

Perubahan Gerakan Formasi Pasukan Dengan Pola Tetap Geometri Berbasis Boids

Fajar Aditiya P¹, Supeno Mardi S.N¹, Mochamad Hariadi³

Jurusan Teknik Elektro

Institut Teknologi Sepuluh Nopember (ITS)

Surabaya, Indonesia

fajarap@gmail.com¹, mardi@ee.its.ac.id², mochar@ee.its.ac.id³

Abstract—Pada saat ini pergerakan dari agen dalam kelompok secara luas digunakan dalam permainan komputer. Teknik dari pergerakan ini biasanya disebut dengan gerak formasi. Dalam tulisan ini, untuk membangun gerak formasi digunakan formasi geometris tetap. Dimana formasi didefinisikan oleh satu set slot, slot ini akan digunakan karakter dari agen NPC. Semua slot dalam formasi didefinisikan relatif terhadap pemimpin. Perilaku diterapkan dalam gerakan ini, fungsi ini untuk mengontrol agen bergerak ke posisi slot, menyesuaikan kecepatan, dan menghindari tabrakan antara teman dan juga kendala. Perilaku ini menerapkan metode Boids. Untuk mengevaluasi kinerja, digunakan tiga parameter: rasio panjang jalan, rata-rata kesalahan posisi dan waktu untuk pembentukan. Kinerja hasil terbaik dari percobaan dengan memutar, diperoleh dari formasi lingkaran. Hasil kinerja terbaik dari percobaan dengan halangan adalah dengan menggunakan formasi kolom.

Keywords—Pergerakan agen; formasi geometris tetap; perilaku Boids.

I. PENDAHULUAN

Formasi pasukan merupakan suatu susunan sekelompok unit yang dengan tujuan tertentu diposisikan untuk membentuk pola atau bentuk tertentu dan diterapkan untuk tujuan taktis yaitu sebagai pola penyerangan atau pertahanan. Dalam formasi pasukan di *game* komputer, karakter dibagi menjadi dua jenis yaitu sebagai pemimpin (*leader*) dan yang lain menjadi pengikut (*follower*). Pemimpin diberi kebebasan bergerak lebih banyak dari yang lainnya, karena diasumsikan digerakkan oleh pemain manusia. Sedangkan setiap pengikut diharuskan mengikuti pemimpinnya kemanapun dia bergerak, dan harus tetap berada dalam formasinya, sambil mereka diberikan perintah. Perintah ini berupa algoritma cerdas yang berisi kemudi untuk mengatur tingkah laku yang disebut *steering behaviour* yang nantinya dipasangkan disetiap karakter pengikut.

Penelitian ini mengembangkan suatu algoritma kemudi tingkah laku (*steering behaviour*) formasi untuk agen *follower*, sehingga agen - agen tersebut dapat dikoordinasi membentuk suatu pola formasi.

II. TEORI PENUNJANG

A. Boids

Tingkah laku formasi dalam alam adalah seperti sekelompok burung di udara atau ikan di air[1]. Mereka

biasanya berkelompok untuk bertahan hidup, untuk menghindari serangan dari hewan pemangsa. Setiap hewan tersebut mempunyai sebuah sensor yang bisa mendeteksi hewan pemangsa, menghindari tabrakan dengan teman dalam kelompoknya dan mempunyai keinginan untuk tetap dekat dengan kelompoknya.

Reynolds[1] mengembangkan sebuah model perilaku sederhana untuk berkelompok yang simulasinya berupa sekelompok burung dan anggota dari kelompok ini kemudian dikenal dengan nama "boids". Dasar tingkah laku untuk boids ini berisi : cara menghindari tabrakan, penyesuaian kecepatan dan gerakan ke pusat kelompok (*flock centering*). Setiap komponen ini dihitung terpisah dan hasilnya digabungkan untuk menciptakan suatu gerakan.

B. Autonomous Characters

Autonomous Characters atau karakter otonom adalah jenis agen otonom yang digunakan dalam animasi komputer dan media interaktif seperti *game* dan *virtual reality*[2]. Agen ini mewakili karakter dalam cerita atau permainan dan memiliki beberapa kemampuan untuk berimprovisasi melakukan tindakan sendiri, dan tidak dikendalikan oleh pemain. Dalam *game* karakter otonom dikenal juga dengan *non-player characters* (NPC).

