

Implementasi Algoritma A* Untuk Mekanisme Gerak Pada Permainan Edukasi *Balloon Shooter* Untuk Siswa Sekolah Dasar

1st Fawwaz Aboeruslan Mulyadi
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

rslnofficial@student.telkomuniversity.a
c.id

2nd Purba Daru Kusuma
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

purbodaru@telkomuniversity.ac.id

3rd Ratna Astuti Nugrahaeni
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

ratnaan@telkomuniversity.ac.id

Abstrak— Game adalah aktivitas yang memiliki konsep menyenangkan dan bermain. Game memiliki alur, cerita, dan karakter. Setiap karakter yang ada dalam sebuah game pasti memiliki unsur Artificial Intelligence. Contoh penggunaan Artificial Intelligence dalam sebuah game ialah pada NPC. Tujuannya adalah agar NPC memiliki pergerakan yang tidak diatur oleh pemain. Dalam sebuah game, ada yang bersifat edukatif dan non-edukatif. Game edukatif memiliki sifat edukatif namun tidak menyenangkan bagi pemain, karena biasanya game edukatif berbentuk seperti quiz. Agar bisa memberikan game edukatif yang menyenangkan dan memiliki nilai edukatif, maka perlu membuat game menyenangkan tapi memiliki aturan edukatif. Salah satu cara agar memberikan sifat menyenangkan dalam sebuah game adalah dengan menerapkan Algoritma dalam unsur game, seperti Algoritma A* pada pergerakan sebuah NPC (Non-Playable Characters), agar NPC tersebut mempelajari jalur untuk pergerakannya tanpa bantuan dari pengembang game. Dengan diterapkan cara ini pada game edukatif yang dibuat yaitu balloon shooter yang berisikan materi bilangan prima, diharapkan game yang dibuat bisa memberikan tantangan yang tentunya menyenangkan bagi pemain. Penerapan algoritma A* dinilai berhasil, ditandai dengan pergerakan balon yang sesuai dengan jalurnya dan bekerja 100% pada saat pengujian. Adapun pengujian fungsi permainan berdasarkan pengujian alpha dinyatakan berjalan sesuai fungsinya dan pengujian beta berdasarkan kuesioner yang telah disebarakan dinyatakan valid dan reliable.

Kata kunci— NPC (Non-Playable Characters), artificial Intelligence, game edukasi, algoritma a*, balloon shooter

I. PENDAHULUAN

Dalam perkembangan game saat ini, fokus dalam pengembangan game berpindah dari kemampuan grafis ke penerapan Artificial Intelligence. Perubahan fokus ini diharapkan dapat meningkatkan minat terhadap game untuk semua kalangan, salah satunya adalah anak sekolah dasar. Banyak game yang ditargetkan untuk anak sekolah dasar memiliki sifat menyenangkan, namun tidak diiringin dengan edukatif yang mengakibatkan game tersebut tidak memiliki nilai edukatif. Apabila game tersebut memiliki nilai edukatif, seringkali dinilai tidak menyenangkan bagi siswa sekolah dasar. Dengan adanya Artificial Intelligence pada game, diharapkan bisa menggabungkan game yang menyenangkan dengan unsur edukatif.

Artificial Intelligence adalah teknik yang digunakan untuk meniru kecerdasan yang dimiliki oleh makhluk hidup maupun benda mati untuk menyelesaikan sebuah persoalan

[1]. AI juga bisa diimplementasikan dalam sebuah game yang bertujuan untuk; meningkatkan tingkat kesulitan; membuat non-playable character; menentukan state game sesuai dengan input pengguna dan sebagainya.

Seringkali game yang menggunakan Artificial Intelligence untuk meningkatkan minat bermain game, namun sedikit dari game tersebut yang memiliki sifat edukatif. Maka dari itu, game yang diciptakan membuat anak-anak senang dan teredukasi, perlu penyematan fitur yang sesuai dengan kapasitas anak sekolah dasar, yaitu dengan menggunakan aturan edukasi dan Artificial Intelligence, agar game tersebut tetap menyenangkan dan memiliki nilai. Salah satunya dengan menggunakan Algoritma A* untuk membantu memberikan jarak/solusi terpendek dalam game, seperti pola gerak Non-Playable Characters.

