

IMPLEMENTASI ALGORITMA A-STAR DALAM MENENTUKAN RUTE PENGEJARAN OPTIMAL PADA PERMAINAN MEMBASMI NARKOBA

THE IMPLEMENTATION OF A-STAR ALGORITHM TO DETERMINE OPTIMAL PURSUE ROUTES IN DRUG ERADICATION GAME

Lukas Tommy ¹, Yohanes Setiawan Japriadi ², Syachriza Hilmaida Habibur ³

^{1,2,3}Program Studi Teknik Informatika, Fakultas Teknologi Informasi, ISB Atma Luhur

¹lukastommy@atmaluhur.ac.id, ²ysetiawanj@atmaluhur.ac.id,
³1611500099@mahasiswa.atmaluhur.ac.id

Abstrak

Narkoba memiliki beberapa dampak negatif seperti ketergantungan, kerusakan otak dan saraf, hingga kematian jika disalahgunakan. Sosialisasi mengenai bahaya penyalahgunaan narkoba di masyarakat oleh pihak kepolisian masih dilaksanakan secara konvensional sehingga kurang efektif terlebih di masa pandemi Covid-19 seperti saat ini. Pada penelitian ini akan diusulkan sebuah permainan Android yang dapat menyampaikan materi bahaya narkoba dengan menarik, interaktif, dan efektif. Pada permainan yang diusulkan, kecerdasan buatan akan diterapkan pada musuh agar dapat mengejar pemain dengan melalui rute yang optimal. Salah satu algoritma penentuan rute terdekat dari lokasi awal menuju tujuan adalah *A-Star*. Algoritma *A-Star* memanfaatkan heuristik dalam mengevaluasi simpul yang ada pada peta berbentuk *grid* sehingga waktu komputasinya lebih singkat dibandingkan Dijkstra. Berdasarkan analisis yang sudah dilaksanakan diketahui bahwa secara keseluruhan kinerja dari algoritma *A-Star* sudah baik dimana tujuh musuh di permainan dapat mengejar pemain secara *real-time* melalui rute yang optimal. Musuh namun tidak mampu untuk bekerja sama dalam mengepung pemain dan cenderung untuk berkumpul di satu titik dengan melalui rute yang sama sehingga efektivitas pengejaran tidak maksimal. Permainan yang diusulkan juga dapat menyampaikan materi mengenai jenis-jenis narkoba dan dampak negatifnya dengan menarik, interaktif, sekaligus efektif melalui *menu* gamepedia sekaligus karakter musuh yang ditampilkan di sepanjang permainan.

Kata kunci: algoritma, A-Star, narkoba, *pathfinding*, permainan

Abstract

Drugs have several negative effects such as dependence, brain and nerve damage, to death if abused. Socialization about the dangers of drug abuse in society by police is still carried out conventionally so that it is less effective especially during the Covid-19 pandemic like today. In this research, an Android game that can convey material on the dangers of drugs in an interesting, interactive, and effective way will be proposed. In the proposed game, an artificial intelligence will be applied to the enemies in order to pursue player through optimal route. One of the algorithms for determining the closest route from the initial location to the destination is A-Star. A-Star algorithm utilize heuristics in evaluating nodes on a grid-shaped map so that its computation time is shorter than Dijkstra's. Based on the analysis that has been carried out, it is known that the overall performance of A-Star algorithm is good where the seven enemies in game can pursue player in real-time through optimal route. The enemies, however, are unable to cooperate in surround player and tend to gather at one point through the same route so that the effectiveness of the pursuit is not maximized. The proposed game can also convey material about the types of drugs and their negative impacts in an interesting, interactive, and effective manner through with gamepedia menu as well as the enemy characters displayed throughout the game.

Keywords: algorithm, A-Star, game, drugs, *pathfinding*

1. PENDAHULUAN

Awalnya narkoba atau napza (singkatan dari narkotika, psikotropika, dan obat terlarang) diperuntukkan sebagai obat bius atau obat bagi penyakit tertentu. Persepsi tersebut kemudian berubah

dikarenakan narkoba sering disalahgunakan untuk rekreasi dan doping [1]. Beberapa komplikasi yang akan timbul dari penyalahgunaan narkoba antara lain ketergantungan, penurunan kesadaran, gangguan psikis, kerusakan otak dan saraf, hingga kematian.

Strategi yang diterapkan oleh pihak kepolisian untuk mencegah terjadinya penyalahgunaan narkoba di masyarakat masih berupa sosialisasi langsung. Strategi ini dirasakan kurang efektif [2] jika dilihat dari masih cukup tingginya jumlah kasus penyalahgunaan narkoba di Indonesia, yaitu menurut data dari BNN (Badan Narkotika Nasional) di sepanjang tahun 2019 terungkap 33.371 kasus dimana sebagian besar adalah generasi milenial (usia remaja hingga dewasa muda) [3]. Hal ini diduga disebabkan oleh peserta dari generasi milenial kurang tertarik untuk membaca brosur dan mendengarkan sosialisasi bahaya narkoba dimana disampaikan secara monoton [2]. Selain itu, di masa pandemi Covid-19 ini, sosialisasi langsung dikhawatirkan dapat membentuk kluster penularan Covid-19 baru dan pembatasan jumlah peserta membuat strategi ini semakin tidak efektif [4].

Untuk mengatasi permasalahan tersebut, diperlukan strategi penyampaian bahaya narkoba yang menarik, efisien, sekaligus interaktif agar partisipasi masyarakat menjadi meningkat. Strategi berupa aplikasi pengenalan jenis narkoba berbasis Android [2] masih kurang menarik karena aplikasi yang dihasilkan tidak jauh berbeda dengan brosur elektronik. Berdasarkan hal tersebut, akan diusulkan *game* / permainan berbasis Android yang memiliki pesan mendidik terkait bahaya narkoba.

Game dapat disisipkan suatu materi pelajaran, sehingga tingkat pemahaman pengguna terhadap materi tersebut menjadi meningkat hanya dengan memainkannya. Selain itu, dengan konsep penyampaian materi dalam bentuk permainan yang interaktif membuat pengguna tidak mudah bosan, sehingga penyampaian materi menjadi lebih efektif [5].

