

IMPLEMENTASI KECERDASAN BUATAN UNTUK NPC PADA PERMAINAN RACING CAR DENGAN ALGORITMA A*

ARTIFICIAL INTELLIGENCE IMPLEMENTATION FOR NPC IN CAR RACING GAME BY USING A* ALGORITHM

Egi Pangestu¹, Dr. Purba Daru Kusuma², Ratna Astuti Nugrahaeni³

^{1,2,3} Universitas Telkom, Bandung

¹egipangestu@student.telkomuniversity.ac.id, ²purbodaru@telkomuniversity.ac.id, ³ratnaan@telkomuniversity.ac.id

Abstrak

Game adalah sebuah permainan yang memiliki alur cerita, aturan, dan karakter didalamnya. Di dalam game pasti terdapat adanya karakter yang dimainkan oleh player dan NPC (Non-Player Character) sebagai pelengkap dalam sebuah game. Tujuan adanya NPC adalah untuk membuat suatu game terasa lebih nyata dan lebih interaktif, terlebih untuk permainan offline atau single player. Pada game balapan misalnya, NPC akan diperbarui jika terdapat penambahan suatu konten di dalam game misalnya lintasan balap yang baru. Agar mendapatkan lebih banyak variasi, maka dilakukan pengembangan dengan menanamkan sistem berbasis kecerdasan buatan terhadap NPC (Non-Playable Character) dengan menggunakan metode algoritma A*. Metode ini menjadikan NPC lebih bervariasi karena setiap nodes yang dirubah maka penentuan rute yang dilakukan oleh NPC pasti berubah. Hasil implementasi dari algoritma A* yang telah diterapkan membuat NPC dapat menentukan jarak terdekat melalui perhitungan target dan jarak terdekat pada setiap checkpoint yang terdapat didepannya. Penerapan algoritma A* ini menambahkan variasi pada game, karena setiap perubahan checkpoint pada lintasan akan merubah rute yang dilalui oleh NPC tersebut.

Kata Kunci: NPC (Non-Player Character), Kecerdasan Buatan, Algoritma A*, Checkpoint, Game.

Abstract

Game is a game that has a storyline, rules, and characters in it. In the game there must be characters that are played by players and NPCs (Non-Player Characters) as a complement in a game. The purpose of having NPCs is to make a game feel more real and more interactive, especially for offline or single player games. In racing games for example, NPCs will be updated if there is additional content in the game, such as a new racing track. To get more variety, development is carried out by embedding an artificial intelligence-based system for NPC (Non-Player Character) using the A* algorithm method. This method makes the NPC more varied because every node that is changed, the route determination made by the NPC must change. The result of the implementation of the A* algorithm that has been applied allows the NPC to determine the closest distance through the calculation of the target and the closest distance to each checkpoint in front of it. The application of the A* algorithm adds variety to the game, because every change in the checkpoint on the trajectory will change the route traversed by the NPC.

Keywords: NPC (Non-Player Character), Artificial Intelligence, Algorithm A*, Checkpoint, Game.

1. Pendahuluan

Game adalah sebuah permainan yang aktifitas didalamnya seperti realita akan tetapi berpura-pura namun ada tujuan yang harus dicapai untuk mendapatkan kemenangan [10]. Dahulu game hanya dijadikan sebagai sebuah hiburan dikala mengisi waktu luang. Sekarang game dapat dijadikan sebagai sarana bisnis, olahraga e-sport dan sebagai sarana pembelajaran. Dengan bantuan teknologi sekarang game dapat dimainkan secara offline ataupun online. Dalam sebuah game pasti akan terdapat karakter yang dimainkan oleh player dan karakter pendukung atau biasa disebut NPC (Non-Player Character).

