

## BAB II

### TINJAUAN PUSTAKA

#### A. Penelitian Terkait

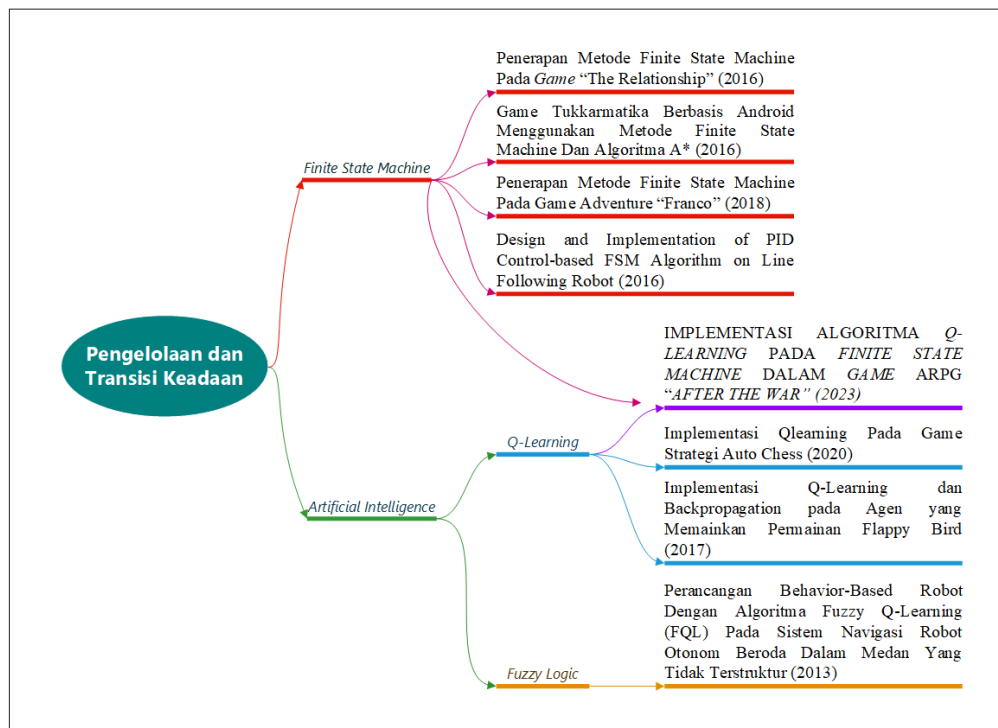
Karya Ilmiah yang digunakan sebagai referensi dalam peneliitan ini ditunjukkan pada tabel penelitian terkait berikut:

Tabel 2. 1 Penelitian terkait

No	Peneliti	Judul	Metode	Hasil Penelitian
1	Miftah Fauzan Rahadian, Addy Suyatno, Septya Maharani (2016)	Penerapan Metode <i>Finite State Machine</i> Pada Game “ <i>The Relationship</i> ”	<i>Finite State Machine (FSM)</i>	Penerapan metode <i>Finite State Machine</i> menghasilkan respon NPC yang berbeda menyesuaikan apa yang telah dimainkan oleh pemain. Hasil akhir <i>game</i> yang ditentukan oleh pemain karena metode <i>Finite State Machine</i> menawarkan pilihan sepanjang permainan berlangsung.
2	Muhammad Murti Hanafi (2016)	Game Tukkarmatika Berbasis Android Menggunakan Metode <i>Finite State Machine</i> Dan Algoritma A*	<i>Finite State Machine (FSM)</i>	Metode <i>Finite State Machine</i> dan Algoritma A* dapat bersinergi dalam <i>Game</i> Tukkarmatika sehingga menjadikan <i>game</i> lebih dinamis.
3	Naiman Fahrudin (2018)	Penerapan Metode <i>Finite State Machine</i> Pada <i>Game</i> Adventure “Franco”	<i>Finite State Machine (FSM)</i>	Hasil Pengujian AI yang ada pada <i>Game</i> Adventure “Franco” berjalan dengan tingkat keberhasilan 100 % sesuai dengan yang diharapkan.