Genre dari permainan yang diusulkan adalah *Third-Person Shooter* (TPS) dengan mempertimbangkan kepopuleran permainan ber-*genre* tersebut (misalkan *Free Fire* dan *Player Unknown Battle Ground* (PUBG)) dibandingkan *genre* lainnya [6]. TPS merupakan jenis permainan tembak-menembak dimana karakter yang dimainkan oleh pemain terlihat di layar selama permainan. Adapun Android dipilih sebagai *platform* sasaran dikarenakan jumlah pengguna perangkatnya jauh lebih banyak jika dibandingkan *platform* lain [7].

Permainan yang diusulkan bersifat *single player* sehingga kecerdasan buatan perlu untuk diterapkan pada karakter musuh agar permainan menjadi menarik dan lebih menantang [8]. Dengan kecerdasan buatan ini, karakter musuh pada permainan dapat mengejar karakter yang digerakkan oleh pemain dengan melalui rute yang optimal (terdekat).

Terdapat beberapa algoritma yang dapat digunakan dalam penentuan rute terdekat dari lokasi awal menuju tujuan, diantaranya *Dijkstra*, *A-star*, *Bellman-Ford*, *Floyd-Warshall*, dan *Dynamic PathFinding Algorithm* [8- 9] dimana yang paling populer adalah *Dijkstra* dan *A-Star*.

Algoritma *Dijkstra* umumnya diterapkan dalam menentukan rute terpendek antar dua titik pada peta sesungguhnya dimana peta ini diwakili dalam bentuk graf. Contoh penerapannya antara lain untuk penentuan rute terpendek lokasi tempat wisata [9-10], SPBU (Stasiun Pengisian Bahan Bakar Umum) [11], dan halte TransJakarta [12]. Algoritma *Dijkstra* juga dapat diterapkan topologi jaringan dan protokol *routing* seperti *Open Shortest Path First* (OSPF) [13]. Algoritma *Dijkstra* adalah jenis dari algoritma *greedy* dan *node* yang sudah ditelusuri tidak dapat ditelusuri kembali, sehingga terkadang rute yang dihasilkan bukan optimum secara global [14]. Selain itu, meskipun algoritma *Dijkstra* selalu dapat menemukan rute, namun proses komputasinya kompleks karena membandingkan biaya jalur satu dengan lainnya [15] sehingga membutuhkan waktu yang lebih lama jika dibandingkan dengan *A-star* [16].

Berbeda halnya dengan *Dijkstra*, algoritma *A-Star* biasanya diimplementasikan dalam mencari jalur terdekat dari titik awal ke titik tujuan pada peta permainan yang direpresentasikan dalam bentuk *grid* [8]. Contoh penerapan *A-Star* adalah seperti untuk NPC (*Non-Playable Character*) dalam

mencari rute terpendek pada lintasan di permainan balap mobil [8] dan menuju sebuah titik dalam permainan Snake 3D [17] dengan menghindari hambatan yang ada. Algoritma *A-Star* juga dapat digunakan sebagai bantuan dalam mencari rute terpendek ke titik *finish* di permainan labirin [18] sekaligus sebagai input dalam penyebaran titik-titik penempatan objek di peta permainan secara acak [19]. *A-Star* menggunakan heuristik dalam proses penentuan rute terpendeknya sehingga waktu komputasinya lebih singkat dibandingkan *Dijkstra* [14-15].

Dalam perkembangannya, beberapa modifikasi telah dilakukan terhadap algoritma *A-Star* seperti memadukannya dengan DPA [8]. Kombinasi ini memungkinkan mobil yang dikendalikan oleh komputer menjadi lebih baik dalam menghindari hambatan dinamis (bergerak) yang muncul di sepanjang permainan. Kombinasi ini namun lebih kompleks dibandingkan *A-Star* biasa sehingga waktu komputasinya sedikit lebih lama [8]. Selain itu, pada permainan yang diusulkan tidak terdapat hambatan dinamis sehingga *A-Star* standar akan lebih cocok untuk diterapkan pada musuh agar dapat mengejar pemain secara *real-time*.

Berdasarkan latar belakang yang telah diuraikan, terdapat beberapa hal penting yang perlu diperhatikan dalam mengenalkan bahaya narkoba melalui permainan yang diusulkan. Pertama, musuh-musuh di permainan berwujud narkoba yang jenisnya bervariasi dan dapat mengejar pemain dengan menggunakan algoritma *A-Star*. Kedua, pemain dapat menggerakkan karakter untuk membasmi musuh-musuh yang ada dengan kontrol layaknya *game* ber-genre TPS populer. Terakhir, permainan dapat menampilkan informasi detail mengenai jenis-jenis dan karakteristik narkoba, serta bahaya yang dikandungnya.

Permainan yang diusulkan diharapkan dapat menjadi media yang menarik, interaktif, serta efektif dalam menyampaikan bahaya penyalahgunaan narkoba kepada masyarakat Indonesia, khususnya generasi milenial. Selain itu juga diharapkan kinerja dari algoritma *A-Star* dalam menentukan rute optimal antara musuh-musuh yang ada menuju karakter pemain dapat dianalisa secara rinci dan dapat berjalan secara *real-time*.

2. METODOLOGI

2.1 Metode Pengumpulan Data

Metode pengumpulan data yang digunakan dalam penelitian ini adalah studi pustaka dimana buku-buku dan penelitian terkait jenis-jenis, karakteristik, dan bahaya dari narkoba dipelajari. Hingga saat ini, terdapat 73 jenis narkoba yang beredar di Indonesia dan 803 jenis narkoba beredar di dunia [3]. Pada penelitian ini hanya 7 diantaranya saja yang paling populer yang akan dibahas, seperti yang dapat dilihat di Tabel 1.

2.2 Algoritma *A-Star*

Algoritma *A-Star* adalah algoritma BFS (*Best First Search*) yang mengkombinasikan *greedy* BFS dengan *uniform cost search*. Fungsi heuristik digunakan *greedy* BFS untuk mengestimasi biaya dari titik awal menuju titik tujuan, sedangkan jarak terpendek dari titik awal menuju titik selanjutnya hingga titik tujuan dipilih oleh *uniform cost search* [20]. Algoritma *A-Star* akan mendapatkan jalur yang lengkap, yaitu solusi selalu didapatkan apabila memang ada, sekaligus optimal dengan heuristik. Selain itu, waktu komputasi *A-Star* cenderung cepat karena titik dengan nilai heuristik kecil (diperkirakan lebih dekat dengan tujuan) akan dievaluasi lebih dulu [16].

