

BAB II

KAJIAN PUSTAKA

A. Kajian Teoritis

1. Game

Game menurut (Imanda, 2022) adalah aktivitas yang kompleks yang mencakup aturan-aturan. Aturan-aturan ini bertujuan untuk membatasi perilaku pemain dan menentukan jalannya permainan. Game bertujuan untuk memberikan hiburan dan meningkatkan konsentrasi.

Kata "game" berasal dari bahasa Inggris yang berarti "permainan". Permainan adalah sesuatu yang digunakan untuk bermain, dengan aturan-aturan tertentu di dalamnya. Menurut Kamus Besar Bahasa Indonesia (KBBI) online, permainan adalah sesuatu yang digunakan untuk bermain; barang atau sesuatu yang dipertandingkan. Permainan tidak hanya sekedar aktivitas untuk mengisi waktu luang, tetapi juga sering kali melibatkan keterampilan, strategi, dan interaksi sosial. (Reno, 2021)

Jadi menurut pembahasan diatas, bisa disimpulkan bahwa Game adalah aktivitas kompleks yang mencakup aturan-aturan untuk membatasi perilaku pemain dan menentukan jalannya permainan dengan tujuan memberikan hiburan dan meningkatkan konsentrasi. Game juga berfungsi sebagai media untuk menyampaikan pesan atau makna tertentu serta dapat berfungsi sebagai alat edukasi dan sarana untuk menyampaikan nilai-nilai dan ide-ide kepada pemainnya. Kata "game" berasal dari bahasa Inggris yang berarti "permainan," yang dalam Kamus Besar Bahasa Indonesia .

(KBBI) diartikan sebagai sesuatu yang digunakan untuk bermain dengan aturan tertentu, dan tidak hanya sekedar mengisi waktu luang tetapi juga melibatkan keterampilan, strategi, dan interaksi sosial.

2. Tower Defense

Menurut (Firmansyah, 2019) Game Tower Defense (TD) adalah salah satu jenis game bergenre strategi yang populer karena dapat melatih otak untuk berpikir taktis dalam mencegah musuh menghancurkan markas pemain. Dalam game ini, pemain harus mengatur dan membangun menara pertahanan di sepanjang jalur yang dilewati oleh musuh untuk menghalangi dan mengalahkan mereka sebelum mencapai tujuan akhir. Musuh yang datang akan mengikuti jalur yang telah dirancang oleh pemain, menuntut strategi yang cermat dan penempatan menara yang efektif untuk berhasil bertahan.

Tower Defense menurut (Yondra et al., 2021) adalah subgenre dari game strategi waktu nyata di mana pemain harus melindungi suatu area dari serangan musuh. Tujuan utamanya adalah untuk mencegah musuh mencapai titik akhir dengan cara menempatkan berbagai jenis menara di sepanjang jalur yang sudah ditentukan. Setiap menara memiliki kemampuan dan efek berbeda, seperti menara yang menyerang dari jarak jauh, menara yang memperlambat gerakan musuh, atau menara yang memberikan kerusakan area. Pemain harus merencanakan dan mengatur strategi penempatan menara dengan hati-hati untuk menghadapi berbagai jenis musuh yang memiliki kekuatan dan kelemahan berbeda. Selain itu, pemain

juga bisa meningkatkan atau meng-upgrade menara untuk meningkatkan efektivitasnya. Kesuksesan dalam game Tower Defense bergantung pada kemampuan pemain untuk mengelola sumber daya dan menyesuaikan strategi sesuai dengan tantangan yang ada.

Berdasarkan pembahasan diatas, dapat disimpulkan bahwa Game Tower Defense (TD) adalah subgenre game strategi yang mengharuskan pemain melindungi markas dari serangan musuh dengan menempatkan menara pertahanan di sepanjang jalur yang dilalui musuh. Game ini melatih pemain berpikir taktis dalam menempatkan berbagai jenis menara yang memiliki kemampuan berbeda untuk menghalangi dan mengalahkan musuh sebelum mencapai tujuan akhir. Kesuksesan dalam Tower Defense bergantung pada strategi yang cermat, pengelolaan sumber daya, dan kemampuan untuk menyesuaikan strategi menghadapi musuh yang beragam serta meng-upgrade menara untuk meningkatkan efektivitas pertahanan.

