



Tesis - TE - 142599

PERGERAKAN OTONOM PASUKAN MENGGUNAKAN ALGORITMA
BOIDS BERBASIS PARTICLE SWARM OPTIMIZATION

Syahri Mu'min
2213205022

PEMBIMBING

Mochamad Hariadi, ST., M.Sc., Ph.D.

Dr. Supeno Mardi Susiki Nugroho, ST., MT.

PROGRAM

BIDANG KEAHLIAN JARINGAN CERDAS MULTIMEDIA

KONSENTRASI TEKNOLOGI PERMAINAN

TEKNIK ELEKTRO

FAKULTAS TEKNOLOGI INDUSTRI

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

SURABAYA

2015



Tesis - TE - 142599

**AUTONOMOUS TROOP MOVEMENT USING BOIDS ALGORITHM
BASE ON PARTICLE SWARM OPTIMIZATION**

Syahri Mu'min
2213205022

SUPERVISOR

Mochamad Hariadi, ST., M.Sc., Ph.D.

Dr. Supeno Mardi Susiki Nugroho, ST., MT.

MAGISTER PROGRAM

INTELEGENT NETWORK EXPERTISE MULTIMEDIA

PROGRAM GAME TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING

FACULTY OF INDUSTRIAL TECHNOLOGY

SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY

SURABAYA

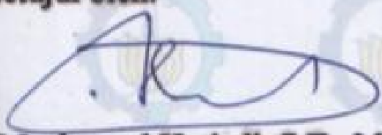
2015

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Teknik (M.T.)
di
Institut Teknologi Sepuluh Nopember


Oleh:
Syahri Mu'min
NRP. 2213205022

Tanggal Ujian : 15 Juni 2015
Periode Wisuda : September 2015


Disetujui oleh:


1. Mochamad Hariadi, S.T., M.Sc., Ph.D.
NIP. 196912091997031002

(Pembimbing I)


2. Dr. Supeno Mardi Susiki N., S.T., M.T.
NIP. 197003131995121001

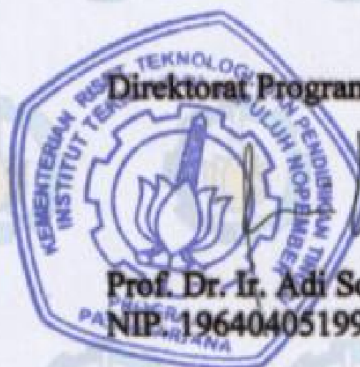
(Pembimbing II)


3. Dr. Surya Sumpeno, S.T., M.Sc.
NIP. 196906131997021003

(Penguji)


4. Dr. Eko Mulyanto Y., S.T., M.T.
NIP. 196806011995121009

(Penguji)



Direktorat Program Pasca Sarjana

Prof. Dr. Ir. Adi Soeprijanto, M.T.
NIP. 196404051990021001

Pergerakan Otonom Pasukan Menggunakan Algoritma Boids Berbasis Particle Swarm Optimization

Nama Mahasiswa : Syahri Mu'min

NRP : 2213205022

Dosen Pembimbing : Mochamad Hariadi, S.T., M.Sc., Ph.D.

Dr. Supeno Mardi Susiki Nugroho, S.T., M.T.

ABSTRAK

Abstrak - Permainan real-time strategi (RTS) merupakan sebuah game yang menarik, layar dipisahkan menjadi peta area, unit, dan bangunan. Bermain game RTS umumnya terdiri dari pemain yang diposisikan di suatu tempat di peta dengan beberapa unit atau bangunan, pemain bergerak dari satu tempat ke tempat yang lain. Kelompok karakter atau pasukan yang bergerak dalam permainan RTS memiliki pendekatan luas, untuk masalah ini menggunakan model boids. Model boids dapat mengendalikan kerumunan dalam formasi untuk melewati halangan dan berkumpul kembali dalam formasi semula.

Halangan dapat menghambat pergerakan pasukan yang mengakibatkan pasukan tertinggal saat menggunakan boids sehingga diperlukan optimasi untuk membuatnya dapat lebih cepat dalam melakukan regrouping formasi. Kemampuan Particle Swarm Optimization (PSO) untuk mencapai posisi optimum menciptakan kemungkinan untuk secara otomatis menghasilkan jalan non deterministic kerumunan pasukan dari satu posisi tertentu dalam mempercepat regrouping formasi. Fungsi tersebut digunakan untuk menggambarkan semua jenis objek dalam sistem simulasi, termasuk target statis, hambatan statis, serta pasukan yang dianggap sebagai partikel dalam mencari cara untuk mencapai solusi terbaik.

Kata Kunci: agen otonom, boids, particle swarm optimization, Real-time game strategi

Autonomous Troop Movement Using Boids Algorithm Base on Particle Swarm Optimization

Name : Syahri Mu'min

NRP : 2213205022

Supervisor : Mochamad Hariadi, S.T., M.Sc., Ph.D.

Dr. Supeno Susiki Mardi Nugroho, S.T., M.T.

ABSTRACT

Abstract - real-time strategy game (RTS) is an interesting game, the screen is separated into area maps, units, and buildings. RTS game play is generally made up of players who are positioned at a spot on the map with some units or buildings, the player moves from one place to another. Groups of characters or troops engaged in RTS games have a broad approach to this problem using models boids. Boids models can control the crowd in formation to break through barriers and reassemble the original formation.

Obstacle can hinder troop movements resulting in forces left behind when using boids so required optimization to make it faster to do regrouping formation. Ability Particle Swarm Optimization (PSO) to achieve optimum position creates the possibility to automatically generate non deterministic way a crowd of troops from one particular position in accelerating the formation regrouping. That function is used to describe all types of objects in the system simulation, including the target static, static obstacles, and the forces which are considered as particles in finding ways to achieve the best solution.

Keywords: autonomous agents, boids, particle swarm optimization, real-time strategy game

KATA PENGANTAR

Alhamdulillah Robbil Alamin...

Alhamdulillah segala puji kehadiran Allah atas segala limpahan Rahmat dan nikmatnya. Dengan nikmat kesehatan selama menjalani masa studi dan selama menyusun Tesis Alhamdulillah buku tesis ini dapat diselesaikan dengan baik dan tepat waktu. Shalawat semoga tetap tercurahkan kehadiran nabi Muhammad SAW yang telah melimpahkan berkah ilmu kepada segenap ummatnya.

Tesis ini disusun sebagai syarat untuk memperoleh gelar Magister Teknik pada bidang konsentrasi Teknologi Permainan, bidang studi Jaringan Cerdas Multimedia, jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember Surabaya.

Terima kasih untuk semua yang telah membantu dan mendukung dalam menyelesaikan tesis ini. Penulis juga mengucapkan terima kasih kepada :

1. Kedua orang tuaku Bapak dan Ibu atas semua amal baik, support, doa serta harapan yang tidak sempat dinikmati “ebok” karena telah mendahului ditengah perjalanan masa studi penulis
2. Semua dosen Elektro ITS khususnya pak Hariadi, pak Uki, pak Surya, Pak Akok dan Prof. Hery yang telah membimbing selama masa studi.
3. Pembimbing tesis Mochamad Hariadi, ST., M.Sc., Ph dan Dr. Supeno Mardi Susiki Nugroho, S.T, M.T. yang telah memberikan wawasan dan ilmu baru kepada saya
4. Teman-teman Game Tech angkatan 2013, kakak tingkat 2012 dan sebelumnya, adik kelas 2014 dan setelahnya. Terima kasih atas kebersamaan dan sharing ilmunya.
5. Teman-teman KSR-PMI Unit UIN Maulana Malik Ibrahim Malang yang turut serta membantu dalam kegiatan.

Penulis sadar akan kekurangan pada buku tesis ini. Maka dari itu penulis mohon maaf dan harapan kedepan akan ada penyempurnaan tema ini. Kritik saran yang membangun kami harapkan untuk penyempurnaan penelitian ini kedepan.

Surabaya, Juni 2015

DAFTAR ISI

LEMBAR PENGESAHAN	i
ABSTRAK	iii
ABSTRACT	v
KATA PENGANTAR.....	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xi
DAFTAR TABEL	xii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	4
1.3. Batasan Masalah	5
1.4. Tujuan	5
1.5. Manfaat Penelitian	5
BAB II KAJIAN PUSTAKA	7
2.1. Permainan <i>Real Time</i> Strategi (RTS)	7
2.2 Pergerakan Pasukan Pada Clash of Clan (CoC)	8
2.3 Swarm Intelegent	12
2.4 Perilaku Kelompok	15
2.5 Perilaku Kendai Agen Otonom.....	19
2.6 Model Boids	21
2.7 Particle Swarm Optimization.....	23
BAB III METODOLOGI PENELITIAN	27
3.1. Desain Agen.....	28
3.2. Desain Environment	28
3.3. Penerapan Model Boids	29
3.4. Penerapan Particle Swarm Optimization	34
3.5 Skenario	37
BAB IV PERCOBAAN DAN SIMULASI KOMPUTER.....	39
4.1. Pergerakan Pasukan dengan Satu Halangan	39

4.2. Pergerakan Pasukan dengan Dua Halangan.....	43
4.3 Pergerakan Pasukan dengan Tiga Halangan	46
4.4 Pergerakan Pasukan dengan Tiga Halangan dalam Lorong.....	45
4.5. Pergerakan Pasukan dengan 4 halangan dinamis dalam lorong.....	53
4.6 Pergerakan Pasukan dengan Variasi Populasi.....	57
BAB V KESIMPULAN DAN SARAN	63
5.1 Kesimpulan	63
5.2 Saran.....	63
DAFTAR PUSTAKA	65
LAMPIRAN	

DAFTAR TABEL

Tabel 2.1 Swarming dari burung, ikan, domba, lebah	14
Tabel 4.1 Posisi Awal Percobaan	39
Tabel 4.2 Hasil Percobaan Pertama Boids	41
Tabel 4.3 Hasil Percobaan Pertama Boids-PSO	42
Tabel 4.4 Hasil Percobaan Kedua Boids.....	44
Tabel 4.5 hasil Percobaan Kedua Boids-PSO	45
Tabel 4.6 hasil Percobaan Ketiga Boids	47
Tabel 4.7 Hasil Percobaan Ketiga Boids-PSO.....	49
Tabel 4.8 Hasil Percobaan Keempat Boids.....	51
Tabel 4.9 Hasil Percobaan Keempat Boids-PSO	52
Tabel 4.10 Hasil Percobaan Kelima Boids	55
Tabel 4.11 Hasil Percobaan Kelima Boids-PSO.....	56
Tabel 4.12 Hasil Percobaan Keenam Boids-PSO	58
Tabel 4.13 Kompleksitas Boids	59
Tabel 4.14 Kompleksitas Boids-PSO.....	60
Tabel 4.15 Kompleksitas Big O	62

