

## BAB II

### KAJIAN PUSTAKA

#### A. Kajian Teoritis

##### 1. *Machine Learning*

*Machine Learning* (ML) adalah cabang dari kecerdasan buatan yang berfokus pada pengembangan algoritma dan model yang memungkinkan komputer untuk belajar berdasarkan data yang diberikan (Bintoro, 2024:1). Sedangkan menurut Sidik et al., (2022:560), *machine learning* merupakan bagian dari ilmu komputer yang bertujuan merancang algoritma agar komputer mampu belajar dari sekumpulan data, sehingga dapat membangun model pembelajaran yang optimal untuk menggali informasi dari kumpulan data.

Nurhalizah et al., (2024:61) mengemukakan bahwa *Machine Learning* merupakan cabang ilmu yang fokus pada pengembangan algoritma dan model statistik, yang memungkinkan komputer menyelesaikan berbagai tugas secara otomatis tanpa arahan langsung dari pengguna.

Berdasarkan dari beberapa pendapat tersebut, dapat disimpulkan bahwa *Machine Learning* merupakan cabang dari kecerdasan buatan dan ilmu komputer dengan fokus utama pada pengembangan algoritma dan model yang memungkinkan sistem komputer untuk belajar dari data. Proses ini memungkinkan komputer menyelesaikan suatu tugas tanpa instruksi eksplisit, dengan memanfaatkan data sebagai dasar pembentukan dari model.

## 2. Reinforcement Learning

Andreanus & Kurniawan, (2018:113) menyatakan bahwa “*Reinforcement learning* adalah bagian dari *Artificial intelligence* yang melatih algoritma dengan sistem *trial and error*.” Sedangkan menurut (Fahrurrozi et al., 2023:74), *Reinforcement Learning* merupakan metode pembelajaran yang memungkinkan sistem berinteraksi langsung dengan proses belajar, di mana algoritma akan memberikan *reward* ketika model menunjukkan kinerja baik, dan *punishment* jika hasil model memburuk.

Menurut Baradja & Tjendrowasono, (2024:192) *Reinforcement Learning* merupakan jenis algoritma dalam pembelajaran mesin yang dijalankan oleh agen untuk mempelajari serta menyempurnakan kebijakan melalui proses uji coba, dengan tujuan memperoleh akumulasi *reward* maksimal dalam jangka panjang pada tugas tertentu.

Berikut adalah macam algoritma dari *Reinforcement Learning* :

### a. Q-Learning

Menurut Mutoffar et al., (2025:217), *Q-Learning* adalah algoritma *Reinforcement Learning* yang termasuk *model-free learning*, digunakan untuk menemukan kebijakan optimal tanpa mengetahui model lingkungan. Algoritma ini memperbarui tabel Q berdasarkan pengalaman agen untuk memaksimalkan total hadiah jangka panjang.

Rumus algoritma *Q-Learning* :

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left[ R + \gamma \max_{a'} Q(s', a') - Q(s, a) \right]$$

b. *Deep Q-Networks (DQN)*

Menurut Mutoffar et al., (2025:222), *Deep Q-Network (DQN)* merupakan pengembangan dari algoritma *Q-Learning* yang menggabungkan *Reinforcement Learning* dan *Deep Learning* dengan menggunakan jaringan saraf tiruan untuk memperkirakan Q-table.

Rumus algoritma Q-Learning :

$$L_t(\theta_t) = E_{s, a, r, s'} [(y_t - Q\theta_t(s, a))^2]$$

Sumber : (Baradja & Irianto Tjendrowasono, 2024:192)

c. *Policy Gradient Methods*

Menurut (Habibi et al., 2023:19), *Policy Gradient* adalah algoritma *reinforcement learning* yang menggunakan gradien untuk memperbarui kebijakan agen berdasarkan nilai *reward* dari tindakan yang diambil. Metode ini memerlukan banyak pengujian dan penyempurnaan agar agen dapat berfungsi optimal dalam lingkungan tertentu.

d. *Proximal Policy Optimization*

Menurut Ardiansyah et al., (2024:23), *Proximal Policy Optimization (PPO)* adalah algoritma *Reinforcement Learning* yang meningkatkan kebijakan secara stabil dan efisien dengan teknik *clip* untuk mencegah perubahan kebijakan yang drastis dan mengatasi varians tinggi serta ketidakstabilan pelatihan pada *policy gradient*.