3. Perlindungan Hutan

Menurut (Aryana, 2021) perlindungan hutan adalah tindakan untuk mempertahankan dan menjaga hak-hak negara, masyarakat, dan individu atas hutan, kawasan hutan, dan hasil hutan, serta investasi dan perangkat yang terkait dengan pengelolaan hutan. Upaya ini melibatkan berbagai langkah untuk memastikan bahwa hutan tetap lestari, sumber daya alamnya dikelola dengan bijak, dan kepentingan semua pihak yang terlibat terlindungi dari eksploitasi atau kerusakan yang merugikan.

Perlindungan hutan merupakan bagian dari kegiatan pengelolaan hutan yang dilaksanakan di wilayah hutan dalam bentuk Kesatuan Pengelolaan Hutan Konservasi (KPHK), Kesatuan Pengelolaan Hutan Lindung (KPHL), dan Kesatuan Pengelolaan Hutan Produksi (KPHP). Tindakan perlindungan ini bertujuan untuk memastikan kelestarian hutan, menjaga biodiversitas, dan mendukung keberlanjutan sumber daya hutan untuk kepentingan generasi sekarang dan mendatang. (Yahya & Idris, 2019)

Perlindungan hutan menurut (Harold Anis, 2021) adalah usaha untuk mencegah terjadinya kerusakan hutan. Terdapat lima golongan kerusakan hutan yang perlu mendapat perhatian dan dicegah, yaitu kerusakan akibat pengerjaan atau pendudukan tanah hutan secara tidak sah, penggunaan hutan yang menyimpang dari fungsinya, dan pengusaha hutan yang tidak bertanggung jawab; kerusakan hutan akibat pengambilan batu, tanah, dan bahan galian lainnya serta penggunaan alat-alat yang tidak sesuai dengan kondisi tanah atau tegalan; kerusakan hutan akibat pencurian kayu dan penebangan tanpa izin; kerusakan hutan akibat penggembalaan ternak dan kebakaran; serta kerusakan hasil hutan akibat perbuatan manusia, gangguan hama dan penyakit, serta kekuatan alam.

Berdasarkan pembahasan diatas, dapat disimpulkan bahwa Perlindungan hutan adalah tindakan untuk menjaga hak-hak negara, masyarakat, dan individu atas hutan, kawasan hutan, dan hasil hutan, serta melindungi investasi dan perangkat pengelolaan hutan. Langkah-langkah ini memastikan kelestarian hutan, pengelolaan sumber daya alam yang

bijak, serta perlindungan dari eksploitasi atau kerusakan. Perlindungan hutan bertujuan mencegah berbagai jenis kerusakan hutan, termasuk pengerjaan atau pendudukan tanah hutan secara tidak sah, penggunaan hutan yang menyimpang dari fungsinya, pengambilan sumber daya tanpa izin, pencurian kayu, penebangan ilegal, penggembalaan ternak, kebakaran, serta kerusakan akibat perbuatan manusia, gangguan hama, penyakit, dan kekuatan alam.

4. Ekosistem

Ekosistem menurut (Bariyah & Sugandi, 2022) adalah sebuah sistem alami yang terbentuk melalui interaksi antara makhluk hidup dan lingkungannya. Di dalam ekosistem, setiap organisme berperan dalam menjaga keseimbangan dan fungsi lingkungan tempat mereka hidup. Interaksi ini mencakup hubungan antara tumbuhan, hewan, mikroorganisme, dan komponen abiotik seperti tanah, air, dan udara. Studi tentang ekosistem, yang dikenal sebagai ekologi, mengkaji bagaimana organisme saling berhubungan dan beradaptasi dengan lingkungan mereka, serta bagaimana faktor-faktor lingkungan mempengaruhi kelangsungan hidup dan distribusi organisme. Ekologi membantu kita memahami kompleksitas alam dan pentingnya menjaga keseimbangan ekosistem untuk keberlanjutan kehidupan di Bumi.

Berdasarkan pembahasan diatas, dapat disimpulkan bahwa Ekosistem adalah sistem alami yang terbentuk melalui interaksi antara makhluk hidup dan lingkungannya, mencakup tumbuhan, hewan,

mikroorganisme, dan komponen abiotik seperti tanah, air, dan udara. Setiap organisme dalam ekosistem berperan menjaga keseimbangan dan fungsi lingkungan. Studi ekologi mengkaji hubungan dan adaptasi organisme dengan lingkungan mereka serta pengaruh faktor-faktor lingkungan terhadap kelangsungan hidup dan distribusi organisme. Ekosistem melibatkan hubungan timbal balik kompleks antara komponen hidup dan tidak hidup yang membentuk suatu sistem ekologi, membantu kita memahami kompleksitas alam dan pentingnya menjaga keseimbangan ekosistem untuk keberlanjutan kehidupan di Bumi.

