

# Implementasi Behaviour Tree Untuk Mengatur Perilaku NPC Musuh Pada Game 2D Platformer `Cyberun`

## *Implementation of Behaviour Tree for Controlling Enemies NPC Behavior on 2D Platformer Game `Cyberun`*

Aribowo Hasti Senoaji<sup>1</sup>, Hanny Haryanto<sup>2</sup>, Sendi Novianto<sup>3</sup>

<sup>1,2</sup>Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro

E-mail: <sup>1</sup>111201811407@mhs.dinus.ac.id, <sup>2</sup>hanny.haryanto@dsn.dinus.ac.id,

<sup>3</sup>sendi.novianto@dsn.dinus.ac.id

### **Abstrak**

*Game Platformer* memiliki tujuan inti untuk menggerakkan karakter sehingga dapat berpindah tempat dari satu titik ke titik lain dalam sebuah arena baik itu dengan gerakan berlari ataupun melompat. Salah satu faktor yang membuat permainan dalam game lebih menarik adalah adanya karakter NPC yang bertugas sebagai musuh. Tentu perancangan NPC musuh yang baik adalah dengan adanya implementasi agen cerdas sehingga mampu membuat karakter NPC musuh berperilaku lebih dinamis. Salah satu metode yang dapat digunakan dalam perancangan agen cerdas ini adalah *Behavior Tree*. *Behavior Tree* merupakan cara penerapan dari suatu pengelompokan yang merupakan proses untuk menentukan fungsi dari pemetaan setiap himpunan *behavior* ke dalam suatu kelompok yang berbentuk seperti pohon (*tree*). Dari perancangan *Behavior Tree* inilah yang kemudian dilakukan pengujian kepada NPC musuh guna mengetahui *output* yang dihasilkan dari berbagai *nodestate* yang terdapat pada NPC musuh seperti Idle, Patrol, Attack, dan Death. *Nodestate* ini diimplementasikan pada masing-masing karakter NPC musuh namun dengan perilaku yang berbeda. Dari pemodelan dengan metode *Behavior Tree* ini menghasilkan *output* perilaku yang dinamis dari NPC musuh yang berbeda-beda sesuai dengan apa yang sudah dipetakan sebelumnya.

Kata kunci : *Game 2D Platformer, NPC, Agen Cerdas, Behaviour Tree*

### **Abstract**

*Platformer games* have the core goal of moving characters so that they can move from one point to another in an arena either by running or jumping. One of the factors that make the game play more interesting is the presence of NPC characters who serve as enemies. Ofcourse a good enemy NPC design is the implementation of intelligent agents so as to make enemy NPC characters behave more dynamically. One method that can be used in designing intelligent agents is the *Behaviour Tree*. The *Behaviour Tree* is a way of implementing a grouping which is a process for determining the function of mapping each set of behaviors into a group that is shaped like a tree. From this *Behaviour Tree* design, tests were then carried out on enemy NPCs in order to find out the output generated from various *nodestates* found on enemy NPCs such as Idle, Patrol, Attack, and Death. This *nodestate* is implemented for each enemy NPC character but with a different behaviour. From modeling with the *Behaviour Tree* method, it produces dynamic behavioral outputs from enemy NPCs that vary according to what has been mapped before.

Keywords: *Game 2D Platformer, NPC, Artificial Intelligent, Behaviour Tree*

## 1. PENDAHULUAN

*Platform Game (game platformer)* merupakan salah satu sub-jenis dari game action yang memiliki tujuan inti untuk menggerakkan karakter sehingga dapat berpindah tempat dari satu titik ke titik yang lain dalam sebuah arena baik itu dengan Gerakan berlari ataupun melompat. Ciri yang dapat dilihat dari *game platformer* adalah desain arena atau dunia berupa bidang yang tidak rata dan juga platform yang memiliki ketinggian yang berbeda [1]. Manuver umum yang dapat dilakukan karakter dalam *game platformer* adalah berjalan, berlari, melompat, memanjat, meluncur, dan bahkan memantul dari satu objek ke objek lain. Sehingga *game* ini juga menuntut pemain untuk bergerak dinamis dalam hal ketepatan dan kecepatan ketika memegang kendali. Jenis platform didalam *game* juga dibedakan kedalam beberapa tipe seperti *Standard Platform*, *Jump Through Platform*, *High Resistance Platform*, dan sebagainya. Salah satu contoh *game platformer* yang umum diketahui orang adalah jenis *mario-like platformer*. Sebagai salah satu pionir dalam penciptaan *game platformer*, jenis *mario-like* ini pun diadaptasi atau dikembangkan menjadi berbagai jenis-jenis *game platformer* lainnya [2].