Rumus algoritma Q-Learning :

$$L^{CLIP}(\theta) = \hat{E}_t \left[ \min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon)\hat{A}_t) \right]$$

Berdasarkan beberapa pendapat di atas, dapat disimpulkan bahwa *Reinforcement Learning* (RL) adalah salah satu cabang dari *Machine Learning* yang menerapkan pendekatan *trial and error* untuk melatih algoritma. Proses ini dijalankan oleh agen yang berinteraksi dengan lingkungan untuk memperoleh *reward* sebagai umpan balik, bukan data input-output langsung. Tujuan utama dari RL adalah menyempurnakan kebijakan tindakan agar memperoleh akumulasi *reward* maksimal dalam jangka panjang.

### 3. *TCG (Trading Card Game)*

*Trading card game* (TCG) adalah jenis permainan kartu strategi yang memungkinkan setiap pemain untuk menyesuaikan dek kartunya sendiri. Yang membuat setiap pemain memiliki susunan kartu yang berbeda saat bertanding (Kuntoro, 2024:376.). Sedangkan menurut Yolanta et al., (2025:1) dalam *Trading Card Game* (TCG), setiap kartu memiliki karakteristik dan kemampuan yang unik. Hal ini menuntut pemain untuk berpikir kreatif dalam menyusun dek serta mengembangkan strategi guna mengalahkan lawan.

Berdasarkan beberapa pendapat tersebut, dapat disimpulkan bahwa *Trading Card Game* (TCG) merupakan permainan kartu strategi yang memiliki fleksibilitas tinggi, karena setiap pemain dapat menyesuaikan dan merancang dek kartunya sendiri sehingga menciptakan keunikan dalam setiap pertandingan. Setiap kartu dalam TCG memiliki karakteristik dan kemampuan yang berbeda, yang menuntut pemain untuk berpikir kreatif dan strategis dalam menyusun dek serta menghadapi lawan.

#### 4. *Turn Based Strategy*

*Turn-based strategy* (TBS) adalah mekanisme permainan, yang dimana pemain dan musuh melakukan aksi secara bergiliran. Pemain terlebih dahulu memilih tindakan untuk karakternya, kemudian sistem menjalankan aksi tersebut dan memberikan giliran kepada musuh (Adinulhaq & Safuan, 2023:15). Menurut Bethsania et al., (2018:21), *turn-based strategy game* adalah jenis permainan di mana pemain menghadapi pertarungan melawan NPC dengan sistem berbasis giliran. Menurut Hakim et al., (2020:278), game dengan genre *turn-based strategy* mengandung unsur strategi seperti permainan catur, di mana pemain harus mempertimbangkan setiap langkah atau aksi yang akan diambil pada setiap gilirannya.

Berdasarkan kutipan-kutipan tersebut, dapat diambil kesimpulan bahwa *Turn-Based Strategy* (TBS) adalah subgenre dari game strategi yang mengharuskan pemain mengambil keputusan secara bergiliran. Permainan ini biasanya melibatkan pertarungan dengan NPC dan menekankan elemen strategi, di mana pemain harus mempertimbangkan setiap langkah secara matang pada setiap giliran, mirip dengan permainan catur.

## 5. *Wayang Kulit*

Wayang merupakan salah satu bentuk kebudayaan Jawa yang telah dikenal sejak sekitar 1500 tahun lalu. Masuknya kebudayaan Hindu ke Jawa memberikan pengaruh terhadap pertunjukan bayang-bayang yang kemudian berkembang menjadi pertunjukan wayang.(Anjani & Fahrudin, 2024:22). Menurut Prasetyo et al., (2023:306), *wayang kulit* merupakan bentuk seni pertunjukan yang menggunakan boneka pipih berwarna-warni yang terbuat dari kulit hewan dan dimainkan oleh seorang dalang yang menyampaikan kisah-kisah seperti *Ramayana* dan *Mahabharata* yang disesuaikan dengan budaya Jawa. Samodro et al., (2023:83) berpendapat bahwa wayang merupakan bentuk seni yang bersifat menyeluruh karena mengandung nilai estetika, fungsi hiburan, serta unsur sakral. Wayang juga dianggap sebagai simbol budaya yang kuat dan menjadi identitas khas masyarakat Jawa, bahkan mencerminkan budaya Indonesia secara umum.

Berdasarkan beberapa pendapat tersebut, dapat disimpulkan bahwa wayang merupakan seni pertunjukan tradisional yang telah dikenal sejak ribuan tahun lalu di Jawa, berkembang dari pengaruh budaya Hindu dan memiliki nilai historis serta budaya yang kuat. Wayang kulit sendiri dimainkan oleh dalang dengan menggunakan boneka pipih dari kulit hewan untuk menyampaikan cerita seperti *Ramayana* dan *Mahabharata* dalam konteks budaya Jawa.