Gerakan agen otonom dapat dibagi menjadi tiga lapisan :

1. **Action Selection** : Ini adalah bagian dari perilaku agen yang bertanggung jawab untuk memilih tujuan dan memutuskan rencana apa untuk mengikuti. Dalam bagian ini misalkan dikatakan "pergi kesana" dan kerjakan A, B lalu C.
2. **Steering** : Lapisan ini bertanggung jawab untuk menghitung lintasan yang diinginkan, diperlukan untuk memenuhi tujuan dan rencana yang ditetapkan oleh pemilihan lapisan tindakan. *Steering behaviors* diterapkan pada lapisan ini. Ini menghasilkan gaya kemudi yang menggambarkan di mana seorang agen harus bergerak dan seberapa cepat untuk pergi tujuan.
3. **Locomotion** : Lapisan bawah, mewakili lebih banyak aspek mekanis gerakan agen. Sebagai contoh, jika kita akan menerapkan mekanisme pergerakan seekor unta, tank, dan ikan mas dan kemudian memberikan perintah agar mereka berjalan ke utara, mereka semua akan menggunakan proses mekanis yang berbeda untuk

menciptakan gerak, walaupun tujuan gerak mereka adalah identik yaitu ke utara.

C. Steering Behaviors (Kemudi Perilaku)

Kemudi untuk NPC dipengaruhi oleh parameter vektor. Kemudi ini kemudian disebut sebagai kemudi perilaku adalah perhitungan geometris dari sebuah vektor untuk mewakili gaya kemudi yang diinginkan atau dengan kata lain merupakan pengembangan gaya kemudi untuk menghasilkan arah gerak tertentu. Kemudi yang digunakan pada penelitian ini adalah *leader following*. Ini adalah perilaku dimana karakter mengikuti salah satu pemimpin. Jika suatu pemimpin telah ditandai maka karakter pengikut akan mengikuti pemimpin tersebut. Para pengikut umumnya berada dekat dengan pemimpin tetapi tidak berkerumun dengan pemimpin. Jika ada lebih dari satu pengikut, mereka akan menghindari tabrakan satu sama lain. Untuk menerapkan perilaku ini digunakan perilaku *arrive* dan *separation*. Perilaku *arrive* fungsinya untuk menjaga jarak dengan pemimpin sehingga gerak karakter tidak melampaui pemimpin. Sedangkan *separation* menjaga agar tidak berkerumunan dengan karakter lainnya

D. Formasi Tetap

Salah satu cara sederhana untuk membentuk formasi adalah dengan menggunakan formasi tetap[4]. Disini formasi didefinisikan sebagai sebuah kumpulan petak (*slot*) yang merupakan lokasi karakter untuk diposisikan.

Satu *slot* ditandai sebagai *slot* pemimpin dan slot lainnya dalam formasi didefinisikan relatif terhadap slot ini. Efektifnya slot pemimpin diasumsikan sebagai titik nol atau pusat dari formasi. Gerak pemimpin diberi kebebasan lebih dari pada yang lainnya, dapat dikontrol dengan kemudi, mengikuti *path*, ataupun dikendali oleh pemain manusia, dan apapun mekanismenya dia tidak diperhitungkan dalam posisi formasi.

Pola formasi yang diposisikan dan diorientasikan pada game menempatkan pemimpin pada slotnya dan kemudian menghadap ke arah yang sesuai. Sebagai pemimpin dari pergerakan pola ini akan bergerak dan berubah dalam permainan. Dan pada gilirannya masing – masing slot akan bergerak serentak.

Setiap penambahan slot dalam formasi dapat diisi oleh karakter tambahan. Posisi masing – masing karakter dapat ditentukan langsung dari geometri formasi, tanpa memerlukan sistem kinematis atau kemudi sendiri.

E. Finite State Machine (FSM)

Finite State Machine (FSM) merupakan metode yang paling populer digunakan pada game. Metode ini seakan menjadi standard dan telah digunakan secara luas dibidang game. Hal ini karena kesederhanaan dan kemudahan FSM diimplementasikan pada game [3][4].

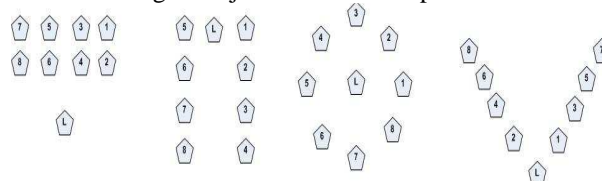
FSM terdiri dari dua elemen utama yaitu *state* (keadaan) dan *transition* (transisi). *State* merupakan keadaan obyek saat ini, sedangkan transisi merupakan hal yang dilakukan agar bisa berpindah dari satu *state* ke *state* yang lain. transisi bisa juga menyatakan kondisi yang harus dipenuhi oleh obyek agar bisa berpindah dari satu *state* ke *state* yang lain.