Berbicara tentang *game* tentu tidak lepas dari adanya sebuah agen pembelajaran berupa *Artificial Intelligence (AI)*. Secara umum AI dalam *game* diimplementasikan ke dalam bentuk *Non-Playable Character (NPC)* atau agen otonom yang disimulasikan memiliki kecerdasan sehingga mampu bertindak secara alami layaknya manusia [3]. NPC merupakan salah satu objek didalam *game* yang dapat berupa monster, hewan, manusia, atau bahkan yang lainnya tetapi objek ini tidak bisa dikendalikan atau dimainkan oleh *player*. NPC dapat dikategorikan baik jika mampu berjalan secara mandiri dan bersifat dinamis atau tidak monoton. Kehadiran NPC yang dinamis ini secara tidak langsung akan membuat *player* merasa tertantang dikarenakan respon tindakan yang diberikan oleh NPC tidak dapat diduga [4]. NPC ini bertugas sebagai musuh sehingga *player* merasa mendapat tantangan untuk dapat menghindari serangan NPC. Salah satu metode yang dapat digunakan untuk merancang atau mengembangkan AI yaitu dengan metode *Behavior Tree*. Dikutip dari [5] menyatakan bahwa *Behavior Tree* merupakan suatu model matematis yang berbentuk seperti pohon (*tree*) yang dibuat guna menampilkan perilaku dari suatu entitas komputer yang diprogram sedemikian rupa. Pengembangan dengan metode ini sangat fleksibel dan dapat diubah, seperti dengan membentuk suatu perilaku dasar, yang kemudian akan menjadi cabang baru sebagai metode alternatif untuk mencapai tujuan.

Terdapat beberapa penelitian sebelumnya yang membahas tentang perancangan kecerdasan buatan untuk NPC dalam *game* dengan menggunakan metode *Behavior Tree*. *Behavior Tree* adalah alternatif modular dari metode *Finite State Machine (FSM)*. *Behavior Tree* menutupi kekurangan FSM yang memiliki kesulitan ketika ada transisi, konsep modularity inilah yang akan membentuk setiap *state* menjadi seperti pohon [6]. Dari jurnal yang dituliskan oleh [7] menyatakan bahwa pemodelan agen cerdas berbasis algoritma *Behavior Tree* terbukti mampu menyajikan permainan yang dinamis serta mampu menambah keseruan dalam bermain bagi pemain yang dituju. Metode ini digunakan untuk menerapkan aksi dari masing-masing karakter NPC musuh dimana karakter NPC harus mampu bergerak berdasarkan berbagai keputusan. [8] *Behavior Tree* tak sebatas digunakan untuk menentukan pergerakan karakter NPC musuh, namun juga dapat digunakan lebih kompleks lagi seperti menentukan sebuah kondisi kapan karakter NPC musuh melakukan suatu tindakan seperti bergerak, menyerang, dsb. Implementasi *Behavior Tree* oleh [5], menuliskan pada penelitiannya tentang pemanfaatan *Behavior Tree* dan *Fuzzy* dalam *game* "War of Zombies" bahwa dengan kedua metode tersebut menghasilkan AI yang lebih cerdas dan efektif dalam mengambil keputusan. Penelitian yang dilakukan oleh [9] tentang 'Penerapan *Behavior Tree* Untuk Pengambilan Keputusan NPC Pada Gim Balap' dituliskan bahwa *Behavior Tree* yang dirancang dengan penambahan implementasi sistem *Waypoint* dan *Raycast* mampu berjalan dengan baik dan sesuai dengan diagram perancangan yang sudah dibuat sebelumnya. Namun dalam pengujian performa NPC mendapat hasil yang kurang baik dikarenakan jumlah NPC musuh yang terlalu banyak dan mengambil tindakan yang cenderung sama, sehingga mengalami penubrukan performa.

Dari penjabaran diatas, penelitian ini dibuat untuk merancang dan memahami agen cerdas pada NPC (*Non-Playable Character*) dengan perilaku dinamis menggunakan metode *Behavior Tree*. Dengan metode ini diharapkan NPC mampu memberikan respon tindakan yang dinamis mengikuti pola rancangan *game*. Sehingga mampu memberikan tantangan kepada *player* dan mengantarkan *flow* permainan yang tidak monoton.

