BAB II
TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Terdapat 3 penelitian sejenis :

Penelitian yang berjudul "Penerapan Pemilihan Model Dinamis Algoritma *Behaviour Tree Decision* dalam *Third Person Game* pada Musuh *Non-Playable Character*" menerangkan, pembuatan *game* ini diperlukan beberapa persiapan yang harus dilakukan, menyiapkan model algoritma yang diperlukan, piranti lunak, hingga sarana lain. Dalam pengembangan model kecerdasan buatan dengan algoritma *behavior tree* ini terdapat *random location*, *chasing player*, *patrolling*, NPC *speed*, *searching* dan *melee attack* (Qowy, Hanafi, Riandi, & Nuraminah, 2021)

Penelitian yang berjudul "Implementasi *Behavior Tree* Pada Perilaku NPC di *Game Sidescroller*" menerangkan, untuk menggunakan *Behavior Tree* untuk mengatur perilaku NPC dalam *game* agar lebih alami dan menarik. NPC memiliki tingkah laku yang bermacam-macam seperti menyerang, berkeliling, mengikuti, dan menghancurkan diri. Algoritma ini membantu mengontrol perilaku dengan menggunakan representasi perilaku dalam bagan berbentuk pohon (Junaidi, Yunus, & Wiguna, 2021).

Penelitian yang berjudul "Membangun *Game* 3d *Side Scroll* Dan Menerapkan Model *Behavior Tree* pada Npc *Enemy* Dalam *Game* (Maverick)" menerangkan, *behavior tree* adalah sebuah cara yang terstruktur dalam pengalihan tugas pada

autonom agent atau AI seperti robot atau *virtual entity* dalam *game* komputer (npc). Gambaran implementasi *behavior tree* pada *game* Maverick terdapat perilaku dari npc *patrol path behavior* dalam *game* ini pada saat tidak mendeteksi adanya *player* yang berada di dekatnya, perilaku *chase and attack* adalah dimana *player* terlihat oleh npc kemudian melakukan pengejaran terhadap *player* dan akan menyerang *player* jika jarak antara npc dan *player* cukup dekat dan perilaku *losing sight* terjadi jika musuh kehilangan *player* atau jika *player* berhasil kabur dari npc. (Pratiwi, Pahrudin, & Nassa, 2021).

Dari 3 penelitian di atas memiliki perbedaan dengan penelitian penulis. Penelitian yang dilakukan qowy dkk menerapkan 6 *task* pada algoritmanya terdapat *random location*, *chasing player*, *patrolling*, *NPC speed*, *searching*, *melee attack* secara garis besar memiliki kemiripan dengan BT penulis, penulis juga menjalankan ke 6 *task* tersebut dan penulis juga menambahkan *task strafe movement* yaitu pergerakan ke samping sambil menghadap target daripada membalikkan tubuh. Pada penelitian junaidi dkk mengambil tipe *game side scrolling* yaitu tipe *game* di mana *player* bergerak melalui alur permainan dalam satu arah untuk menyelesaikan misinya sedangkan penulis mengambil tipe *game Adventures*, *Action*-RPG yaitu *player* akan menjelajahi map untuk menyelesaikan *quest* dan terkait dengan tantangan fisik. Pada penelitian pratiwi dkk mengambil tipe yang sama dengan penelitian kedua *side scroll* dan menggunakan unreal engine versi 4 sebagai *engine* yang dipakai untuk merancang *game* “Maverick” sedangkan penulis mengambil tipe *game Adventures*, *Action*-RPG yang memanfaatkan unreal

engine versi 5 sebagai *engine* yang dipakai untuk merancang *game* "Hunter's Return".