III. METODE PENELITIAN

A. Bentuk Formasi

Bentuk formasi untuk penelitian ini meniru pola formasi dari [2] & [5]. Formasi tersebut adalah diilustrasikan oleh gambar 1, dimana:

- Garis (*line*) – dimana agen berjalan mengikuti pola garis.
- Kolom (*column*) – dimana agen berjalan satu dibelakang agen lainnya.
- Lingkaran – agen berjalan membentuk pola lingkaran yang merupakan modifikasi bentuk *diamond*.
- V – dimana agen berjalan membentuk pola huruf “V”.



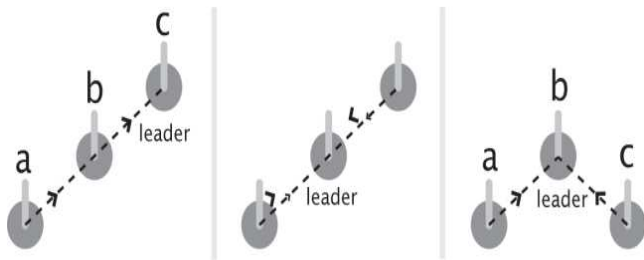
Gambar 1 Bentuk Formasi. Dari Kiri Ke Kanan : garis (*line*), kolom (*column*), lingkaran, dan V.

Pada setiap formasi, terdapat satu agen *leader* dan banyak agen *follower*. Perilaku aktif setiap agen *follower* adalah sama, sedangkan agen *leader* berbeda karena digunakan sebagai acuan titik pusat formasi. Tugas masing – masing agen *follower* adalah secara bersamaan bergerak ke lokasi tujuan, menghindari hambatan, menghindari tabrakan dengan agen lain, dan mempertahankan bentuk formasi. Sedangkan tugas agen *leader* adalah menentukan arah tujuan formasi.

B. Penentuan Titik Formasi

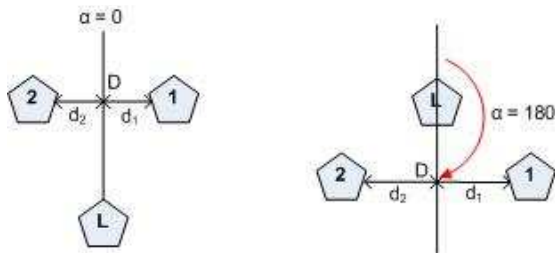
Masing - masing agen menghitung posisi yang tepat dalam formasi berdasarkan lokasi dari agen lainnya dan didefinisikan oleh jarak dan sudut. Teknik yang digunakan pada pembentukan formasi ini adalah menentukan titik referensi *leader*, dimana setiap agen *follower* menentukan posisi di dalam formasi berkaitan dengan agen *leader*. *Leader* tidak ikut dalam mempertahankan bentuk formasi tetapi *follower* bertanggung jawab untuk memelihara formasi.

Orientasi sederhana dari pembentukan formasi akan menghasilkan sebuah garis lurus dari unit *leader*, ke titik lokasi dimana agen akan diposisikan. Bersama – sama dengan unit *leader*, orientasi formasi kemudian mendefinisikan sebuah sistem local koordinat yang menentukan posisi formasi. Lokal koordinat ini dihitung secara terus menerus pada setiap langkah pergerakan. Hubungan pembentukan formasi diilustrasikan pada gambar 2. Panah menunjukkan bagaimana posisi formasi ditentukan. Setiap titik panah menunjukkan dari agen ke referensi terkait yaitu *leader*, kemudian agen menentukan posisi di sisi kiri atau di kanan *leader*. Sedangkan jarak antara agen ditentukan oleh suatu parameter jarak.



Gambar 2 Hubungan pembentukan formasi yang ditentukan dari leader.

Gambar 3 menunjukkan keadaan formasi dimana agen *follower* mengikuti gerak arah pemimpin, dengan posisi di depan dan dibelakang pemimpin.



Gambar 3 Pembentukan formasi relatif terhadap sudut muka leader.