DAFTAR GAMBAR

Gambar 1.1 Pergerakan Pasukan Otonom CoC	1
Gambar 1.2 Pergerakan Halangan Statis.....	2
Gambar 1.3 Pergerakan Boids.....	3
Gambar 1.4 Pergerakan Boids dengan Beberapa Halaman	4
Gambar 2.1 Karakter Goblin.....	10
Gambar 2.2 Ilustrasi Pergerakan Goblin Menuju Target	11
Gambar 2.3 Karakter Giant	11
Gambar 2.4 Karakter Golem	12
Gambar 2.5 Ilustrasi Pergerakan giant dan golem Menuju Target	12
Gambar 2.6. Gerakan Jari pada Tab Saat Deploy Pasukan.....	18
Gambar 2.7 seek dan <i>flee</i>	20
Gambar 2.8 Pursuit dan Evasion.....	21
Gambar 2.9 Obstacle Avoidence.....	21
Gambar 2.10 Separation.....	22
Gambar 2.11 Cohesion.....	23
Gambar 2.12 Alignment.....	23
Gambar 2.13 Ilustrasi Konsep PSO	26
Gambar 3.1 Skema Metodologi Penelitian	27
Gambar 3.2 Formasi Pasukan Boids	28
Gambar 3.3 Lingkungan Halangan	29
Gambar 3.4 Halangan Lorong.....	29
Gambar 3.5 Posisi Agen Pada Formasi Secara Umum	30
Gambar 3.6 Formasi Kotak	31
Gambar 3.7 Deteksi Halangan	31
Gambar 3.8 Pemisahan Pasukan	32
Gambar 3.9 Regrouping Formasi.....	32
Gambar 3.10 Pergerakan Pasukan Dengan Model Boids	33
Gambar 3.11 Agen Terpisah dari Formasi.....	35
Gambar 3.12 Jarak Agen ke Formasi.....	36
Gambar 3.13 Flow Chart PSO	36

Gambar 4.1 Hasil Percobaan Pertama Boids	40
Gambar 4.2 Hasil Percobaan Pertama Boids-PSO	41
Gambar 4.3 grafik Percobaan Pertama Boids-PSO	42
Gambar 4.4 Hasil Percobaan Kedua Boids	43
Gambar 4.5 Hasil Percobaan Kedua Boids PSO	44
Gambar 4.6 Grafik Percobaan Boids PSO	46
Gambar 4.7 Hasil Percobaan Ketiga Boids	47
Gambar 4.8 Hasil Percobaan Ketiga Boids PSO	48
Gambar 4.9 Grafik Percobaan Ketiga Boids PSO	49
Gambar 4.10 Hasil Percobaan Keempat Boids	50
Gambar 4.11 Hasil percobaan keempat boids PSO	52
Gambar 4.12 Grafik percobaan keempat PSO	53
Gambar 4.13 Hasil Percobaan Keempat Boids	54
Gambar 4.14 Hasil Percobaan keempat boids PSO	55
Gambar 4.15 Grafik Percobaan kelima boids PSO	56
Gambar 4.16 Grafik keenam boids PSO	57
Gambar 4.17 Grafik Kompleksitas Boids dan Boids-PSO	61
Gambar 4.18 Loop Program Penelitian	61
Gambar 4.18 Grafik Kompleksitas	62

BAB I

PENDAHULUAN

1.1. Latar Belakang

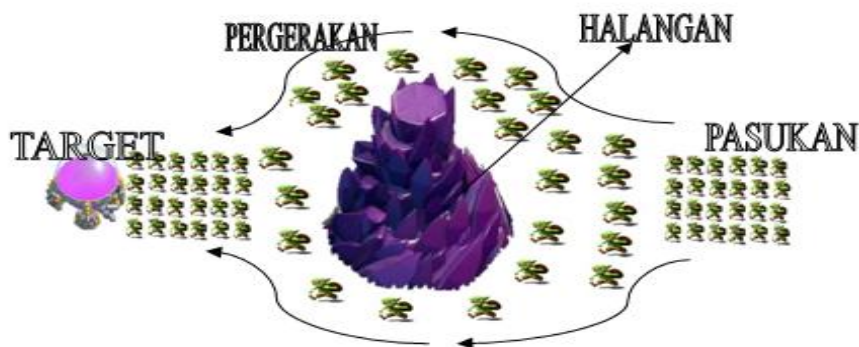
Dalam permainan strategi pemain harus mengendalikan pasukan untuk melakukan pertempuran melawan satu atau lebih lawan. Untuk jenis permainan ini dalam memenangkan permainan, pemain harus menggunakan sumber daya yang dikumpulkan seefisien mungkin dan membangun unit dengan optimal. Beberapa strategi permainan sekarang memungkinkan pasukan untuk bergerak dalam formasi sebagai kelompok yang terorganisir, melalui perencanaan penggunaan jalan tingkat lanjut dan kerja tim[1]. Teknik berkelompok juga dapat digunakan untuk memungkinkan pasukan untuk bergerak bersama-sama dalam formasi terorganisir.

Salah satu contoh game RTS adalah Clash of Clans (CoC) yang dibuat dan disebar oleh Supercell [2]. Pasukan dalam CoC dapat melakukan serangan berkelompok dengan melakukan pergerakan bersama-sama secara bergerombol seperti pada Gambar 1.1.



Gambar 1.1 Pergerakan Pasukan Otonom CoC

Pergerakan berkelompok untuk menuju target dalam permainan CoC terdapat dua tipe yaitu menuju target terdekat dan menuju target tertentu. Pasukan CoC yang menuju target tertentu seperti yang dilakukan oleh goblin, giant, hog rider dan golem dalam menuju target tertentu dapat menghindari halangan yang ada dengan melakukan pemisahan dan setelah melewati halangan kembali bergerombol seperti pada Gambar 1.2.



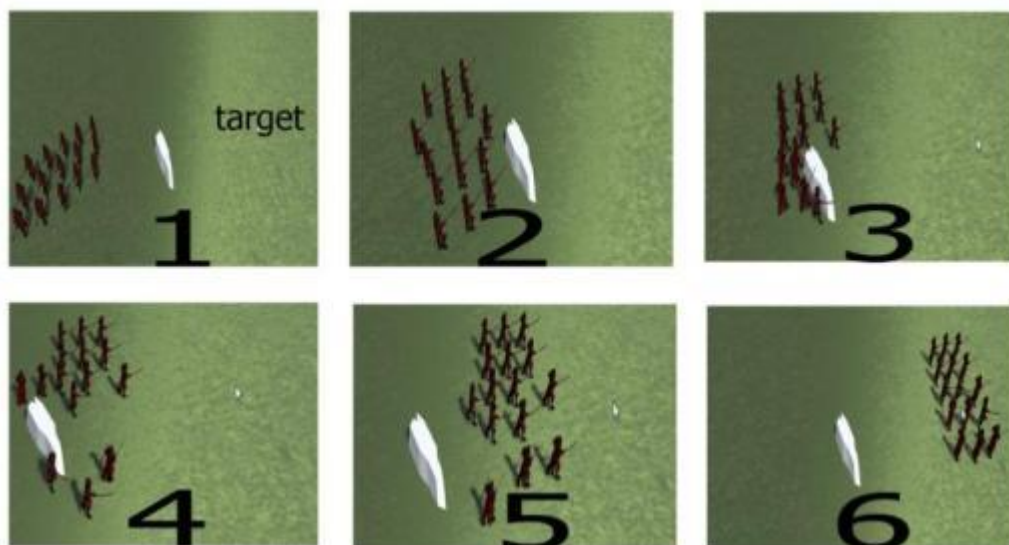
Gambar 1.2. Pergerakan Halangan Statis

Pergerakan pasukan dengan melakukan pemisahan (*separation*), menjaga keselarasan kecepatan (*alignment*) dan menyatu kembali (*cohesion*) adalah program kehidupan buatan yang dikembangkan oleh Craig Reynolds pada tahun 1986 [3] yang mensimulasikan perilaku berkelompok burung dan tiga aturan tersebut disebut dengan algoritma Boids. Menurut Meilany dewi Algoritma Boids dapat digunakan untuk mengendalikan kerumunan agen dalam menghindari halangan statis dan semakin banyak jumlah agen maka semakin lama untuk sampai ke tujuan [4].

Untuk membuat pergerakan pasukan secara otonom yang halus dan fleksibel seperti realita maka diperlukan optimasi dalam pergerakan tersebut. Menurut Ying Ping Chen untuk mengatur perpindahan kerumunan dalam komputer grafis dapat menggunakan particle swarm optimization (PSO) [5]. Setiap agen atau pasukan di inialisasikan dalam bentuk partikel. Setiap partikel menyesuaikan arah bergerak yang sesuai dengan arah keselarasan dan arah kohesi, serta dalam gerakan konvergen, persamaan pembaruan dari versi standar PSO digunakan dalam menentukan gerakan yang konvergen [6]. Dalam simulasi