II. KAJIAN TEORI

A. Game Edukasi

Game merupakan aktivitas yang berhubungan erat dengan sejarah umat manusia. Game terkadang juga dikaitkan dengan konsep "bermain" dan "menyenangkan" yang merupakan karakteristik dasar dari umat manusia. Konsep bermain mendefinisikan aturan dan perintah tertentu, bebas dari kepentingan material dan menciptakan kelompok social [2].

Game edukasi merupakan game yang memiliki unsur edukasi di dalamnya. Belajar melalui game merupakan salah satu solusi yang berguna yang telah diterapkan sepanjang sejarah umat manusia.

Berbagai bentuk dari Game yang bisa digunakan untuk edukasi. Penggunaan game sebagai sarana untuk penyampaian konten edukasi kepada pelajar dinilai lebih efektif. Cara yang paling mudah untuk menggunakan game sebagai *game-based education*, yaitu konten edukasi dengan menerapkan konten game dalam *multiple-choice test*. Adapun cara lain dengan menggunakan konsep game yang sudah ada, namun diberikan unsur edukasi dalam game tersebut [3].

B. Artificial Intelligence

Kecerdasan Buatan (*Artificial Intelligence*) adalah teknologi mensimulasikan kemampuan berpikir dan berbuat layaknya manusia [4]. AI secara mandiri mampu membuat

cara untuk menyelesaikan masalah, mampu menarik kesimpulan dan membuat keputusan.

C. Non-Playable Character

NPC adalah karakter yang tidak dikontrol oleh manusia [5], namun dikontrol oleh game itu sendiri. Dalam sebuah game, karakter dari NPC bisa didefinisikan dalam kode game atau menggunakan Artificial Intelligence. Menggunakan AI dalam NPC bisa menghasilkan karakter NPC yang unik dan bisa berubah seiring waktu. NPC juga bisa bereaksi sebagai bentuk respon dari player [6].

D. Algoritma A* Pathfinding

Pathfinding adalah metode untuk menentukan jalur terpendek. Metode ini memungkinkan mencari jalur terpendek sendiri dari *start node* ke *goal node* [7]. Contoh dari algoritma *pathfinding* berupa Algoritma Dijkstra, Algoritma A*, dan lain-lain.

Algoritma A* adalah salah satu algoritma yang menganalisis input, mengevaluasi sejumlah jalur yang memungkinkan dan menghasilkan solusi. Algoritma A* merupakan algoritma komputer yang digunakan secara luas dalam *graph transversal* dan pencarian jalur bersamaan dengan proses pencarian jalur yang efisien di sekitar yang disebut *node* [8]. Notasi yang digunakan dalam algoritma A* adalah sebagai berikut:

$$f(n) = g(n) + h(n)$$

Keterangan:

$f(n)$ = biaya estimasi terendah

$g(n)$ = biaya dari node awal ke node n

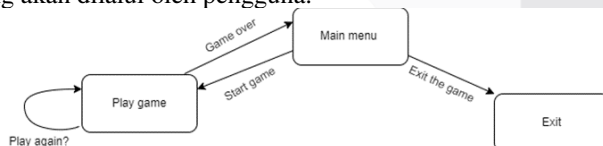
$h(n)$ = perkiraan biaya dari node n ke node akhir.

Terminologi dasar dari algoritma A* adalah titik awal, *node*, *open list*, *closed list*, *price (cost)*, dan *obstacles*. Keuntungan dari algoritma ini adalah menggunakan fungsi *heuristic* sebagai optimasi dengan memberikan nilai pada setiap node yang memandu algoritma untuk mendapatkan solusi yang diinginkan [7]. Kekurangan dari algoritma ini adalah algoritma ini tidak selalu menjamin rute optimal yang dipilih bukan rute yang paling pendek [9].