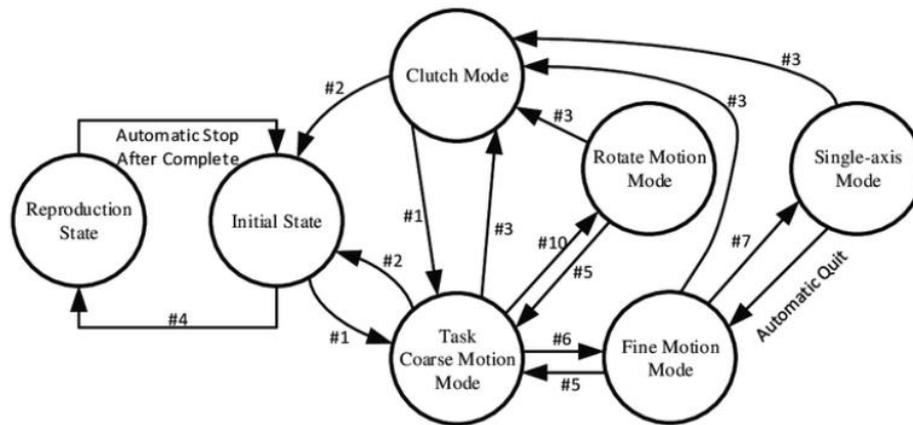
5. Finite State Machine

Menurut (Armedianto Putro et al., 2021) Finite State Machine (FSM) adalah sebuah metode untuk membuat sistem kontrol yang menggambarkan tingkah laku dengan menggunakan tiga elemen utama: State (keadaan), Event (kejadian), dan Action (aksi). Dalam kondisi tertentu, sistem akan berada pada satu state dan kemudian berpindah ke state lain ketika menerima suatu input tertentu yang bisa berasal dari dalam maupun perangkat luar komponen sistem tersebut. FSM digunakan untuk merancang logika pengendalian yang efektif dan terstruktur dalam berbagai aplikasi, seperti game, robotika, dan sistem kendali otomatis.

Finite State Machine (FSM) Menurut (Putro et al., 2021) adalah sebuah metode untuk mengatur tingkah laku sebuah sistem dengan menggunakan prinsip kerja yang melibatkan keadaan (State), kejadian (Event), dan aksi (Action). Cara kerjanya adalah salah satu state akan aktif,

dimulai dari state pertama atau state 0. Sistem dapat berpindah ke state lain jika mendapatkan masukan atau event tertentu, baik yang berasal dari perangkat luar maupun komponen dalam sistem itu sendiri. Setiap state akan menghasilkan aksi yang berbeda-beda sesuai dengan kebutuhan sistem. Aksi yang dilakukan dapat berupa tindakan sederhana atau melibatkan rangkaian proses yang relatif kompleks, tergantung pada desain dan tujuan dari sistem tersebut.

Finite State Machine (FSM), sebuah model matematika yang menggambarkan sistem dengan jumlah state (keadaan) terbatas dan transisi antara state tersebut. Diagram ini menunjukkan berbagai mode operasi seperti Initial State, Clutch Mode, Task Coarse Motion Mode, Fine Motion Mode, Rotate Motion Mode, Single-axis Mode, dan Reproduction State. Setiap mode terhubung melalui transisi yang diindikasikan oleh panah berlabel angka, menunjukkan bagaimana sistem dapat berpindah dari satu state ke state lainnya berdasarkan kondisi tertentu. Misalnya, dari Initial State, sistem dapat berpindah ke Clutch Mode atau Task Coarse Motion Mode, dan seterusnya. FSM ini berguna untuk memodelkan perilaku sistem yang kompleks, seperti dalam sistem kontrol robotik, untuk memastikan setiap langkah operasi dapat diikuti dengan jelas dan konsisten. Penjelasan ini dapat dilihat pada Gambar 2.1.