Adapun fungsi evaluasi dari algoritma *A-Star* dapat dihitung dengan Persamaan (1) dimana $f(n)$ adalah perkiraan total biaya rute melalui n ke tujuan. Sementara itu $g(n)$ merupakan biaya sejauh ini untuk mencapai n dan $h(n)$ adalah perkiraan biaya dari n ke tujuan [15].

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

Tabel 1 Jenis Narkoba dan Dampak Negatifnya[1]

No	Jenis	Dampak Negatif
1	Opium	Penurunan kesadaran, mengantuk, lesu, penglihatan kabur, dan sembelit
2	Morfin	Diabetes, gangguan hormon, tulang rapuh dan mudah patah, resiko infeksi lebih tinggi, dan disfungsi seksual
3	Sabu Sabu	Serangan jantung, tekanan darah tinggi, masalah mulut dan gigi, kerusakan otak, halusinasi, gangguan kecemasan, dan depresi
4	Kokain	Stroke, kejang-kejang, tremor, koma, ketergantungan, depresi, insomnia, gangguan seksual, serangan jantung, gangguan pencernaan dan pernapasan, dan gagal ginjal
5	Ganja	Mengganggu kemampuan berpikir, iritasi paru-paru, serta masalah konsentrasi dan daya ingat
6	Ekstasi	Depresi, konsentrasi menurun, depresi, mudah tersinggung, sulit tidur, penyakit jantung, sakau, dan overdosis
7	Heroin	Mulut kering, badan terasa berat, mual, kulit gatal, sembelit, gangguan pernapasan, kekebalan tubuh menurun, gangguan hati, mandul, gangguan seksual, gangguan saraf

Musuh pada permainan yang diusulkan dapat bergerak ke arah mana saja sehingga digunakan *euclidean distance* untuk menghitung nilai heuristik. *Euclidean distance* adalah jarak di antara dua titik pada ruang *euclidean* [9]. Persamaan (2) digunakan untuk menghitung nilai heuristik dengan *euclidean distance* dimana $h(n)$ adalah jarak antara 2 titik, $n.x$ dan $goal.x$ adalah koordinat sumbu x (horizontal) dari titik saat ini dan tujuan. Sementara itu, $n.y$ dan $goal.y$ adalah koordinat sumbu y (vertikal) dari titik saat ini dan tujuan [17, 20].

$$h(n) = \sqrt{(n.x - goal.x)^2 + (n.y - goal.y)^2} \quad (2)$$

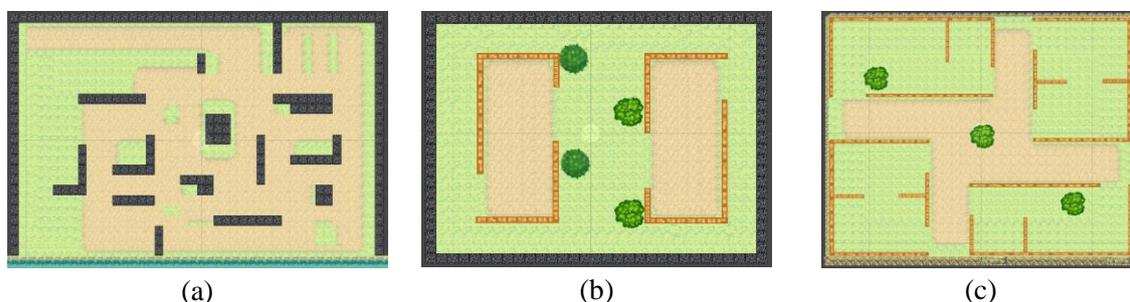
Dalam perhitungan *A-Star* terdapat istilah *open list* dan *closed list*. *Open list* berisikan data titik yang mungkin dilalui dari titik awal maupun titik saat ini. Sementara itu, *closed list* berisikan data titik sebelum titik saat ini yang juga merupakan bagian dari rute terpendek yang telah berhasil diperoleh [21]. Adapun alur proses dari algoritma *A-Star* dalam menentukan rute terpendek menuju titik tujuan adalah sebagai berikut [22]:

- a. Inialisasi *open list* dan masukkan titik awal
- b. Inialisasi *closed list*
- c. Selama *open list* tidak kosong:
 - 1) Temukan titik dengan f terkecil di *open list*, namakan q
 - 2) Keluarkan q dari *open list*
 - 3) Hasilkan 8 suksesor q dan atur *parent* menjadi q
 - 4) Untuk setiap suksesor:
 - a) Hentikan pencarian jika suksesor adalah tujuan
 - b) Jika titik dengan posisi sama dengan suksesor ada di *open list* dan f -nya lebih rendah dari suksesor, lewati suksesor ini
 - c) Jika titik dengan posisi sama dengan suksesor ada di *closed list* dan f -nya lebih rendah dari suksesor, lewati suksesor ini, jika tidak, tambahkan titik ke *open list*
 - 5) Tambahkan q ke *closed list*

2.3 Representasi Peta

Setiap permainan mempunyai peta yang dimanfaatkan sebagai daerah permainan itu berlangsung. Peta ini perlu direpresentasikan agar dapat digunakan oleh algoritma pencarian rute seperti *A-Star* dalam permainan [8]. Representasi peta yang digunakan dalam penelitian ini adalah representasi *grid*. *Grid* dimanfaatkan untuk membagi peta menjadi *cell* teratur yang umumnya berbentuk kotak. Dalam pemrograman, *grid* biasanya diwakili dengan *array* 2 dimensi dimana setiap *cell* ini memiliki nilai 1 atau 0 yang menandakan dapat dilalui atau tidaknya *cell* tersebut.

Adapun *grid* yang digunakan dalam permainan yang diusulkan berbentuk kotak dengan ukuran 25 baris dan 44 kolom dengan beberapa *cell* yang tidak dapat dilalui seperti yang ditunjukkan Gambar 1. Ukuran *grid* tidak terlalu besar dan juga tidak terlalu kecil dikarenakan terdapat 7 musuh komputer yang harus dapat mengejar pemain secara *real-time*. *Grid* yang berukuran terlalu besar akan meningkatkan waktu komputasi dan penggunaan *resource* perangkat secara signifikan [8] sehingga dikhawatirkan kinerja algoritma *A-Star* menjadi kurang baik.



