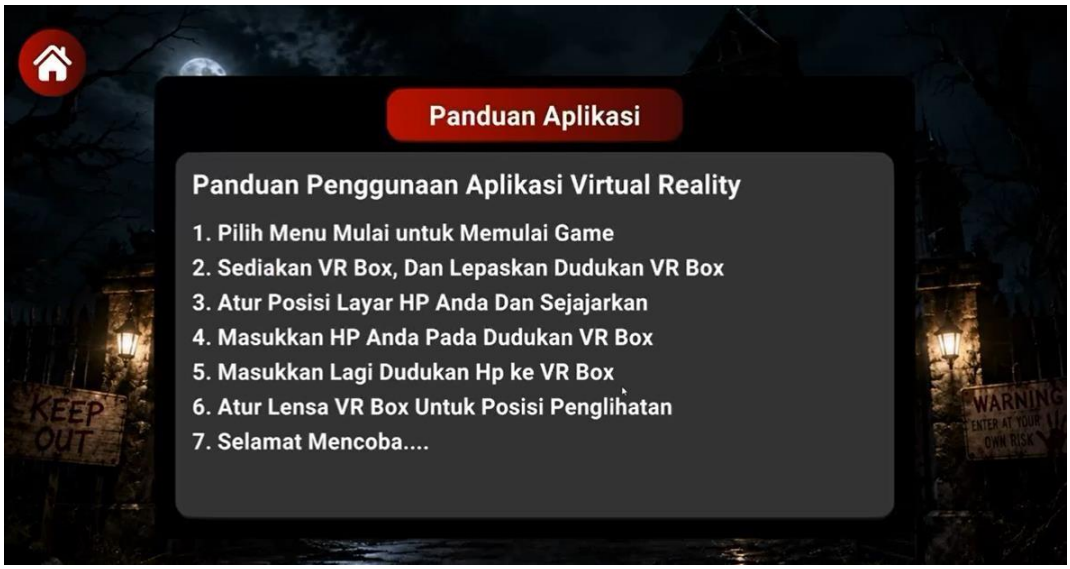


Gambar 4.5 Tentang

Berikut penjelasan pada (Tampilan Tentang) "Informasi Tentang Pengembang Aplikasi" pad game **House of Horror** berikut :

a. **Nama Lengkap**: Feggy Nurwindi, yaitu nama pengembang aplikasi
b. **NIM**: 220280028, yaitu nomor induk mahasiswa pengembang.
c. **Judul**: *Rancang Bangun Virtual Reality Games House of Horror Berbasis Android*, yaitu judul penelitian atau skripsi yang melatarbelakangi pembuatan aplikasi. Judul ini menunjukkan bahwa aplikasi berupa game VR yang berjalan pada platform Android.
d. **Prodi**: Teknik Informatika, yaitu program studi pengembang.
e. **Institusi**: Universitas Muhammadiyah Parepare, yaitu asal perguruan tinggi pengembang.

3. Tampilan Panduan Aplikasi



Gambar 4.6 Panduan

Berikut penjelasan pada (Tampilan Panduan Aplikasi) “Panduan Penggunaan Aplikasi Virtual Reality” game **House of Horror** berikut :

a. **Pilih Menu Mulai untuk Memulai Game**: Pengguna menekan tombol Mulai di menu utama.
b. **Sediakan VR Box, dan Lepaskan Dudukan VR Box**: Pengguna menyiapkan perangkat VR Box, lalu melepas dudukan (holder) tempat HP diletakkan.
c. **Atur Posisi Layar HP Anda dan Sejajarkan**: Layar HP diposisikan sejajar (lurus dan simetris) agar tampilan mata kiri dan kanan tepat.
d. **Masukkan HP Anda pada Dudukan VR Box**: HP dipasang pada dudukan yang sudah dilepas.
e. **Masukkan Lagi Dudukan HP ke VR Box**: Dudukan yang sudah berisi HP dipasang kembali ke badan VR Box.

f. **Atur Lensa VR Box untuk Posisi Penglihatan**: Jarak dan posisi lensa disesuaikan agar gambar terlihat jelas dan nyaman di mata.

g. **Atur Lensa VR Box untuk Posisi Penglihatan**: Jarak dan posisi lensa disesuaikan agar gambar terlihat jelas dan nyaman di mata.

h. **Selamat Mencoba....**: Penutup yang menandakan pengguna siap bermain.

