

1. PENDAHULUAN

1.1 Latar Belakang

Video games merupakan hiburan dan tantangan. Tanpa tantangan, *video games* akan mudah diselesaikan dan membosankan. Namun apabila tantangan terlalu sulit dapat membuat pemain frustrasi dan menyerah (Sofyan, 2019). *Non-player character* (NPC) dalam suatu *game* harus memiliki AI yang mempunyai perilaku yang konsisten, realistis dan mempunyai reaksi yang tepat dengan lingkungan sekitarnya sesuai dengan kondisi NPC (Belle, Gittens, & Graham, 2019). Horror, dalam konteks seni dan hiburan, telah menjadi genre yang telah ada selama berabad-abad. Baik dalam bentuk sastra, film, maupun game, horror bertujuan untuk mengeksplorasi dan memanipulasi emosi manusia, terutama ketakutan dan kecemasan. Membuat sebuah game horror dengan suasana yang efektif dan struktur yang konsisten memiliki tantangannya sendiri. Sama seperti *game* lainnya, sangat penting untuk menjaga Pemain atau *Player* tetap fokus bermain dan tidak bosan. Dalam kasus game horror, menjaga *Player* untuk tetap merasa awas dan takut sangatlah penting serta memperhatikan kedua aspek dalam membangun tema dan praktikalitas yang efektif dalam horror, serta bagaimana game horor sukses lainnya telah mengadopsi informasi ini dalam desain *game* mereka (Allén, 2021).

Dalam dunia *game*, membuat sebuah perilaku pada musuh membutuhkan sebuah metode yang mampu mengambil keputusan yang baik. Beberapa penerapan *artificial intelligence* dalam *video games* seperti Halo 2, Bioshock, Tomb Raider, Resident Evil, The Witcher 3, menggunakan beberapa Teknik salah satunya adalah *Behavior Tree* (BT). *Behavior tree* adalah suatu model matematis berbentuk tree yang dibuat untuk merepresentasikan perilaku dari suatu entitas yang deprogram. *Behavior tree* mirip dengan *finite state machine*, hanya saja tidak seperti *finite state machine* yang berisi keadaan-keadaan, *behavior tree* berisi kumpulan instruksi dan perintah yang harus dijalankan oleh developer karena mudah dimengerti sehingga tingkat kesalahan menjadi minimum. Algoritma *finite state machine* mudah untuk dapat diimplementasikan pada sebuah NPC dalam *video games*, tetapi akan sangat sulit untuk bisa mengontrol *behavior* nya tersebut, terutama karena pertumbuhan jumlah *states* yang akan bertambah (Sekhavat, 2017). Walaupun masalah ini dapat diatasi dengan menggunakan algoritma HFSM, tetapi akan sulit untuk mengaturnya karena FSM dan HFSM yang menggunakan *transition* atau transisi berulang bukanlah solusi yang ideal, terutama modulnya yang tidak bersifat *reusable* (Chang & Zhu, 2017).

Penelitian sebelumnya yang dilakukan oleh (Junaidi, Yunus, & Wiguna, 2021) yaitu menggunakan metode *Behavior Tree* pada perilaku NPC. Metode ini dilakukan dengan memberikan beberapa perilaku atau *behavior* kepada NPC yaitu, *patrol* adalah perilaku di mana NPC berpatroli jika jarak antara NPC dan *player* lebih atau sama dengan 10, *pursue* adalah perilaku NPC yang akan mengejar *player* jika jarak kurang atau sama dengan 10 dan HP dari NPC lebih dari 40 jika di bawah 40 NPC tidak akan melakukan *pursue*, *flee* adalah perilaku di mana NPC akan melarikan diri dari *player* jika HP NPC kurang dari 15, *unique skill* perilaku di mana NPC melakukan serangan terhadap *player* berupa lemparan granat ke arah *Player*. *Task* ini bekerja apabila jarak NPC kurang dari 5, *explode* perilaku di mana NPC meledakkan diri dengan jarak radius 10. *Task* ini bekerja apabila HP NPC kurang dari atau sama dengan 0. Penerapan metode untuk perilaku NPC seperti ini terbilang baik karena, penggunaan *behavior* yang bervariasi serta adaptif dalam situasi tertentu seperti penggunaan *input* HP dan jarak.