Sudut muka *leader* menentukan letak posisi formasi. Jika perhitungan digunakan sudut $\alpha = 0$ posisi formasi berada di depan, sedangkan sudut $\alpha = 180$ formasi terletak di belakang. Parameter jarak (d) antara agen ditentukan dari ID terhadap titik D, pada gambar di atas d_1 bernilai 1 dan d_2 bernilai 2. Karena formasi bergerak mengikuti *leader*, hasil ini kemudian ditambahkan dengan posisi *leader* (L_{posisi}) untuk mendapat posisi formasi baru, cara perhitungannya sebagai berikut :

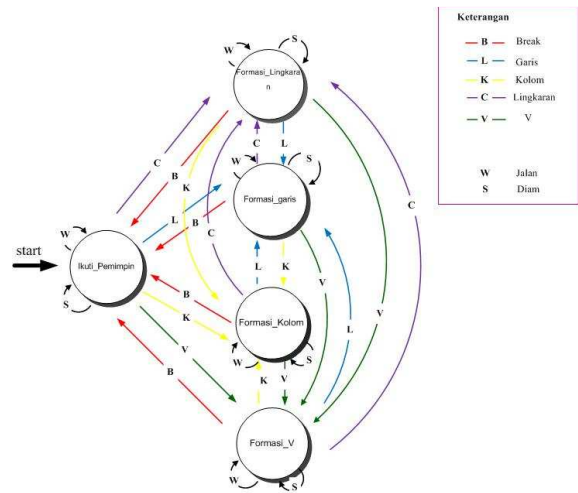
$$F_{posisi} = (\sin \alpha * d, \cos \alpha * d) \quad (1)$$

$$F_{new} = F_{posisi} + L_{posisi} \quad (2)$$

C. Desain FSM NPC

Desain FSM untuk setiap agen NPC *follower* adalah sama. Kelompok NPC tersebut nantinya mengikuti arah gerak pemimpinnya. Pemimpin NPC dapat bergerak bebas ke arah manapun, karena dimainkan oleh pemain, dia juga mempunyai hak kontrol penuh terhadap NPC *follower* untuk membentuk formasi. Di bawah ini merupakan penjelasan dari gambar 4, yang merupakan desain FSM NPC pada penelitian ini:

- Pertama kali agen NPC *follower* mengikuti arah gerak gerakan pemimpin, disini para agen akan mencari sebuah posisi dan menghadap ke arah pemimpin, jika jarak diantara mereka diatas batas yang ditentukan, NPC akan berjalan menuju pemimpin.



Gambar 4 Desain FSM formasi

- *State* berikutnya adalah *state* formasi. *State* ini dibagi menjadi 4 yaitu *state* formasi_ lingkaran, *state* formasi_ garis, *state* formasi_ kolom, dan *state* formasi_ V. saat pemimpin memberi perintah untuk membentuk suatu formasi, misalnya formasi lingkaran, *state* akan berubah menjadi *state* formasi_ lingkaran dan NPC *follower* akan membentuk formasi berbentuk lingkaran. Demikian seterusnya ketika pemimpin memberi perintah suatu bentuk formasi, *state* akan berubah sesuai perintah formasi yang diinginkan. Ketika pemimpin bergerak, bentuk formasi tetap dipertahankan dan mengikuti arah geraknya. Terakhir jika pemimpin memberi perintah istirahat, maka *state* akan kembali ke *state* awal yaitu *ikuti_pemimpin*.

D. Algoritma Formasi

Algoritma pengambilan keputusan agen untuk bentuk formasi adalah sebagai berikut :

```

Algorithm 1. Pengambilan Keputusan
01 IF perintah formasi != null THEN
02     IF perintah formasi = "lingkaran" THEN
03         Proses hitung posisi formasi lingkaran
04     IF perintah formasi = "garis" THEN
05         Proses hitung posisi bentuk formasi garis
06     IF perintah formasi = "kolom" THEN
07         Proses hitung posisi bentuk formasi kolom
08     IF perintah formasi = "V" THEN
09         Proses hitung posisi bentuk formasi V
10 Else jika tidak ada perintah formasi
11     Ikuti pemimpin
    
```

Perintah formasi akan menghasilkan suatu titik posisi yang merupakan lokasi agen ditempatkan pada suatu formasi. Agar dapat menuju titik tersebut agen membutuhkan kemudi yaitu kemudi *arrive*. Dengan kemudi ini agen dapat menyesuaikan kecepatannya terhadap kelompok formasi. Sedangkan jika tidak ada perintah formasi, kemudi yang digunakan adalah kemudi *Leader Following*.