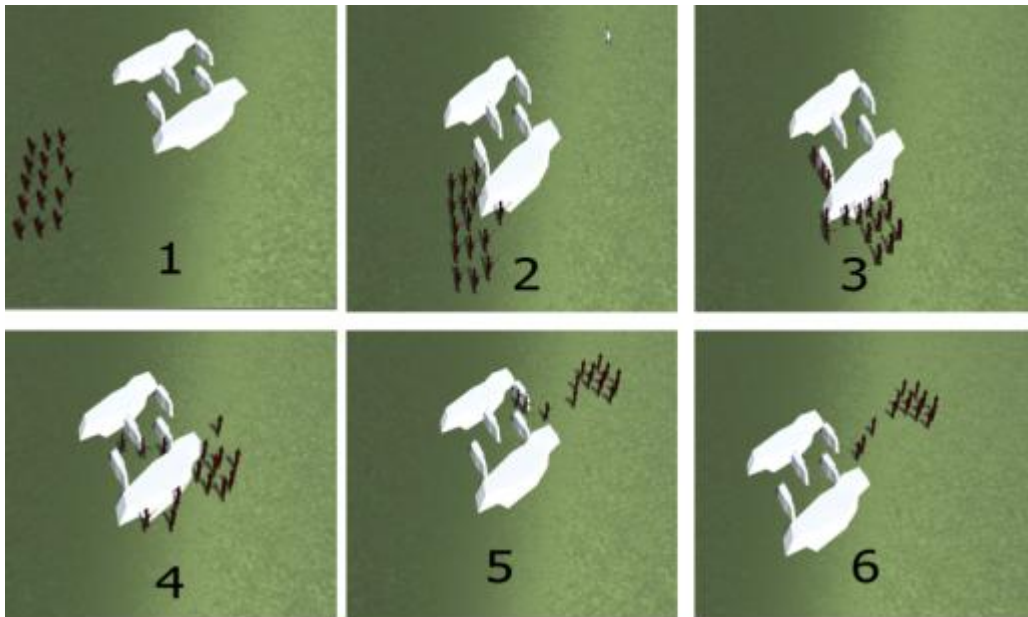
pergerakan pasukan halangan berupa objek sangat dimungkinkan. Setiap pasukan akan berpindah posisi untuk mencari posisi lain yang dapat menghasilkan fungsi tujuan yang lebih baik dengan memanfaatkan komputasi dalam PSO. menurut Saleh Alaliyat flocking boids terlihat lebih baik saat menggunakan PSO dan jauh lebih cepat dari pada menggunakan Algoritma Genetika karena PSO memberikan konvergensi yang lebih cepat dari pada algoritma genetika ketika pasukan berpecah dengan waktu yang lebih lama dalam menemukan posisi optimal dan setelah posisi optimal ditemukan pasukan kembali bergerombol dan menuju tempat tujuan yang telah ditentukan dari pada algoritma genetika [7]..

Pengendalian kerumunan pada algoritma boids jika melewati sebuah halangan masih terlihat halus dan posisi pasukan dapat segera ke formasi semula seperti pada Gambar 1.3. Pada Gambar 1.3 bagian pertama kerumunan pasukan membentuk sebuah formasi kemudian bergerak menuju target yang telah ditentukan. Pada bagian keempat kerumunan pasukan melakukan pemisahan atau *separation* untuk menghindari halangan atau *obstacle avoidance*. Setelah melwati halangan pasukan bergerak menuju pusat kerumunan dan membentuk formasi seperti pada formasi sebelumnya atau yang disebut dengan kohesi. Saat pasukan bergerak bersama setiap pasukan melakukan penyelarasan atau alignmen dengan cara menyelaraskan kecepatan tiap pasukan untuk menuju posisi formasi tiap pasukan.



Gambar 1.3. Pergerakan Boids

Pada Gambar 1.4 pergerakan pasukan tetap menggunakan algoritma boids dengan melakukan pemisahan atau *separation*, *cohesion* dan *alignmen*. Kerumunan pasukan pada saat melewati beberapa halangan untuk menuju target yang telah ditentukan terdapat beberapa agen yang tertinggal seperti pada gambar 4 bagian keempat dan keenam.



Gambar 1.4 Pergerakan Boids dengan Beberapa Halaman

Dalam penelitian ini kami coba mensimulasikan pergerakan pasukan dengan menggunakan algoritma boids berbasis algoritma PSO. Algoritma boids digunakan untuk mengenalkan kerumunan pasukan dalam melakukan pergerakan. Pergerakan pasukan berpindah untuk mencapai tujuan dengan objek statis atau sumber makanan dengan halangan yang dapat berupa objek statis dan bergerak. Untuk dapat menghindari objek tersebut gerombolan pasukan bergerak menghindarinya dengan merubah arah dalam tujuan, pasukan berpencar dengan mencari posisi yang optimal.

1.2. Rumusan Masalah

Kerumunan pasukan dalam permainan RTS bergerak menuju tempat yang telah ditentukan. Untuk menuju tempat tersebut pasukan harus bisa menghindari halangan (obstacle avoidance) yang ada, baik statis maupun dinamis. Model boids

digunakan untuk mengendalikan kerumunan pasukan secara otonom untuk menuju target. Pada saat kerumunan pasukan menghindari beberapa halangan ada beberapa pasukan yang tertinggal sehingga perlu adanya optimasi untuk membuat kerumunan pasukan dapat membentuk formasi kembali setelah melewati halangan.

1.3. Batasan Masalah

Batasan Masalah pada penelitian ini adalah pergerakan pasukan melakukan optimasi penyelarasan atau *velocity matching* saat melewati halangan statis dan dinamis.

1.4. Tujuan

Tujuan dari penelitian ini adalah membentuk formasi kembali (*regrouping formation*) dengan waktu yang lebih cepat dengan PSO karena saat menggunakan boids beberapa pasukan ada yang tertinggal setelah menghindari beberapa halangan.

1.5. Manfaat Penelitian

Pergerakan pasukan dapat melewati halangan pada lingkungan objek secara otonom dan sampai ditempat tujuan dengan formasi yang boids pasukan pada saat di arahkan perpindahannya.

halaman ini sengaja dikosongkan

BAB II

KAJIAN PUSTAKA

2.1. Permainan *Real Time Strategi* (RTS)

Permainan RTS merupakan sub genre strategi video game yang tidak berkembang secara bertahap dan bergantian[1]. Dalam kebanyakan permainan RTS, *gameplay* umumnya cepat dan membutuhkan refleks sangat cepat. Untuk alasan ini, jumlah kekerasan dalam beberapa permainan membuat permainan RTS dekat dengan permainan aksi. *Gameplay* dalam permainan RTS umumnya terdiri dari pemain yang diposisikan di suatu tempat di peta dengan beberapa unit atau bangunan yang mampu membangun unit lain. Tidak selalu permainan RTS mengharuskan pemain untuk membangun tentara (mulai dari regu kecil tidak lebih dari 2 unit, untuk ratusan unit) dan menggunakan mereka untuk mempertahankan diri baik dari bentuk virtual serangan gelombang pasukan atau menghilangkan musuh yang memiliki basis dengan kapasitas produksi unit mereka sendiri. Kadang-kadang permainan RTS akan memiliki sejumlah unit untuk pemain dan untuk mengontrol pembangunan unit.

Pengumpulan sumber daya umumnya fokus utama dari game RTS, tapi judul lain dari *gameplay* signifikansi tempat bergenre tinggi untuk unit digunakan dalam pertempuran, contoh ekstrim yang permainan real-time bergenre taktis. Beberapa judul memaksakan langit-langit pada pasukan simultan nomor, yang menjadi pertimbangan kunci permainan. Contoh permainan real time strategi berbasis mobile yaitu *clash of clans* (CoC).

2.2. Pergerakan Pasukan Pada *Clash of Clan* (CoC)

Clash of Clans adalah permainan populer di IOS dan android yang dibuat oleh "Supercell". Telah tersedia secara internasional di iTunes secara gratis sejak v1.7 rilis pada tanggal 2 Agustus 2012, dan pada Google Play Store sejak 8 Oktober 2013. *Clash of Clans* adalah permainan strategi di mana pemain dapat membangun dan memperluas desa seseorang, membuka prajurit berturut-turut lebih kuat dan

pertahanan, serangan dan penjarahan sumber daya dari desa lain, membuat dan bergabung Clan dan banyak lagi. Meskipun Clash of Clans adalah Free-to-Play, tambahan mata uang dalam game dapat dibeli dengan uang sungguhan dari Apple App Store / Google Play Store dalam bentuk In-App Pembelian. Pasukan dalam CoC dikelompokkan dalam beberapa kelompok yaitu : Tier 1, Tier 2, Tier 3, Dark Elixir dan Heroes

Pasukan pada tier 1 cepat untuk dibuat dan relatif murah. Mereka memiliki nyawa dan hitpoint yang rendah dalam menimbulkan kerusakan jika digunakan secara individu. Pada umumnya, mereka digunakan dalam kawanan besar dan membanjiri musuh dalam jumlah bukan kekuasaan. Hal ini membuat mereka kuat dalam jumlah besar untuk melawan pertahanan musuh dengan target tunggal karena pertahanan yang hanya dapat membunuh satu unit musuh pada suatu waktu sementara pasukan tier 1 menghancurkan segalanya. Namun, mereka tidak juga cocok untuk pertahanan yang dapat menghancurkan pasukan tier 1 dalam satu tembakan seperti canon dan wizard tower.

Pasukan yang ada di Tier 2 lebih kuat daripada troop di Tier 1. Sama pada troop di Tier 1, troops pada Tier 2 juga memiliki keahlian khusus. Seperti Wall Breaker yang hanya bertugas menghancurkan dinding, Wizards meskipun tidak terlalu kuat namun mampu menembakkan api pada musuh dengan kehancuran yang besar dan berpengaruh pada semua area tembak wizard. Giant dan Ballons dirancang menyerang pertahanan musuh sebagai target utama. Ballons merupakan unit terbang pertama yang tersedia di CoC.

Pasukan yang ada pada Tier 3 merupakan pasukan non hero yang paling kuat dan memiliki paling banyak nyawa dalam CoC. Mereka mampu menghancurkan semua desa hanya dengan sedikit pasukan. Namun, troops di Tier 3 membutuhkan biaya yang mahal dan waktu yang lama untuk melatihnya. Pada umumnya, kekuatan 6 Barbarian atau Archer di level 6 setara dengan kemampuan troop di Tier 3 ini. Keunggulan lain troop di Tier 3 adalah mereka tidak mudah dikalahkan oleh Mortars dan Wizard Tower.

Kelompok troop Dark Elixir adalah troops yang memiliki fungsi khusus yang tidak dimiliki oleh troop biasa, seperti troop terbang, perusak dinding, troop yang bertugas membuat kerusakan darat dengan menggunakan senjata yang dilempar dan kekuatan besar. Troop yang menjadi anggota kelompok Dark Elixir adalah Minions, Hog Riders, Valkyries, Golems, Witches dan Lava Hounds.