Gambar 2. 1 *Finite State Machine*

Berdasarkan dari pembahasan diatas, dapat disimpulkan bahwa Finite State Machine (FSM) adalah sebuah metode untuk membuat sistem kontrol yang menggambarkan tingkah laku dengan menggunakan tiga elemen utama: State (keadaan), Event (kejadian), dan Action (aksi). Dalam kondisi tertentu, sistem akan berada pada satu state dan kemudian berpindah ke state lain ketika menerima suatu input tertentu yang bisa berasal dari dalam maupun perangkat luar komponen sistem tersebut. FSM digunakan untuk merancang logika pengendalian yang efektif dan terstruktur dalam berbagai aplikasi, seperti game, robotika, dan sistem kendali otomatis. Cara kerjanya adalah salah satu state akan aktif, dimulai dari state pertama atau state 0. Sistem dapat berpindah ke state lain jika mendapatkan masukan atau event tertentu, baik yang berasal dari perangkat luar maupun komponen dalam sistem itu sendiri. Setiap state akan menghasilkan aksi yang berbeda-beda sesuai dengan kebutuhan sistem.

6. Unity

Menurut (Ramadhanti et al., 2021) Unity adalah sebuah game engine yang dikembangkan oleh Unity Technologies, yang berfungsi sebagai alat bantu bagi para pengembang game dengan kemampuan rendering yang terintegrasi. Sebagai platform pengembangan yang sangat populer, Unity menawarkan berbagai fitur canggih yang memungkinkan pengembang untuk menciptakan permainan dengan grafis yang memukau dan performa yang optimal. Selain kemampuan rendering yang unggul.

Unity Menurut (Fuad Hilmy et al., 2021) adalah aplikasi yang digunakan untuk mengembangkan game multi-platform dan didesain agar mudah digunakan. Unity menawarkan banyak fitur dan integrasi dengan aplikasi profesional, menjadikannya alat yang kuat dan serbaguna bagi para pengembang game. Dengan antarmuka yang user-friendly dan kemampuan untuk membuat game yang dapat berjalan di berbagai perangkat, Unity menjadi pilihan populer di kalangan pengembang game dari berbagai tingkat keahlian.

Berdasarkan pembahasan diatas, dapat disimpulkan bahwa Unity adalah aplikasi pengembangan game yang ideal untuk menciptakan kreasi game dengan berbagai fitur canggih. Unity memiliki kemampuan untuk melakukan authoring pada beberapa platform yang berbeda, membuatnya menjadi pilihan utama bagi para pengembang game. Dengan antarmuka yang intuitif dan dukungan yang luas, Unity memungkinkan pengembang untuk menghasilkan game berkualitas tinggi untuk berbagai perangkat dan

sistem operasi. Unity menawarkan banyak fitur dan integrasi dengan aplikasi profesional, menjadikannya alat yang kuat dan serbaguna bagi para pengembang game. Dengan antarmuka yang user-friendly dan kemampuan untuk membuat game yang dapat berjalan di berbagai perangkat, Unity menjadi pilihan populer di kalangan pengembang game dari berbagai tingkat keahlian.

7. Flowchart

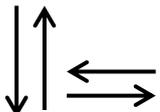
Flowchart menurut (Hasan et al., 2020) Flowchart adalah representasi visual yang menggambarkan urutan langkah-langkah untuk menyelesaikan masalah tertentu. Flowchart digunakan untuk menyajikan algoritma dengan cara yang mudah dipahami, tersusun secara sistematis, dan jelas. Tujuan utama pembuatan flowchart adalah untuk mengilustrasikan secara visual tahapan penyelesaian suatu masalah dengan menggunakan simbol-simbol standar yang memudahkan interpretasi dan implementasi algoritma. Dengan demikian, flowchart membantu dalam memvisualisasikan proses secara efektif dan efisien, sehingga mempermudah pemahaman dan pelaksanaan langkah-langkah yang diperlukan.

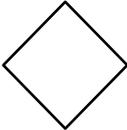
Menurut (Fauzi, 2020) Flowchart adalah cara penulisan algoritma dengan menggunakan notasi grafis. Flowchart berupa gambar atau bagan yang memperlihatkan urutan atau langkah-langkah dari suatu program dan hubungan antar proses beserta pernyataannya. Gambaran ini dinyatakan dengan simbol-simbol khusus, di mana setiap simbol menggambarkan

proses tertentu. Hubungan antara proses-proses ini digambarkan dengan garis penghubung. Dengan menggunakan flowchart, akan lebih mudah untuk memeriksa bagian-bagian yang mungkin terlupakan dalam analisis masalah, sehingga membantu memastikan bahwa semua aspek algoritma telah dipertimbangkan dan diimplementasikan dengan benar.