III. METODE

A. Gambaran Umum Sistem

Berikut merupakan gambaran umum sistem dari *game* yang akan dilalui oleh pengguna.



GAMBAR 1.
GAMBARAN UMUM SISTEM

Halaman *Main Menu* adalah yang pertama kali ditampilkan ketika membuka *game balloon shooter* dan terdapat pilihan untuk memulai permainan, mengatur permainan, dan keluar dari permainan. Setelah memilih mode *play*, maka pemain akan ditemukan menu pilihan tingkat kesulitan yang terdiri dari *easy*, *medium*, dan *hard*. Setelah memilih tingkat kesulitan, maka akan langsung ke area bermain sesuai dengan tingkat kesulitan yang dipilih. Balon yang muncul akan berjumlah banyak dan bergerak sesuai dengan jalur yang dibuat dengan Algoritma A*, namun tidak

semua balon harus ditembak. Terdapat aturan dalam menembak balon yaitu, pemain hanya boleh menembak balon yang bertuliskan angka bilangan prima. Bila pemain menembak balon yang benar, maka skor akan ditambah 1 dan jika pemain menembak balon yang salah maka skor akan dikurangi 1. Apabila waktu yang ditentukan sudah habis, maka waktu bermain sudah habis dan kembali ke menu. Apabila balon yang tersedia sudah habis namun waktu masih tersedia, maka akan muncul kumpulan balon yang baru lagi.

B. Fungsi dan Fitur

Berikut merupakan penjelasan fungsi dan fitur berdasarkan alur *game*:

1. Pada halaman *home*, *player* akan menemukan menu *Start* dan *Exit*.
2. Ketika *player* memilih menu *Start*, maka akan diberi pilihan tingkat kesulitan seperti *Easy*, *Medium*, dan *Hard*. *Player* akan memilih tingkat kesulitan bermain sesuai dengan keinginan.
3. Jika memilih *Easy*, akan diarahkan ke ruang bermain yang tingkat kesulitannya mudah.
4. Jika memilih *Medium*, akan diarahkan ke ruang bermain yang tingkat kesulitannya sedang.
5. Jika memilih *hard*, akan diarahkan ke ruang bermain yang tingkat kesulitannya susah.
6. Ketika *player* memilih *Exit*, maka akan keluar dari *game*.

C. Sistem Kontrol Game

Berikut adalah tabel berisi kontrol *game* yang bisa digunakan saat pembuatan *game*:

TABEL 1.
TABEL KONTROL GAME

| Kontrol | Aksi |
|-------------------------|--|
| <i>Left Click Mouse</i> | Menembak |
| <i>Mouse Aim</i> | Mengarahkan arah tembakan |
| R | Me-reset <i>game</i> apabila ada masalah |

D. Aturan dan Batasan Permainan

Adapun aturan dan batasan dalam permainan yang perlu diketahui oleh pemain adalah sebagai berikut.

1. Waktu bermain dalam 1 area bermain adalah 60 detik.
2. Pemain menembak balon dengan angka prima. Apabila pemain menembak balon yang benar, maka skor akan ditambah 1 per balon yang benar ditembak. Apabila pemain menembak balon yang salah, maka skor akan dikurangi 1 per balon yang salah ditembak
3. Peluru yang digunakan untuk menembaki balon memiliki jumlah tidak terbatas.

