IMPLEMENTATION OF BOIDS ALGORITHM FOR AGENT FORMATION IN WAR GAMES

Ryan Ardian

Teknik Informatika – Universitas Komputer Indonesia
Jl. Dipatiukur 112-114 Bandung
Jalan Dipati Ukur No. 144
Email : amaliah.iput@gmail.com

ABSTRAK

Perang tidak pernah lepas dari formasi, formasi merupakan bagian terpenting dalam strategi perang. *Game* yang mengangkat tema peperangan sudah banyak dibuat, pada *game* sejenis penerapan formasi hanya kepada NPC saja sedangkan agen (*player*) tidak memiliki perilaku khusus, yaitu agen (*player*) hanya bergerak secara tunggal tanpa adanya pasukan sehingga tidak dapat membentuk suatu formasi.

Algoritma *Boids* merupakan sebuah algoritma hasil dari simulasi gerakan berkelompok. Algoritma *boids* didefinisikan sebagai suatu jenis algoritma yang mempresentasikan gerak sebuah kawanan burung. Algoritma *boids* memiliki 3 aturan yaitu, *separation*, *alignment* dan *cohesion*. Berdasarkan aturannya algoritma *boids* cocok diimplementasikan untuk membuat formasi

Pengujian *game* ini menggunakan pengujian *blackbox* dan *whitebox* yang menguji fungsionalitas dari simulasi dan pengujian algoritma *boids* terhadap agen. Hasil pengujian dari *game* ini menunjukan bahwa semakin tinggi nilai inputan pada setiap parameter maka semakin optimal formasi yang terbentuk.

Kata kunci: Algoritma, *boids*, *game*, formasi.

1. PENDAHULUAN

1.1 Latar Belakang Masalah

Boids didefinisikan sebagai suatu jenis algoritma yang mempresentasikan gerak sebuah kawanan burung. *Boids* bereaksi hanya untuk *flockmates* dalam lingkungannya sendiri. Lingkungan ditandai dengan jarak (diukur dari pusat *boids*) dan sudut (diukur dari arah asal *boids*). Karakteristik dasar algoritma *boids* adalah gerak *boids* dihasilkan dari tiga aturan yaitu *cohesion* (pemusatan), *alignment* (penyesuaian kecepatan) dan *separation* (pemisahan) [1].

Permainan tenis meja atau yang lebih dikenal deng Perang tidak pernah lepas dari formasi, formasi merupakan bagian terpenting dalam strategi perang. *Game* yang mengangkat tema peperangan sudah banyak dibuat, salah satunya yaitu *game* "*Space Arcade",* pada *game* "*Space Arcade*" NPC pada game "*space arcade*" dapat membentuk formasi sedangkan agen (*player*) tidak memiliki perilaku khusus, yaitu tidak dapat membentuk formasi. *Game* peperangan pada umumnya menerapkan perilaku khusus kepada NPC saja sedangkan agen (player) hanya bergerak secara tunggal tanpa adanya pasukan sehingga tidak dapat membentuk suatu formasi. Formasi dapat meningkatkan kinerja agen dalam penyerangan, formasi yang nantinya akan digunakan sebagai simulasi yaitu formasi sejajar, formasi sejajar merupakan jenis formasi yang umum digunakan dalam strategi perang, formasi ini optimal dalam penyerangan karena seluruh agen yang terlibat menyerang secara bersamaan. Formasi perang dapat dibentuk dengan memberikan suatu kecerdasan buatan (*Artificial Intelligence*) pada agen dan algoritma yang cocok untuk membuat formasi pasukan perang yaitu algoritma *boids*. Selain dapat membentuk suatu formasi perang, Algoritma *boid* juga dapat member perilaku kepada agen agar dapat melewati rintangan yang nantinya akan diimplementasikan sehingga dapat meminimalisir agen yang gugur.

Berdasarkan dari permasalahan yang telah dipaparkan, beberapa studi literatur, maka diperlukan implementasi algoritma *boids* terhadap kasus pembentukan formasi perang untuk agen pada *game* perang..