IV. HASIL PENGUJIAN SISTEM

Untuk mengevaluasi kinerja pergerakan formasi pada penelitian ini, digunakan 10 agen NPC *follower* dalam setiap formasi, memakai game engine Unity 3D[6]. Kelompok agen ini kemudian diperintahkan bergerak sepanjang 70 meter pertama, lalu berbelok 90 derajat ke kiri dan bergerak lagi sepanjang 70 meter kedua. Disini diamati bagaimana agen berusaha mempertahankan posisi ketika sudut dirubah. Dalam ujicoba ini, tahap pertama ujicoba pergerakan tidak diberi rintangan dahulu, sedangkan untuk perhitungan statistik dilakukan 10 kali simulasi untuk setiap bentuk formasi. Agar menjamin bentuk formasi tepat dan dapat dievaluasi, mereka diperintahkan bergerak lurus dahulu sepanjang 50 meter sebelum titik evaluasi. Jarak 50 meter ini tidak diperhitungkan dalam jarak evaluasi sebesar 140 meter. Perhitungan juga tidak melibatkan *leader*.

Tiga perhitungan yang digunakan dalam evaluasi: **rasio panjang jalan, kesalahan posisi, dan waktu untuk pembentukan**^[5]. Rasio panjang jalan merupakan rata – rata perjalanan dari 10 agen dibagi dengan jarak lurus dari evaluasi (140 meter), sebagai contoh jika rasio dihasilkan 1.02 berarti agen rata – rata bergerak lebih jauh sebesar 2%, disebabkan agen berada dalam formasi. Kesalahan posisi adalah rata – rata perpindahan posisi agen dari posisi titik yang tepat dalam formasi. Sedangkan waktu untuk pembentukan merupakan perhitungan waktu ketika kelompok agen diberi sebuah gangguan yaitu berbelok ke kiri atau rintangan, lalu berusaha kembali ke bentuk formasi semula. Gangguan ini diterapkan juga pada perhitungan kesalahan posisi.

Hasil rata – rata dari percobaan diatas diringkas pada tabel dibawah ini.

Tabel 1 Hasil pergerakan formasi rata - rata tanpa rintangan dengan gangguan putaran 90° ke arah kiri

Jenis Formasi	rasio panjang jalan	kesalahan posisi (meter)	Waktu untuk pembentukan (detik)
Lingkaran	1	0.0	0.0
Garis	1.03	0.81	19

Kolom	1.13	10.55	35.5
V	1.11	9.36	36

Tabel 2 Hasil pergerakan formasi rata - rata dengan rintangan

Jenis Formasi	rasio panjang jalan	kesalahan posisi (meter)	Waktu untuk pembentukan (detik)
Lingkaran	1.03	0.81	26.5
Garis	1.06	5.282	22
Kolom	1.02	0.81	16.1
V	1.02	0.81	30.2

Dari tabel 1 hasil terbaik dari pergerakan formasi tanpa rintangan ditunjukkan dengan nilai terkecil adalah formasi lingkaran. Sedangkan dari tabel 2 pergerakan formasi dengan rintangan, hasil terbaik adalah diperoleh formasi kolom, pada formasi ini waktu yang dibutuhkan untuk kembali ke posisi formasi setelah melewati rintangan, dilakukan dalam waktu yang paling minimal dibanding dengan formasi lainnya.

V. KESIMPULAN

Kami telah membuat metode pergerakan formasi dengan boids dan algoritma formasi, dimana hasil terbaik dari pergerakan formasi tanpa rintangan ditunjukkan dengan nilai terkecil adalah formasi lingkaran. Pada keadaan ini formasi lingkaran menghasilkan rasio panjang jalan sebesar 1, dan kesalahan posisi = 0. Sedangkan untuk pergerakan formasi dengan rintangan, hasil terbaik adalah formasi kolom, dengan rasio 1,02 dan waktu 16,1 detik.

REFERENCES

- [1] Craig W. Reynolds, Flocks, Herds, and Schools: A Distributed Behavioral Model, SIGGRAPH'87, 1987.
- [2] Craig W. Reynolds, Steering Behaviors For Autonomous Characters, 1999
- [3] Chars River Media, *Ai Game Programming Wisdom*, 2002 hlm 272-281.
- [4] Ian Millington And John Funge, *Artificial Intelligence For Games*, Second Edition, 2009, bab Coordinated Movement hlm 144 - 153.
- [5] Tucker Balch And Ronald C. Arkin, Behavior-based Formation Control for Multi-robot Teams, IEEE Transactions On Robotics And Automation, 1999
- [6] Game Engine Unity 3, url: <http://unity3d.com/>