E. Pencarian Rute Algoritma A*

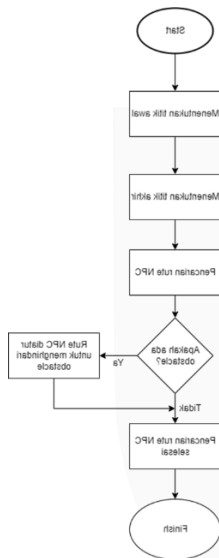
Langkah yang dilakukan dalam potongan program ini adalah menentukan titik akhir jalur NPC menggunakan koordinat kursor *mouse*, lalu dibuat perhitungan agar jalur yg digunakan tidak ada jalur yang miring. Pergerakan NPC akan dimulai dengan *left-click* dari *mouse* yang bersamaan sebagai tombol untuk menembak.

```

/// @description making an a* path
if(! is_moving) {
    var x_cell = floor(mouse_x / global.cell_size);
    var y_cell = floor(mouse_y / global.cell_size);
    var position_x =(x_cell * global.cell_size) + (global.cell_size / 2);
    var position_y = (y_cell * global.cell_size) + (global.cell_size / 2);
    mp_grid_path(global.grid, path, x + (global.cell_size / 2), y + (global.cell_size / 2), position_x, position_y, false);
    is_moving = true;
    path_start(path, 1, path_action_stop, false);
}
    
```

GAMBAR 2. POTONGAN PROGRAM A*

Dalam potongan program yang ada di gambar 3.8 ini berisi tentang program penentuan jalur NPC. Langkah yang dilakukan dalam potongan program ini adalah menentukan titik akhir jalur NPC menggunakan koordinat kursor *mouse*, lalu dibuat perhitungan agar jalur yg digunakan tidak ada jalur yang miring. Pergerakan NPC akan dimulai dengan *left-click* dari *mouse* yang bersamaan sebagai tombol untuk menembak.



GAMBAR 3. DIAGRAM ALUR PENCARIAN RUTE

Pertama yang dilakukan dalam membuat rute adalah menentukan titik awal NPC. Titik awal NPC telah ditentukan secara random dari koordinat x dan koordinat y. Titik akhir dari pergerakan NPC ditentukan dari titik posisi kursor *mouse* berada. Setelah ditentukan titik awal dan titik akhir dari NPC, maka algoritma akan otomatis membuat *path* untuk dilewati oleh NPC. Apabila rute yang ditemukan melewati *obstacle*, maka rute akan diatur untuk menghindari *obstacle*. Algoritma A* akan berjalan mencari rute lagi apabila posisi kursor *mouse* berubah dari posisi sebelumnya.

IV. HASIL DAN PEMBAHASAN

Dalam proses pembuatan permainan ini, implementasi sistem yang dibuat berdasarkan perancangan sebelumnya diterapkan untuk membuat *game*, lalu dari implementasi

tersebut dibuat pengujian seperti pengujian *User Interface*, kontrol *game* dan pengujian jalur NPC.

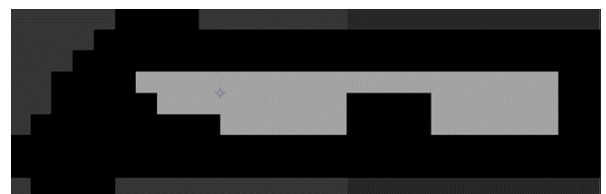
Pengujian yang digunakan menggunakan 2 metode, yaitu *Black-box testing* dan *White-box testing*. *Black-box testing* merupakan metode pengujian yang datanya didapat dari pengujian yang tidak mengetahui *code* dan logika dari programnya. *White-box testing* merupakan metode pengujian yang dilakukan oleh pengujian yang mengetahui *code* dan memastikan apakah *code* yang dijalankan sesuai dengan logika programnya [10].

A. Desain Karakter pada Game

Pada game Balloon Shooter ini, ada beberapa karakter yang dibuat, seperti karakter pistol pada player, dan karakter NPC dalam bentuk balon yang berisi angka prima dan tidak prima.



GAMBAR 4. OBJEK BALON ANGKA



GAMBAR 5. PISTOL

B. Pengujian *User-Interface*

Pengujian *User Interface* dilakukan untuk memastikan apakah tampilan *User Interface* dan tombol-tombolnya yang terdapat pada *game* telah berfungsi dengan baik.