Berdasarkan pembahasan diatas, dapat disimpulkan bahwa Flowchart adalah diagram atau bagan yang digunakan untuk menggambarkan aliran atau urutan suatu algoritma menggunakan notasi grafis. Flowchart memberikan representasi visual yang jelas mengenai proses atau algoritma, dengan setiap langkah diwakili oleh simbol khusus dan panah menunjukkan arah aliran proses. Ini memudahkan pemahaman dan pengecekan langkah-langkah dalam algoritma, membantu memastikan bahwa semua aspek telah dipertimbangkan dan diimplementasikan dengan benar. Meskipun sangat efektif untuk visualisasi algoritma sederhana, flowchart bisa menjadi rumit dan sulit diikuti jika algoritma terlalu panjang atau tidak efisien, sehingga penting untuk menjaga kesederhanaan dan efisiensi dalam penggunaannya.

Tabel 2. 1 Simbol-simbol *Flowchart*

Simbol	Nama	Fungsi
	Garis Alir	Arahan aliran program

	Terminator	Mulai / akhir program
	Proses	Proses perhitungan / proses pengolahan data
	Decision	Perbandingan pernyataan penyelesaian data yang memberikan pilihan untuk langkah selanjutnya
	Input / Output Data	Proses output / input data parameter, informasi
	Document	Mencetak Output seperti document melalui sebuah perangkat keras (printer)
	Disk Storage	Simbol yang menyatakan input berasal dari disk atau output disimpan di disk.

8. UML (Unified Modeling Language)

Menurut (Putra & Andriani, 2019) UML (Unified Modeling Language) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan kebutuhan (requirement), melakukan analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML menyediakan berbagai jenis diagram yang membantu pengembang dan pemangku kepentingan untuk memahami,

merancang, dan mendokumentasikan sistem perangkat lunak dengan cara yang konsisten dan terstruktur. Dengan menggunakan UML, proses komunikasi antar tim menjadi lebih efektif. sehingga memastikan bahwa semua pihak memiliki pemahaman yang sama mengenai sistem yang sedang dikembangkan.

Unified Modeling Language (UML) menurut (Abdillah, 2021) adalah bahasa pemodelan perangkat lunak yang distandardisasi untuk menulis cetak biru perangkat lunak. UML digunakan untuk visualisasi, spesifikasi, konstruksi, dan dokumentasi bagian-bagian dari sistem perangkat lunak. Seperti arsitek yang membuat dokumen cetak biru untuk membangun sebuah bangunan, arsitek perangkat lunak membuat diagram-diagram UML untuk membantu programmer dan developer membangun perangkat lunak. Dengan memahami kosakata yang digunakan dalam UML, kita akan semakin mudah dalam memahami spesifikasi perangkat lunak.

Berdasarkan pembahasan diatas, dapat disimpulkan bahwa Unified Modeling Language (UML) adalah bahasa pemodelan perangkat lunak yang distandardisasi untuk menulis cetak biru perangkat lunak. UML digunakan untuk visualisasi, spesifikasi, konstruksi, dan dokumentasi bagian-bagian dari sistem perangkat lunak. Seperti arsitek yang membuat dokumen cetak biru untuk membangun sebuah bangunan, arsitek perangkat lunak membuat diagram-diagram UML untuk membantu programmer dan developer membangun perangkat lunak. Dengan memahami kosakata yang digunakan

dalam UML, kita akan semakin mudah dalam memahami spesifikasi perangkat lunak.

9. Waterfall

Waterfall adalah model siklus hidup linier berurutan di mana setiap fase harus diselesaikan sebelum fase berikutnya dimulai, tanpa ada tumpang tindih. Fase-fasenya meliputi Analisis, yang melibatkan deskripsi lengkap kebutuhan perangkat lunak; Desain, yang mencakup perencanaan dan pemecahan masalah untuk solusi perangkat lunak; Implementasi, yang merujuk pada konversi spesifikasi desain menjadi program yang dapat dieksekusi; Pengujian, yang memastikan perangkat lunak memenuhi kebutuhan awal melalui proses debugging; dan Pemeliharaan, yang melibatkan modifikasi perangkat lunak setelah pengiriman untuk memperbaiki kesalahan dan meningkatkan kinerja. (Thesing et al., 2021)

Menurut (Senarath, 2021), metodologi Waterfall adalah pendekatan manajemen proyek yang mengikuti proses yang jelas dari awal hingga akhir, di mana hasil yang diharapkan ditentukan sejak awal oleh klien. Proyek direncanakan secara rinci dari awal hingga selesai, dengan paket kerja, tanggung jawab, dan tenggat waktu yang sudah ditetapkan. Fokus utama metode ini adalah mengimplementasikan rencana awal dengan seakurat mungkin untuk memastikan stabilitas, struktur yang jelas, sumber daya yang dapat diprediksi, dan perencanaan yang terdokumentasi dengan baik.