4. Tampilan Mulai



Gambar 4.7 Mulai

Berikut penjelasan pada (Tampilan Mulai) “Misi Utama” game **House of Horror** berikut :

a. **Cari 6 kunci yang tersebar di rumah ini**: Tujuan utama pemain, yaitu mengumpulkan 6 kunci.

b. **Waktu 5 menit untuk mencari semua kunci**: Batas waktu permainan, yang menambah tekanan bagi pemain.

c. **Jika terlambat, setan-setan akan datang menghampiri dan game berakhir**: Konsekuensi kegagalan (game over) bila waktu habis.

d. **Cari jalan keluar melalui pintu be...**: Tujuan akhir, yaitu menemukan pintu keluar setelah kunci terkumpul. Teks pada gambar terpotong, sehingga kelanjutan kalimatnya tidak terbaca jelas.

5. Menjalankan Misi

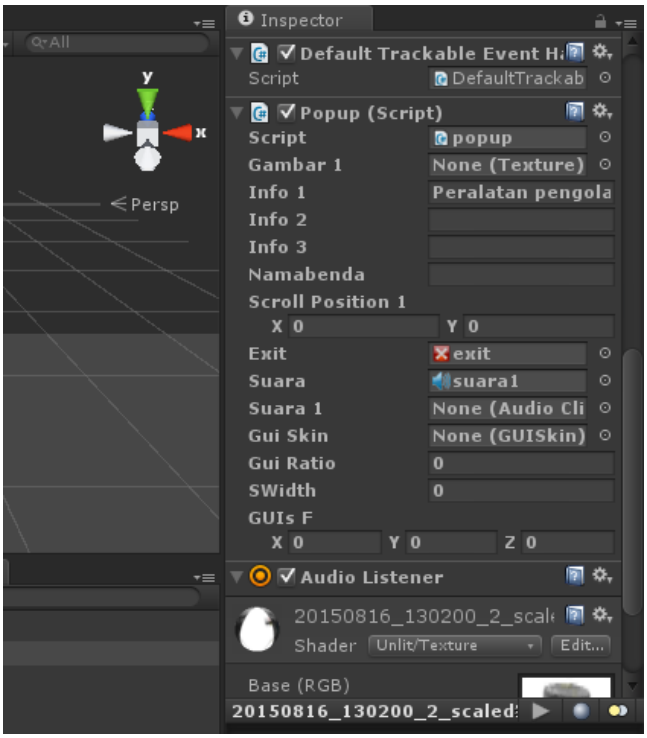


Gambar 4.8 Misi

Berikut penjelasan pada (Tampilan Menjalankan Misi) pada game **House of Horror** berikut :

a. **Penghitung kunci "Kunci: 0 / 6"**: Menunjukkan jumlah kunci yang sudah terkumpul dari total 6. Pada gambar nilainya masih 0, artinya pemain baru memulai.
b. **Teks instruksi "Temukan 6 kunci dan melarikan diri dari rumah"**: Pengingat tujuan misi yang tampil di bagian atas layar.
c. **Ikon interaksi (lingkaran biru kecil di tengah)**: Tampak berupa penanda (reticle/pointer) yang menunjukkan objek yang dapat diinteraksikan atau titik bidik pemain.