Ada pula penelitian sebelumnya yang dilakukan oleh (Mahendranta, Jonemaro, & Akbar, 2019) yaitu menerapkan *decision making* pada NPC dengan menggunakan metode *Hierarchical Finite State Machine* (HFSM). Perbedaan dari FSM dan HFSM adalah HFSM memiliki *super-state* atau *clustering* dan juga *sub-state*. Di mana *clustering* bisa menggabungkan beberapa *state* menjadi satu *state* dan mengurangi *transition* yang berlebih. Kekurangan dari metode ini adalah, perubahan *transition* harus dilakukan satu-persatu, dan jika ada perubahan terhadap sebuah *state*, maka *state* yang berhubungan harus diubah kondisinya untuk menyesuaikan logika untuk banyak konteks yang berbeda, solusi yang bisa digunakan adalah dengan membuat setiap *state* modular atau berdiri sendiri sehingga lebih mudah digunakan untuk logika yang berbeda dan metode *Behavior Tree* adalah solusinya.

Pada penelitian ini akan dilakukan penerapan seperti penelitian-penelitian sebelumnya dengan metode *Behavior Tree* menggunakan tools *Unreal Engine*. Dengan menggunakan metode ini, penelitian dilakukan dengan tujuan meningkatkan tantangan dalam *game* dan memberikan kemampuan kepada NPC musuh *behavior* atau perilaku yang variatif terhadap situasi atau kondisi yang lebih kompleks. Jadi diharapkan *Player* tidak akan merasa bosan dengan kemampuan AI NPC musuh yang monoton serta bisa ditebak.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan di atas, maka rumusan masalahnya adalah:

1. Apakah metode *Behavior Tree* yang digunakan dapat membuat perilaku NPC yang variatif dalam situasi tertentu?

1.3 Tujuan Skripsi

Berdasarkan perumusan masalah di atas maka tujuan dari penelitian ini adalah:

1. Menerapkan metode *Behavior Tree* untuk menghasilkan suatu kecerdasan buatan yang mampu mengeluarkan aksi yang variatif pada perilaku musuh dalam *game horror*
2. Mengimplementasikan *game horror* kedalam bentuk *game desktop*

1.4 Ruang Lingkup

Ruang Lingkup dibatasi pada:

1. Pada penelitian ini hanya mengembangkan algoritma pemrograman sesuai dengan metode *Behavior Tree*.
2. *Tools* yang digunakan adalah *Unreal Engine 5*, dengan *art style* Full 3D.
3. Nama dari *Game* yang akan dibuat adalah "*Dark Dream*". *Player* akan terjebak didalam dunia mimpi yang tidak pernah berakhir dan harus mencari sumber dari mimpi buruk ini untuk keluar. Dengan *concept art* berupa perkotaan, suasana malam hari dengan penggunaan cahaya minimal. Beberapa mekanisme yang akan dibuat:
 - *Exploration*: *Player* harus mengeksplorasi kota untuk mencari *clue* dan menyelesaikan *puzzle*. *Player* memiliki banyak jenis pergerakan mulai dari berjalan, berjalan jongkok, lari santai, lari cepat, melompat, dan memanjat bangunan dengan ketinggian tertentu.
 - *Puzzle Solving*: *Player* akan menemukan beberapa jenis *puzzle* atau teka-teki yang harus diselesaikan dengan cara eksplorasi untuk mencari *clue* atau petunjuk.
 - *Ranged Combat*: *Player* bisa menggunakan senjata jarak jauh seperti senapan yang bisa ditemukan ketika sedang bereksplorasi untuk melindungi diri. Senapan juga memiliki peluru dengan jumlah terbatas.
4. User Interface seperti *inventory* dan *status (health, stamina)*.