1.2 Perumusan Masalah

Berdasarkan latar belakang masalah maka dirumuskan sebuah masalah yaitu bagaimana memberikan kemampuan terhadap agen membentuk barisan formasi perang menggunakan algoritma *boids*..

1.3 Maksud dan Tujuan

Berdasarkan permasalahan yang diteliti, maka maksud dari penelitian tugas akhir ini adalah mengimplementasikan algoritma *boids* untuk formasi agen pada *game* perang.

Tujuan yang akan dicapai dalam implementasi algoritma *boids* untuk formasi agen pada *game* perang terdiri dari :

1. Memberikan kemampuan pada agen untuk membentuk formasi berdasarkan parameter yang diinputkan
2. Memberikan kemampuan kepada agen untuk menghindari rintangan

1.4 Batasan Masalah

Batasan masalah dalam implementasi algoritma *boids* untuk formasi agen pada game perang adalah sebagai berikut :

1. Formasi perang hanya pada agen (*player*)
2. Formasi yang akan diimplementasikan adalah formasi sejajar.
3. Terdapat 3 parameter : *Cohesion*, *Alignment* dan *Separation.*
4. Data keluaran yang dihasilkan berupa pergerakan agen dalam formasi dan perilaku agen dalam melewati rintangan.
5. Pemodelan perancangan sistem menggunakan pemodelan UML.

1.5 Metodologi Penelitian

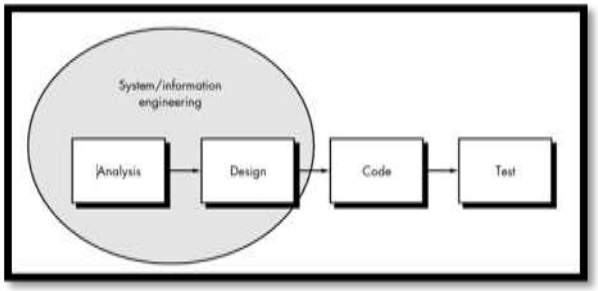
Metode penelitian yang digunakan adalah metode deskriptif. Metode penelitian ini mempunyai dua metode yaitu :

1. Metode Pengumpulan Data.
2. Metode Pengembangan Perangkat Lunak.

Metode pengumpulan data yang digunakan untuk mengumpulkan informasi dalam implementasi algoritma *boids* untuk formasi agen pada *game* praeng ini adalah sebagai berikut :

a. Studi Literatur
b. Observasi

Metode pembangunan perangkat lunak yang digunakan adalah model *waterfall* yang bisa dilihat pada gambar 1.



Gambar 1 Ilustrasi Model Waterfall [2]