6. Rancangan Input



Gambar 4.9 Input Data Aplikasi

C. Pengujian Sistem

1. *Black Box*

Tabel 4.2. Black Box

No	Skenario Pengujian	Input / Aksi Pengguna	Proses Sistem yang Diuji	Output yang Diharapkan	Hasil
1	Memulai permainan	Pemain menekan tombol Start Game	Sistem memuat data permainan dan menginisialisasi game	Permainan berhasil dimulai dan menampilkan lingkungan rumah horor	Berhasil
2	Menampilkan menu utama	Pemain membuka aplikasi/game	Sistem menjalankan halaman awal permainan	Menu utama tampil dengan pilihan permainan	Berhasil
3	Mengatur pengaturan permainan	Pemain mengubah volume suara atau kontrol permainan	Sistem menyimpan perubahan pengaturan	Pengaturan berhasil diterapkan sesuai pilihan pemain	Berhasil
4	Menjelajahi lingkungan rumah	Pemain menekan tombol navigasi untuk bergerak	Sistem membaca input pemain dan memperbarui posisi karakter	Karakter dapat berjalan dan berpindah area dalam rumah	Berhasil
5	Menggunakan senter	Pemain mengaktifkan fitur senter	Sistem mengaktifkan pencahayaan tambahan	Area gelap menjadi terlihat dengan bantuan senter	Berhasil
6	Membuka pintu	Pemain mendekati pintu dan melakukan interaksi	Sistem memeriksa status pintu dan akses ruangan	Pintu terbuka dan pemain dapat memasuki ruangan baru	Berhasil

7	Berinteraksi dengan objek	Pemain menekan tombol interaksi pada objek tertentu	Sistem mendeteksi objek yang dipilih	Objek dapat digunakan atau memberikan informasi	Berhasil
8	Mengambil item	Pemain menemukan dan mengambil item	Sistem menyimpan item ke inventori pemain	Item berhasil masuk ke inventori	Berhasil
9	Membaca petunjuk	Pemain membuka catatan/petunjuk yang ditemukan	Sistem menampilkan informasi petunjuk	Informasi petunjuk tampil pada layar	Berhasil
10	Memunculkan jump scare	Pemain memasuki area pemicu hantu	Sistem melakukan pengecekan kondisi kejadian horor	Karakter hantu muncul dan efek jump scare berjalan	Berhasil
11	Tidak memicu jump scare	Pemain berada di area tanpa kondisi pemicu	Sistem mengecek kondisi tetapi tidak memenuhi syarat	Pemain dapat melanjutkan eksplorasi tanpa jump scare	Berhasil
12	Menyelesaikan puzzle	Pemain memasukkan solusi puzzle yang benar	Sistem melakukan validasi jawaban	Puzzle berhasil diselesaikan dan area baru terbuka	Berhasil
13	Puzzle gagal	Pemain memasukkan solusi yang salah	Sistem melakukan pengecekan jawaban	Sistem menampilkan pesan gagal dan pemain mencoba kembali	Berhasil
14	Membuka area baru	Pemain berhasil menyelesaikan puzzle	Sistem memberikan akses lokasi berikutnya	Area baru dapat dimasuki pemain	Berhasil

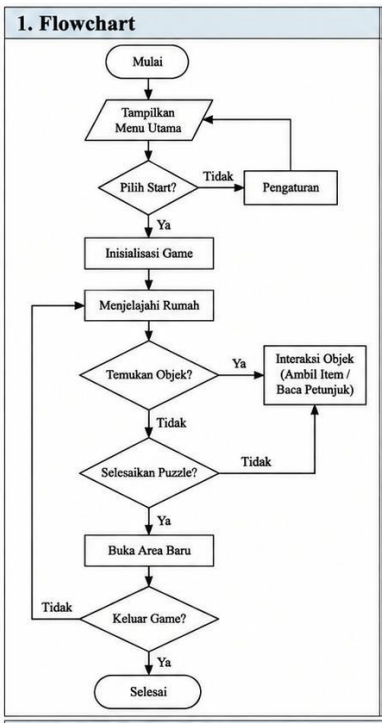
15	Keluar dari permainan	Pemain memilih menu keluar	Sistem menghentikan sesi permainan	Game tertutup dan kembali ke menu utama	Berhasil

2. *White Box*

Pengujian White Box merupakan metode pengujian perangkat lunak yang dilakukan dengan melihat struktur internal program, seperti alur logika program, fungsi, percabangan, kondisi, dan proses eksekusi kode. Pengujian ini bertujuan untuk memastikan bahwa setiap bagian algoritma dalam sistem telah berjalan sesuai dengan rancangan.