Gambar 1. Peta Permainan yang Diusulkan (a) *Level 1* (b) *Level 2* (c) *Level 3*

3. PEMBAHASAN

3.1 Analisis Kebutuhan Pengguna

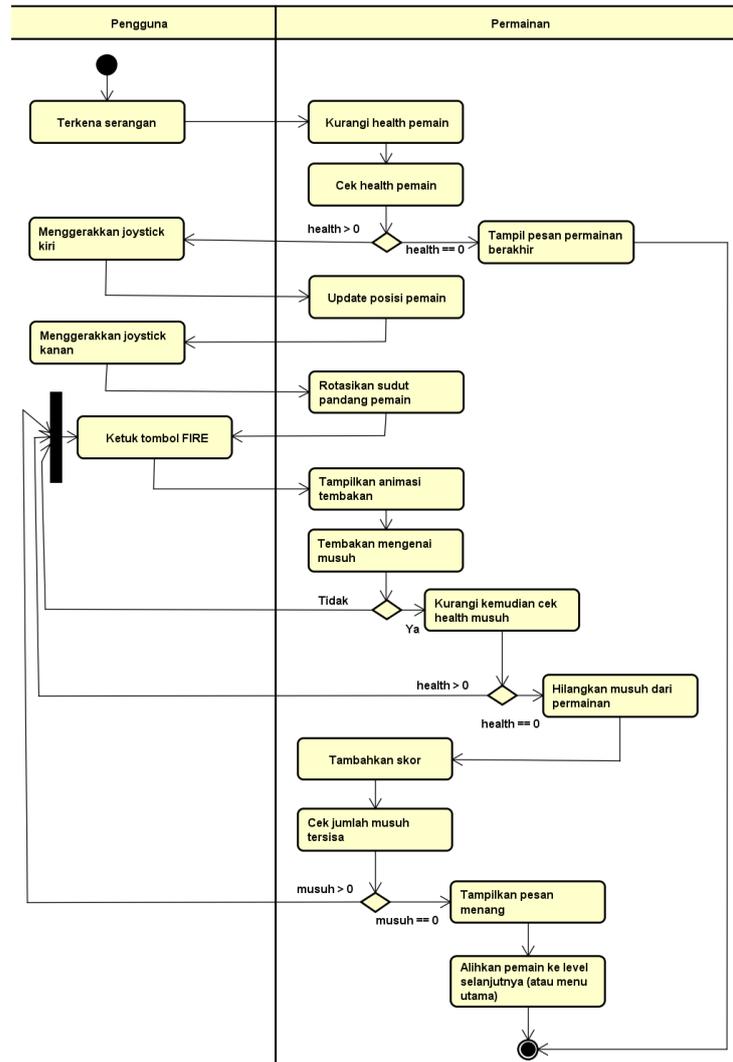
Analisis kebutuhan pengguna dari permainan usulan direpresentasikan dengan *use case diagram* yang ditunjukkan Gambar 2. Pengguna pada menu utama dapat melihat skor tertinggi, melihat informasi terkait jenis-jenis narkoba dan dampak negatifnya, serta keluar dari permainan. Selain itu saat permainan berlangsung, pengguna dapat menggerakkan karakter, merotasi sudut pandang karakter, menembak sekaligus terkena serangan musuh yang ada.



Gambar 2. *Use Case Diagram* Permainan Usulan

3.2 Analisis Sistem Usulan

Analisa *gameplay* dari permainan usulan direpresentasikan dengan *activity diagram* yang ditunjukkan Gambar 3 dimana *activity diagram* ini menunjukkan aliran *gameplay* dari permainan secara garis besar.



Gambar 3. Activity Diagram Gameplay dari Permainan yang Diusulkan

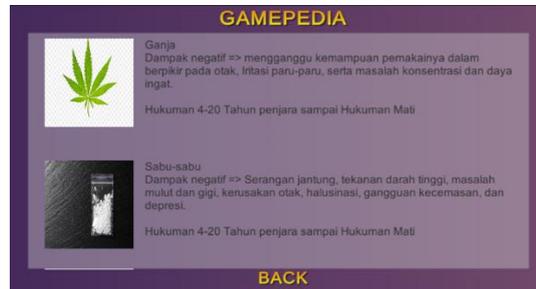
3.3 Tampilan Layar

Gambar 4 merupakan tampilan halaman permainan dari permainan yang diusulkan. Pada halaman ini terdapat beberapa objek seperti karakter pemain, karakter musuh, *bar* nyawa pemain, skor, jumlah musuh yang tersisa, tombol *joystick* kiri untuk menggerakkan karakter pemain, tombol *joystick* kanan untuk mengarahkan pandangan pemain, tombol *pause* untuk memberikan jeda, dan tombol *fire* untuk menembak musuh. Pada permainan ini terdapat 3 *level* dengan peta berbeda dimana semakin tinggi *level*, kecepatan perpindahan musuh akan semakin bertambah. Karakter musuh yang berjumlah 7 ini akan digerakkan secara otomatis menuju pemain dengan algoritma *A-Star*.



Gambar 4. Tampilan Halaman Permainan

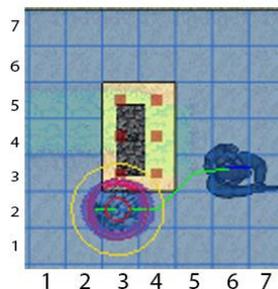
Gambar 5 merupakan tampilan halaman *gamepedia* yang berisikan informasi mengenai narkoba. Informasi ini berupa foto, nama, dampak negatif (lihat Tabel 1), serta hukuman penjara bagi pengguna narkoba tersebut.