No	Peneliti	Judul	Metode	Hasil Penelitian
4	Made Santo Gitakarma, Gede Nurhayata (2013)	Perancangan Behavior-Based Robot Dengan Algoritma <i>Fuzzy Q-Learning</i> (Fql) Pada Sistem Navigasi Robot Otonom Beroda Dalam Medan Yang Tidak Terstruktur	Algoritma <i>Q-Learning</i>	Penerapan <i>Fuzzy Q-Learning</i> memberikan hasil yang relatif lebih baik dari <i>Q-Learning</i> dalam jumlah penghargaan yang diterima dan performa navigasi robot mandiri, meski algoritma FQL membutuhkan waktu yang lebih lama untuk dieksekusi. <i>Fuzzy Q-Learning</i> yang termodifikasi dapat menghemat memori robot tidak memberikan penurunan signifikan pada performa robot, sehingga MFQL masih dapat digunakan sebagai media simplifikasi.
5	Fajar Salam (2020)	Implementasi Qlearning Pada <i>Game</i> Strategi <i>Auto Chess</i>	Algoritma <i>Q-Learning</i>	<i>Reinforcement Learning</i> yang dikombinasikan dengan <i>Q-Learning</i> dapat mempersingkat waktu pelatihan dengan hasil yang lebih optimal dibandingkan dengan <i>Basic Reinforcement Learning</i> dengan efektifitas waktu dan hasil yang diperoleh lebih baik daripada <i>Basic Reinforcement Learning</i>
6	Ardiansyah, Ednawati Rainarli (2017)	Implementasi <i>Q-Learning</i> Dan <i>Backpropagation</i> Pada Agen Yang Memainkan Permainan <i>Flappy Bird</i>	Algoritma <i>Q-Learning</i>	Implementasi <i>Q-Learning</i> yang dikombinasikan dengan <i>Backpropagation</i> dapat membuat waktu pembelajaran agen untuk memainkan <i>Flappy Bird</i> lebih cepat hingga 92% dan dapat mengurangi bobot yang disimpan di memori hingga 94%, jika dibandingkan dengan penggunaan <i>Q-Learning</i> saja.
7	Adnan Rafi Al Tahtawi, Yoyo Somantri, Erik Haritman3 (2016)	<i>Design And Implementation Of PID Control-Based FSM Algorithm On Line Following Robot</i>	<i>Finite State Machine (FSM)</i>	Algoritma FSM berbasis PID telah berhasil dirancang dan diimplementasikan pada <i>line follower robot</i> . Hasil pengujian menunjukkan bahwa dengan menggabungkan kontroler PID dan algoritma FSM, robot pengikut garis dapat merespons dengan baik.

Perbedaan penelitian ini dengan penelitian – penelitian sebelumnya adalah penelitian ini memiliki fokus utama pada implementasi algoritma *Q-Learning* dalam kerangka *Finite State Machine* (FSM). Pendekatan ini bertujuan untuk meningkatkan responsivitas dan kecerdasan suatu sistem, khususnya dalam konteks yang memanfaatkan FSM. Penelitian ini mencoba menggabungkan konsep *Q-learning*, yang merupakan metode pembelajaran mesin, dengan FSM untuk menghasilkan suatu sistem yang dapat belajar dan beradaptasi dengan lingkungan atau situasi tertentu. Berikut disajikan diagram *mind map* untuk memperjelas deskripsi diatas.



Gambar 2. 1 *Mind Map* Penelitian

## B. Landasan Teori

### 1. *Game*

Dalam bahasa Indonesia, istilah *Game* memiliki arti sebagai permainan. Menurut KBBI, permainan memiliki arti sesuatu yang digunakan untuk bermain, barang atau sesuatu yang dipermainkan. Konsep permainan dalam *game* juga merujuk pada konsep "kelincahan

intelektual" atau "intelektualitas permainan." Sementara itu, kata "game" dapat diartikan sebagai suatu arena di mana pemain membuat keputusan dan melibatkan aksi. Terdapat target-target yang ingin dicapai oleh pemain. Tingkat "kelincahan intelektual" dapat menjadi ukuran sejauh mana *game* tersebut menarik untuk dimainkan secara optimal [10].

*Game* merupakan media yang digunakan untuk menyampaikan suatu pesan kepada masyarakat umum dalam bentuk permainan yang dapat menghibur [11]. Pada penelitian ini *game* yang digunakan sebagai media implementasi merupakan *game* 3D yang ber-genre *Action Role Playing Game* (ARPG) yang berjudul "After The War" yang dikembangkan sendiri oleh penulis.

## 2. *Finite State Machine*

*Finite State Machine* (FSM) atau finite-state automaton (FSA, plural: automata), finite automaton, atau sederhananya state machine, adalah model komputasi matematis. FSM adalah mesin abstrak yang dapat berada tepat di salah satu dari sejumlah keadaan terbatas pada waktu tertentu. *Finite State Machine* adalah sebuah metodologi perancangan sistem kontrol yang menggambarkan tingkah laku atau prinsip kerja sistem dengan menggunakan tiga hal berikut, State (Keadaan), Event (kejadian) dan action (aksi). FSM dapat berubah dari satu keadaan ke keadaan lain sebagai respons terhadap beberapa masukan; perubahan dari satu keadaan ke keadaan lain disebut transisi [12].