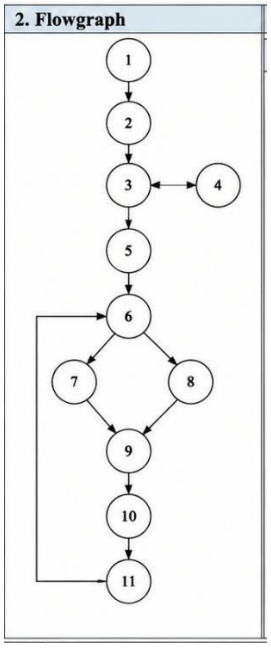
a. Pengujian Alur Program Memulai Permainan

Proses ini digunakan untuk memastikan bahwa sistem dapat menjalankan permainan setelah pengguna memilih menu mulai.



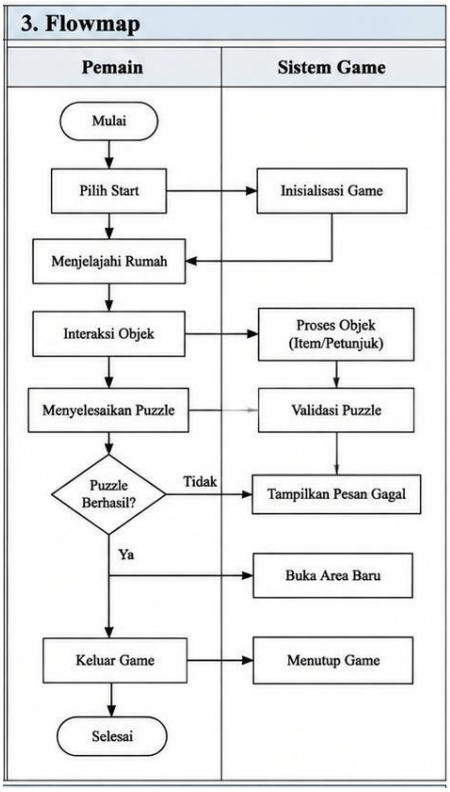
Gambar 4.10 *Flowchart*

b. Flowgraph



Gambar 4.11 *Flowgrapht*

c. Flowmap



Gambar 4.12 *Flowmap*

d. *Cyclomatic Complexity*

Cyclomatic Complexity digunakan untuk menghitung jumlah jalur independen yang terdapat pada alur program. Perhitungan ini bertujuan untuk mengetahui tingkat kompleksitas logika sistem dan menentukan jumlah minimal kasus pengujian yang harus dilakukan.

1. Rumus Cyclomatic Complexity

$$V(G) = E - N + 2$$

Keterangan:

- V(G) = Nilai Cyclomatic Complexity

- E = Jumlah Edge (garis/alur penghubung antar node)
- N = Jumlah Node (titik proses atau aktivitas)

2. Identifikasi Node dan Edge

Berdasarkan flowgraph Game House of Horror, terdapat:

Jumlah Node (N) = 11

Tabel 4.3. Node

Node	**Proses**
1	Mulai
2	Menampilkan Menu Utama
3	Inisialisasi Game
4	Pengaturan Game
5	Menjelajahi Rumah
6	Pemeriksaan Interaksi/Puzzle
7	Interaksi Objek (Item/Petunjuk)
8	Penyelesaian Puzzle
9	Validasi Puzzle
10	Keluar Game
11	Selesai

Jumlah Edge (E) = 13

Daftar hubungan antar node:

1 → 2

2 → 3

3 → 5

2 → 4

4 → 2

5 → 6

6 → 7

6 → 8

7 → 9

8 → 9

9 → 5

9 → 10

10 → 11

3. Perhitungan Cyclomatic Complexity

Diketahui:

- E = 13
- N = 11

Maka:

$$V(G) = E - N + 2$$

$$V(G) = 13 - 11 + 2$$

$$V(G) = 4$$

4. Hasil Cyclomatic Complexity

$$\boxed{V(G) = 4}$$

Nilai Cyclomatic Complexity = 4 menunjukkan bahwa terdapat 4 jalur independen yang harus diuji pada sistem Game House of Horror.