Berdasarkan pembahasan diatas, dapat disimpulkan bahwa Waterfall adalah model siklus hidup linier berurutan di mana setiap fase

harus diselesaikan sebelum fase berikutnya dimulai, tanpa tumpang tindih. Fase-fasenya meliputi Analisis, Desain, Implementasi, Pengujian, dan Pemeliharaan. Proyek direncanakan secara rinci dari awal hingga selesai, dengan paket kerja, tanggung jawab, dan tenggat waktu yang sudah ditetapkan. Fokus utamanya adalah mengimplementasikan rencana awal dengan seakurat mungkin untuk memastikan stabilitas, struktur yang jelas, sumber daya yang dapat diprediksi, dan perencanaan yang terdokumentasi dengan baik.

10. Black Box

Menurut (Supriyono, 2020) Black box testing adalah metode pengujian perangkat lunak yang mengevaluasi fungsionalitas sistem berdasarkan spesifikasi tanpa memperhatikan struktur internalnya. Metode ini menekankan pada pemberian input dan observasi output untuk memastikan bahwa sistem berfungsi sesuai yang diharapkan. Salah satu teknik yang digunakan adalah Boundary Value Analysis (BVA), yang menguji nilai input pada batas maksimum dan minimum. Dengan pendekatan ini, black box testing memverifikasi bahwa perangkat lunak beroperasi dengan benar dari perspektif pengguna akhir, tanpa perlu mengetahui detail kode internalnya.

Menurut (Indriyanti, 2023) Black box testing adalah metode pengujian perangkat lunak yang mengevaluasi fungsionalitas sistem berdasarkan spesifikasi tanpa memperhatikan struktur internalnya. Pengujian ini dilakukan dengan memasukkan data input dan memeriksa

output untuk memastikan sistem berfungsi sesuai yang diharapkan. Beberapa teknik yang digunakan termasuk Equivalence Partitioning, Boundary Value Analysis, dan Robustness Testing. Metode ini efektif untuk mengidentifikasi kelemahan dan bug dalam sistem, memastikan perangkat lunak siap digunakan oleh pengguna akhir. Dalam pengujian ini, tester bertindak sebagai pengguna akhir yang hanya mengetahui fungsi dan spesifikasi yang harus dipenuhi tanpa memahami struktur internal perangkat lunak.

Black box testing menurut (Totade et al., 2023) adalah teknik pengujian perangkat lunak yang menilai fungsionalitas aplikasi tanpa memperhatikan struktur internal kodenya. Teknik ini dilakukan dengan memberikan input dan mengamati output untuk memastikan perangkat lunak berfungsi sesuai dengan persyaratan. Fokus utamanya adalah pada perspektif pengguna akhir, memastikan perangkat lunak berperilaku sesuai dengan spesifikasi yang diinginkan. Sejak 1970-an, black box testing telah berkembang dengan teknik seperti equivalence partitioning dan boundary value analysis untuk meningkatkan akurasi. Metode ini menjadi bagian integral dari siklus hidup pengembangan perangkat lunak, terutama dengan munculnya metodologi agile dan DevOps, serta penggunaan otomatisasi dan kecerdasan buatan untuk meningkatkan efisiensi dan cakupan pengujian.

Berdasarkan pembahasan diatas, dapat disimpulkan bahwa black box testing adalah metode pengujian perangkat lunak yang mengevaluasi

fungsionalitas sistem berdasarkan spesifikasi tanpa memperhatikan struktur internalnya. Pengujian ini dilakukan dengan memberikan input dan memeriksa output untuk memastikan sistem berfungsi sesuai dengan yang diharapkan. Teknik ini mencakup berbagai pendekatan seperti equivalence partitioning dan boundary value analysis untuk meningkatkan akurasi dan efektivitas pengujian. Black box testing berfokus pada perspektif pengguna akhir, memastikan perangkat lunak memenuhi persyaratan dan berfungsi dengan baik dalam situasi dunia nyata. Seiring perkembangan teknologi, metode ini telah berintegrasi dengan otomatisasi dan kecerdasan buatan, meningkatkan efisiensi dan cakupan pengujian. Dengan demikian, black box testing tetap menjadi komponen penting dalam siklus hidup pengembangan perangkat lunak, membantu memastikan kualitas dan keandalan perangkat lunak sebelum digunakan oleh pengguna akhir.

