

**PATHFINDING PADA LINGKUNGAN STATIS BERDASARKAN
ARTIFICIAL POTENTIAL FIELD DENGAN FLOCKING
BEHAVIOR UNTUK NON-PLAYER CHARACTER FOLLOWER
PADAGAME**

Paulus Harsadi

Program Studi Teknik Informatika, STMIK Sinar Nusantara Surakarta

keshigomu.paul@gmail.com

Abstract

Artificial Intelligence in video games are an essential to provide a challenge to the players. One of them makes the character or Follower NPC (non-player character Follower) in video games such as human or animal behavior indeed. Among the many techniques in Artificial Intelligence, pathfinding is one of the popular techniques studied than the other. In own pathfinding is to avoid obstacles that is one of the main problem. Algorithm which solves this problem is Astarpathfinding, but it has a disadvantage in determining way in the large area. AstarPathfinding requires high computing time. In this research, the proposed algorithm which is able to handle the pathfinding process by using the function attractive potential field of Artificial Potential Field to perform pathfinding to the purpose and the flocking behavior techniques to avoid the obstacle. Based on the test results through environments static simulation, it can be gotten that the proposed algorithm has a faster computation time than the comparison algorithm, Astarpathfinding algorithm.

Kata Kunci: Artificial Intelligence, Game, pathfinding, Artificial Potential Field, Flocking behavior

I. PENDAHULUAN

*Artificial Intelligence (AI) di dalam video game merupakan hal penting untuk memberikan tantangan kepada pemain, salah satunya membuat karakter didalam video game seperti perilaku manusia atau hewan sesungguhnya (Ian Millington and John Funge, 2009). Beberapa masalah yang muncul dalam penerapan AI pada sebuah game meliputi *learning, planning, natural language processing*, dan *pathfinding*. *Pathfinding* merupakan salah satu yang paling banyak dan umum diteliti dari pada*

teknik yang lain dikarenakan *pathfinding* merupakan teknik dasar yang wajib dan paling banyak diaplikasikan dalam sebuah video game (Adi Botea, et al., 2013).

Penerapan *pathfinding* meliputi analisa sebuah peta untuk menemukan nilai terbaik dalam perjalanan dari satu titik ke titik yang lain. Lintasan terbaik disini dapat diartikan banyak nilai dan menggunakan kriteria seperti lintasan terpendek, nilai lintasan paling sedikit, serta lintasan yang aman (Peter Yap, 2002). Salah satu hal yang utama dalam *pathfinding* adalah menghindari halangan. Karena dengan kemampuan menghindari halangan maka kita akan mendapatkan hasil yang lebih baik dalam melakukan *pathfinding*.

Penelitian dalam metode pencarian khususnya *pathfinding* dalam video game sudah cukup banyak dilakukan (Adi Botea, et al., 2013). Algoritma Astar (P. Hart, et al., 1968), atau *Astar pathfinding* dan variasi perkembangan dari *Astar pathfinding* sudah sangat melekat dalam AI game dan merupakan metode yang paling banyak digunakan di video game (Peter I, et al., 2013).

Penelitian ini mengusulkan algoritma *pathfinding* untuk menghindari halangan statis memanfaatkan *flocking behavior* dari *boid* tetapi hanya digunakan sebagai metode untuk menghindari halangan, sedangkan untuk bergerak menuju ke tujuan yaitu *player* menggunakan fungsi *attractive potential field* yang dimiliki oleh *Artificial Potential Field*.

Algoritma *pathfinding* yang diusulkan akan di implementasikan ke *Non-player Character (NPC) Follower* pada video game agar pergerakan menuju ketujuan yaitu *player* bisa lebih maksimal. *NPC Follower* sendiri adalah karakter di game yang merupakan aliansi dari *player* atau musuh tetapi di jalankan oleh komputer.

Pembahasan paper ini disusun berdasarkan urutan sebagai berikut. Bagian 1 berisi pendahuluan. Bagian 2 menjelaskan rumusan masalah. Bagian 3 mengemukakan tujuan. Bagian 4 menguraikan metodologi yang digunakan. Bagian 5 eksperimen dan hasil dari eksperimen dilakukan. Bagian terakhir yaitu kesimpulan.