NPC adalah sebuah karakter didalam game yang membuat sebuah game terasa lebih hidup dan interaktif. Keberadaan NPC sangat berpengaruh dalam sebuah game, terutama jika game tersebut dimainkan secara offline atau single player. Pada game balapan yang akan dibuat nantinya, keberadaan NPC sangat penting karena bertindak sebagai lawan yang harus dikalahkan oleh player. NPC akan bertindak sesuai rules yang telah diatur oleh pengembang game dan akan bergerak pada lintasan yang telah ditentukan.

Dengan algoritma A* akan menambah banyak variasi pada jalur balapan yang dilalui NPC karena setiap perubahan nodes atau checkpoint pada jalur tersebut maka akan berubah juga rute yang dilewati oleh NPC. Selain itu penerapan algoritma A* sering digunakan untuk mencari rute tercepat pada sebuah game sehingga pada NPC yang akan ditanamkan sistem kecerdasan buatan ini diharapkan NPC dapat mencari rute tercepat pada lintasan yang terdapat pada game tersebut.

2. Landasan Teori

2.1 Game

Game adalah sebuah permainan yang berisi kumpulan aturan yang dapat memberikan situasi bersaing baik itu dengan player lain ataupun dengan NPC pada game tersebut untuk mendapatkan kemenangan [1]. Dari beberapa jenis genre game yang ada, game racing menjadi pilihan pada topik penelitian ini. Jenis genre ini mengendalikan sebuah kendaraan untuk memenangkan dalam sebuah balapan pada lintasan yang terdapat pada game tersebut. Ada beberapa elemen yang wajib ada dalam sebuah game diantaranya elemen visual, elemen audio, dan gameplay [2].

Gameplay pada game dapat dibedakan menurut genre yang ada saat ini. Genre juga memiliki tugas untuk membuat batasan bagi para pengembang game agar bisa berkreasi lebih spesifik. Terkadang didalam satu game terdapat lebih dari satu genre yang mana disebut hybrid genre. Beberapa jenis genre game antara lain Action Games, Strategy games, Role-Playing Games, Sports Games, Racing/Driving Games, Simulation/Building Games [3].

2.2 Artificial Intelligence

Artificial intelligence atau sering disebut dengan kecerdasan buatan adalah kecerdasan yang diterapkan pada suatu mesin agar dapat bertindak layaknya seperti manusia. Pada game, kecerdasan buatan diterapkan pada NPC untuk digunakan sebagai lawan pemain [4]. Dari setiap algoritma yang ada, algoritma yang digunakan kali ini untuk diimplementasikan pada NPC adalah algoritma A*.

2.3 NPC (Non-Player Character)

NPC adalah sebuah karakter yang terdapat dalam game dan ikut berperan aktif terhadap kegiatan yang dilakukan oleh player, namun NPC tidak bisa dikendalikan oleh player [1]. Karakter tersebut dikendalikan oleh komputer dengan di implementasikannya kecerdasan buatan pada NPC sehingga karakter tersebut dapat melakukan pergerakan [7]. Pada game racing car ini, karakter NPC ini adalah sebuah mobil yang menjadi lawan player ketika dilintasi balap nanti.

2.4 Algoritma A*

Algoritma A* adalah algoritma Best First Search (BFS) yaitu menggabungkan fungsi biaya Uniform Cost Search dan Greedy Best First Search. Biaya yang diperhitungkan ialah biaya sebenarnya ditambah dengan biaya perkiraan. Jika dituliskan dalam notasi matematika sebagai berikut:

$$f(n) = g(n) + h(n) \quad (1)$$

Dimana:

$f(n)$ = fungsi evaluasi

$g(n)$ = biaya yang dikeluarkan dari keadaan awal sampai keadaan n

$h(n)$ = estimasi biaya untuk sampai pada suatu tujuan mulai dari n

Dengan melakukan perhitungan seperti berikut, algoritma A* dinilai optimal [5]. Metode ini membuat keseimbangan yang baik antara kecepatan dan ketepatan pada game [6].