B. Kajian Empiris

Berikut ini merupakan penelitian yang relevan untuk dijadikan acuan dalam penelitian ini:

1. Penelitian yang dilakukan berjudul “Rancang Bangun Game 2D Shooter Platformer Menggunakan Metode Finite State Machine”. Penelitian ini bertujuan untuk mengembangkan game 2D Shooter Platformer yang menarik dan edukatif menggunakan Unity 3D, mengintegrasikan metode Finite State Machine (FSM) untuk pengontrolan NPC (Non-playable Character). Dalam penelitian ini, FSM digunakan untuk mengatur perilaku NPC seperti musuh dan bos dengan memanfaatkan

konsep state, event, dan action yang membantu dalam penciptaan gameplay yang dinamis dan interaktif. Game ini dikembangkan menggunakan bahasa pemrograman C# dan platform pengembangan Unity 3D. Pengujian game dilakukan melalui feedback dari pengguna yang menilai kinerja AI dan responsivitas game. Hasil penelitian menunjukkan bahwa penggunaan FSM dalam game membantu dalam meningkatkan interaksi pemain dengan game dan memberikan pengalaman belajar yang interaktif melalui gameplay yang menantang. (Nopriansyah et al., 2019)

2. Penelitian yang dilakukan berjudul "Implementasi Game Edukasi Bahasa Lampung Dialek A dan Dialek O Berbasis Android". Penelitian ini bertujuan untuk memperkenalkan dan melestarikan Bahasa Lampung kepada anak-anak usia dini melalui media game edukatif. Metode yang digunakan adalah Game Development Life Cycle (GDLC), yang merupakan pendekatan iteratif meliputi tahapan inisialisasi, produksi, dan pengujian. Game dikembangkan menggunakan Construct 2 dan Adobe Photoshop untuk desain, menargetkan platform Android. Kriteria pengujian yang diaplikasikan adalah berdasarkan ISO 25010, mencakup aspek functionality, efficiency, usability, dan portability. Hasil penelitian menunjukkan bahwa game ini efektif dalam meningkatkan kecakapan bahasa serta keterlibatan anak-anak dengan Bahasa Lampung, dengan penilaian positif pada semua aspek pengujian. (Imanda, 2022)