## 2. METODE PENELITIAN

Pada penelitian ini mengusulkan metode *Behavior Tree* dan memiliki beberapa tahapan. Berikut merupakan tahapan – tahapan dalam perancangan NPC (*Non-Playable Character*) dalam game:

### 2.1 Prosedur Pengumpulan Data

Pengumpulan data bertujuan sebagai acuan yang menunjang atau mendukung penelitian dan pembuatan *game*. Pengumpulan data dilakukan dengan memberikan kusioner dalam bentuk google form kepada para responden. Isi kuesioner meliputi ketertarikan responden terhadap *game* berjenis *platformer*, pengetahuan tentang NPC, dan juga pentingnya kehadiran NPC dalam bentuk musuh khususnya pada *game platformer*. Mayoritas responden mengetahui dan memainkan *game* berjenis *platformer* baik itu *platform* PC, konsol, maupun *mobile*.

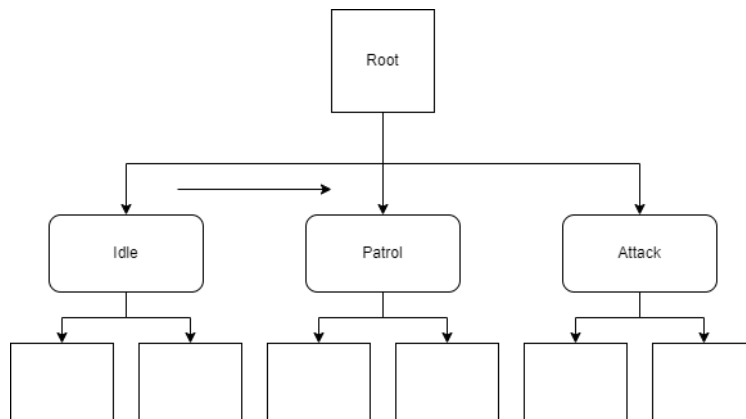
### 2.2 Analisa Data

Metode analisis data yang digunakan dalam perancangan NPC pada *game platformer* adalah metode kualitatif. Metode kualitatif merupakan metode untuk menganalisa suatu permasalahan dengan menggunakan data bertipe deskriptif atau kalimat untuk dilakukan analisis sehingga dapat mendapatkan kesimpulan [10]. Metode ini digunakan karena data yang diperoleh bersifat induktif, yang berarti menganalisis melalui suatu data yang kemudian dapat dikembangkan menjadi pola tertentu untuk menjadi suatu hipotesis. Parameter utama yang digunakan dalam perancangan game ini adalah pergerakan player yang meliputi jarak antara karakter pemain dengan NPC musuh. Parameter tersebut digunakan untuk merancang implementasi AI yang akan disematkan kepada NPC musuh. Sehingga diharapkan mampu menghasilkan perilaku NPC yang dinamis, agar secara tidak langsung membuat permainan lebih menantang dan tidak monoton.

### 2.3 Metode Behavior Tree

Metode ini digunakan untuk memetakan berbagai composite node yang ada pada NPC musuh. Dengan terbentuknya berbagai node, akan dengan mudah menganalisis dan merancang perilaku yang sesuai untuk kebutuhan NPC musuh. Dibawah ini adalah node yang akan disertakan ke dalam game. Melalui node ini akan diketahui perilaku yang akan diterapkan oleh AI NPC musuh dalam game.

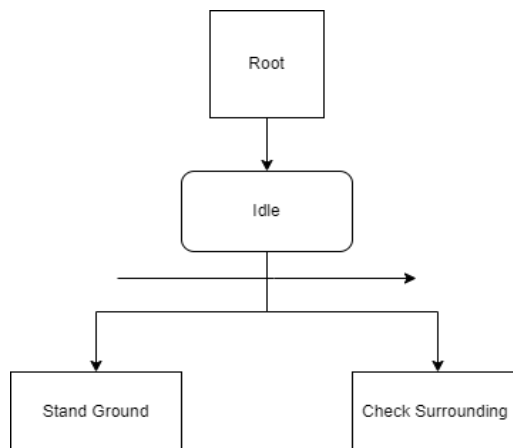
a. Behavior



Gambar 1 Tree Node Behavior

Pada *tree* diatas disebutkan ada 3 *condition node* utama yaitu Idle, Patrol, dan Attack. Masing-masing dari node tersebut juga memiliki beberapa *leaf node* baik berupa *action* maupun *condition*. Terdapat pula *control node* berupa *sequence* yang berarti *node* tersebut dieksekusi secara berurutan.