Pada beberapa kasus pencarian rute menggunakan algoritma A*, algoritma ini dapat menemukan pencarian rute tanpa adanya kendala atau terjebak oleh suatu halangan. Algoritma A* bekerja jika cost perkiraan tidak lebih besar dari cost sebenarnya. Ketika cost perkiraan lebih besar maka rute yang akan ditemukan bukanlah rute yang tercepat. Jadi node dengan nilai terendah akan diperiksa lebih awal pada $g(n) + h(n)$. Maka ketika fungsi tersebut terpenuhi, pencarian rute dengan algoritma A* dapat optimal [5].

2.5 Implementasi Algoritma A*

Pada semua game pasti akan ada aturan main. Aturan tersebut digunakan untuk mempermudah pencarian dan kebebasan pada para peneliti atau pengembang. Game bisa menghasilkan beberapa pencarian ruang. Dengan hal itu dibutuhkan suatu teknik yang dikenal dengan nama heuristic dan berkaitan dengan penelitian tentang kecerdasan buatan.

Fungsi heuristic memfokuskan pencarian pada node-node yang mengarah pada node tujuan. Lalu pencarian dilakukan dengan memeriksa waktu yang dibutuhkan menuju node tujuan. Dengan begitu jumlah node yang diperiksa dapat diminimalisasikan sehingga waktu yang digunakan tidak terlalu banyak. Tapi keadaan terburuk dari algoritma A* ini adalah ketika semua node pada jalur diperiksa dan tidak ditemukannya jalur terbaik [5].

2.6 Teori Pengukuran dan Pengujian R-Validation

Pada setiap penelitian diperlukan data yang dapat menjawab tujuan penelitian tersebut. Data yang benar harus diuji validitas dan reliabilitasnya agar sesuai dengan keadaan yang sebenarnya. Validitas adalah penilaian pada elemen-elemen yang diproses dengan analisis rasional. Reliabilitas adalah pengukuran dari percobaan alat yang diuji [11].

Dalam penelitian ini diukur nilai validitas dan reliabilitas dari pertanyaan yang terdapat pada kuisioner untuk menilai valid atau tidaknya. Untuk menilai kuisioner tersebut valid atau tidak diperlukan rumus perhitungan yaitu r hitung > r banding sebagai berikut:

$$r = \frac{n \sum XY - (\sum X) (\sum Y)}{\sqrt{(n \sum X^2 - (\sum X)^2) (n \sum Y^2 - (\sum Y)^2)}} \tag{2}$$

Dimana:

N: Jumlah Responden

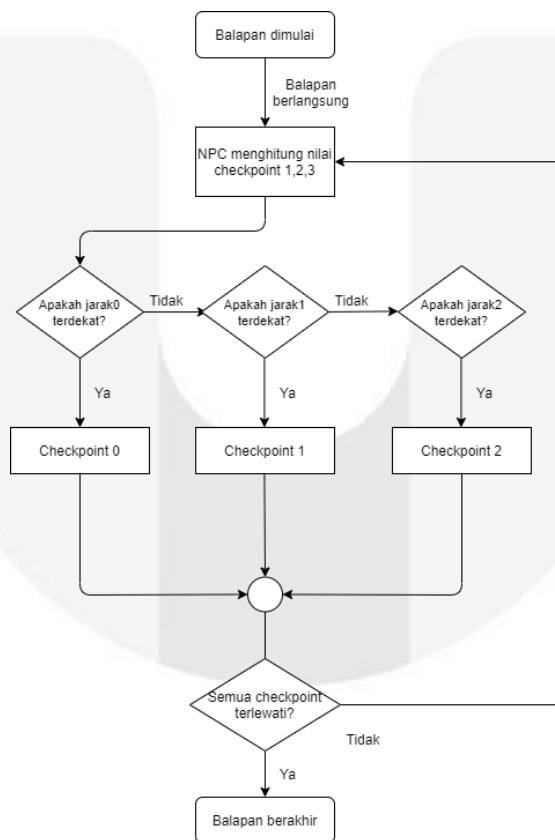
X: Jumlah nilai soal per responden

Y: Jumlah nilai total per responden

3. Pembahasan

3.1 Gambaran Umum Sistem

Sistem yang diimplementasikan berupa kecerdasan buatan menggunakan algoritma A* yang dimana sistem tersebut bekerja mencari rute terpendek untuk dilalui oleh NPC.