Gambar 5. Tampilan Halaman Gamepedia

3.4 Analisis Kinerja Algoritma A-Star

Pada permainan ini algoritma *A-Star* digunakan oleh setiap musuh untuk menentukan rute pengejaran optimal antara lokasi musuh dengan lokasi pemain saat ini pada *grid* yang ada. Algoritma *A-Star* memerlukan *grid* x,y yang meliputi seluruh peta permainan. Setiap *cell* di *grid* ini menjadi titik (simpul) dan terbagi menjadi 2, yaitu titik yang bisa dilalui ataupun tidak bisa dilalui. Setelah itu rute pengejaran yang optimal akan ditentukan dengan melalui titik-titik yang dapat dilalui tersebut. Sebagai contoh, salah satu musuh berada di $cell(3,2)$ dan pemain berada di $cell(6,3)$ seperti yang ditunjukkan Gambar 6.



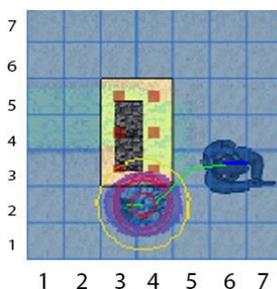
Gambar 6. Langkah ke-1 *Pathfinding* oleh *A-Star*

Berdasarkan Gambar 6, algoritma *A-Star* akan mengevaluasi *cell-cell* di sekitarnya dengan urutan searah jarum jam. Mula-mula masukkan $cell(4,2)$ ke dalam *closed list* dikarenakan $cell$ atas (3,3) dan kanan atas (4,3) dari $cell$ saat ini dilewati karena merupakan titik yang tidak dapat dilalui. Perhitungan untuk $cell$ yang titik koordinatnya semakin jauh dari $cell$ pemain seperti (2,3), (2,2), dan (2,1) juga akan dilewati karena pasti memiliki nilai estimasi yang lebih besar dari $cell$ lain. Mula-mula nilai $cell(4,2)$ yang dilalui jika musuh bergerak ke arah kanan sebanyak satu *cell* akan dihitung seperti yang ditunjukkan Tabel 2.

$Cell(4,2)$ berikut nilai f -nya lalu ditambahkan ke *open list* dan sekarang dilakukan perhitungan terhadap nilai $cell(4,1)$ dan (3,1) dimana kedua $cell$ ini berikut nilai f -nya ditambahkan ke *open list*. Pada $cell(4,1)$ nilai g ditambahkan 1.414 (hasil dari $\sqrt{(1)^2 + (1)^2}$) karena musuh bergerak diagonal. Hal ini berarti musuh memprioritaskan untuk bergerak horisontal atau vertikal dibandingkan diagonal jika nilai heuristiknya sama atau sedikit lebih buruk. Sekarang dari ketiga $cell$ yang ada di *open list* akan dipilih $cell$ dengan f terendah, yaitu $cell(4,2)$. $Cell$ ini kemudian dikeluarkan dari *open list* agar tidak dihitung kembali. Musuh saat ini berada di $cell(4,2)$ seperti yang ditunjukkan Gambar 7.

Tabel 2 Langkah ke-1 Perhitungan Nilai Cell dengan A-Star

Cell Saat Ini	Cell yang Dievaluasi	Nilai	Dipilih
(3,2)	(4,2)	$g = 0 + 1 = 1$ $h = \sqrt{(4 - 6)^2 + (2 - 3)^2} = \sqrt{5} = 2.236$ $f = 1 + 2.236 = 3.236$	(✓)
	(4,1)	$g = 0 + 1.414 = 1.414$ $h = \sqrt{(4 - 6)^2 + (1 - 3)^2} = \sqrt{8} = 2.828$ $f = 1.414 + 2.828 = 4.242$	
	(3,1)	$g(3,1) = 0 + 1 = 1$ $h(3,1) = \sqrt{(3 - 6)^2 + (1 - 3)^2} = \sqrt{13} = 3.606$ $f(3,1) = 1 + 3.606 = 4.606$	



Gambar 7. Langkah ke-2 Pathfinding oleh A-Star

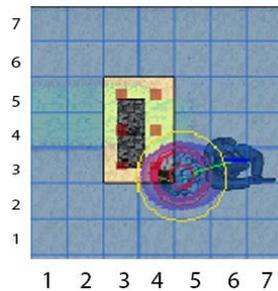
Ulangi langkah-langkah di atas untuk cell(5,3), (5,2), (5,1), dan (4,1) dimana hasil perhitungan keempat cell ini dapat dilihat pada Tabel 3. Cell yang sudah ada di closed list seperti cell(3,2) tidak dievaluasi kembali untuk menghindari perulangan tanpa titik akhir.

Tabel 3 Langkah ke-2 Perhitungan Nilai Cell dengan A-Star

Cell Saat Ini	Cell yang Dievaluasi	Nilai	Dipilih
(4,2)	(5,3)	$g = 1 + 1.414 = 2.414$ $h = \sqrt{(5 - 6)^2 + (3 - 3)^2} = \sqrt{1} = 1$ $f = 2.414 + 1 = 3.414$	(✓)
	(5,2)	$g = 1 + 1 = 2$ $h = \sqrt{(5 - 6)^2 + (2 - 3)^2} = \sqrt{2} = 1.414$ $f = 2 + 1.414 = 3.414$	
	(5,1)	$g = 1 + 1.414 = 2.414$ $h = \sqrt{(5 - 6)^2 + (1 - 3)^2} = \sqrt{5} = 2.236$ $f = 2.414 + 2.236 = 4.65$	
	(4,1)	$g = 1 + 1 = 2$ $h = \sqrt{(4 - 6)^2 + (1 - 3)^2} = \sqrt{8} = 2.828$ $f = 2 + 2.828 = 4.828$	

Berdasarkan Tabel 3, nilai awal dari g bernilai 1 dikarenakan cell saat ini (4,2) berjarak 1 cell dari titik awal (3,2). Cell (5,3), (5,2), dan (5,1) berikut nilai f-nya kemudian ditambahkan ke open list sedangkan nilai f dari cell(4,1) lebih buruk daripada f sebelumnya sehingga dilewati. Dari empat titik

yang ada di *open list*, dipilih titik dengan *f* yang terendah dan paling awal dihitung, yaitu *cell* (5,3). *Cell* (4,2) sudah ditelusuri sehingga dimasukkan ke *closed list*. Musuh saat ini berada di *cell* (5,3) seperti yang ditunjukkan Gambar 8.