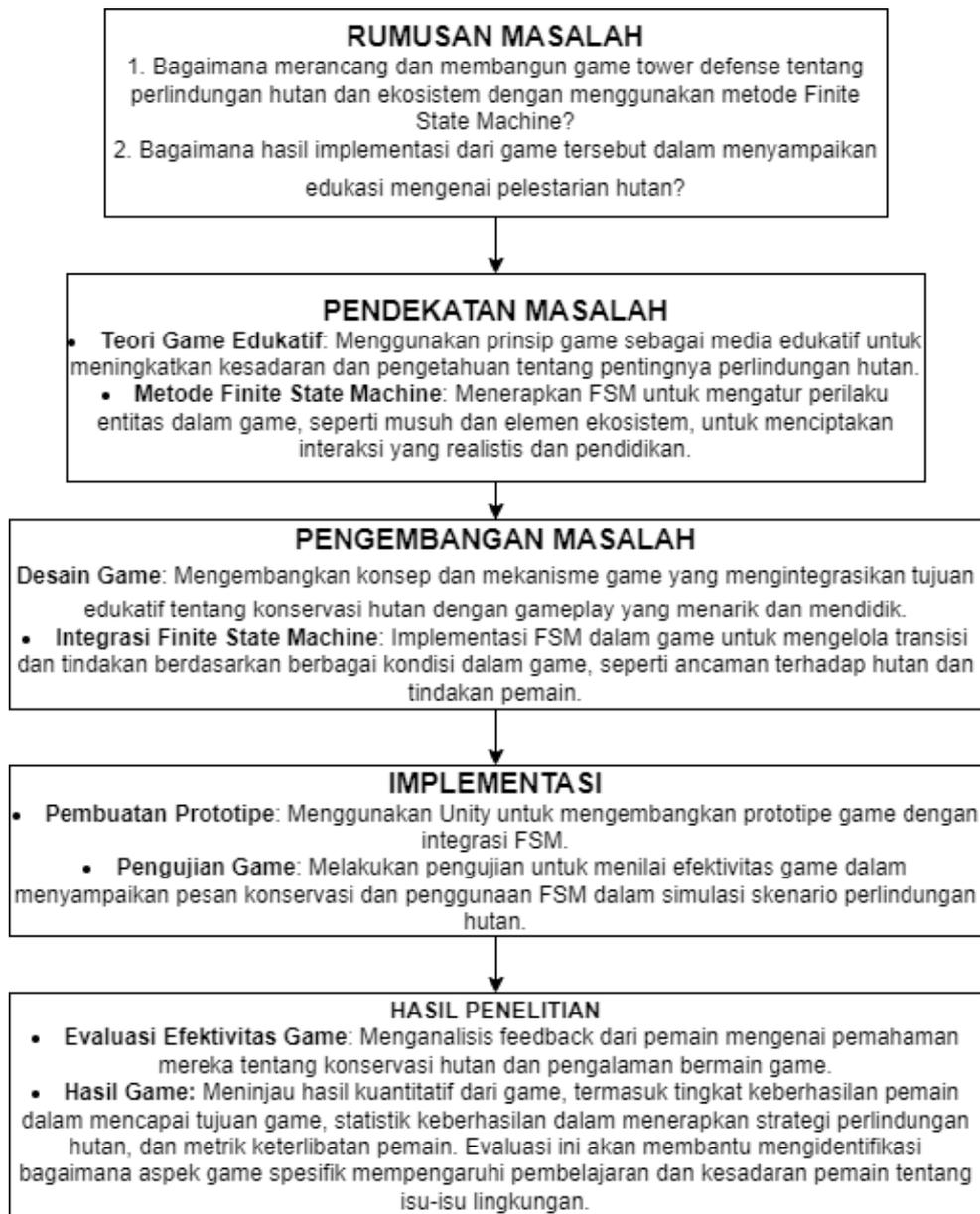
3. Penelitian yang dilakukan berjudul "Kombinasi Metode Finite State Machine dan Game-Based Learning pada Game 'Escape from COV-Madness'". Penelitian ini bertujuan untuk mendidik masyarakat tentang protokol kesehatan melalui game, memanfaatkan metode Finite State Machine untuk mengontrol NPC, dan Game-Based Learning untuk menyediakan teka-teki edukatif tentang COVID-19. Game ini dikembangkan menggunakan Unity3D dan Adobe Illustrator, dan ditujukan untuk desktop. Berdasarkan hasil pengujian yang melibatkan 12 responden dengan rentang usia yang beragam, 100% responden menyukai game dan merasa mendapatkan nilai moral, sementara 91.7% menyatakan bahwa kontrol dalam game ini berjalan dengan baik. Penelitian ini menunjukkan efektivitas game sebagai alat sosialisasi dan edukasi tentang pentingnya protokol kesehatan selama pandemi COVID-19. (Armedianto Putro et al., 2021)
4. Penelitian yang dilakukan berjudul "Penerapan Metode Finite State Machine pada Game 'Escape from Punk Hazard'". Penelitian ini bertujuan untuk mengembangkan game 3D adventure berbasis Android yang menggunakan Unity sebagai game engine, menggabungkan metode Finite State Machine dan Fuzzy Logic untuk mengontrol NPC dalam game. Metode Finite State Machine diterapkan pada semua NPC, sedangkan Fuzzy Logic khusus untuk NPC Boss di setiap level. Hasil pengujian menunjukkan bahwa 100% responden menyukai game ini, 81.8% menyatakan bahwa kontrol dan gameplay berjalan dengan lancar,

dan 90.9% menyatakan bahwa NPC berjalan dengan baik. (Putro et al., 2021)

5. Penelitian yang dilakukan berjudul "Pembuatan Game Balap Kelinci dengan Unity Berbasis Android". Penelitian ini bertujuan untuk mengembangkan game balap yang menghibur dengan karakter kelinci yang didesain secara 3D, menggabungkan gameplay yang menantang dengan visual yang menarik. Game ini dibangun menggunakan Unity dan karakter-karakternya dimodelkan dengan Blender. Proses pengujian melibatkan penggunaan berbagai perangkat Android untuk memastikan kompatibilitas dan performa. Hasil dari pengujian menunjukkan bahwa game ini disukai oleh pengguna, dengan kontrol yang dinilai lancar dan gameplay yang menarik, memberikan pengalaman yang positif bagi pemainnya. (Reno, 2021)

C. Kerangka Berfikir

Berdasarkan latar belakang dan kajian teori diatas maka kerangka berfikir penelitian ini dapat dilihat pada Gambar 2.2.



Gambar 2. 2 Kerangka Berpikir