5. Jalur Independen Pengujian

Tabel 4.4. Independen

Jalur	Urutan Node	Skenario
Jalur 1	1-2-3-5-6-7-9-10-11	Pemain memulai game, eksplorasi rumah, menemukan objek, menyelesaikan proses, dan keluar dari permainan
Jalur 2	1-2-3-5-6-8-9-10-11	Pemain langsung menyelesaikan puzzle tanpa mengambil objek tambahan
Jalur 3	1-2-3-5-6-8-9-5-6	Puzzle gagal sehingga pemain kembali melakukan eksplorasi
Jalur 4	1-2-4-2-3-5-6-7-9-10-11	Pemain melakukan pengaturan terlebih dahulu sebelum memulai permainan

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil analisis dan pengujian terhadap sistem permainan Game House of Horror, dapat disimpulkan bahwa sistem telah memiliki alur proses yang terstruktur mulai dari proses memulai permainan, pengaturan awal, eksplorasi lingkungan rumah, interaksi dengan objek, penggunaan item, penyelesaian puzzle, hingga proses keluar dari permainan. Perancangan sistem menggunakan Use Case Diagram, Activity Diagram, dan Sequence Diagram berhasil menggambarkan hubungan antara pemain dengan komponen sistem serta alur komunikasi yang terjadi selama permainan berlangsung.

Hasil pengujian Black Box menunjukkan bahwa seluruh fitur utama permainan dapat berjalan sesuai dengan fungsi yang diharapkan, seperti memulai permainan, menjelajahi rumah, menggunakan senter, mengambil item, membaca petunjuk, memunculkan efek *jump scare*, menyelesaikan puzzle, dan keluar dari permainan. Pengujian ini membuktikan bahwa sistem mampu memberikan respons yang sesuai terhadap setiap input yang diberikan oleh pengguna.

Pada pengujian White Box, analisis dilakukan menggunakan metode Flowchart, Flowgraph, Flowmap, Cyclomatic Complexity, dan Matrix Jalur Independen. Berdasarkan perhitungan Cyclomatic Complexity diperoleh nilai

V(G)=4, yang menunjukkan terdapat empat jalur independen yang harus diuji untuk memastikan setiap percabangan logika program berjalan dengan benar. Hasil tersebut menunjukkan bahwa struktur logika permainan memiliki tingkat kompleksitas yang masih rendah dan mudah untuk dilakukan pengujian maupun pengembangan lebih lanjut.

Secara keseluruhan, sistem Game House of Horror telah memenuhi kebutuhan dasar sebuah permainan horor berbasis eksplorasi dengan mekanisme interaksi, pemecahan puzzle, serta elemen kejutan berupa *jump scare*. Rancangan dan hasil pengujian dapat menjadi dasar untuk pengembangan fitur permainan yang lebih kompleks pada tahap berikutnya.

B. Saran

Berdasarkan hasil analisis dan pengujian pada Game House of Horror, pengembangan sistem selanjutnya disarankan untuk meningkatkan variasi gameplay agar pengalaman bermain menjadi lebih menarik dan tidak berulang. Penambahan jenis puzzle, misi, serta tantangan baru dapat memberikan tingkat kesulitan yang lebih beragam bagi pemain. Selain itu, mekanisme karakter hantu atau sistem *Artificial Intelligence (AI)* dapat dikembangkan agar memiliki perilaku yang lebih dinamis, sehingga kemunculan *jump scare* tidak hanya berdasarkan lokasi tertentu, tetapi dapat menyesuaikan dengan aktivitas dan kondisi pemain selama permainan berlangsung.