## 6. *Unity*

Unity merupakan sebuah alat pengembangan yang terintegrasi dan digunakan untuk membuat game, merancang arsitektur bangunan, serta membangun simulasi (Nuaja et al., 2022:2). Sedangkan menurut Mongi et al., (2018:2), Unity merupakan *game engine* yang dikembangkan oleh Unity Technologies Inc., dan berfungsi sebagai alat terintegrasi untuk pembuatan game, perancangan arsitektur bangunan, serta simulasi.

Unity menyediakan API *open source* bernama Unity ML-Agents yang berfungsi untuk menyimulasikan lingkungan pembelajaran dalam melatih NPC. Metode yang digunakan adalah *reinforcement learning*, yang memberikan *reward* kepada NPC guna mencapai hasil yang optimal (Haq et al., 2022:16).

Berdasarkan kutipan-kutipan tersebut, dapat disimpulkan bahwa Unity adalah game engine yang dikembangkan oleh Unity Technologies Inc., dan berfungsi sebagai alat pengembangan terintegrasi yang digunakan untuk membuat game, merancang arsitektur, serta membangun simulasi. Unity juga mendukung pengembangan kecerdasan buatan melalui API open source bernama *Unity ML-Agents*, yang memungkinkan simulasi lingkungan pembelajaran bagi NPC menggunakan metode *reinforcement learning* untuk mencapai performa optimal melalui sistem *reward*.

## 7. Visual Studio Code

Visual Studio Code merupakan sebuah software editor kode yang bersifat gratis dan *open source*, dikembangkan oleh Microsoft, dan tersedia untuk berbagai platform seperti Windows, macOS, Linux, bahkan dapat dijalankan melalui *web browser*. (Hidayah & Rofiqoh, 2024:382). Sedangkan menurut Romzi & Kurniawan, (2020:3), Visual Studio Code adalah editor kode sumber yang ringan namun memiliki kemampuan tinggi, dengan dukungan bawaan untuk berbagai bahasa pemrograman dan ekosistem ekstensi yang sangat beragam seperti .NET dan Unity. Menurut jubilee enterprise, (2019:3) Visual Studio adalah software Integrated Development Environment (IDE) yang dikembangkan oleh microsoft dan digunakan untuk mengembangkan program,software,aplikasi dan juga mobile app.

Berdasarkan kutipan-kutipan tersebut, dapat disimpulkan bahwa *Visual Studio Code* (VS Code) adalah editor kode sumber yang ringan, gratis, dan open source, dikembangkan oleh Microsoft serta mendukung berbagai platform seperti Windows, macOS, Linux, bahkan melalui browser. VS Code memiliki fitur bawaan yang mendukung berbagai bahasa pemrograman serta ekosistem ekstensi yang luas, termasuk untuk .NET dan Unity. Sementara itu, *Visual Studio* merupakan *Integrated Development Environment (IDE)* yang digunakan untuk pengembangan perangkat lunak, aplikasi desktop, maupun mobile.

## 8. *Game Devopment Life Cycle*

*Game Development Life Cycle* (GDLC) adalah metodologi pengembangan aplikasi berbasis game yang menggunakan pendekatan iteratif, terdiri dari enam tahap mulai dari inialisasi konsep, pra-produksi, produksi, pengujian, perilsan beta, hingga rilis final aplikasi (Wahyu, 2022:83). Menurut Krisdiawan & Darsanto, (2019:32), *Game Development Life Cycle* (GDLC) merupakan proses pengembangan game dengan pendekatan iteratif yang mencakup enam fase, yaitu inialisasi, pra-produksi, produksi, pengujian, beta, dan rilis, yang dikelompokkan dalam tiga proses utama: inialisasi, produksi, dan rilis. Menurut Prasetyo et al., (2021:106), *Game Development Life Cycle* (GDLC) merupakan metode yang digunakan dalam pengembangan game, dimulai dari tahap awal seperti pembuatan ide dan konsep, hingga tahap akhir yaitu saat game dirilis ke publik.

Berdasarkan kutipan-kutipan tersebut, dapat disimpulkan bahwa *Game Development Life Cycle* (GDLC) adalah metode atau proses iteratif dalam pengembangan game yang terdiri dari enam tahap utama, yaitu inialisasi konsep, pra-produksi, produksi, pengujian, perilsan beta, dan rilis final. Proses ini dapat dikelompokkan ke dalam tiga fase besar: inialisasi, produksi, dan rilis. GDLC dimulai dari ide awal hingga game siap dirilis ke publik, dengan pendekatan yang sistematis untuk memastikan kualitas dan kelengkapan produk akhir.