Gambar 1.1 Tampilan UI

- **Inventory:** untuk mengatur peralatan dan senjata yang *Player* temukan.
 - **Status (health, stamina):** sebagai indikator jumlah *health* (darah) dan *stamina*. *Health* bisa berkurang ketika *Player* diserang oleh NPC atau terjatuh dari ketinggian tertentu dan untuk menambah *health*, *Player* bisa mencari dan menggunakan *medkit* (obat). Jumlah *stamina* akan berkurang cepat ketika *Player* sedang berlari cepat dan ketika stamina habis *Player* tidak akan bisa berlari cepat dan otomatis masuk ke mode berlari santai serta jumlah stamina akan bertambah Kembali.
5. Daftar senjata yang bisa digunakan oleh *Player*. Setiap senjata memiliki jenis *ammo* atau peluru tersendiri.

- *Assault Rifle*



Gambar 1.2 Assault Rifle

Sumber: Marketplace. (2004-2024). *Assault Rifle*.

<https://www.unrealengine.com/marketplace/en-US/product/military-weapons-dark>

- *Pistol*



Gambar 1.3 Pistol

Sumber: Marketplace. (2004-2024). *Pistol*.

<https://www.unrealengine.com/marketplace/en-US/product/military-weapons-dark>

- *Sniper*



Gambar 1.4 Sniper

Sumber: Marketplace. (2004-2024). *Sniper*.

<https://www.unrealengine.com/marketplace/en-US/product/military-weapons-dark>

6. Jenis NPC yang akan ada di dalam *Game* berjumlah 5, yaitu:

- *Ghoul*



Gambar 1.5 Ghoul

Sumber: Marketplace. (2004-2024). *Ghoul*.

<https://www.unrealengine.com/marketplace/en-US/product/undead-pack-01>

- *Lich*



Gambar 1.6 Lich

Sumber: Marketplace. (2004-2024). *Lich*.

<https://www.unrealengine.com/marketplace/en-US/product/undead-pack-01>

- *Zombie*



Gambar 1.7 Zombie

Sumber: Marketplace. (2004-2024). *Zombie*.

<https://www.unrealengine.com/marketplace/en-US/product/undead-pack-01>

- *Goblin*



Gambar 1.8 Goblin

Sumber: Marketplace. (2004-2024). *Goblin*.

<https://www.unrealengine.com/marketplace/en-US/product/undead-pack-01>

- *Skeleton*



Gambar 1.9 Skeleton

Sumber: Marketplace. (2004-2024). *Skeleton*.

<https://www.unrealengine.com/marketplace/en-US/product/undead-pack-01>

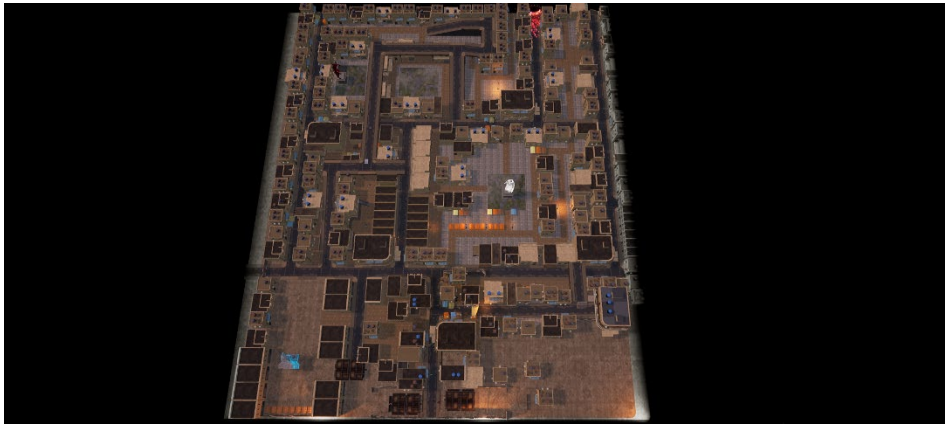
7. Game “Dark Dream” memiliki 3 *stage* atau dunia yang memiliki desain dan scenario masing-masing. Di setiap *stage* Player akan diberikan bantuan seperti *ammo* atau peluru dan *medkit* tetapi dengan jumlah yang terbatas, jumlah *ammo* dan *medkit* yang Player miliki tidak akan kembali ke jumlah maksimum ketika sebuah *stage* berhasil diselesaikan, misalnya jika jumlah maksimum peluru dari senjata *Assault Rifle* adalah 360 kemudian Player menggunakan 100 peluru untuk menyelesaikan Stage 2, berarti Player memiliki sisa 260 peluru ketika lanjut ke Stage 3. Ketika Player kalah atau *health* berada di angka 0, Player akan diberikan pilihan untuk mengulangi permainan dari Stage 1 atau mengulang Stage sekarang.
- Stage 1 : Perkenalan, Player akan diberi informasi untuk mengendalikan pergerakan karakter, cara mengambil barang, serta menggunakannya. Setelah skenario Perkenalan selesai, lanjut ke scenario *Survival* di mana Player harus bertahan hidup serta mengalahkan 100 NPC musuh yang muncul di Stage. Jenis NPC musuh yang ada di Stage ini adalah *Ghoul* dan *Goblin*. Stage akan didesain memiliki ukuran yang luas dengan banyak jenis bangunan dan jalanan yang lebar.