Berikutnya adalah kelompok pasukan Heroes yang terdiri dari Barbarian King dan Archer Queen. Mereka adalah pasukan yang kekal atau immortal. Mereka memiliki kekuatan terbesar namun setiap pemain hanya memiliki 1 pasukan Heroes saja. Meskipun Heroes tidak dapat mati, tapi mereka tetap dapat terluka dan membutuhkan waktu istirahat untuk menyembuhkan luka mereka. Saat melakukan serangan dan untuk menjaga agar Heroes tidak terluka, Heroes perlu dibantu oleh pasukan yang lain. Pasukan yang biasa digunakan untuk membantu serangan Heroes adalah Barbarian dan Archer. Setiap Barbarian King dan Archer Queen dapat mengeluarkan Barbarian dan Archer untuk membantu mereka menyerang lawan.

Berdasar medan gerakannya, ada 2 jenis pasukan dalam CoC, yaitu unit darat dan unit udara. Semua unit udara dalam CoC tidak mempunyai kemampuan untuk menghindari serangan yang diberikan padanya. Mereka akan bergerak langsung dari posisi awal mereka dideploy menuju posisi target tanpa memperhatikan arah serangan lawan. Hal ini membuat troop unit terbang mudah ditembak. Sedangkan unit darat bergerak sesuai dengan tugas yang diberikan padanya.

Ada 2 unit darat berdasarkan targetnya, yaitu unit darat yang memiliki target khusus dan unit darat yang tidak memiliki target khusus. Unit darat yang memiliki tujuan khusus akan melewati semua bangunan yang bukan targetnya. Jika saat bergerak menuju target unit darat terbentur dengan dinding, maka mereka akan menghancurkan dinding tersebut sebelum menuju bangunan targetnya. Unit darat yang tidak memiliki tujuan khusus akan menghancurkan setiap bangunan lawan yang ditemuinya. Penelitian ini akan meneliti pergerakan kelompok troop yang mempunyai tujuan khusus menghancurkan pertahanan lawan dan sumber daya lawan. Hog Rider, Giant, golem adalah jenis pasukan yang mempunyai tujuan

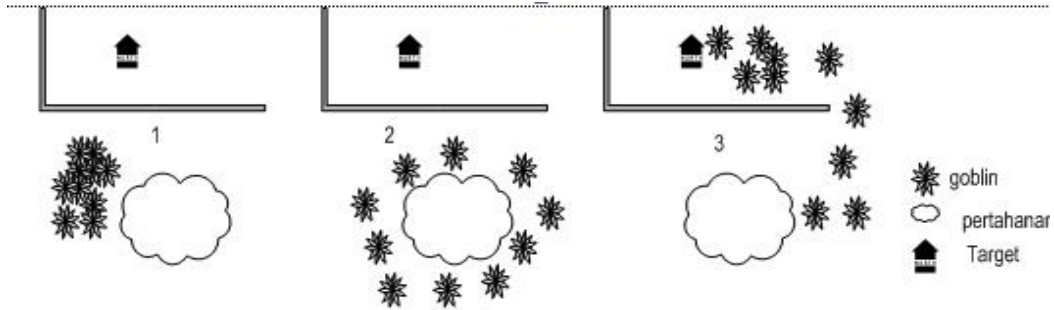
menghancurkan pertahanan lawan. sedangkan Goblin mempunyai misi menghancurkan bangunan penyimpan sumber daya lawan.

Karakter Goblin digambarkan sebagai makhluk hijau kecil dengan telinga runcing lebar, mata merah dengan pupil berwarna hijau dan hidung merah. Goblin memakai celana dan sepatu coklat serta membawa karung yang berisi sumber daya yang dicuri dari lawan. Goblin merupakan karakter tercepat dalam CoC. Bangunan sumber daya adalah target utama Goblin, seperti Gold, Elixir dan penyimpanan. Karena Goblin mengutamakan menyerang bangunan sumber daya, maka mereka akan mengabaikan semua pertahanan lawan, rentan terhadap serangan dan lemah jika digunakan tanpa back up dari troop lain. Gambar 2.1 adalah perubahan karakter Goblin di beberapa level game CoC.



Gambar 2.1 Karakter Goblin

Saat dikirim secara berkeompok, dari posisi awal mereka Goblin akan langsung mencari posisi bangunan sumber daya terdekat. Mereka akan menuju bangunan sumber daya yang sama dan mengabaikan bangunan lain. Jika bangun sumber daya yang diserbu pertama kali sudah mereka hancurkan, maka mereka akan pecah menjadi beberapa kelompok. Masing-masing kelompok akan mencari sumber daya lain yang bisa ditemukan. Jika sumber daya ada di balik dinding, maka Goblin harus menghancurkan dinding itu. Gambar berikut adalah ilustrasi pergerakan Goblin dari posisi awal dikirim ke posisi sumber daya yang diincar.



Gambar 2.2 Ilustrasi Pergerakan Goblin Menuju Target

Giant dan golem adalah pasukan berbadan besar, gerakan agak lambat dan memiliki hit point yang tinggi terdapat pada Gambar 2.3 dan Gambar 2.4. Troop ini mengincar pertahanan musuh seperti Canon, Wizard Tower dan Archer Tower. Dalam jumlah yang besar mereka dapat menghancurkan sebuah desa. Karena hitpoint Giant sangat besar, pada umumnya player menempatkan mereka terlebih dahulu untuk melindungi pasukan lain yang lebih lemah.

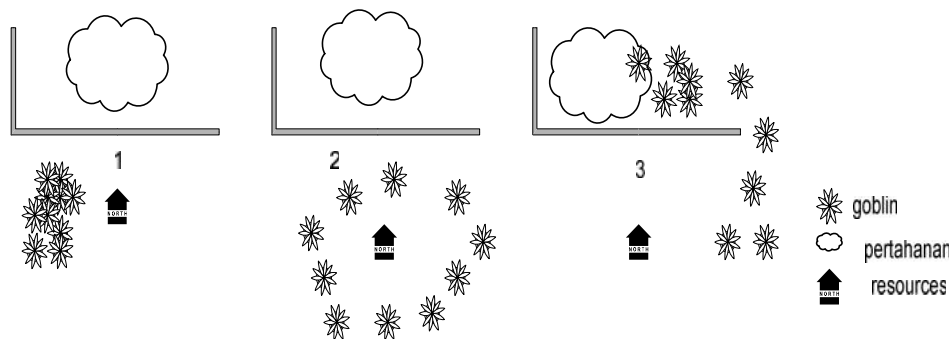


Gambar 2.3 Karakter Giant



Gambar 2.4 Karakter Golem

Hampir sama dengan Goblin, Giant juga dapat menyerang dalam kelompok. Mereka cenderung akan menyerang bangunan yang sama terlebih dahulu. Jika bangunan yang pertama dituju sudah berhasil dihancurkan, maka mereka juga akan dibagi menjadi beberapa kelompok untuk mencari target bangun pertahanan lain untuk dihancurkan. Saat bergerak menuju target, Giant tidak diberi kecerdasan untuk menghindari serangan lawan. Selain itu Giant juga tidak diberi kecerdasan untuk menjaga jarak antar agen, sehingga saat menggerombol di tempat yang sama mereka akan terlihat saling tumpang tindih. Gambar 2.5 adalah ilustrasi pergerakan Giant dari posisi awal mereka dideploy sampai menuju target.



Gambar 2.5 Ilustrasi Pergerakan giant dan golem Menuju Target

2.3 Swarm Intelligent

Swarm Intelligence (SI) merupakan cabang dari *Artificial Intelligence* (AI) yang digunakan untuk memodelkan perilaku kolektif dari hewan social di alam seperti koloni semut, lebah atau burung. Meskipun setiap agen dalam swarm cenderung memiliki kemampuan yang sederhana tetapi mereka saling berkomunikasi dan melakukan pembagian tugas yang jelas untuk kelangsungan hidup mereka. Interaksi social antar anggota swarm dapat dilakukan secara langsung maupun tidak langsung. Interaksi langsung dilakukan dengan melakukan kontak visual atau suara, seperti pada saat lebah pekerja (*employed bee*) mempresentasikan informasi tentang jumlah *nectar* pada *food source* yang dikunjunginya di lantai dansa (*dance area*) dengan melakukan tarian (*waggle dance*) yang ditujukan pada lebah pengamat (*onlooker bee*). Sedangkan interaksi tidak langsung terjadi pada saat ada anggota





kelompok yang melakukan perubahan pada lingkungannya dan anggota yang lain berperilaku sesuai perubahan lingkungan itu. Contoh interaksi tidak langsung adalah ketika semut berkomunikasi dengan temannya dengan meninggalkan jejak feromon pada jalur yang dia lewati saat menuju sumber makanan, kemudian semut lain akan bergerak mengikuti jejak feromon itu [8].

Tidak semua kelompok agen dapat dikatakan memiliki kecerdasan (*intelligent*). Kelompok agen dapat dikatakan cerdas jika memiliki *self-organization* dan pembagian tugas yang jelas. *Self-organization* adalah ciri utama dari sistem *swarm* yang menghasilkan perilaku kolektif yang berasal dari interaksi lokal antar agen (Bonabeau et al. 1999). 4 karakteristik *self-organization* menurut Bonabeau adalah :

1. *Positive feedback* : tiap anggota kelompok merasa nyaman berada di dalam kelompoknya. Sebagai contoh, proses rekrutmen dan penguatan formasi koloni semut saat mengikuti jejak yang telah dibentuk oleh anggota di depannya.
2. *Negative feedback* : sebagai penyeimbang *positive feedback* dan membantu menstabilkan pola formasi koloni. *Negative feedback* diperlukan untuk menghindari kejenuhan yang mungkin terjadi pada saat proses pencarian makanan.
3. *Fluctuations* : berjalan secara acak, melakukan kesalahan, saling bertukar tugas secara acak sesama anggota koloni merupakan kreatifitas yang penting. *Randomness* (mengacak) kadang kala diperlukan untuk menemukan solusi baru.
4. *Multiple interactions* : anggota koloni menggunakan informasi yang datang dari anggota koloni lain sehingga informasi menyebar ke seluruh jaringan.

Untuk menyempurnakan 4 karakteristik *self-organization* di atas, Bonabeau menambahkan bahwa pembagian tugas untuk menyelesaikan pekerjaan secara bersama-sama juga merupakan ciri penting sebuah koloni dapat disebut memiliki kecerdasan [9].