## 9. Pengujian Blackbox

*black box testing* merupakan metode pengujian perangkat lunak yang berfokus pada aspek fungsionalitas, dengan tujuan untuk mengidentifikasi berbagai kesalahan fungsi yang tidak tepat, kesalahan antarmuka, struktur data, performa, serta inisialisasi dan terminasi. (Wijaya & Astuti, 2021:23). Sedangkan menurut Jaya, (2018:45), *Black-Box Testing* adalah teknik pengujian perangkat lunak yang mementingkan spesifikasi fungsional tanpa memperhatikan struktur kontrol program, sehingga fokus utamanya berada pada informasi domain dari perangkat lunak tersebut. Menurut Febriyanti et al., (2021:2), metode *Blackbox Testing* digunakan untuk menguji perangkat lunak tanpa perlu memperhatikan detail internalnya, dengan cara mencoba program yang telah dibuat melalui pengujian input pada setiap form yang tersedia.

Dari kutipan-kutipan tersebut, dapat disimpulkan bahwa *Black Box Testing* adalah metode pengujian perangkat lunak yang berfokus pada aspek fungsionalitas sistem tanpa mempertimbangkan struktur internal atau kode program. Pengujian ini bertujuan untuk mengevaluasi apakah fungsi-fungsi perangkat lunak berjalan sesuai spesifikasi, serta untuk mengidentifikasi kesalahan pada antarmuka, struktur data, performa, dan proses inisialisasi maupun terminasi, dengan cara menguji input dan output pada setiap fitur atau form yang tersedia.

## 10. Pengujian SUS

*System Usability Scale* (SUS) adalah metode pengujian pengguna yang bersifat cepat namun tetap dapat diandalkan. Diperkenalkan oleh John Brooke pada tahun 1986, metode ini dapat digunakan untuk menilai berbagai jenis produk, termasuk situs web dan aplikasi (Ramadhan et al., 2019:140). Menurut Kosim et al., (2022:3), *System Usability Scale* (SUS) merupakan alat ukur untuk menilai *usability* dari suatu produk, aplikasi, atau sistem. SUS memiliki keunggulan berupa kemudahan penggunaan karena hanya terdiri dari 10 pertanyaan, bersifat teknologi agnostik sehingga dapat diterapkan pada berbagai jenis antarmuka. Ependi et al., (2019:66) berpendapat bahwa *System usability scale* (SUS) merupakan tehnik pengujian yang menggunakan cara yang melibatkan pengguna akhir (end user) dalam prosesnya.

Berdasarkan kutipan-kutipan tersebut, dapat disimpulkan bahwa *System Usability Scale* (SUS) adalah metode evaluasi *usability* yang dikembangkan oleh John Brooke pada tahun 1986, dan digunakan untuk menilai tingkat kegunaan dari berbagai jenis produk, seperti aplikasi dan situs web. SUS terdiri dari 10 pertanyaan sederhana yang memudahkan proses pengujian, serta bersifat teknologi agnostik sehingga dapat diterapkan pada berbagai jenis sistem. Pengujian ini melibatkan langsung pengguna akhir untuk memperoleh penilaian yang objektif terhadap pengalaman penggunaan.

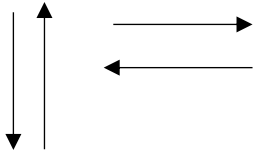
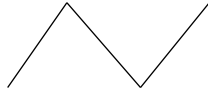
## 11. Flowchart

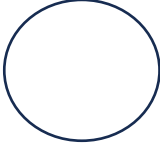

Bagan alir (*flowchart*) merupakan teknik analitis bergambar yang digunakan untuk menggambarkan berbagai aspek sistem informasi secara jelas, ringkas, dan logis. Bagan ini mencatat bagaimana proses bisnis dijalankan serta bagaimana dokumen mengalir dalam suatu organisasi (Tuasamu et al., 2023:499). Sedangkan menurut Zalukhu et al., (2023:62), *flowchart* merupakan representasi diagram alir dari algoritma dalam suatu program yang menunjukkan arah alur logika atau jalannya program tersebut.

Menurut Syamsiah, (2019:87), *flowchart* atau diagram alir merupakan bagan yang menunjukkan alur logis dalam prosedur atau program sistem. Flowchart digunakan untuk menggambarkan langkah-langkah pemecahan masalah dengan simbol-simbol standar yang mudah dipahami dan digunakan.