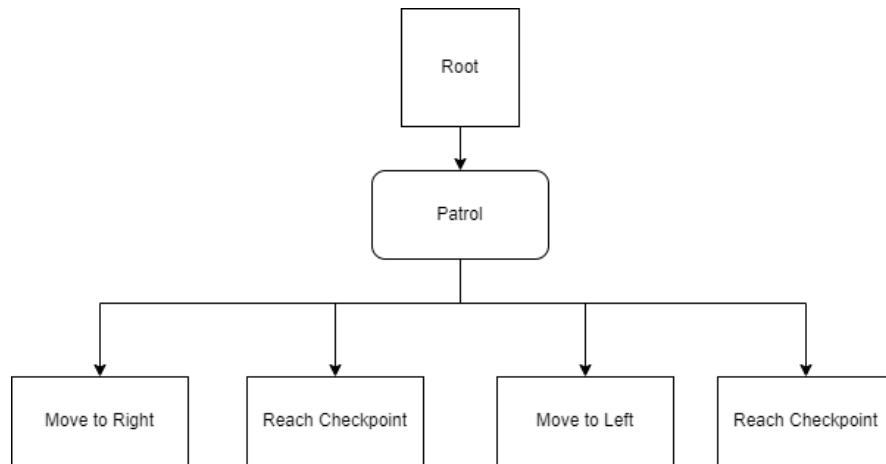
b. Idle



Gambar 2 Tree Node Idle

*Tree* untuk *parent node* Idle yang berjenis *condition node*. Memiliki dua buah *leaf node* yang keduanya berjenis *action node*. *Node* pertama bernama ‘Stand Ground’ dimana dalam *node* ini karakter NPC musuh akan berdiri secara statis. *Node* selanjutnya ‘Check Surrounding’ dieksekusi secara berurutan setelah *node* sebelumnya dieksekusi. Pada *node* ‘Check Surrounding’ ini karakter NPC musuh juga secara *realtime* memantau area sekitar karakter dengan dibekali fitur *Field of View (FOV)*.

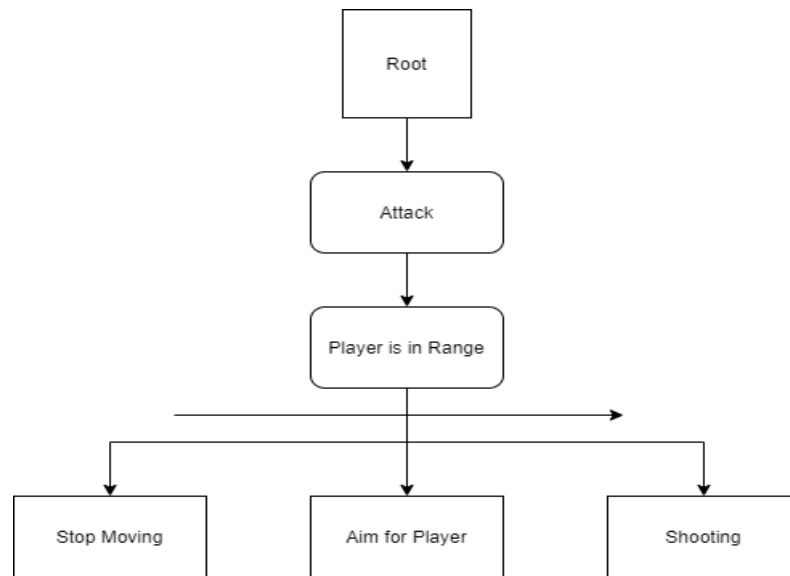
c. Patrol



Gambar 3 Tree Node Patrol

Parent node selanjutnya adalah Patrol, terdiri dari empat leaf node yaitu 'Move to Right', 'Move to Left', dan juga 2 node 'Reach Checkpoint'. Node ini akan dieksekusi oleh karakter dengan pergerakan mondar-mandir pada titik tertentu seolah karakter NPC musuh ini melakukan kegiatan patroli.

d. Attack



Gambar 4 Tree Node Attack

Pada tree node Attack ini terdapat sebuah condition node bernama 'Player is in Range' yang berarti kondisi jika player memasuki area jangkauan dari karakter NPC musuh yang akan dibekali dengan Field of View yang berfungsi sebagai detector. Setelah kondisi tersebut terpenuhi maka akan mengeksekusi node selanjutnya yang berupa 'Stop Moving' dimana karakter NPC musuh akan berhenti bergerak setelah menyadari kehadiran karakter player. Kemudian 'Aim for Player' dan juga 'Shooting', dimana kedua node ini merupakan

tindakan yang diambil oleh karakter NPC musuh yang berupa penyerangan terhadap karakter player. Ketiga *leaf node* ini berjalan secara berurutan.