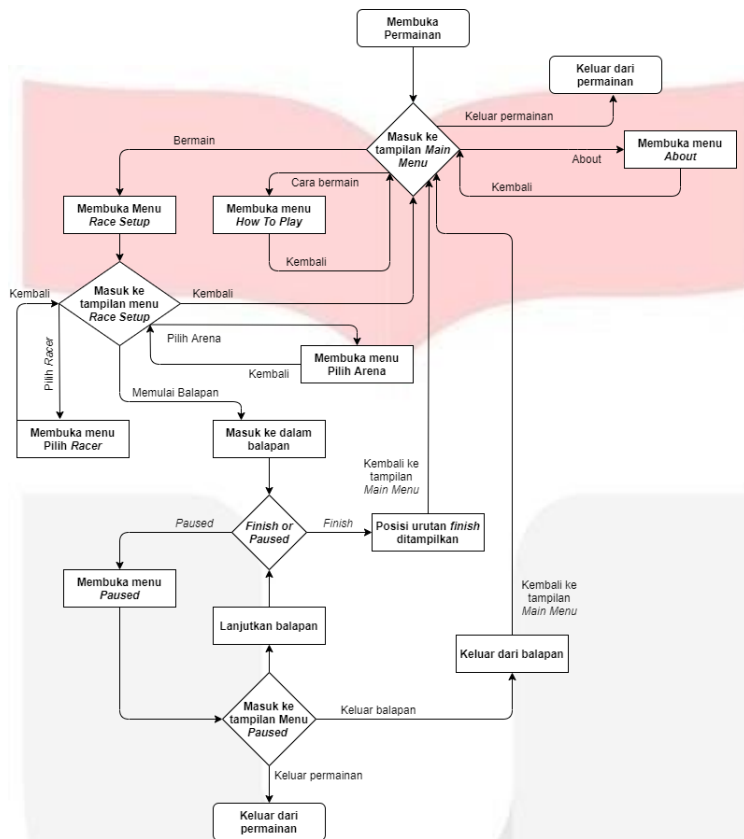
Gambar 1 Gambaran Umum Sistem

Pada gambar 1 menunjukkan bagaimana cara kerja NPC dari mulainya balapan hingga mencapai garis finish. Pertama ketika balapan dimulai, NPC akan menghitung nilai dari ketiga checkpoint yang ada didepannya setelah itu NPC menentukan jarak dari ketiga kondisi jarak pada ketiga checkpoint. Apabila salah satu jarak terpenuhi maka akan dipilih checkpoint tersebut untuk dilalui oleh NPC. Lalu akan ada kondisi dimana NPC akan mengecek apakah semua checkpoint sudah terlewati semuanya, jika belum terlewati maka proses akan pertama akan diulang sedangkan jika semua checkpoint sudah terlewati semuanya maka game selesai.

3.2 Perancangan Sistem Game

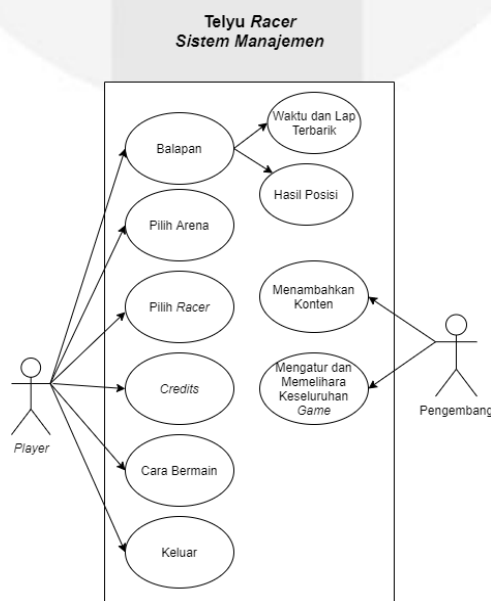
3.2.1. Flowchart

Flowchart sistem di bawah ini dirancang untuk memahami bagaimana alur sistem dari game ini bekerja.