Gambar 1.11 Tampilan Stage 1

- Stage 2 : *Stealth*, Player harus bergerak tidak terdeteksi oleh NPC musuh, jika terdeteksi Player harus bersembunyi untuk menghindari NPC karena NPC akan diatur untuk memiliki jumlah *Health* dan *Damage* yang besar. *Puzzle* atau teka-teki, untuk menyelesaikan Stage ini Player harus menyelesaikan 2 jenis teka-teki dan setiap teka-teki memiliki petunjuk masing-masing bagaimana cara menyelesaikannya, setiap petunjuk ada disekitar Stage dan Player harus mencarinya. Jenis NPC musuh yang ada di Stage ini adalah *Zombie* dan *Skeleton*. *Skeleton* dibagi menjadi 2 yaitu, *Skeleton Sword* dan *Skeleton Bow*. Stage akan

didesain memiliki banyak bangunan dan dekorasi untuk memberikan banyak jalur yang bisa dilalui oleh *Player* maupun titik patroli dari setiap NPC.



Gambar 1.12 Tampilan Stage 2

- Stage 3 : *Boss*, di Stage terakhir ini tugas *Player* adalah mengalahkan 1 NPC dengan menggunakan semua senjata yang *Player* dapatkan dari Stage 1 dan 2, NPC yang ada di Stage ini adalah *Lich*. Stage akan didesain dengan jumlah bangunan yang sedikit serta area yang cukup kecil.



Gambar 1.13 Tampilan Stage 3

8. Di setiap *Stage*, jenis AI yang digunakan beragam:
 - Stage 1 : NPC memiliki kemampuan penglihatan dan pendengaran, penglihatan adalah kemampuan NPC untuk mengejar *Player* jika sudah masuk ke dalam radius penglihatan dan menyerang jika sudah masuk ke radius serangan. Pendengaran adalah kemampuan NPC untuk mendengar suara *Player* yang sedang menembakkan senjata.