2. ISI PENELITIAN

2.1 Landasan Teori

1. Algoritma merupakan salah satu cabang ilmu komputer yang membahas prosedur penyelesaian suatu permasalahan. Algoritma adalah urutan langkah-langkah dalam menentukan suatu masalah. Algoritma juga dapat didefinisikan dengan deretan langkah komputasi yang mentransformasikan masukan (*input*) menjadi keluaran (*output*). Jadi algortima adalah serangkaian urutan langkah-langkah atau prosedur untuk menyelesaikan suatu masalah dengan memproses nilai masukan menjadi nilai keluaran.
   Algoritma dapat ditulis dalam bentuk *pseudocode* ataupun *flowchart*. *Pseudocode* berasal dari kata *pseudo* yang artinya mirip atau menyerupai dan *code* berarti kode sehingga *pseudocode* adalah kode yang menyerupai kode program sebenarnya. Sedangkan *flowchart* merupakan bentuk gambar atau diagram yang mempunyai aliran satu atau dua arah secara sekuensial [3].
2. Boids merupakan sebuah algoritma hasil dari simulasi gerakan berkelompok yang membentuk suatu barisan dengan gerak yang elegan. *Boids* bereaksi hanya untuk *flockmates* dalam lingkungannya sendiri. Lingkungan ditandai dengan jarak (diukur dari pusat *boids*) dan sudut (diukur dari arah asal *boids*), sedangkan *flockmates* di luar lingkungan akan diabaikan. Pada pertengahan tahun 1980 Craig Reynolds menerapkan prinsip-prinsip *A-Life* untuk fenomena kawanan burung terbang yang terkoordinasi. Tantangannya adalah untuk mengungkap aturan sederhana bahwa setiap burung (atau *boid*) bias mengikuti gerakan berkelompok dengan sempurna sebagai sebuah perilaku. *Boids* memiliki tiga aturan, yaitu *separation, cohesion,* dan *alignment* [4].
   1. Separation
      Tiap anggota ke tiap anggota kelompok cenderung menghindari tubrukan dengan kelompoknya. Tujuan ini disebut *separation* atau *collision avoidance*. Ini mengarahkan *boids* untuk menghindari *overcrowding* pada *flockmate* lokal Jika gaya untuk mengarahkan/mengemudikan ini diterapkan, maka *flock* akan menghilang. Solusi sederhana untuk gaya ini adalah berbelok saat bertemu dengan agen yang lainnya.
   2. Cohesion
      Cohesion, atau pemusatan *flock*, merupakan gaya kemudi yang memindahkan *boids* menuju pusat *flock* yang terlihat. Tindakan ini melengkapi pemisahan / jika kemudi ini diaplikasikan, *flockmates* akan bersatu kedalam posisi tunggal. *Cohesion* dihitung dari pusat yang terlihat secara langsung. Pusat ini terhubung kepusat kepadatan dari semua *boids* yang terlihat
   3. Alignment
      Steering force terakhir, *alignment* atau penyesuaian kecepatan (*velocity)*, penting bagi *boids* untuk mengikuti gaya dorong. *Boids* menjaga agar

tetap sejajar dengan kecepatan *flockmate*-nya. Laju kecepatan adalah ukuran dari vektor. Oleh karena itu, *boids* akan mempercepat atau memperlambat secara otomatis tergantung *flockmate*-nya. Jika percepatan *boids* terlalu banyak maka akan memindahkan *visibility flockmates* dan keluar secepatnya.

3. ANALISIS DAN PERANCANGAN

3.1 Analisis Masalah

Implementasi algoritma *boids* untuk formasi agen pada *game* perang memiliki tujuan yaitu membentuk suatu formasi sehingga gerak dari agen menjadi beraturan, umumnya pada *game* perang lainnya agen (*player*) hanya bertindak sendiri tanpa adanya pasukan ataupun agen (*player*) lebih dari satu namun pergerakannya tidak beraturan dan tidak membentuk suatu formasi. Algortima *boids* disamping mampu membentuk suatu formasi juga memiliki karakteristik lainnya yaitu dapat memberikan perilaku kepada agen (*player*) agar dapat menghindar dari rintangan, agen (*player*) akan menghindari rintangan pada saat bergerak secara beraturan, agen akan berpisah dari formasi awal pada saat melewati rintangan dan akan kembali membentuk suatu formasi ketika rintangan sudah berhasil dilalui.

3.2 Analisis Masukan

Analisis masukan yang dilakukan dalam penelitian ini adalah menganalisis parameter dalam implementasi algoritma *boids* untuk formasi agen pada *game* perang, parameter tersebut antara lain jarak dan kecepatan.