Gambar 8. Langkah ke-3 *Pathfinding* oleh *A-Star*

Cell (5,3) kemudian dikeluarkan dari *open list* dan 8 suksesornya, yaitu *cell* di sekitarnya dibangkitkan lalu dihitung nilai *f*-nya seperti ditunjukkan Tabel 4. Saat mengevaluasi nilai *cell*(6,3), diketahui bahwa nilai heuristiknya adalah 0 atau dengan kata lain *cell* ini merupakan titik tujuan sehingga pencarian dihentikan karena solusi berupa rute terpendek sudah ditemukan. *Cell*(6,3) memiliki nilai *f* sebesar 3.414 yang adalah sama dengan nilai *f* dari *cell* *parent*-nya, yaitu *cell* (5,3) sehingga *A-Star* berhasil memprediksi bahwa *cell*(5,3) mengarah ke solusi yang ada. Nilai *f* akan berubah apabila terdapat hambatan atau titik yang tidak dapat dilalui. *Backtracking* kemudian dilakukan agar diperoleh rute pengejaran optimal mulai dari titik tujuan ke titik awal. *Parent* dari *cell*(6,3) adalah (5,3), *parent* dari *cell* (5,3) adalah (4,2), *parent* dari *cell*(4,2) adalah (3,2) dan *cell*(3,2) tidak memiliki *parent* karena adalah titik awal. Rute pengejaran optimal dari *cell*(3,2) menuju (6,3) adalah dengan melalui (4,2) dan (5,3) seperti yang ditunjukkan garis hijau pada Gambar 6.

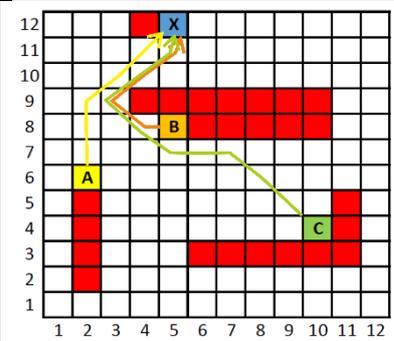
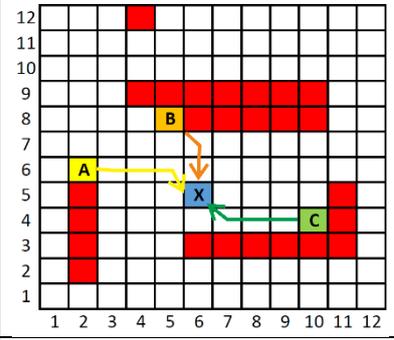
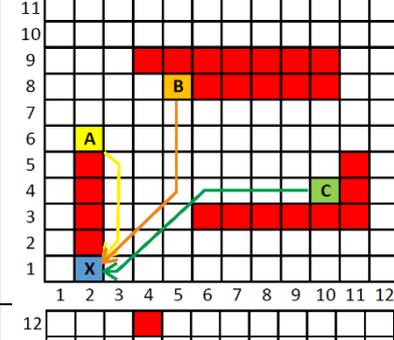
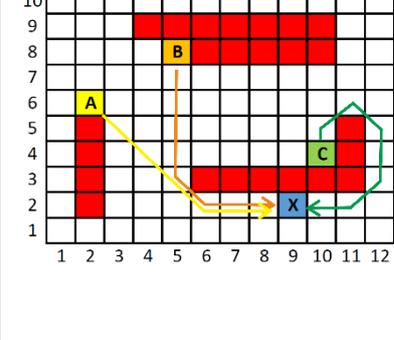
Tabel 4 Langkah ke-3 Perhitungan Nilai *Cell* dengan *A-Star*

Cell Saat Ini	Cell yang Dievaluasi	Nilai	Dipilih
(5,3)	(5,4)	$g = 2.414 + 1 = 3.414$ $h = \sqrt{(5 - 6)^2 + (4 - 3)^2} = \sqrt{2} = 1.414$ $f = 3.414 + 1.414 = 4.828$	
	(6,4)	$g = 2.414 + 1.414 = 3.828$ $h = \sqrt{(6 - 6)^2 + (4 - 3)^2} = \sqrt{1} = 1$ $f = 3.828 + 1 = 4.828$	
	(6,3)	$g = 2.414 + 1 = 3.414$ $h = \sqrt{(6 - 6)^2 + (3 - 3)^2} = \sqrt{0} = 0$ $f = 3.414 + 0 = 3.414$	(✓)

3.5 Pengujian Kinerja *A-Star*

Agar proses pengujian tidak terlalu kompleks, pengujian kinerja *A-Star* akan dilakukan pada *subgrid* berukuran 12 baris dan 12 kolom dari *grid* permainan *level* 1 sesungguhnya yang berukuran 25 baris dan 44 kolom. Pada *subgrid* ini terdapat *cell* yang tidak dapat dilalui (ditandai dengan warna merah). Selain itu, jumlah musuh yang ada pada *subgrid* adalah 3 dari 7 musuh dan posisi karakter pemain tidak dipindah selama pengujian berlangsung. Pengujian dilakukan sebanyak 4 kali dengan posisi awal musuh (A, B, C) adalah tetap dan posisi pemain (X) bervariasi. Hasil pengujian kinerja *A-Star* setiap musuh dalam menentukan rute optimal untuk mengejar pemain dapat dilihat pada Tabel 5.

Tabel 5 Hasil Pengujian Kinerja A-Star pada *Subgrid Level 1*

<i>Subgrid</i>	<i>Cell Pemain</i>	<i>Rute Pengejaran oleh Musuh</i>	<i>Jumlah Langkah</i>	<i>Ket.</i>
	(5,12)	A: (2,6) → (2,7) → (2,8) → (2,9) → (3,10) → (4,11) → (5,12) B: (5,8) → (4,8) → (3,9) → (4,10) → (5,11) → (4,12) C: (10,4) → (9,5) → (8,6) → (7,7) → (6,7) → (5,7) → (4,8) → (3,9) → (4,10) → (5,11) → (4,12)	6 5 10	Optimal Optimal Optimal
	(6,5)	A: (2,6) → (3,6) → (4,6) → (5,6) → (6,5) B: (5,8) → (6,7) → (6,6) → (6,5) C: (10,4) → (9,4) → (8,4) → (7,4) → (6,5)	4 3 4	Optimal Optimal Optimal
	(2,1)	A: (2,6) → (3,5) → (3,4) → (3,3) → (3,2) → (2,1) B: (5,8) → (5,7) → (5,6) → (5,5) → (5,4) → (4,3) → (3,2) → (2,1) C: (10,4) → (9,4) → (8,4) → (7,4) → (6,4) → (5,3) → (4,2) → (3,1) → (2,1)	5 7 8	Optimal Optimal Optimal
	(9,2)	A: (2,6) → (3,5) → (4,4) → (5,3) → (6,2) → (7,2) → (8,2) → (9,2) B: (5,8) → (5,7) → (5,6) → (5,5) → (5,4) → (5,3) → (6,2) → (7,2) → (8,2) → (9,2) C: (10,4) → (10,5) → (11,6) → (12,5) → (12,4) → (12,3) → (11,2) → (10,2) → (9,2)	7 9 8	Optimal Optimal Optimal