Tabel 2.1 Swarming dari burung, ikan, domba, lebah

No.	Nama	<i>Swarming</i>
1	Burung	
2	Ikan	
3	Domba	
4	Lebah	

2.4 Perilaku Kelompok

Perilaku berkelompok adalah perilaku yang ditunjukkan ketika sekelompok agen bergerak bersama sama menuju sebuah tujuan tertentu dalam foramsi tertentu. Pergerakan kelompok ini meniru pergerakan kelompok makhluk yang ada di alam. . Ada persamaan dengan perilaku kawanan burung, perilaku kawanan ikan, perilaku kerumunan serangga, dan perilaku kawanan hewan darat.

NPC atau *Non Playable Character* adalah karakter dalam game yang perilakunya tidak dikontrol oleh manusia/*player*. Perilaku NPC dibagi menjadi 3, yaitu strategis (*strategic*), taktik (*tactical*) dan reaktif (*reactive*). Perilaku strategi digunakan untuk mencapai tujuan jangka panjang, misalnya mengamankan wilayahnya. Setiap NPC selalu memiliki tujuan jangka panjang dan jangka pendek. Perilaku taktis digunakan untuk mencapai tujuan jangka pendek yang lebih spesifik lagi. Sedangkan perilaku reaktif adalah reaksi sederhana sesuai dengan persepsi audio visualnya pada saat itu, seperti melompat, berjalan, membidik atau menembak[10].

Dalam game pergerakan kelompok digunakan untuk berbagai macam pergerakan NPC maupun player yang bertujuan permainan dalam game lebih menarik. Menyerang maupun bertahan adalah salah satu aktifitas dalam game yang berjenis RTS seperti contohnya game Clash of Clan. Aktifitas ini membutuhkan formasi pasukan dalam bentuk kelompok untuk melakukan serangan terhadap musuh maupun bertahan terhadap musuh ataupun melarikan diri dari musuh. Agen didalam kelompok tersebut diharapkan mempunyai kecerdasan sehingga dapat menentukan pergerakannya dalam melakukan aktifitas menyerang, bertahan, maupun melarikan diri. Jika diambil contoh dari satu aktifitas yaitu menyerang maka dalam melakukan penyerangan agen bisa menyerang sendirian maupun menyerang dalam formasi kelompok. Dalam serangan kelompok, agar pergerakan dapat sukses melakukan serangan, terlihat almah dan tidak mudah ditebak, maka *swarm inteligent* perlu ditanamkan di agen dalam kelompok

Simulasi komputer dan model matematika yang telah dikembangkan untuk meniru perilaku berkelompok burung secara umum dapat diterapkan juga untuk perilaku berkelompok spesies lain. Algoritma Boid banyak digunakan untuk mensimulasikan pergerakan kelompok agen, yang sering terlihat dalam perilaku kawanan burung atau kawanan ikan dalam melakukan pergerakan dalam kelompoknya. Seperti halnya program simulasi sekelompok agen, maka setiap agen berbasis boid memiliki perilaku yang berbeda yang muncul akibat reaksi buatan yang dibuat dan ditanamkan didalamnya.

Dari perspektif modeller matematika, berbondong bondong adalah gerakan kolektif dari sejumlah besar individu dalam kelompok dan merupakan perilaku hewan kolektif yang dipunyai oleh banyak makhluk hidup seperti burung, ikan, domba, dan serangga

Dalam game strategi (RTS), player harus mengontrol pasukan untuk berperang melawan musuh. Emas, elixir atau trofi dikumpulkan player untuk menguatkan unit, membangun hall/barak dan merakit pesawat atau aset lain agar menang saat perang melawan musuh. Pengumpulan emas, elixir atau trofi dapat dilakukan dengan menyerang unit lawan atau menambang di tambang emas/elixir. Jika pengumpulan emas/elixir diperoleh dari hasil mengalahkan lawan, maka player harus mengkoordinir pasukan saat melakukan serangan pada lawan.

Ada 2 jenis game strategi, yaitu *turn based* dan *real time*. Pada *turn-base strategy games* player dan lawan secara bergantian mengeluarkan perintah untuk unit mereka, seperti bermain catur. Baik player maupun lawan bergantian menjalankan pasukannya. Sedangkan *real time strategy game* player dan computer mengkoordinir pasukannya secara bersamaan (real-time).

Ada 2 jenis NPC dalam game strategy, yaitu *strategic* NPC dan *unit* NPC. Strategic NPC digunakan untuk mengendalikan tentara lawan. Mereka harus bisa mengatur strategi sama seperti yang player lakukan. Mereka juga mengumpulkan sumber daya yang ada dalam environment selama game dijalankan. Strategic NPC harus dapat melakukan melee attack sebgus player, tetapi di akhir game mereka

harus membiarkan player yang menang.

Sedangkan Unit NPC adalah tentara atau karakter tunggal yang menjadi anggota pasukan player maupun strategic NPC. Artinya, unit NPC ini pasukan yang digerakkan oleh player maupun strategic NPC. Unit NPC harus memiliki kecerdasan agar dapat melaksanakan perintah player maupun perintah strategic NPC. Perintah itu dapat berupa perintah menyerang, mengumpulkan sumber daya atau membangun gedung. Unit NPC juga harus mampu merencanakan rute jalan dan mengikuti rute itu untuk mencapai target mereka dan untuk melaksanakan tugas mereka secara efektif sementara pada saat yang bersamaan mereka juga harus bereaksi terhadap perubahan lingkungan [10].

Pada umumnya, untuk mengarahkan unit NPC player harus men-deploy mereka ke lokasi yang ditentukan oleh player. Sebagai contoh kasus, pada game CoC untuk mengirimkan pasukan menyerbu musuh yang memasuki wilayah, player harus mendeploy pasukan dulu ke lokasi musuh. Sehingga, jika player tidak mendeploy unit NPC untuk menyerang musuh yang masuk ke wilayahnya maka dengan mudah unit NPC musuh dapat menghancurkan wilayah itu. Gambar 2.6 adalah gerakan tangan player menyentuh tab saat dia mengirim pasukannya secara manual ke lokasi yang dia inginkan.

Untuk menghindari hal tersebut, dibutuhkan unit NPC berbasis agen otonom yang dapat bereaksi secara otonom terhadap setiap serangan yang masuk ke wilayah player. Artinya, dengan adanya NPC berbasis agen otonom player tidak perlu secara manual mendeploy pasukan ke lokasi musuh, karena unit NPC player akan menyerang secara otomatis setiap NPC lawan yang masuk ke wilayahnya.

Agen otonom adalah sebuah sistem komputer yang hidup di dalam lingkungan buatan. Dia beraksi sesuai dengan agenda yang ditanamkan padanya sehingga dia bisa tahu harus melakukan apa jika menerima suatu rangsangan. (Franklin dan Graesser 1997). [11]

Agen otonom harus mampu beraksi sesuai dengan sensor yang dia terima. Artinya agen tidak hanya berada dan menjadi bagian dari sebuah lingkungan buatan,

tetapi menjadi pasangan dari lingkungan buatan itu. Arsitektur dan mekanisme agen harus terhubung dengan lingkungannya sehingga dapat merasakan setiap tujuan yang dirancang untuknya dan beraksi untuk memenuhi tujuan itu. (Maturana 1975, Maturana dan Varela 1980, Varela 1991). [11]



Gambar 2.6. Gerakan Jari pada Tab Saat Deploy Pasukan

Agen otonom sesuai dengan namanya harus bersifat otonom, mandiri, reaktif, proaktif. Biasanya agen otonom terhubung dengan suatu server, sehingga mereka dapat berkomunikasi dengan agen lain dengan bahasa yang telah disepakati antar agen itu (ACL = *Agent Communication Language*). Maksud dari sifat otonom adalah agen dapat bertindak tanpa adanya control dari manusia atau intervensi lain sesuai dengan kecerdasan buatan yang ditanamkan padanya. Agen juga harus bersifat reaktif, artinya agen mampu secara terus-menerus berinteraksi dengan lingkungannya dan mampu memberikan respon yang tepat terhadap setiap perubahan yang terjadi di lingkungannya. Selain bersifat reaktif agen juga harus pro-aktif, artinya agen harus mampu mengambil inisiatif sendiri, tidak menunggu sesuatu terjadi sesuatu terlebih dahulu baru bertindak. Sebagai contoh sifat pro aktif agen adalah jika ada musuh memasuki wilayahnya, agen tidak perlu menunggu musuh melepaskan tembakan dulu baru menyerang, tetapi agen akan proaktif menyerang

setiap musuh yang memasuki wilayahnya [10]. Selanjutnya pada penelitian ini akan digunakan istilah agen untuk menyebut NPC berbasis agen otonom.

Setiap agen dalam game selalu memiliki tujuan (*goal*), seperti : menyerang musuh, tetap hidup, menangkap avatar. Agen atau aotonomous NPC juga mampu merasakan (*sensing*) perubahan pada lingkungannya, seperti kemampuan melihat halangan, mendengarkan suara. Dan agen juga dapat bertindak (*acting*) sesuai perubahan lingkungan yang ditemuinya, seperti : melompat untuk menghindari halangan, makan. agen diberi semacam indera untuk dapat mengindra lingkungannya dan untuk berkomunikasi dengan agen lain. Gambar 2.3 adalah ilustrasi interaksi agen/NPC dengan lingkungannya.

Saat bergerak di dalam lingkungannya, sekelompok agen harus mengatur pergerakannya agar tidak berbenturan dengan agen lain. *Steering behavior* bertujuan untuk membantu agen bergerak secara realistik di lingkungan buatan di mana agen hidup. Pergerakan ini diciptakan mengacu pada informasi lokal tentang posisi agen lain yang berdekatan dengannya.

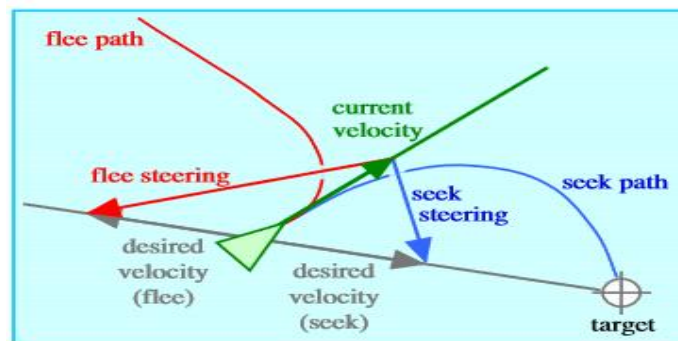
2.5. Perilaku Kendali Agen Otonom

Untuk menghasilkan pergerakan pasukan yang realistis, kita harus mengetahui terlebih dahulu karakteristik dasar dari pasukan atau agen tersebut. Karakteristik agen atau pasukan dapat dinyatakan sebagai berikut [12]:

- Perilaku (*behaviour*) : kemampuan respon perilaku inividu, kelompok atau spesies terhadap lingkungan.
- Kecerdasan (*Intelegant*) : kemampuan kecerdasan untuk mengerti kondisi baru.
- Otonom (*autonomy*) : kemampuan untuk mengendalikan diri.
- Adaptasi (*adaptation*) : kemampuan bertahan hidup dari lingkungan yang tidak dapt diprediksi.
- Persepsi (*Perseption*); kemampuan kesadaran terhadap lingkungan berdasarkan fisik.