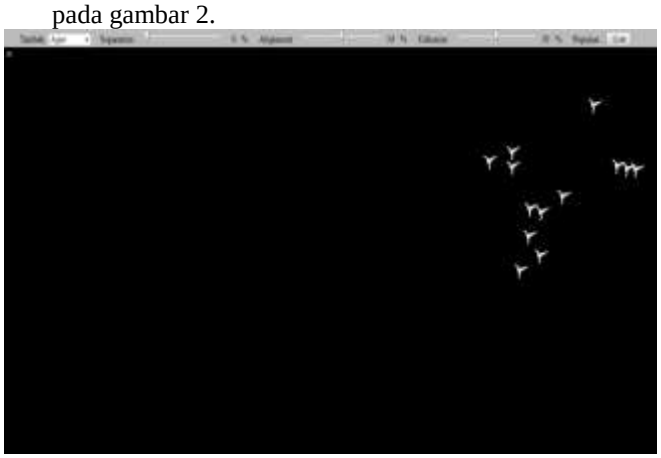
1. Parameter Jarak
   Agen (*player*) bergerak menyesuaikan jarak agar tetap terjaga dengan agen lainnya sehingga tidak akan terjadi tabrakan antar agen. Satuan jarak antar agen yaitu *pixel*.
2. Parameter Kecepatan
   Agen (*player*) menyesuaikan kecepatan antar agen lainnya sehingga formasi tetap terjaga dan tidak berubah-ubah

3.3 Analisis Algoritma Boids

Algoritma *boids* pada pengembangan *game* ini digunakan untuk menentukan variasi perilaku yang dihasilkan oleh agen (*player*), dengan diterapkannya algoritma *boids*, tiap-tiap agen (player) dapat merubah perilaku berbasis perubahan variable masukan menjadi perilaku yang akan dirancang menggunakan algoritma *boids.* Dalam pembentukan formasi terdapat 3 proses input, yaitu *Separation, Alignment* dan *Cohesion*. Nilai-nilai yang diinputkan pada 3 proses tersebut dapat mempengaruhi sejauh mana formasi bekerja dengan baik. Berikut contoh penerapan *separation, alignment* dan *cohesion* pada *prototype* yang akan dibuat.

1. Separation
   Range nilai masukan : 0 – 100 %

   Jika *separation* = 0%, maka masing-masing agen tidak dapat menjaga jarak antar agen lainnya sehingga kemungkinan terjadi tabrakan antar agen sangat besar.
   Berikut contoh penerapan parameter *separation* bernilai 0% yang dapat dilihat pada gambar 2.



Gambar 2 *Separation* 0%

   Jika *separation* > 0% dan <= 100%, maka masing-masing agen dapat menjaga jarak antar agen lainnya sehingga kemungkinan terjadi tabrakan dapat diminimalisir, semakin besar nilai input maka kemungkinan tabrakan semakin kecil.
   Berikut contoh penerapan parameter *separation* bernilai 100% yang dapat dilihat pada gambar 3.



Gambar 3 *Separation* 100%

2. Alignment
   Range nilai masukan : 0 – 100%

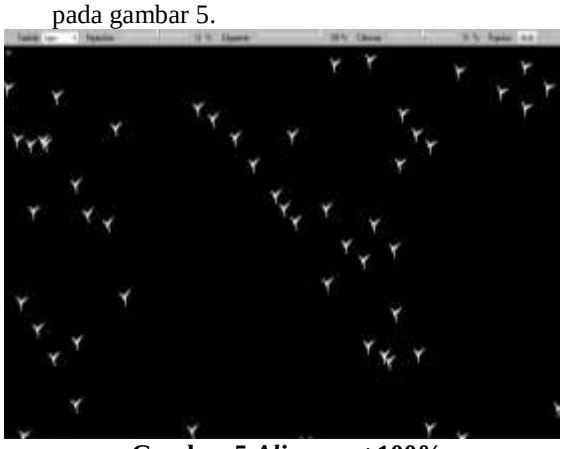
   Jika Alignment = 0%, maka masing-masing agen tidak dapat menjaga kecepatan masing-masing agen sehingga gerakan tiap-tiap agen tidak akan selaras.
   Berikut contoh penerapan parameter *alignment* bernilai 0% yang dapat dilihat pada Gambar 4.



Gambar 4 *Alignment* 0%

Jika Alignment > 0% dan <= 100%, maka masing-masing agen dapat menjaga kecepatan masing-masing agen sehingga gerakan tiap-tiap agen agen akan selaras, semakin besar nilai input maka semakin baik penyesuian kecepatan tiap-tiap agen terhadap formasinya.