3.6 Keterbatasan Algoritma Usulan

Terdapat permasalahan pada algoritma *A-Star* yang telah diimplementasikan, yaitu tidak terdapat koordinasi antara karakter musuh satu dengan yang lain dalam mengejar dan mengepung pemain. Musuh-musuh yang ada memiliki kecenderungan untuk menuju dan berkumpul di satu titik

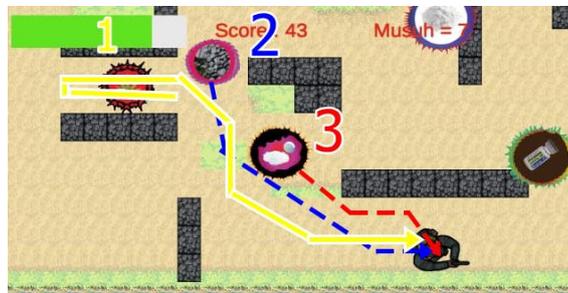
yang sama dengan melalui rute yang sama pula seperti yang ditunjukkan Tabel 5 dan Gambar 9. Hal ini membuat musuh mudah untuk dikalahkan dengan mengumpulkan mereka terlebih dahulu di satu rute sehingga membuat permainan menjadi kurang menantang.



Gambar 9. Musuh-Musuh Berkumpul di Satu Titik

Berdasarkan Gambar 9, musuh yang cerdas seharusnya dapat bekerja sama dan mempertimbangkan kondisi sekitar, misalkan beberapa musuh memilih rute pengejaran di sebelah kiri pemain meskipun lebih jauh. Rute ini namun akan lebih optimal karena dapat menutup rute pelarian karakter pemain dan mengharuskan pemain menembak di dua arah berbeda sehingga membuat permainan menjadi lebih sulit dan menantang.

Eksperimen untuk mengatasi keterbatasan ini sudah dilakukan, salah satunya dengan membuat *cell* yang ditempati musuh lainnya menjadi tidak dapat dilalui. Hal ini diharapkan dapat membuat musuh yang posisinya lebih jauh untuk memilih jalur alternatif. Strategi ini namun membuat musuh bingung dan melakukan gerakan yang tidak perlu seperti yang ditunjukkan garis kuning di Gambar 10 dengan kondisi karakter pemain tidak digerakkan.



Gambar 10. Rute Pengejaran Berbeda oleh Setiap Musuh

Berdasarkan Gambar 10, mula-mula musuh ke-1 bergerak ke kanan karena diprediksi rute tersebut optimal dan bisa dilalui. Musuh lainnya juga bergerak pada waktu yang sama dan musuh ke-2 menempati *cell* di rute pengejaran musuh ke-1 sehingga membuat *cell* tersebut tidak dapat dilalui. Hal ini membuat rute pengejaran musuh ke-1 berubah dan dengan mempertimbangkan posisi musuh ke-3 maka dipilihlah rute kiri. Setelah posisi musuh ke-2 cukup jauh dari posisinya semula seperti yang ditunjukkan Gambar 10, musuh ke-1 akan kembali mengambil rute kanan untuk mengejar pemain karena *cell* yang awalnya tidak dapat dilalui sudah dapat dilalui.

4. KESIMPULAN

Penyampaian informasi mengenai dampak negatif penyalahgunaan narkoba kepada pengguna dapat dilakukan secara menarik, interaktif, sekaligus efektif melalui permainan yang diusulkan. Hal ini dikarenakan materi yang tersirat pada media hiburan akan lebih cepat dan mudah untuk dikuasai karena penyampaiannya tidak membosankan.

Analisis secara mendetail sudah dilaksanakan terhadap kinerja dari algoritma *A-Star* dalam menentukan rute optimal antara posisi musuh-musuh yang ada menuju posisi karakter pemain. Secara keseluruhan kinerja dari algoritma *A-Star* sudah baik dimana rute yang dipilih adalah rute yang terdekat dan musuh dapat menghindari hambatan statis yang ada selama permainan. Selain itu, musuh dapat mengejar karakter pemain secara *real-time* tanpa *delay* bahkan pada ponsel *entry-level* berspesifikasi rendah seperti Redmi 3S Prime yang dirilis 5 tahun yang lalu. Algoritma *A-Star* yang diusulkan namun memiliki keterbatasan seperti musuh cenderung untuk memilih rute pengejaran yang sama dan tidak ada kerja sama antara musuh satu dengan lain dalam mengepung pemain.

Algoritma *A-Star* yang telah diusulkan dapat diperbaiki lagi misalkan dengan membuat *cell* yang ditempati oleh musuh lain menjadi tidak dapat dilalui hanya jika mereka berjarak kurang dari 6 *cell* dari *cell* pemain. Algoritma *A-Star* juga perlu untuk mempertimbangkan gerakan dan hasil perhitungan *A-Star* musuh lain agar dapat saling bekerja sama dan mengepung pemain dari berbagai posisi. Penggunaan *dynamic weighting A-Star* juga dapat dipertimbangkan agar musuh dapat memprioritaskan *cell* strategis di peta permainan. Perbaikan-perbaikan terhadap *A-Star* namun dapat meningkatkan waktu komputasi secara signifikan, sehingga perlu dikombinasikan dengan *iterative deepening* atau *bounded memory*. Hal ini dilakukan agar pengejaran oleh musuh tetap bisa dilakukan secara *real-time* meskipun harus dengan mengorbankan sedikit akurasi karena pencarian menjadi tidak terlalu dalam. Solusi alternatif lainnya adalah melakukan prakomputasi terhadap nilai *f* dari setiap *cell* sebelum permainan dirilis ke masyarakat. Hal ini dimungkinkan karena peta, hambatan, dan karakter-karakter yang ada di permainan bersifat statis.