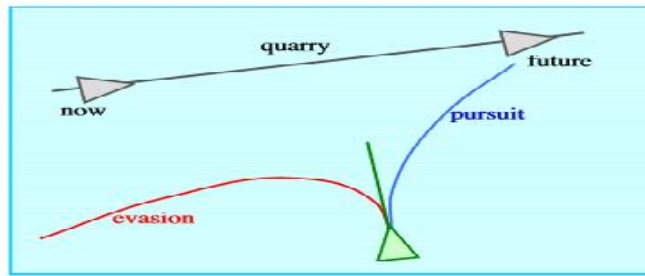
- Daya ingat (*memory*): proses memanggil kembali sesuatu yang pernah dikerjakan.
- Emosi (*emotion*) aspek reaktif dari kesadaran dan perasaan.
- Kesadaran (*Consciousness*): kualitas dari kesadaran dengan sensasi, emosi, kemauan dan gagasan.
- Kebebasan (*freedom*): kemampuan perilaku agen yang tidak dapat diprediksi

Perilaku sendiri dapat dinyatakan sebagai sekumpulan aksi dari manusia atau hewan yang didasari oleh kemauannya [13]. Sekumpulan aksi menjadi perilaku kendali untuk melakukan pergerakan secara bergerombol dengan tujuan yang telah ditentukan. Perilaku seek memungkinkan unit atau agen untuk bergerak menuju titik. Perilaku flee adalah kebalikan dari perilaku seek unit lari dari titik tertentu . Pada efek dari seek dan flee perilaku yang akan ditampilkan dalam Gambar 2.7.



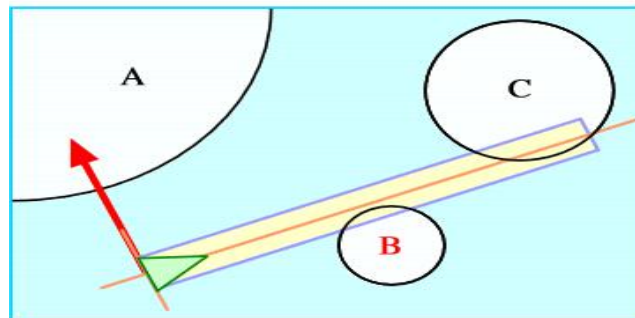
Gambar 2.7 seek dan flee

Pursuit itu mirip dengan perilaku seek tapi fakta bahwa itu berlaku untuk titik bergerak agen lain. Perilaku ini memperhitungkan prediksi pergerakan titik untuk mengikuti. Perilaku menghindar memungkinkan agen untuk melarikan diri dari titik bergerak. Dalam Gambar 2.8 perilaku evasion dan pursuit dibandingkan. Sebuah perilaku yang sedikit berbeda adalah perilaku pengejaran offset yang memungkinkan unit untuk mengejar unit lain menjaga jarak tertentu offset antara mereka .



Gambar 2.8 *pursuit dan evasion*

Perilaku penghindaran rintangan memberikan karakter kemampuan untuk manuver dalam lingkungan yang berantakan dengan menghindari sekitar hambatan. Ada perbedaan penting antara menghindari rintangan dan perilaku melarikan diri. *Flee* akan selalu menyebabkan karakter untuk menjauhkan diri dari lokasi tertentu, sedangkan menghindari rintangan mengambil tindakan hanya jika hambatan terdekat terletak langsung di depan karakter dalam Gambar 2.9

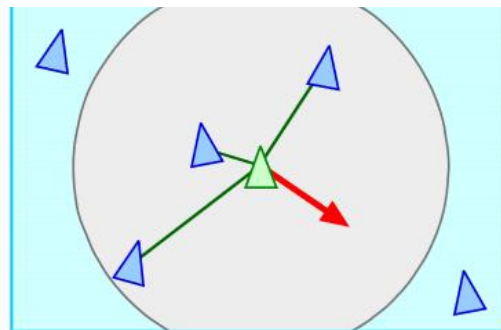


Gambar 2.9 Obstacle Avoidance

2.6. Model Boids

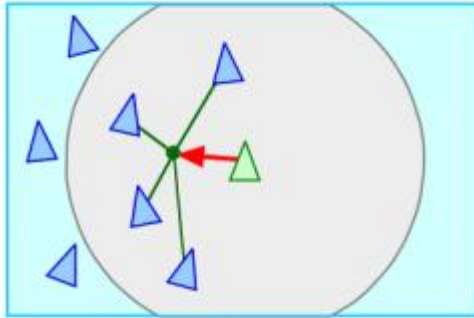
Boids adalah program kehidupan buatan , yang dikembangkan oleh Craig Reynolds pada tahun 1986 , yang mensimulasikan perilaku berkelompok burung [5]. Algoritma ini menunjukkan struktur dari implementasi khas boidss. Tiga aturan algoritma boidss terdapat perubahan kecepatan dan arah boids. Setelah kecepatan dan jalannya boidss telah diperbarui oleh aturan Reynolds kita dapat memperbarui posisi dari boidss.

Perilaku pemisahan kemudi memberikan karakter kemampuan untuk menjaga jarak pemisahan tertentu dari karakter lain di dekatnya. Hal ini dapat digunakan untuk mencegah karakter dari berkerumun bersama-sama . Untuk menghitung kemudi dalam pemisahan, pertama pencarian dilakukan untuk menemukan karakter lain dalam lingkungan tertentu . Ini mungkin pencarian yang lengkap dari semua karakter dalam dunia simulasi, atau menggunakan semacam partisi ruang dalam skema caching untuk membatasi pencarian ke karakter lokal . Untuk masing-masing karakter di dekatnya, gaya tolak dihitung dengan mengurangi posisi karakter kita dan karakter di dekatnya, normalisasi , dan menerapkan bobot $1/r$ yang artinya, posisi vektor offset skala oleh $1/r^2$. Perhatikan bahwa $1/r$ hanya pengaturan yang telah bekerja dengan baik , bukan nilai fundamental . Lihat Gambar 2.10



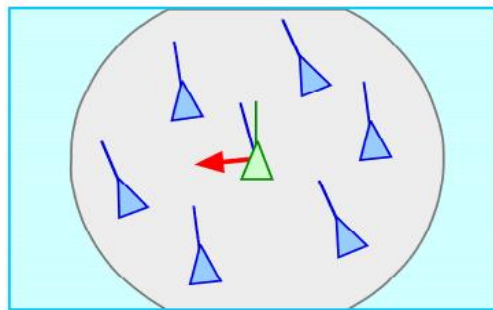
Gambar 2.10 Separation

Perilaku kemudi cohesion memberikan karakter kemampuan untuk berpadu dengan karakter lain di dekatnya. Kemudian untuk kohesi dapat dihitung dengan menemukan semua karakter dalam lingkungan lokal seperti dijelaskan di atas untuk pemisahan, menghitung posisi rata-rata atau pusat gravitasi karakter di dekatnya. Kekuatan kemudi dapat diterapkan dalam arah posisi rata-rata dalam mengurangi posisi karakter kita dari posisi rata-rata, seperti dalam boidss model asli, atau dapat digunakan sebagai target untuk mencari perilaku kemudi. Lihat Gambar 2.11



Gambar 2.11 *Cohesion*

Perilaku kemudi keselarasan memberikan karakter kemampuan untuk menyesuaikan diri karakter lain di dekatnya, seperti yang ditunjukkan pada Gambar 2.12. Kemudi untuk penyelarasan dapat dihitung dengan menemukan semua karakter dalam lingkungan lokal seperti dijelaskan di atas untuk pemisahan, rata-rata bersama kecepatan atau bergantian, vektor satuan ke depan karakter di dekatnya. Rata-rata ini adalah kecepatan yang diinginkan dan sebagainya vektor kemudi adalah perbedaan antara rata-rata dan kecepatan karakter kita saat ini atau bergantian. Kemudi ini akan cenderung untuk mengubah karakter kita sehingga sejajar dengan kelompok tetangga.



Gambar 2.12 *Alignment*

2.7. Particle Swarm Optimization

Swarm intelligence (SI) berasal dari studi koloni, atau kawanan organisme sosial. Studi tentang perilaku sosial organisme (individu) di kawanan diminta desain optimasi dan clustering algoritma yang sangat efisien. Sebagai contoh, studi simulasi anggun, namun tak terduga, koreografi ternak burung menyebabkan desain algoritma

optimasi segerombolan partikel, dan studi dari mencari makan yang perilaku semut mengakibatkan algoritma semut. Optimasi partikel swarm (PSO) adalah pendekatan optimasi stokastik, mencontoh perilaku sosial kawanan burung. PSO adalah prosedur pencarian berbasis populasi di mana individu, disebut sebagai partikel, dikelompokkan menjadi ribuan. Setiap partikel dalam segerombolan merupakan solusi kandidat untuk masalah optimasi.

Dalam PSO sebuah sistem, setiap partikel "diterbangkan" melalui ruang pencarian multidimensi, menyesuaikan posisinya di ruang pencarian menurut pengalaman sendiri dan dari tetangga partikel. Oleh karena itu partikel A memanfaatkan posisi terbaik dihadapi dengan sendirinya dan posisi terbaik dari tetangganya untuk memposisikan diri ke arah solusi optimal. Efeknya adalah bahwa partikel "terbang" ke arah yang optimal, sementara masih mencari wilayah yang luas sekitar solusi terbaik saat ini. Kinerja masing-masing partikel (yaitu "kedekatan" dari partikel ke minimum global) diukur menurut kebugaran yang telah ditetapkan Fungsi yang terkait dengan masalah yang dipecahkan. Aplikasi PSO termasuk fungsi pendekatan, clustering, optimasi struktur mekanik, dan pemecahansistem persamaan.