Penelitian sebelumnya dalam *pathfinding* yaitu menggunakan algoritma *Astar pathfinding* yang menggunakan nilai heuristic untuk menghitung bobot dalam setiap area di lingkungan game. *Astar pathfinding* merupakan metode pencarian dengan agen tunggal atau *single agent* yang menggunakan fungsi *heuristic* untuk memperkirakan nilai atau harga untuk perjalanan dari titik awal ke titik tujuan. *Astar pathfinding* memiliki kelemahan jika lingkungan pencarian yang digunakan cukup

luas Astar pathfinding mengalami “bottlenecks” atau ketersediaan memori untuk memproses tidak sesuai dengan kebutuhan dikarenakan tingkat waktu komputasi terlalu tinggi. Hal ini terjadi karena *Astar pathfinding* merupakan algoritma breadth-first search (Nathan R & Strurtevant, 2012).

Rumusan dari penelitian ini adalah sulitnya melakukan metode *pathfinding* yaitu pergerakan pada *non-player character (NPC) follower* dalam game dari satu tempat ketempat yang lain berdasarkan objek lain dan keadaan lingkungan dimana lingkungan adalah lingkungan statis yaitu sebuah lingkungan dengan halangan yang tetap baik posisi dan keberadaannya.

Tujuan dari penelitian ini adalah penerapan metode *pathfinding* pada *non-player character (NPC) follower* agar mampu bergerak dari satu tempat ketempat lain dimana lingkungan video game yang digunakan bersifat statis.

II. TINJAUAN PUSTAKA

2.1. Pathfinding

Pathfinding dalam *artificial intelligence model* terletak antara *decision making* dan *movement*. *Pathfinding* digunakan untuk membuat keputusan untuk bergerak oleh objek atau karakter dari satu tempat ke tempat lain berdasarkan keadaan peta dan object lainnya. *Pathfinding* sendiri dapat dibedakan menjadi dua, statis atau dinamis, yaitu *global path searching* yang mengacupada pencarian di dunia statis, dan *local path searching* pada lingkungan dinamis (Chen, Siyuan, et al., 2009).

2.2. Flock Behavior

Pergerakan dari sebuah kawanan dimana setiap anggota kawanan tersebut seakan-akan memiliki kecerdasan dalam bergerak dan berinteraksi dengan tetangganya dalam kawanan sering disebut sebagai *flocking behavior*. Boid merupakan salah satu konsep perilaku kawanan yang diusulkan oleh Reynolds (Reynolds, 1987) dan umum digunakan dalam bidang game dan animasi karena konsep yang diusulkan tergolong sederhana tetapi menghasilkan simulasi kawanan yang begitu natural.

2.3. Artificial Potential Field

Artificial Potential Field (APF) pertama kali digunakan dalam bidang robot path planning pada tahun 1986 oleh Khatib (Khatib, 1986), dimana

ide dasarnya adalah membentuk gaya virtual yang di formulasikan dalam lingkungan robot.

Metode ini menempatkan robot dalam lingkungan sebagai robot yang bergerak berdasarkan medan gaya virtual yang dibuat secara manual dimana halangan (obstacle) akan menghasilkan gaya tolak atau repulsive force pada robot sedangkan tujuan (goal) menghasilkan gaya tarik atau attractive force. Total medan gaya yaitu repulsive force dan attractive force mampu mengontrol robot untuk bergerak menghindari halangan dan mencapai tujuan atau target (Li Zhou & Wei Li, 2014).

III. METODE PENELITIAN

Metode penelitian yang dilakukan pada penelitian ini adalah metode penelitian eksperimen, tahapannya yaitu:

1. Perancangan

Tahap perancangan akan terdiri dari perancangan static environment, Non-player character Follower dan player dan algoritma pathfinding yang diusulkan.

2. Implementasi

Tahap ini akan melakukan implementasi dari perancangan yang dilakukan menggunakan Unity 3D game engine. Unity 3D merupakan aplikasi pembuatan game yang memiliki dukungan platform banyak dan bersifat gratis.