Pengembangan berikutnya juga dapat memperhatikan aspek visual dan audio dengan meningkatkan kualitas grafis, pencahayaan, efek suara, serta animasi agar

suasana horor yang dibangun menjadi lebih realistis dan imersif. Penambahan alur cerita yang lebih mendalam melalui dialog, catatan, atau petunjuk tersembunyi juga dapat memperkuat pengalaman pemain dalam memahami latar belakang permainan. Selain itu, pengujian sistem dapat diperluas menggunakan metode lain seperti User Acceptance Testing (UAT) dan pengujian performa untuk mengetahui tingkat kenyamanan pengguna serta memastikan sistem dapat berjalan dengan baik pada berbagai perangkat. Dengan adanya pengembangan tersebut, diharapkan *Game House of Horror* dapat memiliki kualitas permainan yang lebih baik, interaktif, dan mampu memberikan pengalaman bermain yang lebih menarik bagi pengguna.

DAFTAR PUSTAKA

Amalia, F. (2024). Teknologi Augmented Reality (Ar) Dan Virtual Reality (Vr): Mengubah Cara Kita Berinteraksi Dengan Dunia. *Jurnal Teknologi Pintar*.

Adhitya Wibawa (2015), Pemanfaatan *Augmented Reality* Sebagai Media Marker, Penerbit Andi

Arifianto, R. 2017. Materi *Flowchart. From* https://rahmatarifianto.wordpress.com/2014/11/20/pengertian-*Flowchart*-dan-jenis-jenisnya.html.

Basri, M., & Kasmaida, K. (2025). Pemberdayaan UMKM Takoyaki: Pemanfaatan Augmented Reality (AR) Untuk Katalog Produk Interaktif Dan Pemasaran Digital. Abdimas Toddopuli: Jurnal Pengabdian Pada Masyarakat, 6(2), 378-385.

Filus, Teo (2017),Pemrograman C++. Jakarta Penerbit: Andi

Filus, Teo. 2017. Pengenalan Pemrograman C#. From https://www.codepolitan. com/pengenalan-bahasa-pemrograman-c-587effa1cb95b (25 Nov 2021).

Komputer, Wahana (2013), Pengenalan Android, Jakarta, Penerbit:ANDI

Hermawan dan Hariadi. 2015. Pemanfaatan *Augmented Reality* Sebagai Media Informasi Kampus Menggunakan Brosur. E-Journal Tek. Inform., vol. 13, no. 1, pp. 30–37.

Ismayani, Ani. 2020. Membuat Sendiri Aplikasi Augmented Reality. Jakarta Pusat : Elex Media Komputindo.

Kadir, Abdul. 2013. Teori dan Aplikasi Pengolahan Citra. Yogyakarta: Andi Offset.

Komputer, Wahana. 2013. Step by Step Menjadi Pemrograman Android. Yogyakarta : Andi.

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

N, Sora. 2015. Pengertian *UML* dan Jenis - jenisnya. *From http://www*. pengertianku.net/2015/09/pengertian-*uml*-dan-jenis-jenisnya-serta-contoh-diagramnya.*html.* (11 Nov 2021).

Nadira, Sujaini dan Pratiwi. 2016. Implementasi *Augmented Reality* Pada Brosur Teknik Informatika Universitas Tanjungpura Menggunakan Metode *Marker*. Teknik Informatika., vol. 5, no. 1, pp. 40–48.

Putra, Adhitya Wibawa. 2015. Apa Itu Vuforia. From https://teknojurnal. com/vuforia/#:~:text=Vuforia%20ini%20sendiri%20merupakan%20SDK, yang%20memiliki%20teknologi%20Augmented%20Reality.

Wafiah, A., Suwardoyo, U., & Syawal, F. (2021). Aplikasi Mendeteksi Jenis Harga Assesorieshandphone Menggunakan Augmented Reality. *Jurnal Sintaks Logika*, *1*(1), 26-32.

Wafiah, A. (2021). Informasi Obat Cair Untuk Anak Menggunakan Augmented Reality Berbasis Android. *Jurnal Sintaks Logika*, *1*(2), 107-112.

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

Zaky, Ali. 2016. Animasi Karakter Dengan Blender dan Unity. Jakarta Pusat :Elex Media Komputindo.