2.2 Game

Game adalah permainan interaktif komputer yang dikendalikan oleh mikroprosesor. *Game* komputer menggunakan satu atau beberapa perangkat, biasanya dalam bentuk tombol atau kombinasi antara *joystick*, *mouse* dan *keyboard*. Terdapat beberapa istilah dalam video *game* seperti platform dan genre. Platform mengacu pada perangkat atau alat yang digunakan untuk bermain video *game* dan genre mengacu pada klasifikasi video game berdasarkan kesamaan karakteristiknya. (Herdiana & Hermanto, 2019).

2.3 Jenis-Jenis Game

Terdapat banyak jenis-jenis *game* dan variasinya, mulai dari media dalam menjalankannya berbeda dari tahapan dalam bermain dan seberapa banyak *player*. Terdapat berbagai jenis permainan yang dibedakan berdasarkan jenis permainan yang biasanya dimainkan pada komputer atau handphone (Ridoi, 2018).

2.3.1. Action Games

Jenis permainan aksi melibatkan tantangan fisik, permainan teka-teki, permainan balap. Terkadang mencakup masalah ekonomi sederhana seperti mengoleksi barang.

2.3.2. Adventure Games

Game Petualangan memfokuskan pada penjelajahan dalam menyelesaikan teka-teki biasanya juga melibatkan masalah konseptual dan tantangan fisik, meskipun jarang terjadi.

2.3.3. Role-Playing Games

Game jenis RPG biasanya memiliki fokus pada strategi, *logic*, penjelajahan melibatkan misteri dan masalah ekonomi dengan cara mengumpulkan barang jarahan dan mendagangkannya untuk memperkuat perlengkapan.

2.3.4. Slide Scrolling Games

Game jenis *Slide Scrolling* menampilkan tokoh yang melakukan pergerakan ke samping dengan latar belakang yang bergerak seiring dengan gerakan karakter.

2.3.5. Puzzle Games

Puzzle Game bertujuan untuk menyelesaikan sebuah masalah tertentu. Hampir seluruh tantangannya melibatkan masalah logika dan biasanya terbatas oleh waktu.

2.4 Action-RPG (ARPG)

Action-RPG adalah sebuah *game* RPG yang berperan dalam varian *game* kompetisi yang telah pasti menjumpai pertarungan-pertarungan dan melawan penguasa. (Kumparan.com, 2022).

2.5 Artificial Intelligence (AI)

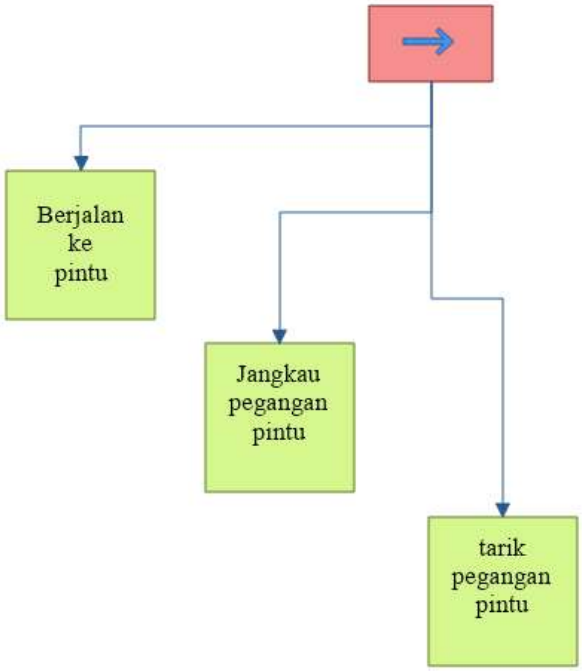
Kecerdasan buatan (Artificial Intelligence/AI) adalah kecerdasan yang di miliki oleh suatu entitas buatan. Kecerdasan diciptakan dan dimasukkan ke dalam suatu mesin (Komputer) agar dapat melakukan pekerjaan seperti yang dapat dilakukan manusia. Beberapa macam bidang yang menggunakan kecerdasan buatan antara lain adalah sistem pakar dan permainan komputer (*game*). (Sugianto & Utama, 2021)

2.6 Behavior Tree (BT)

BT adalah sebuah metode untuk mengendalikan aksi NPC. Metode ini memanfaatkan pohon berakar dengan setiap daun pohon menandakan aksi dan kondisi dan setiap simpul lainnya disebut *decorator* (akan disebut "dekorator"), *sequence* (akan disebut "urutan"), atau *selector* (akan disebut "pengatur") berfungsi untuk memilih aksi apa yang dapat diambil atau dikerjakan oleh NPC (Marcello, 2020).