TABEL 2. TABEL KONTROL UI

| No | Deskripsi | Output | Kondisi |
|----|----------------------|---|-----------|
| 1 | Tombol <i>Start</i> | Menuju ke halaman pilihan <i>Difficulty</i> | Berfungsi |
| 2 | Tombol <i>Exit</i> | Keluar dari <i>game</i> | Berfungsi |
| 3 | Tombol <i>Easy</i> | Menuju ke area bermain model <i>easy</i> | Berfungsi |
| 4 | Tombol <i>Medium</i> | Menuju ke area bermain model <i>medium</i> | Berfungsi |
| 5 | Tombol <i>Hard</i> | Menuju ke area bermain model <i>hard</i> | Berfungsi |
| 6 | Tombol <i>Menu</i> | Kembali menuju halaman <i>Menu</i> dari halaman <i>Times up</i> | Berfungsi |

Dari tabel diatas dapat dilihat bahwa dari pengujian *user interface* sudah berfungsi dengan baik dan tidak ditemukan error pada saat dilakukan pengujian.

C. Pengujian NPC

Pengujian NPC dilakukan dengan cara melakukan uji coba gerak balon pada titik awal dan titik akhir yang sudah ditentukan.



GAMBAR 6.
SKENARIO PENGUJIAN NPC

Pengujian ini dilakukan sebanyak 5 kali untuk setiap objek NPC. Tiap pengujian memiliki titik awal dan titik akhir yang berbeda. Berikut merupakan koordinat dan hasil percobaan dari pengujian.

TABEL 3.
TITIK KOORDINAT PENGUJIAN

| No | Percobaan | Titik koordinat awal(x,y) dan tujuan (x,y) |
|----|-------------|--|
| 1 | Percobaan 1 | (32,64), (678,439) |
| 2 | Percobaan 2 | (678,439), (1323,39) |
| 3 | Percobaan 3 | (1323,39), (1323, 695) |
| 4 | Percobaan 4 | (1329, 695), (32, 702) |
| 5 | Percobaan 5 | (32, 702), (678, 439) |

Hasil dari percobaan tersebut menunjukkan bahwa dari 5 kali percobaan tiap objek NPC, semua objek dapat melewati jalur yang telah ditentukan dan menghasilkan nilai performa 100%.

D. Pengujian Beta

Pengujian beta digunakan mengetahui kualitas aplikasi yang digunakan oleh pengguna, agar pengguna bisa memberikan kritik terhadap *game* yang diuji. Kritik tersebut

akan digunakan oleh pengembang *game* untuk diperbaiki [11]. Pengujian ini akan dilakukan dengan kuesioner *google form*.

Pengujian pengguna dilakukan berdasarkan pertanyaan yang diajukan kepada pengguna. Responden akan diberi link kuesioner yang berisi video demo *gameplay* dari *game Balloon Shooter* dan menjawab pertanyaan kuesioner.

Tabel 4 dibawah adalah deskripsi relasi antara jawaban kuisisioner dengan klasifikasinya. Pada tugas akhir ini, jawaban dikelompokkan menjadi 5 kelompok. Kelompok tersebut meliputi sangat tidak setuju, tidak setuju, kurang setuju, setuju, dan sangat setuju. Adapun distribusi masing-masing pilihan adalah setara, yaitu 20%.

TABEL 4.
RELASI ANTARA SKOR JAWABAN DENGAN KLASIFIKASI

| Jawaban | Keterangan |
|-------------|---------------------|
| 0 – 19.99% | Sangat tidak setuju |
| 20 – 39.99% | Tidak setuju |
| 40 – 59.99% | Kurang setuju |
| 60 – 79.99% | Setuju |
| 80 – 99.99% | Sangat setuju |

Hasil survei mengukur lima aspek usability, yaitu: *learnability* (LR), *memorability* (MF), *efficiency* (EF), *error* (ER), dan *satisfaction* (SF) [12]. Daftar pertanyaan survei meliputi tampilan, sound, fitur, tingkat pemahaman, keramahan fitur, dan relasi antara permainan dengan edukasi.