Berikut macam – macam simbol dari flowchart :

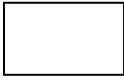

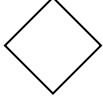
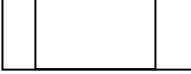

Tabel 2.1 Simbol Arus (FlowDirectionSymbols)

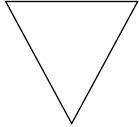

Simbol	Nama	Fungsi
	Flow direction	Berfungsi menghubungkan simbol.
	Communication link	Berfungsi untuk transmisi data.

Simbol	Nama	Fungsi
	Connector	Menyatakan sambungan proses di halaman yang sama.
	Offline connector	Menyatakan sambungan proses ke halaman berbeda.

Sumber : (Rosaly et al., 2019:5)


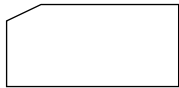

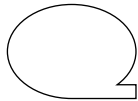
Tabel 2.2 Simbol Proses (Processing Symbols)



Simbol	Nama	Fungsi
	processing	Menunjukkan pengolahan yang dilakukan pada komputer.
	Manual operation	Menunjukkan pengolahan yang tidak dilakukan komputer.
	Decision	Memilih proses berdasarkan kondisi tertentu.
	Predefined process	Mempersiapkan penyimpanan yang sedang digunakan.
	Terminal	Memulai atau mengakhiri program.

Simbol	Nama	Fungsi
	Offline storage	Menunjukkan data akan disimpan pada media.
	Manual input	Menginputkan data secara manual.

Sumber : (Rosaly et al., 2019:6)

Tabel 2.3 Simbol I/O (Input-Output)

Simbol	Nama	Fungsi
	Input/Output	Menyatakan input dan output.
	Punched card	Menyatakan masukan dan keluaran yang berasal dari card.
	Disk Storage	Menyatakan masukan dan keluaran dari disk.
	Magnetic Tape	Menyatakan keluaran dan masukan yang berasal dari pita magnetis

Simbol	Nama	Fungsi
	Document	Menyatakan keluaran dan masukan berasal dari dokumen.
	Display	Menyatakan keluaran melalui layar monitor.

Sumber : (Rosaly et al., 2019:7)

## 12. *Unified Modeling Language (UML)*

*Unified Modeling Language (UML)* adalah metode pemodelan visual yang digunakan dalam proses perancangan dan pengembangan perangkat lunak berorientasi objek (Prihandoyo, 2018:12). Sedangkan menurut Wahyuni et al., (2022:856), *Unified Modeling Language (UML)* merupakan metode pemodelan visual yang digunakan dalam perancangan sistem berorientasi objek, serta berfungsi sebagai bahasa standar untuk visualisasi, perancangan, dan pendokumentasian sistem.

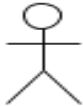


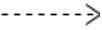

Menurut Binangkit et al., (2023:1430), *Unified Modeling Language (UML)* adalah bahasa standar yang digunakan untuk menggambarkan, menjelaskan, dan membangun perangkat lunak dalam pengembangan sistem berbasis objek. UML juga berperan sebagai alat bantu dalam proses perancangan, yang mencakup berbagai jenis diagram seperti *use case*, *activity diagram*, *sequence diagram*, dan *class diagram*.

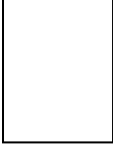



Berikut macam – macam diagram dari UML :

a. Use case diagram

*Use case* diagram digunakan untuk menggambarkan persyaratan suatu sistem, yaitu bagaimana sistem seharusnya digunakan. Komponen utamanya terdiri dari aktor, *use case*, dan subjek (sistem). (Destriana et al., 2022:7).

Tabel 2.4 Use case diagram

Simbol	Nama	Fungsi
	<i>Actor</i>	Peran <i>user</i> saat berinteraksi dengan use case
	<i>Dependency</i>	Perubahan satu elemen memengaruhi elemen bergantung.
	<i>Generalization</i>	Pewarisan perilaku dari induk ke anak.
	<i>Include</i>	Use case sumber menyertakan use case lain secara eksplisit.
	<i>Association</i>	Hubungan antar objek.






Simbol	Nama	Fungsi
	<i>System</i>	Representasi sistem secara terbatas.
	<i>Use Case</i>	Urutan aksi sistem untuk hasil bagi aktor.
	<i>Collaboration</i>	Kerja sama elemen.
	<i>Note</i>	Elemen fisik yang mencerminkan sumber daya.