Behavior Tree memiliki keunggulan dalam penggambaran tindakan pemilihan keputusan seperti pohon perilaku yang terhubung, mempermudah perancang dalam mengatur dan mengubah bagian tertentu. Ini memungkinkan

berbagai perilaku NPC untuk ditangani dengan mudah. Konsep multi perilaku NPC dapat diterapkan dengan mudah menggunakan BT. (RIFQI, 2020).



Gambar 2.1 Contoh BT membuka pintu

2.7 Game Engine

Game engine merupakan sebuah *software* yang dirancang untuk membuat dan mengembangkan sebuah *game*. *Game engine* ini memberikan kemudahan dalam menciptakan sebuah konsep dari *game* yang akan dibuat. Seperti *rendering*, *physics*, arsitektur suara, A.I, dan lain-lain. Berikut dibawah ini fungsi-fungsi dasar *game engine*. *Rendering* 2D maupun 3D, *physics engine*, pengatur audio, *scripting*, pengatur dan penampilan sebuah animasi, *networking*, *streaming* data, pengaturan memori, pengaturan grafis. (Lufthansa, 2022)

2.8 Unreal Engine

Unreal Engine adalah seperangkat alat pengembangan yang berkualitas tinggi dan menerapkan teknologi *real-time*. Bisa digunakan pada berbagai platform seperti *mobile*, desktop, konsol, *augmented reality* dan *virtual reality*. Perangkat ini mengizinkan *creator* untuk menggunakan teknologi unreal untuk membangun visualisasi fotorealistik, *interactive product design*, *movie*, *virtual production*, dan *animated entertainment*. (Warta Ekonomi, 2022).



Gambar 2.2 Unreal Engine

2.9 Blueprint

Sistem *Blueprint Visual Scripting* di Unreal Engine adalah sistem *scripting gameplay* yang lengkap yang didasarkan pada konsep membuat elemen *gameplay* pada Unreal Editor dengan menggunakan antarmuka berbasis *node*. Banyak bahasa *scripting* yang digunakan secara umum untuk menentukan klasifikasi dan kelas berorientasi objek pada *engine*. Dalam Unreal Engine, objek sering didefinisikan

menggunakan *Blueprint* dan disebut "Cetak Biru". Sistem ini sangat adaptif dan handal karena memberikan kemampuan desainer untuk menggunakan konsep dan alat yang biasanya disediakan bagi pemrogram. Selain itu markup khusus Cetak Biru yang diimplementasikan pada Unreal Engine C++ memungkinkan pemrogram untuk merancang sistem dasar yang dapat dikembangkan lebih jauh oleh perancang. (Epic Games, 2022).

2.10 Flowchart

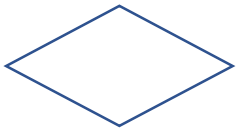
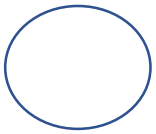
Flowchart adalah representasi visual dari langkah-langkah penyelesaian masalah yang diwujudkan dalam simbol-simbol khusus. *Flowchart* ini akan menampilkan urutan logis dari program. *Flowchart* ini tidak hanya berfungsi sebagai sarana komunikasi, tetapi juga diperlukan sebagai panduan untuk memahami komponen-komponen *flowchart* dengan lebih baik. (Khesya, 2021)

2.11 Simbol-Simbol *Flowchart*

Simbol yang terdapat pada *flowchart* terlihat pada tabel di bawah ini :

Tabel 2.1 Simbol-Simbol *Flowchart*

No	Simbol	Keterangan
1		Simbol *input* / *output* Simbol data / informasi : masukkan / keluaran.