Studi dari koloni semut telah memberikan kontribusi dalam kelimpahan dengan set algoritma cerdas. Pemodelan feromon menyetorkan oleh semut dalam pencarian mereka untuk terpendek jalan untuk sumber makanan mengakibatkan pengembangan algoritma optimasi jalur terpendek. Aplikasi lain dari optimasi koloni semut termasuk optimasi routing yang dalam jaringan telekomunikasi, pewarnaan graf, penjadwalan dan memecahkan kuadrat masalah penugasan. Studi bangunan sarang semut dan lebah mengakibatkan pengembangan clustering dan algoritma optimasi struktural.

Particle Swarm Optimization adalah salah satu metode optimasi yang terinspirasi dari perilaku model boids yang kemudian tiap objek disederhanakan menjadi sebuah partikel. Suatu partikel dalam ruang memiliki posisi yang dikodekan sebagai vektor koordinat. Vektor posisi ini dianggap sebagai keadaan yang sedang ditempati oleh suatu partikel di ruang pencarian. Setiap posisi dalam ruang pencarian

merupakan alternatif solusi yang dapat dievaluasi menggunakan fungsi objektif. Setiap partikel bergerak dengan kecepatan v . Particle Swarm Optimization pertama kali dimunculkan pada tahun 1995 diciptakan oleh James Kennedy dan R.C Eberhart[14], sejak saat itulah para peneliti banyak menurunkan dan mengembangkan metode PSO.

Sistem PSO diinisialisasi oleh sebuah populasi solusi secara acak dan selanjutnya mencari titik optimum dengan cara memperbarui tiap hasil pembangkitan. Metode optimasi yang didasarkan pada *swarm intelligence* ini disebut algoritma *behaviorally inspired* sebagai alternatif dari algoritma genetika, yang sering disebut *evolution-based procedures*. Dalam konteks optimasi multivariabel, kawanan diasumsikan mempunyai ukuran tertentu atau tetap dengan setiap partikel posisi awalnya terletak di suatu lokasi yang acak dalam ruang multidimensi. Setiap partikel diasumsikan memiliki dua karakteristik posisi dan kecepatan. Setiap partikel bergerak dalam ruang tertentu dan mengingat posisi terbaik yang pernah dilalui atau ditemukan terhadap sumber makanan atau nilai fungsi objektif. Setiap partikel menyampaikan informasi atau posisi bagusnya kepada partikel yang lain dan menyesuaikan posisi dan kecepatan masing-masing berdasarkan informasi yang diterima mengenai posisi yang bagus tersebut.

Dengan demikian perilaku kawanan burung akan didasarkan pada kombinasi dari 3 faktor simpel berikut:

1. Kohesi - terbang bersama
2. Separasi - jangan terlalu dekat
3. Penyesuaian (*alignment*) - mengikuti arah bersama

Partikel-partikel PSO bergerak melalui penelusuran ruang menuju suatu potensial solusi. Pergerakan PSO dipengaruhi oleh nilai *velocity* yang dinamis. Nilai *velocity* ini dipengaruhi oleh tiga faktor, yaitu nilai *velocity* partikel sebelumnya, posisi partikel terbaik dalam populasi (Gbest) dan posisi terbaik yang pernah diraih oleh partikel itu sendiri (Pbest). Pada awalnya persamaan *velocity* PSO adalah seperti berikut ini:

$$v = v + c1 (pbest-x) + c2 (gbest-x)$$

Keterangan:

v = velocity sebuah partikel.

$c1$ dan $c2$ = correction factor.

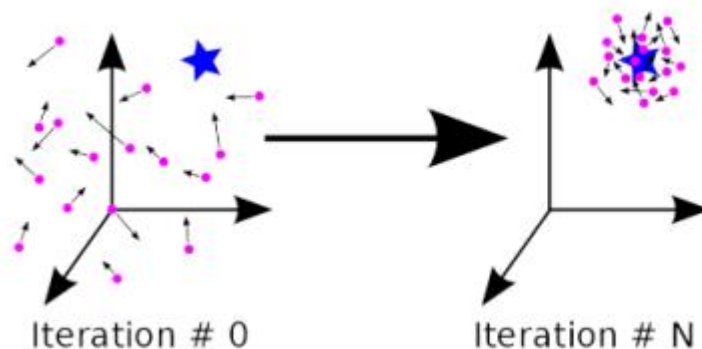
$pbest$ = posisi terbaik partikel.

$gbest$ = posisi partikel terbaik dari seluruh partikel yang ada di populasi.

Pada tahun 1998 Eberhart dan Shi menambahkan *inertia weight* (w) sebagai faktor yang mengendalikan besarnya pengaruh *velocity* sebelumnya terhadap *velocity* saat ini [15]. *Inertia weight* juga digunakan sebagai kontrol yang dapat menyeimbangkan antara eksplorasi global dan eksploitasi lokal. Sehingga persamaan *velocity* menjadi:

$$v = w*v + c1*rand()*(pbest-x) + c2*rand()*(gbest-x)$$

Model ini akan disimulasikan dalam ruang dengan dimensi tertentu dengan sejumlah iterasi sehingga di setiap iterasi, posisi partikel akan semakin mengarah ke target yang dituju untuk dilakukan hingga maksimum iterasi dicapai atau bisa juga digunakan kriteria penghentian yang lain.



Gambar 2.13 Ilustrasi Konsep PSO

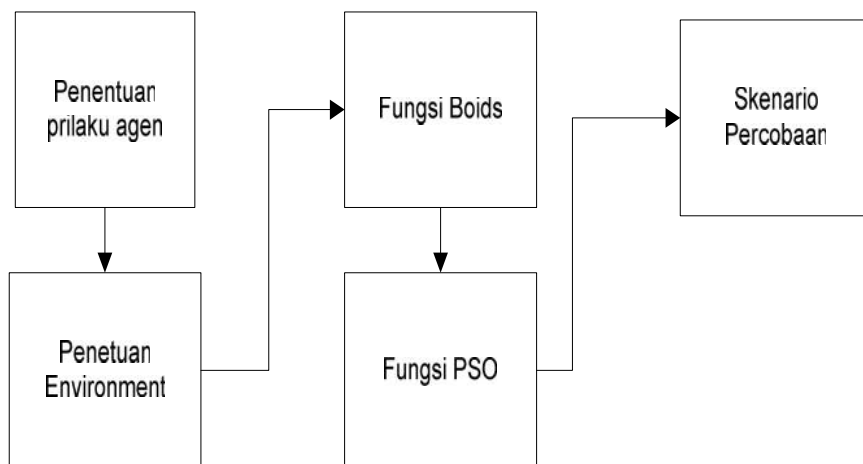
BAB III

METODOLOGI PENELITIAN

Goblin, golem dan Giant adalah 3 contoh pasukan darat yang bergerak berdasarkan tujuan khusus. Target utama golem dan giant adalah pertahanan lawan seperti Canon, Wizard Tower dan Archer Tower sedangkan target Goblin adalah sumber daya. Saat bergerak bersama-sama menuju target pasukan ini melewati halangan untuk menuju target yang telah ditentukan dan tiap pasukan sejenis mempunyai kecepatan yang sama sehingga pada saat dideploy bergantian pasukan tidak membentuk kelompok melainkan berbaris memanjang.

Agan dalam game membutuhkan *path planning* untuk menentukan jalur menuju posisi berikutnya atau menuju targetnya. Penelitian ini dilakukan untuk membuat *path planning* sekelompok agen otonom yang bergerak bersama menuju target secara bersama-sama yang konvergen dalam sebuah formasi dan kecepatan. Hasil dari penelitian ini dapat diterapkan untuk menentukan pergerakan pasukan agar lebih kelihatan halus atau *smooth*.

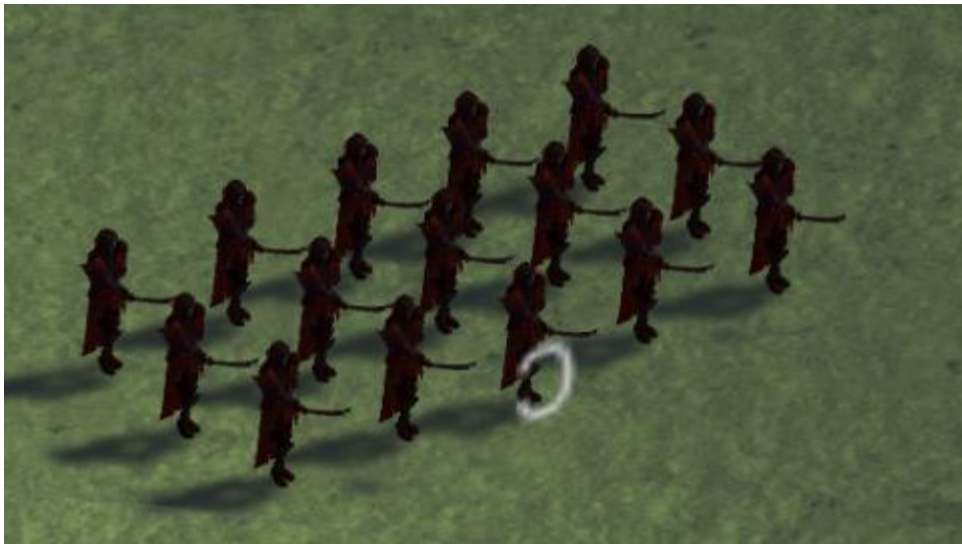
Gambar 3.1 adalah diagram alur langkah-langkah pelaksanaan penelitian ini. Diawali dengan penentuan desain perilaku agen sampai dengan visualisasi skenario pegeran pasukan dalam menghindari halangan dengan terget yang telah ditentukan.