Sumber : (Destriana et al., 2022:7)

#### b. Activity diagram

*diagram aktivitas* merupakan representasi visual dari alur kerja yang mencakup aktivitas, tindakan, pilihan, pengulangan, dan *concurrency*. Dalam *Unified Modeling Language* (UML), diagram ini digunakan untuk menjelaskan aktivitas sistem komputer maupun proses dalam organisasi. (Destriana et al., 2022:10).

Tabel 2.5 Activity diagram

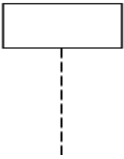
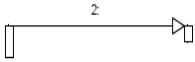
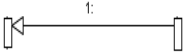
Simbol	Nama	Fungsi
	<i>Activity</i>	Interaksi antar kelas antarmuka.
	<i>Action</i>	Eksekusi dari suatu aksi sistem.
	<i>Initial Node</i>	Titik awal objek dibentuk.
	<i>Activity Final Node</i>	Titik akhir objek dihancurkan.
	<i>Fork Node</i>	Satu aliran menjadi beberapa aliran.

Sumber : (Destriana et al., 2022:11)

### c. Sequence diagram

*sequence diagram* digunakan untuk menggambarkan interaksi dalam sistem dengan menekankan urutan pesan yang dipertukarkan antar objek, serta menunjukkan spesifikasi kejadian yang terjadi pada masing-masing *lifeline* (Destriana et al., 2022:14).

Tabel 2.6 Sequence diagram




Simbol	Nama	Fungsi
	<i>LifeLine</i>	Objek antarmuka yang saling berinteraksi.
	<i>Input Message</i>	Komunikasi antar objek yang memuat informasi.
	<i>Output Message</i>	Pesan yang dihasilkan sistem sebagai respon.

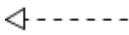
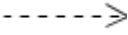

Sumber : (Destriana et al., 2022:11)

#### d. Class diagram

*Class diagram* merupakan representasi visual dari struktur sistem yang menggambarkan jenis-jenis kelas yang dibentuk, serta menunjukkan alur kerja dari basis data dalam sistem tersebut. (Destriana et al., 2022:19).

Tabel 2.7 Class diagram

Simbol	Nama	Fungsi
	<i>Generalization</i>	Hubungan pewarisan sifat antara objek induk dan anak.
	<i>Class</i>	Kumpulan objek dengan atribut dan operasi yang sama.
	<i>Collaboration</i>	Urutan aksi sistem untuk mencapai hasil bagi aktor.

Simbol	Nama	Fungsi
	<i>Realization</i>	Implementasi nyata dari suatu operasi oleh objek.
	<i>Dependency</i>	Perubahan satu elemen memengaruhi elemen lain yang bergantung padanya.
	<i>Association</i>	Hubungan antara dua objek.

Sumber : (Destriana et al., 2022:20)

### 13. Storyboard

*Storyboard* merupakan representasi grafis yang awalnya digunakan untuk menggambarkan urutan adegan dalam produksi film dan televisi. Dalam rekayasa perangkat lunak, *storyboard* dimanfaatkan untuk memvisualisasikan urutan interaksi pengguna melalui antarmuka pengguna (*user interface*) (Nendya et al., 2023:2). Sedangkan menurut Imbar et al., (2021:111), *storyboard* adalah visualisasi ide dari aplikasi yang akan dikembangkan, yang bertujuan untuk memberikan gambaran awal mengenai bentuk dan alur aplikasi yang akan dihasilkan. Menurut Rizqi & Baihaqi, (2024:41), *storyboard* merupakan rangkaian sketsa atau naskah yang menggambarkan alur cerita, dan digunakan untuk memberikan gambaran rinci mengenai aplikasi yang sedang dirancang.

Berdasarkan kutipan-kutipan tersebut, dapat disimpulkan bahwa *storyboard* adalah representasi visual yang digunakan untuk menggambarkan alur interaksi dan struktur dari sebuah aplikasi. Awalnya digunakan dalam produksi film, dalam konteks rekayasa perangkat lunak *storyboard* berfungsi untuk memvisualisasikan ide, bentuk, serta alur antarmuka pengguna secara rinci. Rangkaian sketsa ini membantu tim pengembang memahami dan merancang aplikasi secara lebih terarah sebelum proses implementasi dimulai.