2		*Processing simbol* Simbol proses yang diolah oleh komputer.
3		Simbol *Decision* Simbol kondisi.
4		*Terminator* simbol Simbol *start* atau *end*.
5		*Flow direction* simbol Simbol yang dipergunakan sebagai penghubung.
6		Simbol *predefine* proses Simbol *sub program*.
7		*Connector* simbol Simbol penyambung proses pada halaman yang sama.
8		*Connector* simbol Simbol penyambung proses pada halaman yang berbeda.
9		Simbol *display* Simbol *output* menampilkan layar dan lainnya.

2.12 Waterfall Model

Langkah-langkah penting dari model *waterfall model* mewakili tahap-tahap fondasi. Menurut (Sommerville, 2011) dalam (Putra, 2019) :

1. "Analisis dan definisi persyaratan"

   Prasarana, masalah target program diputuskan berdasarkan diskusi bersama *system users* dan didefinisikan dengan jelas.

2. "Desain sistem dan *software*"

   Alur *system design* menempatkan kebutuhan *hardware* atau *software* dengan merancang sistem secara kelengkapan Arsitektur. Perancangan *software* menyertakan pengidentifikasian dan konseptualisasi dasar sistem *software* dan relasinya.

3. "Implementasi dan pengujian unit"

   Pada tahapan ini desain *software* direalisasikan seperti halnya sekumpulan program atau program unit. Uji coba unit mencakup verifikasi setiap unit dapat memenuhi spesifikasinya.

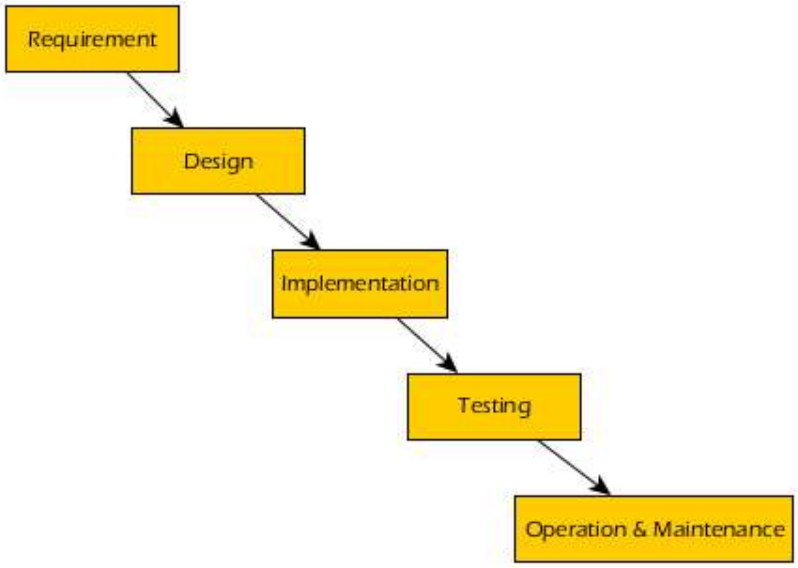
4. "Integrasi dan pengujian sistem"

   Unit atau program program individu diintegrasikan dan diuji sebagai sistem yang lengkap untuk memastikan bahwa persyaratan *software* telah terpenuhi. Setelah pengujian, sistem *software* dikirim ke pelanggan.

5. "Operasi dan pemeliharaan"

   Pada umumnya (walaupun tidak harus), ini merupakan periode durasi terlama. Sistem dipasang dan dipergunakan dengan efektif. Servis

mencakup perbaikan kesalahan yang tidak ada pada tahap awal *life cycle*, peningkatan pengaplikasian sistem unit dan peningkatan *system service*.



Gambar 2.3 *Waterfall* Model