### 3. HASIL DAN PEMBAHASAN

Berdasarkan rancangan Behavior Tree yang telah dibuat, dilakukanlah pengujian kepada NPC musuh guna mengetahui output yang dihasilkan. Penentuan output ini ditentukan oleh parameter utama yang dijadikan sebagai trigger yang berupa aksi dan juga jarak karakter player terhadap NPC musuh. Nodestate yang terdapat pada NPC musuh meliputi Idle, Patrol, Attack, dan Death. Hasil perancangan agen cerdas dengan metode Behavior Tree pada karakter NPC musuh ditampilkan pada tabel dibawah ini :

Tabel 1 Hasil Rancangan Behavior Tree Musuh Jenis Drone Enemy

Perilaku	Jarak Player > 5px	Jarak Player < 5px	Tertembak Player
Patrol	Patroli	Berhenti bergerak	Mati
Attack	Patroli	Menembak	Mati
Death	Mati	Mati	Mati

Tabel 2 Hasil Rancangan Behavior Tree Musuh Jenis Turret Enemy

Perilaku	Jarak Player > 5px	Jarak Player < 5px	Tertembak Player
Idle	Diam di tempat	Diam di tempat	Mati
Attack	Diam di tempat	Menembak	Mati
Death	Mati	Mati	Mati

Tabel 3 Hasil Rancangan Behavior Tree Musuh Jenis Hover Enemy

Perilaku	Jarak Player > 8px	Jarak Player < 8px dan > 5px	Jarak Player < 5px	Tertembak Player
Idle	Kembali dan diam di tempat	Bergerak ke arah <i>player</i>	Berhenti bergerak	Mati
Attack	Kembali dan diam di tempat	Bergerak ke arah <i>player</i>	Menembak	Mati
Chasing	Kembali dan diam di tempat	Bergerak ke arah <i>player</i>	Berhenti bergerak	Mati
Death	Mati	Mati	Mati	Mati

Guna mengetahui hasil dari pengalaman pemain ketika memainkan game “Cyberun”, dilakukanlah survei dengan menggunakan metode *Game Experience Questionnaire* (GEQ). Kuesioner ini berisi 10 pertanyaan dan diberikan kepada 10 responden yang telah memainkan

game ini. Penilaian evaluasi terdiri dari angka 1 sampai dengan 5, dengan penjabaran penilaian sebagai berikut :

- Angka 1 : Sangat Tidak Menarik
- Angka 2 : Tidak Menarik
- Angka 3 : Cukup
- Angka 4 : Menarik
- Angka 5 : Sangat Menarik

Tabel 4.4 Hasil Evaluasi Terhadap Pengguna

No.	Pertanyaan	Nilai					Rata-Rata
		1	2	3	4	5	
1	Apakah <i>gameplay</i> menarik	0	1	3	4	2	3,7
2	Apakah mudah dimainkan	0	0	3	4	3	4,0
3	Apakah merasa bosan	3	1	4	2	0	2,5
4	Apakah merasa senang	0	2	4	3	1	3,3
5	Apakah berkesan	0	1	3	4	2	3,7
6	Apakah merasa tertantang	0	1	4	3	2	3,6
7	Apakah merasa menikmati	0	0	4	4	2	3,8
8	Apakah lupa waktu	2	2	4	2	0	2,6
9	Apakah merasa lelah	3	2	4	1	0	2,3
10	Apakah menyukai <i>game</i> ini	0	1	4	4	1	3,1

Berdasarkan hasil dari tabel diatas dapat diketahui bahwa nilai tertinggi ada pada kemudahan game ketika dimainkan yaitu dengan nilai 4,0 (Baik). Sedangkan nilai terendah ada pada rasa lelah ketika memainkan game, dengan nilai 2,3.