3. Pengujian

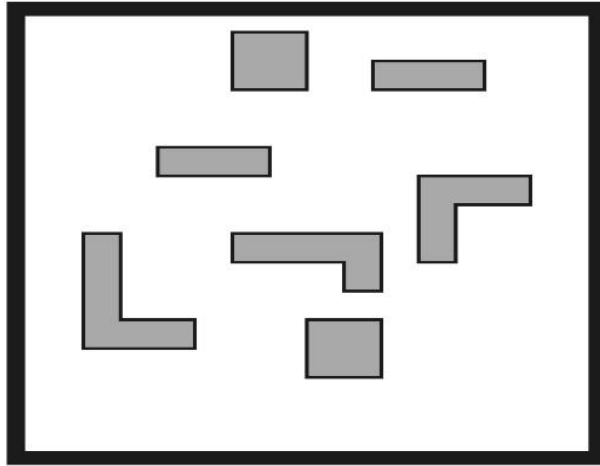
Pengujian dalam penelitian ini menggunakan pengujian waktu tempuh karena pengukuran *pathfinding* sendiri adalah kecepatan waktu tempuh dari titik awal sampai ke titik akhir.

IV. HASIL DAN PEMBAHASAN

4.1. PERANCANGAN

4.1.1. Static environment

Simulasi *pathfinding* akan dilakukan pada lingkungan statis dimana akan diwujudkan dalam bentuk sebuah peta dalam video game. Perancangan static environment dapat dilihat pada Gambar 1.



Gambar 1. Lingkungan dengan halangan kompleks

4.1.2. Karakter *NPC follower* dan *player*

Perancangan *NPC follower* dan *player* nantinya menggunakan 3d karakter *soldier* yang memegang senjata dengan bentuk *avatar* sebagai berikut :



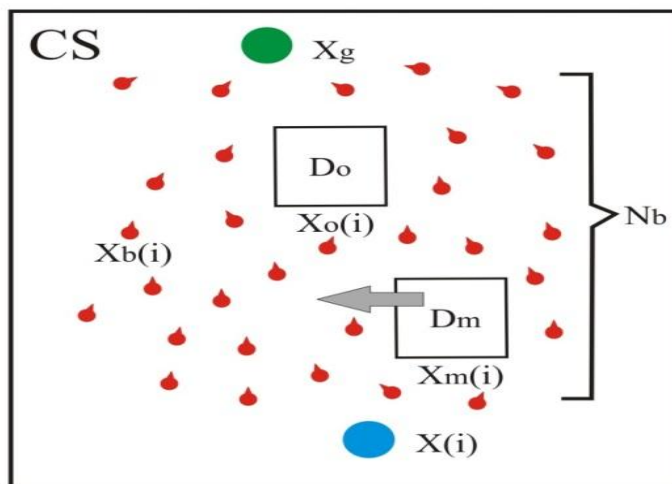
Gambar 2. Avatar Soldier untuk *Non-player Character* dan *player*

4.1.3. Algoritma *pathfinding*

Perilaku kawanan *boid* (*boid flocking behavior*)

Penelitian ini mengasumsikan bahwa kawanan dengan sejumlah variabel *boid* N_b . Pada waktu tertentu dalam proses yaitu i , *boid* memiliki posisi $X_b(i)$ dengan $k \in I_{N_b} \equiv \{1, 2, \dots, N_b\}$. *Non-player character Follower* terletak pada posisi $X(i)$ dan posisi target yaitu $Player X_g$. Pada waktu yang dipilih i kawanan didistribusikan melalui R^2 (Jarak ke target). Di ruang tersebut ada distribusi D_m sebagai halangan dinamis dan menetapkan posisinya sebagai $X_m(i)$ dengan $m \in I_{N_m} \equiv \{1, 2, \dots, N_m\}$.

Kemudian domain D_o sebagai halangan statis dan X_o sebagai posisi halangan statis. Ruang konfigurasi (CS) adalah area R^2 setelah dikurangi domain D_o dan D_m sedemikian rupa sehingga *boid* dan *NPC Follower* dianggap sebagai titik bergerak di CS. *NPC Follower* ketika proses berlangsung dan para *boid* dianggap sebagai partikel bergerak.