Adapun penilaian berdasarkan aspek *usability* sebagai berikut :

1. Penilaian terhadap aspek *Learnability* yaitu menjelaskan tingkat kemudahan pengguna terhadap *game balloon shooter* dalam memenuhi task dasar memiliki nilai sebesar 92% berdasarkan pertanyaan “Apakah *game balloon shooter* mudah dipahami?”, maka hasil tersebut menunjukkan bahwa aspek *learnability* telah terpenuhi.
2. Penilaian terhadap aspek *memorability* yaitu menjelaskan tingkat kemudahan pengguna dalam menggunakan rancangan yang baik memiliki nilai sebesar 90% berdasarkan pertanyaan “Apakah tampilan *game* yang telah dibuat berkesan dan menarik?”, maka hasil tersebut menunjukkan bahwa aspek *memorability* telah terpenuhi.
3. Penilaian terhadap aspek *efficiency* yaitu menjelaskan tingkat kecepatan pengguna dalam menyelesaikan task yang telah dipelajari memiliki nilai sebesar 87,33% berdasarkan pertanyaan “Apakah fitur pada *game* ramah pengguna?”, maka hasil tersebut menunjukkan bahwa aspek *memorability* telah terpenuhi.
4. Penilaian terhadap aspek *error* yaitu menjelaskan jumlah error yang dilakukan pengguna dan cara memperbaiki error tersebut memiliki nilai sebesar 92% berdasarkan pertanyaan “Apakah fitur pada *game* ini berfungsi dengan baik?”, maka hasil

tersebut menunjukkan bahwa aspek *error* telah terpenuhi.

5. Penilaian terhadap aspek *satisfaction* yaitu mengenai tingkat kepuasan pengguna terhadap game yang dirancang sebesar 88,11%, maka hasil tersebut menunjukkan bahwa game memiliki aspek *satisfaction* yang baik.

E. Uji Validitas

Pengujian validitas dilakukan dengan korelasi produk *moment pearson* dengan rumus [13]:

$$r = \frac{N\sum XY - \sum X \sum Y}{\sqrt{[N\sum X^2 - (\sum X)^2][N\sum Y^2 - (\sum Y)^2]}}$$

Keterangan:

r = koefisien korelasi *pearson*

N = jumlah responden

X = skor pertanyaan per responden

Y = skor total per responden

Berdasarkan persamaan diatas, pertanyaan dikatakan valid apabila nilai $r > r_{tabel}$.

Berikut ini adalah tabel yang telah dihitung dengan persamaan diatas, kemudian dibandingkan dengan nilai r_{tabel} yang bertaraf signifikan 5% dengan banyaknya jumlah responden yaitu 30 orang.

TABEL 5.
TABEL UJI VALIDITAS

| Pertanyaan | r | r_{tabel} | keterangan |
|------------|-------------|-------------|------------|
| 1 | 0,702675146 | 0,361 | VALID |
| 2 | 0,714814567 | 0,361 | VALID |
| 3 | 0,690161172 | 0,361 | VALID |
| 4 | 0,713327353 | 0,361 | VALID |
| 5 | 0,651815563 | 0,361 | VALID |
| 6 | 0,684226013 | 0,361 | VALID |

Berdasarkan nilai r pada table diatas, menunjukkan bahwa $r_{setiap\ pertanyaan} > r_{tabel}$ sehingga dapat dikatakan bahwa pengujian yang dilakukan adalah valid.

F. Uji Reliabilitas

Uji reliabilitas dilakukan menggunakan metode *Cronbach's Alpha* dengan persamaan sebagai berikut [14]:

$$r_{11} = \left[\frac{k}{k-1} \right] \left[1 - \frac{\sum s_i^2}{s^2} \right]$$

Keterangan:

r_{11} = koefisien reliabilitas alpha

k = jumlah pertanyaan

$\sum s_i^2$ = varian nilai setiap pertanyaan

s^2 = varian total

Berdasarkan persamaan diatas, perhitungan yang didapat kemudian dibandingkan dengan nilai r_{tabel} yang bertaraf signifikan 5% dengan banyaknya jumlah responden yaitu 30 orang.