*Finite State Machine* digunakan untuk mendukung interaksi yang terdiri dari serangkaian state yang menentukan keputusan dari setiap state untuk dapat berpindah dari state satu ke state lainnya. Sehingga akan membentuk perilaku karakter NPC (*Non Player Character*) yang dapat memberikan tindakan sesuai dengan masukan dari player [13].

### 3. *Artificial Intelligence*

*Artificial Intelligence* atau kecerdasan buatan adalah sistem komputer yang mampu melakukan tugas-tugas yang biasanya membutuhkan kecerdasan manusia. Teknologi ini dapat membuat keputusan dengan cara menganalisis dan menggunakan data yang tersedia di dalam sistem. Proses yang terjadi dalam *Artificial Intelligence* mencakup *learning*, *reasoning*, dan *self-correction*. Proses ini mirip dengan manusia yang melakukan analisis sebelum memberikan keputusan [14].

Penelitian ini menggunakan pendekatan *reinforcement learning* dengan subbidang *machine learning*. Hal ini bertujuan agar NPC pada *game* dapat berkembang seiring berjalannya permainan.

### 4. *Machine Learning*

*Machine learning* adalah teknologi yang memungkinkan mesin untuk belajar secara otomatis tanpa perlu diprogram secara eksplisit. Teknologi ini merupakan salah satu cabang dari kecerdasan buatan (AI) yang sangat menarik perhatian, karena mesin dapat belajar layaknya manusia. Cara kerja *machine learning* melibatkan pengumpulan data yang relevan, persiapan data untuk analisis, pemilihan model pembelajaran mesin yang sesuai, dan pelatihan algoritma dengan data yang ada. melalui proses pembelajaran, mesin dapat membuat keputusan atau melakukan tugas berdasarkan pola yang ditemukan dalam data [15].

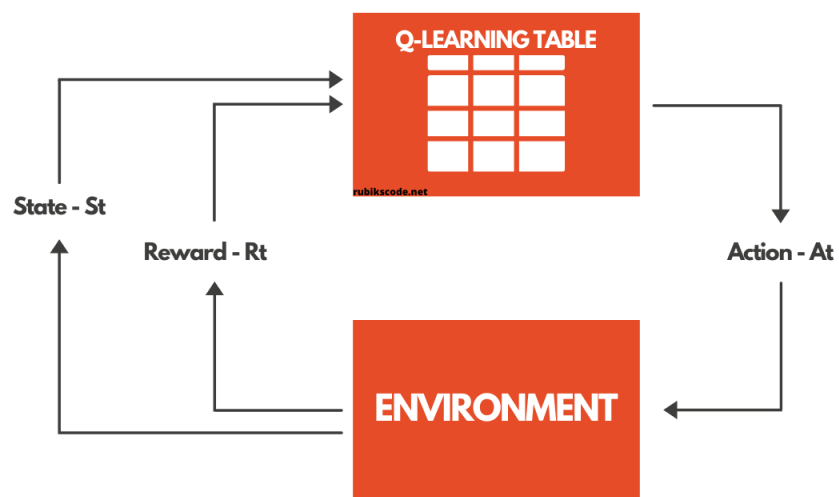
Salah satu pendekatan yang umum digunakan dalam pengembangan *game* menggunakan *Machine learning* adalah penggunaan algoritma pembelajaran mesin untuk meningkatkan kecerdasan buatan (AI) dalam permainan. Dalam penelitian oleh Brown et al. [16], dijelaskan bagaimana teknik-teknik seperti *Reinforcement Learning* digunakan untuk melatih agen cerdas yang dapat belajar dan beradaptasi dengan lingkungan *game*. Algoritma *Reinforcement Learning* mampu mencapai

hasil yang optimal dalam lingkungan yang dinamis dan kompleks, sehingga cocok untuk diterapkan dalam pengembangan perilaku karakter dalam *game*. Dengan algoritma ini, karakter dalam *game* dapat belajar untuk mengambil keputusan yang optimal dalam situasi yang berubah-ubah, serta menyesuaikan strategi berdasarkan informasi baru yang diperoleh dari lingkungan [17].