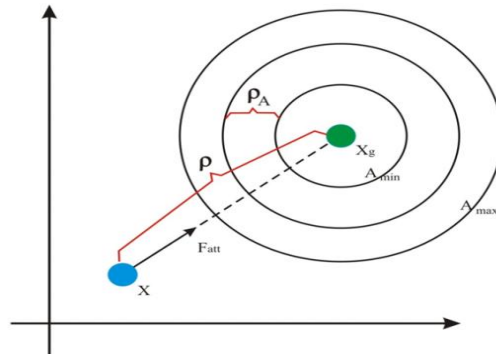


Gambar 3. Pergerakan kawanan *boid*

Dari 3 peraturan dasar yaitu *alignment*, *separation*, dan *cohesion* kemudian penambahan 2 peraturan yaitu *obstacle avoidance* dan *goal attraction*, maka titik untuk *pathfinding* dapat dihasilkan dari pergerakan setiap *boid* dalam menghindari rintangan dan menuju ke target. Hasil titik tersebut dapat digunakan oleh *Artificial Potential Field* untuk menghindari halangan secara dinamis menuju ke target.

Improved attractive potential field untuk pathfinding

Metode *Artificial Potential Field (APF)* dalam penelitian ini yang digunakan hanya fungsi *attractive potential field*. *Attractive potential field* dalam penelitian ini digunakan sebagai gaya tarik yang dihasilkan oleh target yaitu *player* kepada *non-player character (NPC) follower*. Gaya tarik ini akan semakin besar menarik *NPC follower* ketika jarak antara *NPC follower* dan *player* atau tujuan semakin jauh dalam lingkungan game.



Gambar 4. Model *Improved Attractive Potential Field*

Fungsi *attractive potential field* yang diadaptasi dari robot di definisikan sebagai berikut (Changfan Zhang et al., 2014) :

$$U_{att}(X) = \frac{1}{2} k_1 \rho^2 (X, X_g) \quad (1)$$

Dimana :

- X = (x,y) adalah vector posisi *NPC follower*
- X_g = Lokasi *player*
- U_{att} = Nilai *Attractive potential field*
- $\rho(X, X_g)$ = Jarak antara *NPC follower* dengan *player*
- k_1 = Konstanta *attractive potential field* (nilai positif)

Gaya *attractive* F_{att} adalah negatif gradien dari U_{att} di definisikan sebagai berikut (Changfan Zhang et al., 2014) :

$$F_{att} = -\nabla[U_{att}(X)] = k_1 \rho(X, X_g) \quad (2)$$

Dimana :

F_{att} = gaya *attractive*

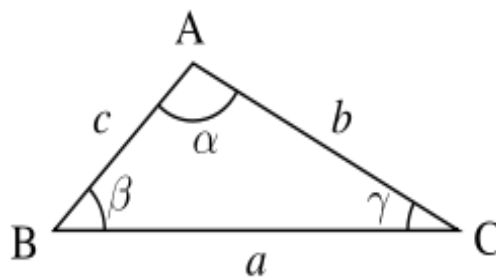
$\rho(X, X_g)$ = Jarak antara *NPC follower* dan *player*

k_1 = Konstanta *attractive potential field* (nilai positif)

Obstacle avoidance

NPC follower dalam proses *obstacle avoidance* atau menghindari halangan yang ada baik statis maupun dinamis dalam metode ini tidak memanfaatkan gaya *repulsive* dari *Artificial potential field* tetapi dengan memanfaatkan posisi dari pergerakan kawanan *boid*.

Hukum kosinus adalah salah satu hukum yang diterapkan dalam segitiga atau ilmu *trigonometry* yang dapat digunakan untuk menentukan nilai baik sudut, maupun panjang sisi dalam sebuah segitiga (Hazewinkel & Mitchiel, 2001).