DAFTAR PUSTAKA

- [1] Silalahi, D. H.. 2019. *Penanggulangan Tindak Pidana Penyalahgunaan Narkotika*. 1st ed. Medan: EnamMedia.
- [2] Arif, A.. 2020. Aplikasi Pengenalan Jenis Narkoba Berbasis Android Pada Badan Narkotika Nasional Kota Pagar Alam. *Indones. J. Comput. Sci.* **9:1** 53–64.
- [3] Pimpinan Pusat Gerakan Nasional Anti Narkoba. 2020. *Jagalah Dirimu dan Keluargamu dari Api Narkoba*. 1st ed. Jakarta: Majelis Ulama Indonesia.
- [4] Tumiwa K K, dkk. 2021. *Tetap Kreatif dan Inovatif di Tengah Pandemi Covid-19 (Jilid 2)*. 1st ed. Pekalongan: PT. Nasya Expanding Management.
- [5] Wibawanto, W.. 2020. *Game Edukasi RPG (Role Playing Game)*. 1st ed. Semarang: LPPM UNNES,
- [6] Ahdiyati M A, dan Irwansyah. 2018. Analisis Keterlibatan Komunitas dalam Industri Permainan Daring di Indonesia. *Interak. J. Ilmu Komun.* **7:2** 105–115.
- [7] Tommy L, Kirana C, dan Lindawati V. 2019. Recommender System dengan Kombinasi Apriori dan Content-Based Filtering pada Aplikasi Pemesanan Produk. *J. Teknoinfo*. **13:2** 84–95.
- [8] Sazaki Y, Satria H, Primanita A, dan Syahroyni M. 2018. Analisa Perbandingan Algoritma A* dan Dynamic Pathfinding Algorithm dengan Dynamic Pathfinding Algorithm untuk NPC pada Car Racing Game. *J. Teknol. Inf. dan Ilmu Komput.* **5:1** 95–103.
- [9] Umar R, Yudhana A, dan Prayudi A. 2021. Analisis Perbandingan Algoritma Dijkstra, A-Star, dan Floyd Warshall dalam Pencarian Rute Terdekat pada Objek Wisata Kabupaten Dompu. *J. Teknol. Inf. dan Ilmu Komput.* **8:2** 227–234.
- [10] Juniawan F P, dan Sylfania D Y. 2020. Penentuan Rute Terpendek Tujuan Wisata di Kota Toboali Menggunakan Algoritme Dijkstra Berbasis Web. *J. Teknol. Inf. dan Ilmu Komput.* **7:**

1 211–218.

- [11] Tommy L, dan Japriadi Y S. 2018. Implementasi Algoritma Dijkstra pada Aplikasi Pencarian Jalur Terpendek Lokasi SPBU di Pangkalpinang Berbasis Android. in *Konferensi Nasional Sistem Informasi*. 1018–1023.
- [12] Dewi R N, Atmodjo D W P, dan Purwaningsih M. 2020. Aplikasi Penentuan Rute dan Waktu Tempuh ke Halte Transjakarta Terdekat dengan Algoritme Dijkstra Berbasis Location Base System. *J. Teknol. Inf. dan Ilmu Komput.* **7:4** 653–660.
- [13] Musril H. A.. 2017. Penerapan Open Shortest Path First (OSPF) untuk Menentukan Jalur Terbaik dalam Jaringan. *J. Elektro dan Telekomun. Terap.* **4:1** 421–431.
- [14] Attamimi I, Yahya W, dan Hanafi M H.. 2017. Analisis Perbandingan Algoritma Floyd-Warshall dan Dijkstra untuk Menentukan Jalur Terpendek Pada Jaringan Openflow. *J. Pengemb. Teknol. Inf. dan Ilmu Komput.* **1:12** 1842–1849.
- [15] Wayahdi M R, Ginting S H N, dan Syahputra D. 2021. Greedy, A-Star, and Dijkstra's Algorithms in Finding Shortest Path. *Int. J. Adv. Data Inf. Syst.* **2:1** 45–52.
- [16] Zainulhayat L, dan Purwanto T H. 2017. Perbandingan Rute Optimum Hasil Perhitungan Algoritma Dijkstra dan A-Star untuk Sirkulasi Paket Jasa Ekspedisi JNE di D.I. Yogyakarta. *J. Bumi Indones.* **6:3** 1–10.
- [17] Maaruf K. C.. 2016. Kecerdasan Buatan Menggunakan Algoritma A Star (A*) dalam Permainan Ular Tangga (Snake 3D). in *Seminar Nasional Teknologi Informasi dan Multimedia*. 19–24.
- [18] Widodo W, dan Ahmad I. 2017. Penerapan Algoritma A Star (A*) pada Game Petualangan Labirin Berbasis Android. *Khazanah Inform. J. Ilmu Komput. dan Inform.* **3:2** 57–63.
- [19] Umami N A, Agustina I, dan Fauziah. 2018. Rancang Bangun Game Android Adventure Finding Diamond dengan Unity 3D Menggunakan Metode Dynamic Weighting A*. *JOINTECS (Journal Inf. Technol. Comput. Sci.)* **3:1** 45–50.
- [20] Hermanto D, dan Dermawan S.. 2018. Penerapan Algoritma A-Star Sebagai Pencari Rute Terpendek pada Robot Hexapod. *J. Nas. Tek. Elektro.* **7:2** 122–129.
- [21] Putra A B W, Rachman A A, Santoso A, dan Mulyanto. 2020. Perbandingan Hasil Rute Terdekat Antar Rumah Sakit di Samarinda Menggunakan Algoritma A*(star) dan Floyd-Warshall. *J. Sisfokom (Sistem Inf. dan Komputer)*. **9:1** 59–68.
- [22] Colls M G, dan Martínez de Osés F X.. 2016. A Ship Routing System Applied at Short Sea Distances. *J. Marit. Res.* **13:2** 3–6.