## 5. Algoritma *Q-learning*

Algoritma *Q-Learning* adalah salah satu metode dalam *Reinforcement Learning* yang digunakan untuk membuat agen belajar mengambil keputusan dalam suatu lingkungan untuk mencapai tujuan tertentu [18]. Algoritma ini dikembangkan oleh Christopher J.C.H Watkins pada tahun 1989.

Konsep dasar dari *Q-Learning* melibatkan pembuatan *Q-table* yang menyimpan nilai-nilai *Q* untuk setiap kombinasi dari *state* (keadaan) dan *action* (aksi) yang mungkin dilakukan oleh agen. Nilai *Q* menggambarkan seberapa baik suatu aksi dilakukan dalam suatu keadaan tertentu, dengan tujuan untuk memaksimalkan total *reward* (hadiah) yang diperoleh oleh agen seiring waktu [18].



Gambar 2. 2 Siklus Pembelajaran Algoritma *Q-Learning*

Berikut *Pseudocode* dari Algoritma *Q-Learning*:

**Procedure** *QLearning*  
**Algoritma :**  
**Learning Process**  
Inisialisasi *Q-table*  
Set Hyperparameter  
Inisialisasi keadaan awal (*state*)  
Pilih tindakan (*action*)  
Terapkan tindakan dan peroleh *reward* (*reward, next\_state*)  
**Update Q-Value**  
Apakah *next\_state* merupakan keadaan terminal?  
Jika Yes (Ya) maka :  
     $Q(\text{state}, \text{action}) \leftarrow (1 - \alpha) \cdot Q(\text{state}, \text{action}) + \alpha \cdot \text{reward}$   
Jika No (Tidak) maka :  
     $Q(\text{state}, \text{action}) \leftarrow (1 - \alpha) \cdot Q(\text{state}, \text{action}) + \alpha \cdot (\text{reward} + \gamma \cdot \max(Q[\text{next\_state}, \text{semua\_tindakan}]))$   
Kembali ke Inisialisasi keadaan awal  
**EndProcedure**

Formula update *Q-value* dalam *Q-Learning* [18]:

$$Q(s, a) = (1 - \alpha) \cdot Q(s, a) + \alpha \cdot (r + \gamma \cdot \max_{a'} Q(s', a'))$$

Dimana:

- $Q(s, a)$  adalah nilai  $Q$  untuk keadaan  $s$  dan aksi  $a$ ,
- $\alpha$  adalah tingkat pembelajaran (*learning rate*),
- $r$  adalah hadiah (*reward*) yang diterima setelah melakukan aksi  $a$  dalam keadaan  $s$ ,
- $\gamma$  adalah faktor diskon untuk mengatur pengaruh nilai  $Q$  masa depan,
- $s'$  adalah keadaan berikutnya setelah melakukan aksi  $a$ , dan
- $\max_{a'} Q(s', a')$  adalah nilai  $Q$  tertinggi untuk keadaan  $s'$ .

## 6. ARPG

*Role-playing games* (RPGs) adalah salah satu genre *video game* yang menekankan peran pemain sebagai karakter di dalam dunia permainan. Ada berbagai jenis RPG, dan salah satu jenis yang cukup populer adalah *Action RPG* (ARPG). ARPG menggabungkan elemen-

elemen RPG tradisional dengan *gameplay real-time action*, memberikan pengalaman yang dinamis kepada pemain [19].

Dalam ARPG, pemain sering kali memiliki kebebasan untuk menentukan perkembangan karakter mereka melalui pemilihan keterampilan, atribut, dan perlengkapan. Ini menciptakan pengalaman bermain yang lebih personal dan disesuaikan dengan preferensi pemain. Selain itu, ARPG sering kali menawarkan pilihan moral yang dapat mempengaruhi perkembangan cerita dan hubungan dengan karakter lain di dalam permainan, menambahkan elemen RPG tradisional [20].

## 7. Unity

Unity adalah sebuah platform pengembangan permainan atau *game engine* dan simulasi visual. Dikembangkan oleh Unity Technologies, platform ini telah menjadi tulang punggung bagi sejumlah besar *game*, aplikasi AR/VR, simulasi, dan konten interaktif lainnya. Kesederhanaan penggunaannya bersanding dengan kekuatan dan fleksibilitas, menjadikannya pilihan utama bagi pengembang di seluruh spektrum industri kreatif.

Unity dapat dijalankan pada microsoft Windows dan MAC OS X, dan permainan yang dihasilkan dapat dijalankan pada Windows, MAC, Xbox 360, PlayStation 3, Nintendo Wii, iPad, iPhone, Android dan Linux. Unity 3D juga dapat menghasilkan permainan untuk browser dengan menggunakan plugin Unity Web Player. Unity 3D juga memiliki kemampuan untuk mengekspor permainan yang dibangun untuk fungsionalitas Adobe Flash 3D [21].