#### 4. KESIMPULAN DAN SARAN

Berdasarkan permasalahan dan pembahasan diatas, maka dapat diperoleh kesimpulan, sebagai berikut:

1. Pengembangan agen cerdas dengan menggunakan metode *Behavior Tree* dapat menghasilkan pemodelan perilaku NPC musuh yang berbeda-beda di tiap jenis NPC musuhnya.
2. Sistem memberikan hasil tentang perilaku NPC musuh secara otomatis dan dinamis berdasarkan tindakan yang dilakukan oleh *player*.
3. *Game* yang dirancang mampu dimainkan dengan mudah oleh pemain dan memiliki *gameplay* yang menarik.

Dalam melakukan penelitian ini, masih banyak terdapat kekurangan yang dapat diperbaiki dan bahkan dikembangkan lagi di penelitian kedepan. Saran yang dapat dilakukan untuk penelitian selanjutnya adalah :

1. Pemodelan perilaku pada NPC musuh dapat dibuat lebih kompleks sehingga pemain akan lebih merasa tertantang.
2. Penggunaan agen cerdas pada penelitian ini bisa dikombinasikan dengan metode algoritma yang lain seperti Fuzzy, A\*, dan lain sebagainya. Agar menghasilkan AI yang lebih dinamis dan juga lebih menarik. Sehingga mendapatkan pengalaman bermain *game* yang lebih *immersive*.

#### DAFTAR PUSTAKA

- [1] I. Pelle, "Apa Itu Genre Game Platformer? - Esportsnesia," Apa Itu Genre Game Platformer? Accessed: Oct. 18, 2023. [Online]. Available: <https://esportsnesia.com/penting/apa-itu-genre-game-platformer/>
- [2] N. Abdillah, "Ada 4 Jenis dan 11 Genre Game, yang Mana Favorit Kamu?," Pricebook. Accessed: Oct. 18, 2023. [Online]. Available: <https://www.pricebook.co.id/article/review/3593/ada-4-jenis-dan-11-genre-game-yang-mana-favorit-kamu>
- [3] M. L. Setiawan, A. Arbansyah, and S. H. Suryawan, "Penerapan Algoritma A\* Dan Behaviour Trees Untuk Perilaku Non-Player Character(NPC) Pada Game 'The Last Hope' Berbasis Android Menggunakan Unity 2D," *J. CoSciTech Comput. Sci. Inf. Technol.*, vol. 4, no. 2, Art. no. 2, Sep. 2023, doi: 10.37859/coscitech.v4i2.5418.
- [4] U. Nurhasan, H. Pradibta, and F. Z. Alhaddad, "Analisis Perilaku Non-Playable Character Pada Game Menggunakan Fuzzy Sugeno," vol. 19, no. 3, pp. 308–320, 2020.
- [5] M. P. Christie and Andreas, "Pemanfaatan Behavior Tree dan Fuzzy Waypoint Tactic pada Game Strategi " War of Zombies ",," vol. 2, pp. 40–46, 2020.
- [6] E. M. Trianto, H. Junaedi, and H. Sutiksno, "AGEN CERDAS BERBASIS CONTROLLER FUZZY PADA PERMAINAN STRATEGI PERTEMPURAN DENGAN BEHAVIOR TREE," pp. 1–8, 2017.
- [7] B. A. Qowy, F. Hanafi, M. A. Riandi, and A. Nuraminah, "Penerapan Pemilihan Model Dinamis Algoritma Behaviour Tree Decision dalam Third Person Game pada Musuh Non-Playable Charater," *J. Tek. Inform. Dan Elektro*, vol. 3, no. 1, pp. 32–37, 2021.
- [8] B. Sugianto and G. P. Utama, "IMPLEMENTASI ALGORITMA PATHFINDING DAN DECISION TREE DALAM PEMBUATAN VIDEO GAME BERGENRE THIRD PERSON SHOOTER," vol. 4, no. 2, pp. 7–14, 2021.
- [9] F. Soefana, E. Muhammad, A. Jonemaro, and M. A. Akbar, "Penerapan Behavior Tree Untuk Pengambilan Keputusan Non-Player Character Pada Gim Balap," *J. Pengemb. Teknol. Inf. Dan Ilmu Komput.*, vol. 5, no. 5, pp. 1886–1890, 2021.
- [10] N. R. Akbar, E. M. A. Jonemaro, and T. Afirianto, "Evaluasi User Experience Pada Game Hearthstone Dengan Menggunakan Metode Game Experience Questionnaire," *J. Pengemb. Teknol. Inf. Dan Ilmu Komput.*, vol. 2, no. 12, Art. no. 12, Aug. 2018.