## **B. Kajian Empiris**

Penelitian oleh Fitrianto et al., (2021) berjudul “*Pengembangan Augmented Reality Card Game Amogasakti untuk Melestarikan Budaya Wayang Indonesia*”. Latar belakang penelitian ini adalah semakin menurunnya minat masyarakat terhadap budaya wayang yang mengancam keberlangsungannya. Tujuan penelitian ini adalah mengembangkan media edukatif melalui permainan kartu berbasis Augmented Reality untuk memperkenalkan tokoh-tokoh wayang dalam bentuk visual 3D. Pengembangan dilakukan dengan model Four-D (Define, Design, Develop, Disseminate) dan hasil validasi menyatakan bahwa sistem sangat layak digunakan dari aspek perangkat lunak dan visualisasi. Kelebihan penelitian ini adalah fokus pada pelestarian budaya lokal melalui teknologi, namun tidak mengangkat aspek kecerdasan buatan (AI) atau gameplay dinamis yang dapat belajar dari pemain.

Penelitian oleh Kuntoro (2024) berjudul “*Implementasi Penggunaan Artificial Intelligence dalam Pembuatan Collectible Card Game Berlisensi Nonkomersial*”. Penelitian ini mengembangkan ilustrasi karakter kartu pada

game CCG menggunakan teknologi text-to-image AI untuk menggantikan ilustrator manusia. AI digunakan untuk membuat aset visual dari deskripsi teks secara otomatis, sehingga mempercepat proses desain karakter kartu. Tujuan utama penelitian ini adalah mendemonstrasikan kemampuan AI dalam pembuatan visual, bukan pada kecerdasan dalam bermain. Penelitian ini relevan dalam pemanfaatan AI pada game, namun tidak mencakup AI adaptif dalam gameplay atau integrasi dengan mekanisme strategi seperti Reinforcement Learning. Penelitian oleh Arga et al., (2018) berjudul “*Pengembangan Permainan Trading Card Augmented Reality Bertema Perang Baratayuda*”. Fokus penelitian ini adalah mengangkat cerita tradisional Baratayuda ke dalam game kartu dengan tampilan Augmented Reality. Game ini dikembangkan menggunakan metode Iterative with Rapid Prototyping dan bertujuan mengenalkan budaya Jawa kepada generasi muda. Sistem belum memiliki AI yang kompleks, dan gameplay masih bersifat linier dan tergantung input pemain. Penelitian ini mendukung pelestarian budaya melalui media interaktif, tetapi belum menjawab tantangan strategi dinamis dalam permainan.

Dalam penelitian ini mengusulkan *implementasi Reinforcement Learning pada game Trading Card bertema wayang berbasis turn based strategy*, dengan pendekatan berbasis Unity ML-Agents. Penelitian ini bertujuan tidak hanya untuk melestarikan budaya lokal, tetapi juga menerapkan AI yang dapat belajar dan beradaptasi terhadap strategi pemain. Berbeda dengan penelitian sebelumnya, sistem AI dalam skripsi ini bersifat dinamis dan berkembang berdasarkan reward dari setiap interaksi dalam permainan.

Kelebihannya adalah integrasi budaya lokal, gameplay strategi, dan teknologi AI adaptif secara menyeluruh.