Gambar 2 Flowchart Sistem

3.2.1. Use Case Diagram



Gambar 3 Use Case Sistem

Di atas merupakan *use case diagram* dari *game* yang telah dibuat, dimana bila sebagai *player* kita diberikan akses untuk dapat bermain balapan, lalu kita juga dapat melakukan pemilihan terhadap arena dan juga mobil yang ingin digunakan serta menu tambahan seperti *credits*, *how to play*, dan *quit*.

4. Implementasi dan Pengujian Sistem

4.1 Pengujian R-Validation

Pada uji validitas bertujuan mengukur valid atau tidaknya pertanyaan pada kuesioner yang telah diberikan kepada *tester*. Berikut merupakan hasil pengujian validitas yang telah dilakukan.

Tabel 1 Hasil Validitas

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
$N \sum XY - \sum X \sum Y$	439	521	134	60	1139	1199	1102	1006
$N \sum X^2 - (\sum X)^2$	130	102	60	298	310	304	268	250
$N \sum Y^2 - (\sum Y)^2$	55230	55230	55230	55230	55230	55230	55230	55230
r_{hitung}	0.1638344 86	0.21950 7 88	0.07361 0819	0.01478 9572	0.27526 7764	0.29261 3844	0.28643 539	0.2707325 38
$r_{banding}$	1.720742903							
Kriteria	Sangat Rendah	Rendah	Sangat Rendah	Sangat Rendah	Rendah	Rendah	Rendah	Rendah
Keterangan	Valid							

Berdasarkan hasil pengujian validitas yang telah dilakukan pada setiap pertanyaan maka diperoleh hasil yaitu setiap pertanyaan telah dinyatakan valid.

Tabel 2 Hasil Pengujian Reliabilitas

\sum Varian	Varian Total	r_{11}	$r_{banding}$	Kriteria	Keterangan
3.255198488	10.58601134	0.791428571	1.720742903	Tinggi	Reliable

Berdasarkan hasil pengujian reliabilitas di atas diperoleh hasil yaitu setiap pertanyaan pada kuesioner dinyatakan *reliable*.

4.2 Pengujian Performa NPC

Berikut ini adalah hasil pengujian beserta data persentase dari kemenangan yang dimiliki oleh NPC dapat dilihat pada Tabel 3 dan 4 di bawah ini:

Tabel 3 Hasil Pengujian Performans

No	Kelas Uji	Poin Uji	Persentase
1	Arena 1	Apakah NPC sudah tepat melewati <i>checkpoint</i> yang diharapkan?	70%
		Apakah performa NPC sudah baik ketika menentukan rute?	60%
2	Arena 2	Apakah NPC sudah tepat melewati <i>checkpoint</i> yang diharapkan?	65%
		Apakah performa NPC sudah baik ketika menentukan rute?	50%

Pada tabel 3 dilakukan pengujian NPC pada arena 1 dan arena 2 untuk melihat apakah ada perbedaan diantara kedua arena tersebut terhadap NPC. Pada arena 1 NPC sudah dapat menentukan *checkpoint* yang dilewati akan tetapi pada kondisi tertentu terdapat NPC yang stuck di pembatas lintasan dikarenakan NPC tidak dilengkapi dengan gerak mundur, akan tetapi setelah itu NPC akan kembali melanjutkan balapannya.