TABEL 6.
TABEL UJI RELIABILITAS

| r_{11} | r_{tabel} | Keterangan |
|-------------------|-------------|-----------------|
| 0.777512897986035 | 0,361 | <i>Reliable</i> |

Berdasarkan nilai yang ditunjukkan pada table diatas $r_{11} > r_{tabel}$, dapat disimpulkan bahwa pengujian yang dilakukan adalah *reliable*.

V. KESIMPULAN

Melihat hasil dari pengujian dan analisis yang dilakukan maka dapat disimpulkan bahwa pergerakan NPC dapat dibuat menggunakan Algoritma A* dan bekerja 100% pada saat pengujian.

Adapun hasil dari dari survei melalui kuesioner, implementasi unsur edukasi pada *game Balloon Shooter* cukup baik ditandai dengan persentase dari hasil survei aspek *usability* yaitu 87,33%.

REFERENSI

- [1] A. Ahmad, "Yayasan Cahaya Islam, Jurnal Teknologi Indonesia," *Mengenal Artificial Intelligence, Machine Learning, Neural Network, dan Deep Learning*, 2017.
- [2] H. A. Fatta, Z. Maksom and M. H. Zakaria, "Game-based Learning and Gamification: Searching for Definitions," p. 41, 2018.
- [3] Z.-Y. Liu, Z. A. Shaikh and F. Gazizova, "Using the Concept of Game-Based Learning in Education," vol. 15, pp. 54-55, 2020.
- [4] A. P. Malik, M. Pathania and V. K. Rathaur, "Overview of artificial intelligence in medicine," *Journal of Family Medicine and Primary Care*, vol. 8, p. 2328, 2019.
- [5] V. M. Petrovic, "Artificial Intelligence and Virtual Worlds - Toward Human-Level AI Agents," *IEEE Access*, vol. 6, p. 39977, 2018.
- [6] U. N. Rupasingha, "Assess the awareness of Artificial Intelligence (AI) in Computer," pp. 10-11, 2020.
- [7] A. Candra, M. A. Budiman and R. I. Pohan, "Application of A-Star Algorithm on Pathfinding Game," *Journal of Physics: Conference Series*, p. 1, 2021.
- [8] S. D. H. Permana, K. B. Yogha Bintoro, B. Arifitama and A. Syahputra, "Comparative Analysis of Pathfinding Algorithms A *, Dijkstra, and BFS on Maze Runner Game," *International Journal Of Information System & Technology*, vol. 1, no. 2580-7250, p. 1, 2018.

- [9] D. Foead, A. Ghifari, B. M. Kusuma, N. Hanafiah and E. Gunawan, "A Systematic Literature Review of A* Pathfinding," *Procedia Computer Science*, vol. 179, p. 509, 2021.
- [10] Z. Zhang and J. Qian, "Test Suite Augmentation via Integrating Black- and White-box Testing Techniques," *International Journal of Performability Engineering*, vol. 14, p. 1324, 2018.
- [11] J. J. Lee-Jayaram, B. W. Berg, A. Sy and K. M. Hara, "Alpha and Beta Testing During a Faculty Development Course," *Simulation in Healthcare: The Journal of the Society for Simulation in Healthcare*, vol. 14, no. 1, p. 43, 2019.
- [12] L. Amelia and D. Novita, "Analisis Usability Aplikasi Pengisian KRS Online STMIK XYZ Palembang Menggunakan Use Questionnaire," *Jurnal Informasi dan Komputer*, vol. 7, pp. 18-19, 2019.
- [13] F. Yusup, "Uji Validitas dan Reliabilitas Instrumen Penelitian Kuantitatif," *Jurnal Tarbiyah: Jurnal Ilmiah Kependidikan*, vol. 7, p. 19, 2018.
- [14] F. Yusup, "Uji Validitas dan Reliabilitas Instrumen Penelitian Kuantitatif," *Jurnal Tarbiyah: Jurnal Ilmiah Kependidikan*, vol. 7, pp. 21-22, 2018.