### **C. Kerangka Berpikir**

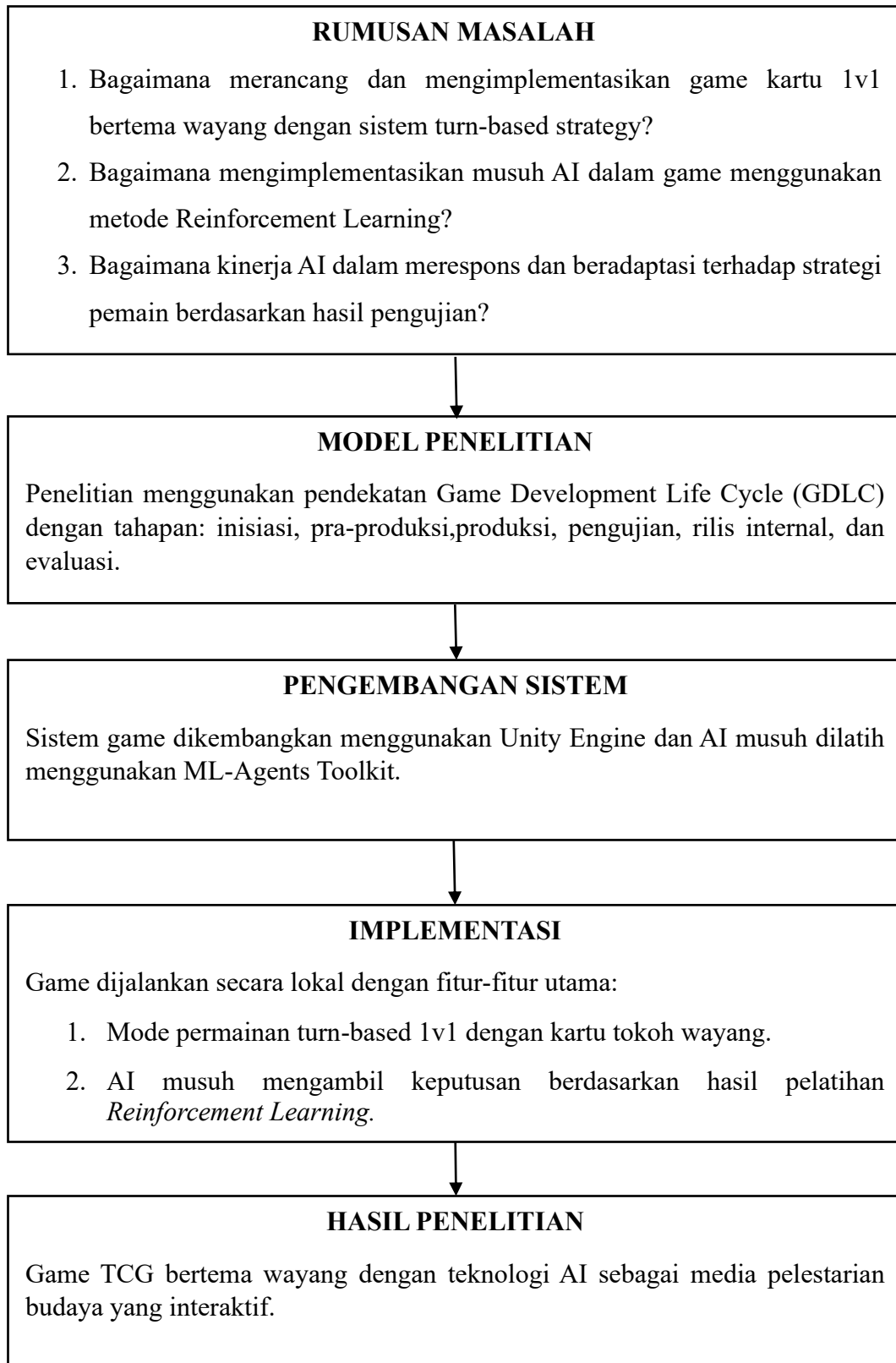
Pada penelitian yang telah dilakukan sebelumnya, sebagian besar masih terbatas pada pengembangan permainan kartu (Trading Card Game/TCG) dengan pendekatan visualisasi budaya, seperti yang dilakukan oleh Fitrianto et al., (2021) dan Arga et al., (2018), yang berfokus pada aspek edukatif dan pelestarian budaya melalui Augmented Reality. Meskipun berhasil mengenalkan budaya wayang kepada generasi muda, penelitian-penelitian ini belum mengintegrasikan AI adaptif dalam gameplay.

Penelitian oleh Kuntoro (2024) menunjukkan pemanfaatan teknologi Artificial Intelligence dalam pembuatan ilustrasi kartu pada game CCG. Namun, penggunaan AI dalam konteks tersebut hanya terbatas pada penciptaan aset visual, bukan perilaku adaptif musuh dalam permainan. Dari berbagai penelitian tersebut, terlihat adanya kekurangan dalam aspek interaktivitas dan adaptasi musuh dalam gameplay, khususnya pada pengembangan AI yang mampu merespons strategi pemain secara cerdas dan dinamis.

Berangkat dari permasalahan tersebut, penelitian ini bertujuan untuk mengembangkan permainan TCG berbasis turn-based strategy yang mengangkat tokoh-tokoh wayang sebagai elemen budaya, dengan menerapkan metode Reinforcement Learning untuk membentuk AI musuh yang adaptif. Teknologi Unity ML-Agents Toolkit digunakan untuk melatih agen AI agar mampu belajar

dari hasil permainan dan meningkatkan kualitas pengambilan keputusan berdasarkan sistem reward.

Dengan pendekatan ini, game yang dikembangkan tidak hanya berfungsi sebagai media edukatif dan pelestarian budaya lokal, tetapi juga menghadirkan tantangan gameplay yang lebih interaktif dan realistis. Penelitian ini diharapkan dapat memberikan kontribusi baru dalam pengembangan game edukatif berbasis AI adaptif, serta mengisi kekosongan yang belum dijawab oleh penelitian-penelitian terdahulu.



Gambar 2.1 Kerangka Berfikir