- Stage 2 : Sama seperti Stage 1 dengan tambahan NPC memiliki kemampuan untuk berpatroli di titik-titik yang sudah ditentukan, jika NPC melihat atau mendengar sesuatu maka NPC akan menuju ke daerah tersebut untuk menginvestigasi. Jika dalam durasi tertentu tidak ada kejadian lanjutan, NPC akan kembali ke titik awal patroli dan melanjutkan aktivitasnya. Kemampuan selanjutnya adalah untuk NPC *Skeleton Sword* memiliki kesempatan untuk memblokir serangan yang diterima dari *Player* misalnya, *Player* menembakkan 10 peluru dengan senjata *Assault Rifle* tetapi ada kemungkinan beberapa peluru tersebut tidak memberikan *damage* atau mengurangi *health* dari *Skeleton Sword*. Kemudian *Skeleton Sword* memiliki kemampuan serangan lompat atau *JumpAttack* dari radius yang cukup jauh serta kemampuan *SpinAttack* atau serangan berputar yang memberikan serangan berulang ke *Player*.
 - Stage 3 : Sama seperti Stage 2 tanpa kemampuan berpatroli, NPC memiliki kemampuan untuk menghindari serangan yang dilakukan oleh *Player*. Ketika *health* dari NPC sudah dibawah 20%, NPC akan bersembunyi selama beberapa saat untuk *Restore* atau menyembuhkan diri sendiri sebanyak 50% dari jumlah maksimum *health*, NPC hanya mampu melakukan *Restore* sebanyak 2 kali. NPC juga memiliki kemampuan untuk menyerang dari jarak jauh serta kemampuan untuk teleportasi atau kemampuan berpindah tempat dengan cepat. Kemampuan selanjutnya adalah melakukan *GroundSmash* merupakan kemampuan yang akan digunakan ketika *Player* terlalu mendekati NPC, *GroundSmash* juga memiliki jumlah *damage* yang cukup besar
9. Asset-asset yang digunakan di download dari *Unreal Marketplace*, *Sketchfab*, *CGTrade*.
 10. *Game* ini hanya dimainkan di PC berbasis *desktop* dengan penggunaan *keyboard* dan *mouse*, dan sistem operasi *Windows*.
 11. Jumlah pemain berupa *single player* (satu pemain).

1.5 Metodologi Penelitian

Langkah-langkah dalam pengerjaan skripsi:

1. Studi Literatur
 - 1.1. Mempelajari *Unreal Engine 5*
 - 1.2. Mempelajari cara kerja metode *Behavior Tree*
2. Perencanaan dan Pembuatan Game
 - 2.1. Perancangan Desain Game

- 2.2. Perancangan Gameplay
- 2.3. Perancangan *Behavior Tree*
- 2.4. Pembuatan *game* dengan menggunakan *Unreal Engine 5*
- 3. Penerapan Game pada *Unreal Engine 5*
 - 3.1. Membangun *Stage Design* pada *Game*
 - 3.2. Menerapkan *Behavior Tree* ke dalam NPC yang ada pada *game* sebagai AI NPC
- 4. *Testing Game*
 - 4.1. Melakukan uji coba *game*
- 5. Pengujian dan Analisis Program
 - 5.1. Menganalisa kinerja NPC berdasarkan *Behavior Tree* yang sudah diterapkan
 - 5.2. Menganalisa hasil output dari Program
- 6. Pengambilan Kesimpulan
 - 6.1. Membuat kesimpulan tentang hasil penelitian dari Analisa yang telah dilakukan
- 7. Pembuatan Laporan
 - 7.1. Membuat laporan berdasarkan hasil yang telah diperoleh

1.6 Sistematika Penulisan

Dalam penyusunan laporan skripsi ini, sistematika penulisan yang digunakan adalah sebagai berikut:

- 1. BAB I : PENDAHULUAN

Bab ini merupakan pendahuluan terkait penelitian yang berisi latar belakang, perumusan masalah, tujuan dari penelitian, ruang lingkup penelitian, metodologi penelitian, dan sistematika penulisan.
- 2. BAB II : LANDASAN TEORI

Bab ini merupakan landasan teori yang digunakan sebagai dasar penelitian.

Bab ini berisi metode, teori – teori dan penelitian sebelumnya untuk mendukung pembuatan skripsi.
- 3. BAB III : ANALISA DAN DESAIN SISTEM

Bab ini merupakan analisa dan desain sistem yang meliputi desain interface dan desain database.
- 4. BAB IV : IMPLEMENTASI SISTEM

Bab ini merupakan implementasi sistem dan pembuatan aplikasi dari sistem yang telah dibuat pada bab sebelumnya.
- 5. BAB V : PENGUJIAN SISTEM

Bab ini merupakan pengujian sistem yang membahas aplikasi yang telah dibuat pada bab sebelumnya.

6. BAB VI : KESIMPULAN DAN SARAN

Bab ini merupakan kesimpulan dari aplikasi yang telah dibuat dan saran bagi pengembangan aplikasi di kemudian hari.