## 8. C#

C# (dibaca "C sharp") adalah bahasa pemrograman yang memainkan peran sentral dalam dunia pengembangan perangkat lunak modern. Dikembangkan oleh Microsoft, C# menawarkan kombinasi kekuatan, kesederhanaan, dan fleksibilitas yang menjadikannya pilihan



utama untuk berbagai aplikasi, mulai dari pengembangan permainan hingga pembuatan aplikasi desktop dan web.

C# adalah salah satu dari banyak bahasa yang bisa dipakai untuk pemrograman .NET. Kelebihan utama bahasa ini adalah sintaksnya yang mirip C, namun lebih mudah dan lebih bersih. C# sebagai bahasa pemrograman untuk Framework .NET memiliki ruang lingkup penggunaan yang sangat luas. Pembuatan program dengan user interface Windows maupun console dapat dilakukan dengan C#. Karena Framework .NET memberikan fasilitas untuk berinteraksi dengan kode yang *unmanaged*, penggunaan library seperti DirectX 8.1 dan OpenGL dapat dilakukan. C# juga dapat digunakan untuk pemrograman *web site* dan *web service* [22].

## 9. GDLC (*Game Development Life Cycle*)

*Game Development Life Cycle* merupakan proses pengembangan dari sebuah *game* yang lebih mengutamakan aspek interaktif yang memiliki enam fase pengembangan, dimulai dari fase inisialisasi/pembuatan konsep, *pre-production*, *production*, *testing*, *beta* dan *release* [23].

### a. *Initiation*

Pada tahap ini akan dilakukan pembuatan konsep *game* yang akan dibangun, berupa analisa akan seperti apa *game* itu keseluruhan, inisiasi akan menghasilkan konsep *game* dan deskripsi sederhana.

### b. *Pre-production*

*Pre-production* adalah awal dari *production cycle* yang berurusan dengan *game design*. *Pre-production* adalah tahap yang vital sebelum proses *production* dimulai, karena pada tahap ini dilakukan perancangan *game*, dan rencana produksi *game*.

### c. *Production*

*Game design* dan *prototype* yang ada pada *pre-production* disempurnakan pada produksi. Artinya, tahap ini memiliki fokus pada menerjemahkan rancangan *game design*, *concept art*, dan aspek – aspek lainnya menjadi unsur penyusun *game*.

d. *Testing*

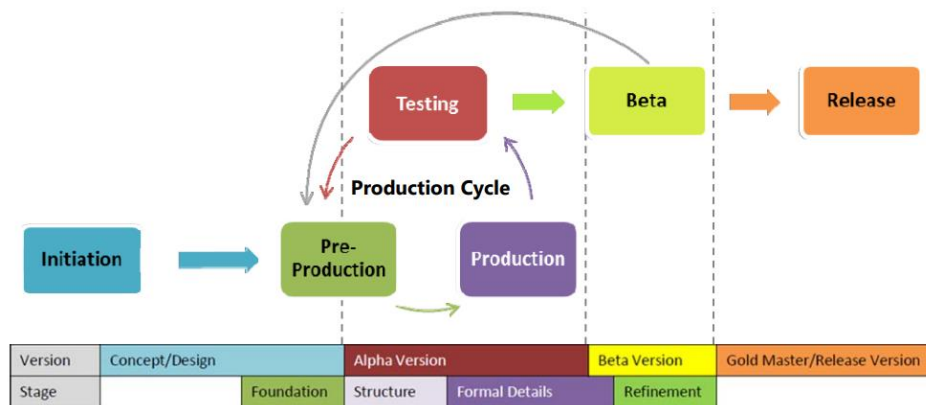
Tahap *testing* atau pengujian merupakan tahap dimana akan dijalankan serangkaian evaluasi untuk memastikan kualitas, keberfungsian, dan pengalaman pengguna yang optimal.

e. *Beta*

*Eksternal testing*, dikenal dengan istilah *beta testing* dilakukan untuk menguji keberterimaan *game* dan untuk mendeteksi berbagai *error* dan keluhan dari *third party tester*.

f. *Release*

*Release* adalah tahap dimana *final build* dari *game* resmi dirilis. Tahap ini dilakukan setelah *game* lulus dari *beta testing* yang menandakan *game* tersebut siap untuk dirilis ke publik.



Gambar 2. 3 *Game Development Life Cycle*