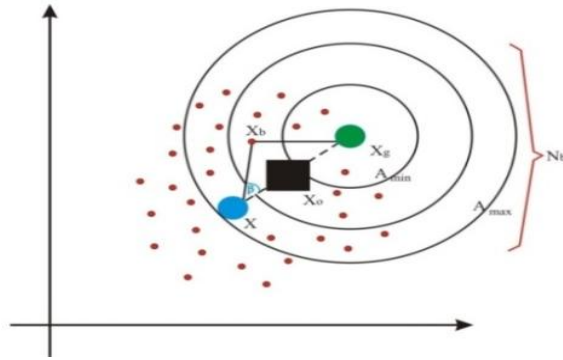
Gambar 5. Notasi sisi dan sudut dalam suatu segitiga

Berdasarkan notasi pada gambar 3, maka dapat diasumsikan bahwa titik B adalah posisi *NPC follower*, titik A adalah posisi *boid*, dan titik C adalah posisi *player*.

Berikut aturan kosinus untuk menghitung sudut β [12] :

$$\cos \beta = \frac{a^2 + c^2 - b^2}{2ac} \quad (3)$$

Aturan cosinus yang digunakan adalah penentuan sudut β . *Boid* dalam area akan menghitung nilai sudut β berdasarkan posisi *NPC follower* dan *player* kemudian akan diurutkan berdasarkan nilai sudut β .



Gambar 6. Perhitungan sudut β antara *Boid*, *NPC Follower* dan *Player*

Semakin besar sudut β , maka posisi *boid* semakin menjauh dengan halangan sehingga bisa dipakai *NPC follower* sebagai tujuan sementara menghindari halangan, setelah itu maka gaya *attractive* dapat diterapkan kembali.

4.2. IMPLEMENTASI

4.2.1. Implementasi *flocking behavior*

Implementasi *boid* yaitu dengan membuat sebuah *game object* berupa *prefab* lingkaran pipih. Setiap aturan dan *variable* dalam *boidikan* diberikan nilai sehingga terbentuk *flocking behavior* sesuai dengan yang diinginkan. Dalam penelitian ini nilai aturan tersebut adalah :

- | | |
|-----------------------------------|------|
| a. Speed / Kecepatan | : 3 |
| b. Max Speed / Maksimal Kecepatan | : 5 |
| c. Separation | : 6 |
| d. Aligment | : 2 |
| e. Cohesion | : 5 |
| f. Force Obstacle Avoidance | : 25 |

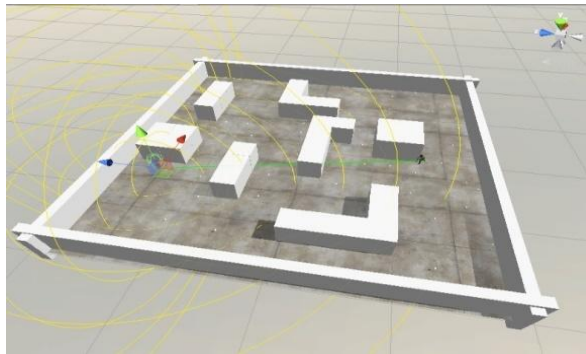
Area sebaran untuk *boid* sendiri dibatasi dengan area yang disebut dengan *Swarm radius* sehingga tidak keluar dari lingkungan dinamis. Untuk nilai lebar dan panjangnya di tentukan dengan nilai *swarm bounds*. Nilai *spwan radius* dalam penelitian ini adalah 15 sedangkan nilai *swarm bounds* 35 x 35.

4.2.2. Implementasi *pathfinding* pada *NPC follower*

Proses implementasi algoritma *pathfinding* dalam penelitian ini yaitu, pertama berdasarkan nilai radius dan range *potential field* yang berada di *script potential field* yang di implementasikan ke *player*, maka *boiddan NPC follower* akan digolongkan berdasarkan area masing – masing sesuai range nilai gaya *attractive potential field*-nya.

Kedua, pada *script FollowBehaviorPF* yang diimplementasikan ke *NPC follower*, gaya *attractive potential field* dihitung untuk melakukan pergerakan. Jika *NPC follower* menemukan halangan ketika melakukan pergerakan menuju tujuan yaitu *player* maka *NPC follower* akan menggunakan *boid* dalam area *NPC follower* berada untuk memilih *boid* sebagai perubahan lintasan menghindari halangan.

Hasil pada implementasi lingkungan statis, dapat dilihat pada Gambar 7.