Berikut contoh penerapan parameter *alignment* bernilai 100% yang dapat dilihat pada gambar 5.



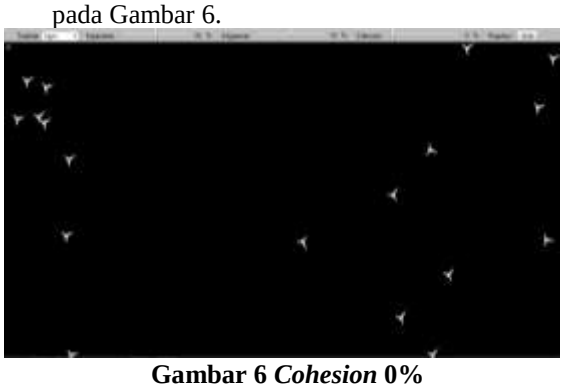
Gambar 5 *Alignment* 100%

3. Cohesion

Range nilai masukan : 0 – 100%

Jika *Cohesion* = 0%, Maka masing-masing agen tidak menjaga jarak dengan agen lainnya sehingga terpisah dari formasi.

Berikut contoh penerapan parameter *cohesion* bernilai 0% yang dapat dilihat pada Gambar 6.



Gambar 6 *Cohesion* 0%

Jika *Cohesion* = 0% dan <= 100%, Maka masing-masing agen akan menjaga jarak agar tetap dekat dengan agen lainnya sehingga tidak terpisah dari formasi, semakin besar nilai input maka kemampuan agen menyesuaikan jarak dengan agen lainnya semakin baik.

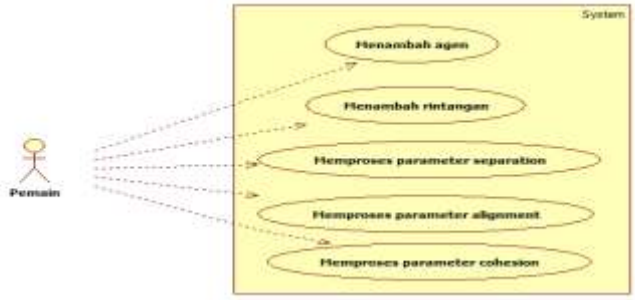
Berikut contoh penerapan parameter *cohesion* bernilai 100% yang dapat dilihat pada Gambar 7.



Gambar 7 *Cohesion* 100%

3.4 Use Case Diagram Simulasi

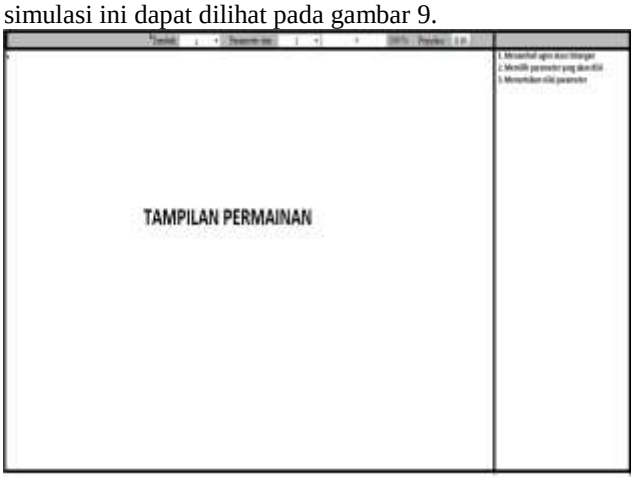
Use case diagram simulasi bisa dilihat pada gambar 8.



Gambar 8 *Use Case* Diagram Simulasi

3.5 Antarmuka Menu Utama

Perancangan antarmuka menu utama dari simulasi ini dapat dilihat pada gambar 9.