Lalu pada arena 2 dilakukan kembali pengujian pada NPC dan hasil yang didapatkan pada pengujian yang pertama yaitu NPC sudah tepat melewati *checkpoint* yang diinginkan. Akan tetapi karena arena 2 tidak memiliki pembatas lintasan yang dimana NPC keluar dari jalur karena dengan cara tersebut NPC mendapatkan jalur yang lebih dekat dari pada mengikuti jalur lintasan.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Berdasarkan hasil analisis terhadap beberapa skenario pengujian yang telah dilakukan maka dapat disimpulkan:

1. Algoritma A* berhasil membuat NPC menentukan checkpoint yang harus dilewati untuk menentukan rute tercepat pada lintasan yang digunakan. Performa dari NPC tersebut dapat dinilai dengan presentase sebesar 60% -70% pada game Telyu Racer dan dapat mengimbangi player ketika balapan sedang berjalan.
2. Terdapat masalah pada NPC yang terjebak di pembatas jalan pada lintasan, hal ini disebabkan karena NPC belum dilengkapi dengan gerak atau jalan mundur.

5.2 Saran

Adapun saran untuk pengembangan tugas akhir ini untuk dilakukannya penelitian lebih lanjut yaitu:

1. Membuat tampilan dari game Telyu Racer lebih menarik dari segi grafik, level kesulitan, dan memperbanyak jenis lintasan.
2. Untuk penelitian berikutnya mobil NPC perlu dilengkapi gerak mundur untuk menghindari stuck pada pembatas jalan yang terdapat dilintasan.

Referensi

- [1] Athhariq, A., & Putra, D. A. (2018). Penentuan Pergerakan Non-Player Character Menggunakan Algoritma a* Pada Game Action- Role-Playing Game. *Jurnal Infomedia*, 2(2), 35–40. <https://doi.org/10.30811/v2i2.516>
- [2] Pramono, A. (2015). Algoritma Pathfinding A* Pada Game RPG Tanaman Higienis. *Jurnal Edukasi Dan Penelitian Informatika (JEPIN)*, 1(2). <https://doi.org/10.26418/jp.v1i2.12517>
- [3] Tjahyadi, M., Sinsuw, A., Tulenan, V., & Sentinuwo, S. (2015). Prototipe Game Musik Bambu Menggunakan Engine Unity 3D. *Jurnal Teknik Informatika*, 4(2), 1–6. <https://doi.org/10.35793/jti.4.2.2014.6990>
- [4] Billy, B., Kuswardayan, I., & Khotimah, W. N. (2017). Implementasi Artificial Intelligence pada game Defender of Metal City dengan menggunakan Finite State Machine. *Jurnal Teknik ITS*, 6(2), 640–645. <https://doi.org/10.12962/j23373539.v6i2.25151>
- [5] Herwanto, P., & Tris. (2016). Rancang Bangun Game 3D “Ena Burena” Dengan Algoritma a* Dan Collision Detection Menggunakan Unity 3D Berbasis Desktop Dan Android. *Jurnal Informasi*, VIII(1), 1–22.
- [6] Sazaki, Y., Satria, H., Primanita, A., & Syahroyni, M. (2018). Analisa Perbandingan Algoritma A* dan Dynamic Pathfinding Algorithm dengan Dynamic Pathfinding Algorithm untuk NPC pada Car Racing Game. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 5(1), 95. <https://doi.org/10.25126/jtiik.201851544>
- [7] Maxmilano, R. (2020). Implementasi Algoritma Boids Dan Collision Avoidance Pada Pergerakan NPC Dalam Game 2D Berbasis Web. 021, 1–19.
- [8] Gao, J., Tsao, H.-S. J., & Wu, Y. (2003). *Testing and Quality Assurance for Component-based Software*. Nordwood: ARTECH HOUSE, INC.
- [9] Nelson, S. (2003). United States of America Patent No. NASA/CR-2003- 212806
- [10] Muhammad, A., Nafi'iyah, N., & Ismi, A. (2017). *Game Balap Perahu Berbasis Android*. 1(1), 479–486.
- [11] Yusup, F. (2018). Uji Validitas dan Reliabilitas Instrumen Penelitian Kuantitatif. *Jurnal Tarbiyah : Jurnal Ilmiah Kependidikan*, 7(1), 17–23. <https://doi.org/10.18592/tarbiyah.v7i1.2100>