Gambar 7. Implementasi lingkungan game

4.3. PENGUJIAN

Pengujian dilakukan terhadap algoritma yang diusulkan yaitu *Improved attractive potential field* (IAPF) dengan perbandingan algoritma *Astar pathfinding*. Untuk melihat hasil yang akurat maka dilakukan sebanyak 50 kali untuk melihat rata-rata dari kedua algoritma tersebut. Berikut hasil rata-rata implementasi dalam satuan detik.

Tabel 1. Hasil pengujian lingkungan game

Astar Pathfinding	IAPF
9.067002	7.653611

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan penelitian dalam *pathfinding* yang telah dilakukan diatas dapat diambil kesimpulan sebagai berikut :

- a. Penelitian ini dilakukan untuk mengurangi waktu komputasi dalam permasalahan *pathfinding* dengan penerapan *Artificial Potential Field (APF)* yaitu fungsi *attractive potential field* saja disebut *Improvedattractive potential field* untuk bergerak ketujuan dan memanfaatkan pergerakan kawanan *boid* untuk menghindari halangan dalam lingkungan statis, sehingga dapat mempercepat waktu untuk menuju ketujuan.
- b. Simulasi diterapkan dalam lingkungan game dan hasil dari pengujian didapat bahwa algoritma *pathfinding* yang diusulkan lebih unggul dibanding dengan algoritma *AStar pathfinding*, unggul selisih 1.4 detik.

5.2 Saran

- a. Penggunaan fungsi *attractive potential field* dan *flocking behavior* yaitu *boid* dapat dikembangkan tidak hanya untuk lingkungan yang bersifat statis tetapi bersifat dinamis sehingga interaksi dengan objek lain dalm game lebih terlihat nyata.
- b. Algoritma yang diusulkan bisa di kembangkan untuk group NPC follower, sehingga tidak hanya satu *NPC follower* saja yang melakukan *static pathfinding*.

DAFTAR PUSTAKA

- Adi Botea, Bruno Bouzy, Michael Buro, Christian Bauckhage, and Dana Nau. 2013. *Pathfinding in Games*. Dags tuhl Follow- Ups, Vo.6, pp.21-31.
- Changfan Zhang, Mangang Niu, Han Wu, Jing He, Yuanyuan Xiao. 2014. A Multi-motor Synchronous Control Algorithm for Artificial Potential Field with Adjacent Attractive Force. *26th Chinese Control and Decision Conference (CCDC)*. pp.820-825.
- Chen, Siyuan, et al. 2009. Fast Path Searching in Real Time 3D Game. *Global Congress on Intelligent Systems*
- C.W. Reynolds. 1987. Flocks, herds and schools: a distributed behavioral model. *ACM SIGGRAPH Computer Graphics*, pp. 25–34.

- Hazewinkel, Michiel, ed. 2001. *Cosine theorem*, Encyclopedia of Mathematics, Springer.
- Ian Millington and John Funge. 2009. *Artificial Intelligence for Game : second editon*. Morgan Kaufmann Publishers.
- Khatib. 1986. Real-time Obstacle Avoidance for Manipulators and Mobile Robots. *International Journal of Robotics Research*. Vol.5,No.1, pp.90-98.
- Li Zhou, Wei Li. 2014. Adaptive Artificial Potential Field Approach for Obstacle Avoidance Path Planning. Seventh International Symposium on Computational Intelligence and Design, IEEE. pp.429 -432.
- Nathan R. Sturtevant. 2012. Moving Path Planning Forward. MIG 2012, LNCS 7660. pp. 1–6.
- Peter I. Cowling, Michael Buro, Michal Bida, Adi Botea, Bruno Bouzy, Martin V. Butz, Philip Hingston, bHéctor Muñoz-Avila, Dana Nau, and Moshe Sipper. 2013. *Search in Real-Time Video Games*. Dags tuhl Follow- Ups, Vo.6, pp. 1-19.
- Peter Yap. 2002. Grid-Based Path-Finding, *R. Cohen and B. Spencer (Eds) : AI 2002*. Springer. pp. 44-55.
- P. Hart, N. Nilsson, and B. Raphael. 1968. A formal basis for the heuristic determination of minimum cost paths, *IEEE Transactions on System Sciences and Cybernetics*. SSC-4(2):100–107.