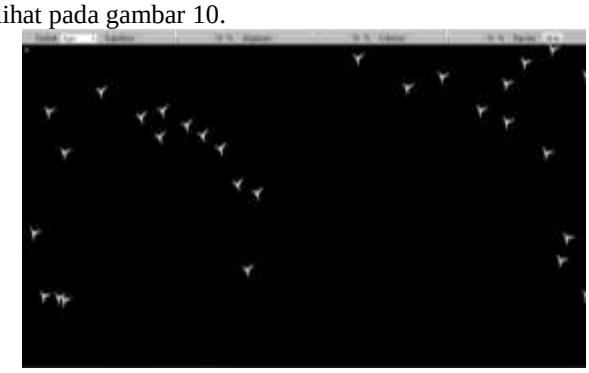


Gambar 9 Antarmuka menu utama

4. IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Implementasi antarmuka dari menu utama dapat dilihat pada gambar 10.



Gambar 10 Antarmuka Menu Utama

4.2 Implementasi Perangkat Keras

Perangkat keras yang digunakan untuk melakukan implementasi dan pengujian simulasi ini yaitu sebagai berikut :

Tabel 1 Implementasi Perangkat Keras

No	PerangkatKeras	Keterangan
1	Prosesor	Kecepatan 2.4 GHz
2	Monitor	Monitor 17", resolusi 1024x768
3	VGA On-board	Kecepatan 512 MB
4	Optical Drive DVD-ROM	Standard
5	Keyboard	Standard
6	Mouse	Standard
7	Memori	1GB

4.3 Implementasi Perangkat Lunak

Adapun spesifikasi perangkat lunak yang dibutuhkan untuk mengimplementasikan perangkat lunak diantara lain sebagai berikut :

Tabel 2 Implementasi Perangkat Lunak

No	PerangkatLunak	Keterangan
1	SistemOperasi	*Windows* 7
2	*Tools* Program	*Adobe Flash* CS5

4.4 Pengujian Whitebox

Untuk memastikan algoritma berjalan dengan baik maka akan dilakukan uji coba bagaimana masing-masing agen meng-*update* posisi sehingga membentuk formasi. Pengujian *whitebox* dapat dilihat pada Tabel 3.

Tabel 3 Hasil Uji Whitebox

Path No	Node(n)	Hasil yang Diharapkan	Hasil Sesuai Uji Kasus	Keterangan
1	1-2-3-4-5-6-7-8-9	Masing-masing boids dapat di-update posisinya	Masing-masing boids dapat di-update posisinya	[√] alur terlewati [ ] alur tidak terlewati
2	1-2-3-9	*Boids* tidak dapat di-update posisinya karena boid kosong	*Boids* tidak dapat di-update posisinya karena boid kosong	[√] alur terlewati [ ] alur tidak terlewati

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil implementasi dari hasil pengujian algoritma *boids* yang dilakukan pada *game* perang dapat diambil kesimpulan bahwa :

1. Algoritma *boids* dapat berjalan dan diterapkan pada *game* perang ini.
2. Algoritma *boids* teruji dapat memberi kemampuan pada agen untuk membentuk formasi.

5.2 Saran

Berdasarkan hasil dari keseluruhan simulasi, maka dapat diberikan saran-saran yaitu penambahan jenis formasi yang nantinya akan diterapkan dan mengembangkan kemampuan agen pada saat melewati rintangan agar posisi antar agen tetap terjaga sehingga tidak terjadi tabrakan.

DAFTAR PUSTAKA

[1] C. Reynolds. (1987), *Flocks, Herd, and Schools* : *a distributed behavior model*, *Proceedings of* ACM SIGGRAPH ’87.

[2] Rosa A.S, M. Shalahudin. (2011), Modul Pembelajaran (rekayasa perangkat lunak), Modula, Bandung

[3] Suyanto, ST., Msc. (2007), *Artifial Intelligence: Searching, Reasoning, Planning.* Bandung: Informatika Bandung.

[4] Suyanto, ST., Msc. (2007), *Artifial Intelligence: Searching, Reasoning, Planning.* Bandung: Informatika Bandung.

.