Gambar 3.1 Skema Metodologi Penelitian

3.1.Desain Agen

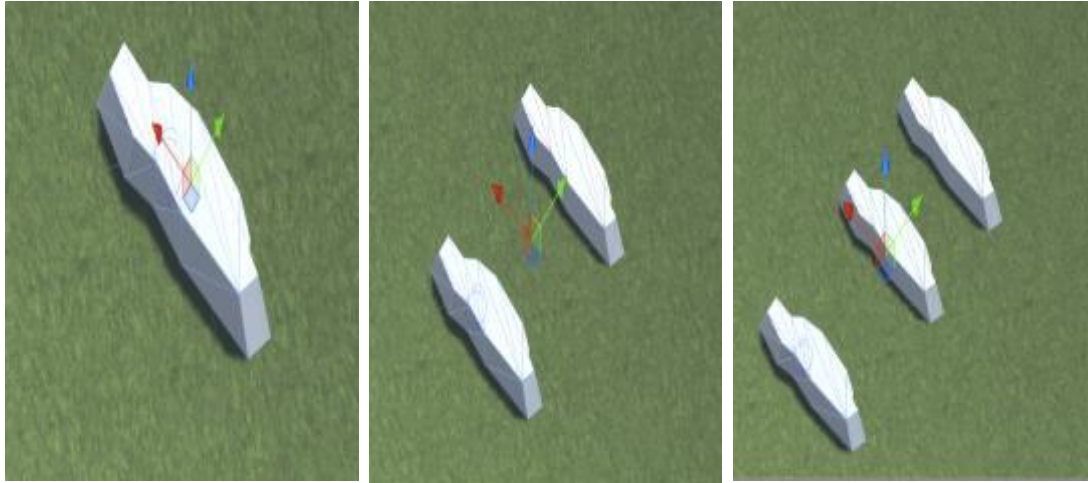
Pasukan yang digunakan dalam skenario ini menggunakan 3 aturan perilaku boids yaitu *separation*, *cohesion*, dan *alignment*. Skenario dalam simulasi ini menggunakan formasi kotak dengan seorang pemimpin dengan tanda "C" seperti pada Gambar 3.2. Setiap agen yang bukan pemimpin memiliki perilaku kendali *obstacle avoidance* dan *leader following*. Perilaku kendali agen menggunakan optimasi PSO untuk menyelaraskan kecepatan untuk menuju ke titik konvergen.



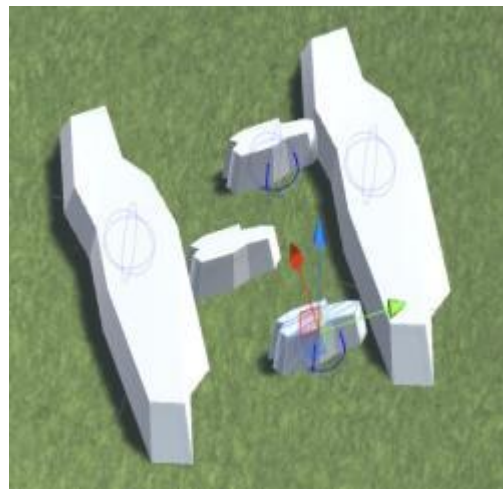
Gambar 3.2 Formasi Pasukan Boids

3.2. Desain Environment

Area lingkungan pada simulasi percobaan ini pada umumnya berbentuk datar dengan penambahan beberapa halangan berupa objek sebagai tanda halangan yang harus dihindari oleh kerumunan agen atau pasukan dalam melakukan pergerakan menuju target yang telah ditentukan. Enviroment pada simulasi percobaan ini menggunakan beberapa posisi halangan dan jumlah halangan dengan variasi yang berbeda yaitu dengan menambahkan sebuah halangan, dua halangan dan tiga halangan seperti pada Gambar 3.3. Skenario environment selanjutnya pasukan melewati lorong dengan posisi halangan yang berbeda dengan sebuah halangan, dua halangan dan tiga halangan seperti pada Gambar 3.4



Gambar 3.3 Lingkungan Halangan



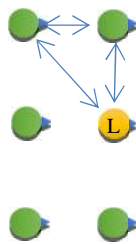
Gambar 3.4 Halangan Lorong

3.3. Penerapan Model Boids

Dasar *flocking* melalui tiga tahapan yaitu separasi, penyesuaian dan kohesi. Model boids dalam penelitian ini digambarkan dalam alur diagram penelitian seperti pada Gambar 3.2 Pergerakan pasukan secara bergerombol untuk menuju target melalui beberapa tahapan. Halangan dimungkinkan dalam pergerakan pasukan, setelah halangan dapat terdeteksi pasukan merubah arah dengan melakukan pemisahan. Jalur yang dilalui pasukan dapat berbeda dan jaraknya juga tidak sama.

Posisi setiap pasukan perlu diketahui oleh antar pasukan untuk melakukan penyelarasan agar pergerakan pasukan tampak realistis. Penyelarasan atau *velocity matching* pasukan menggunakan optimasi partikel swarm untuk menentukan kecepatan dan posisi yang sesuai untuk menjaga jarak antar pasukan. Halangan yang bergerak membutuhkan waktu yang lebih lama dalam menemukan nilai optimal dalam penyelarasan.

Pada Gambar 3.10 kecepatan dan inisialisasi boids ditentukan dengan posisi formasi pasukan kotak dan mempunyai kecepatan yang sama dari setiap agen pasukan. Model berkelompok terdiri dari tiga perilaku kemudi sederhana yang menggambarkan bagaimana sebuah manuver agen boids berdasarkan posisi dan .kecepatan *flockmates* terdekatnya. Kecepatan dalam pasukan dalam penelitian ini kami buat sama dengan posisi pasukan berbentuk formasi dan secara umum formasi dapat sebagai berikut digambarkan pada Gambar 3.5. Formasi yang digunakan pada penelitian ini adalah formasi kotak dan jumlah agen ada lima belas pasukan dengan seorang pemimpin dengan inisial L seperti pada Gambar 3.6



Gambar 3.5 Posisi Agen Pada Formasi Secara Umum

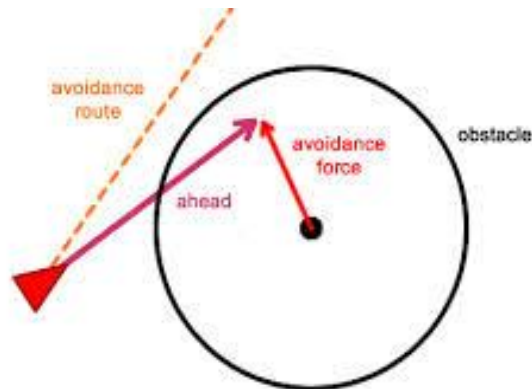
Dimana :

- () : Jarak agen dengan agen disamping
- () : Jarak agen dengan agen ke belakang
- () : Sudut agen

	14	9	2	
	13	8	1	
	12	7	L	
	11	6	3	
	10	5	4	

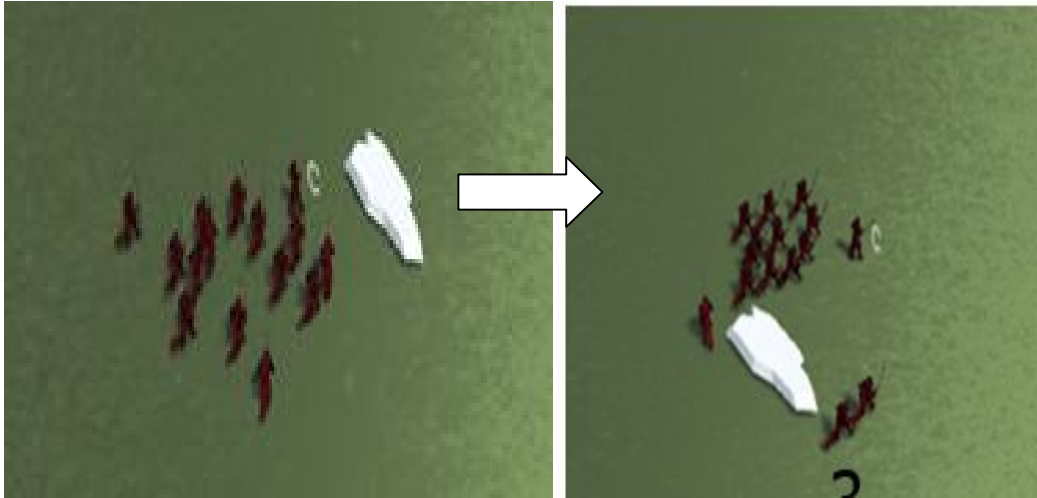
Gambar 3.6 Formasi Kotak

Setelah pasukan membentuk formasi pasukan bergerak menuju target yang telah ditentukan. Jika pasukan menemui halangan pasukan akan melakukan penghindaran halangan dengan perilaku kendali separasi. Halangan pada penelitian ini memiliki radius 1 satuan Unity3d seperti pada Gambar 3.7.



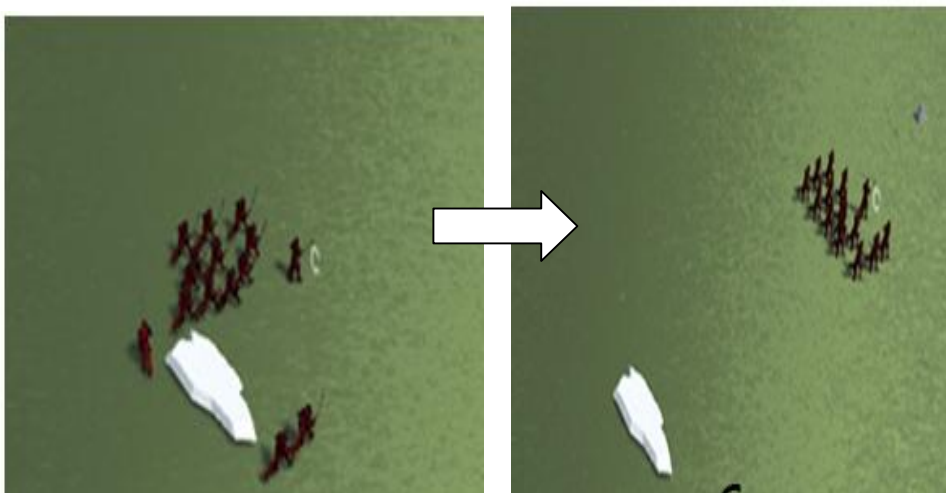
Gambar 3.7 Deteksi Halangan

Saat melakukan separasi pasukan bergerak menuju target dengan melakukan pemisahan dengan memperhatikan jarak antar pasukan. Formasi pasukan yang semula berbentuk kotak berubah untuk dapat melewati halangan berupa objek yang harus dilewati. Setiap pasukan bergerak menjauh dari objek yang akan dilewati sesuai dengan posisi masing-masing agen pasukan agar dapat melewati halangan seperti pada gambar 3.8.

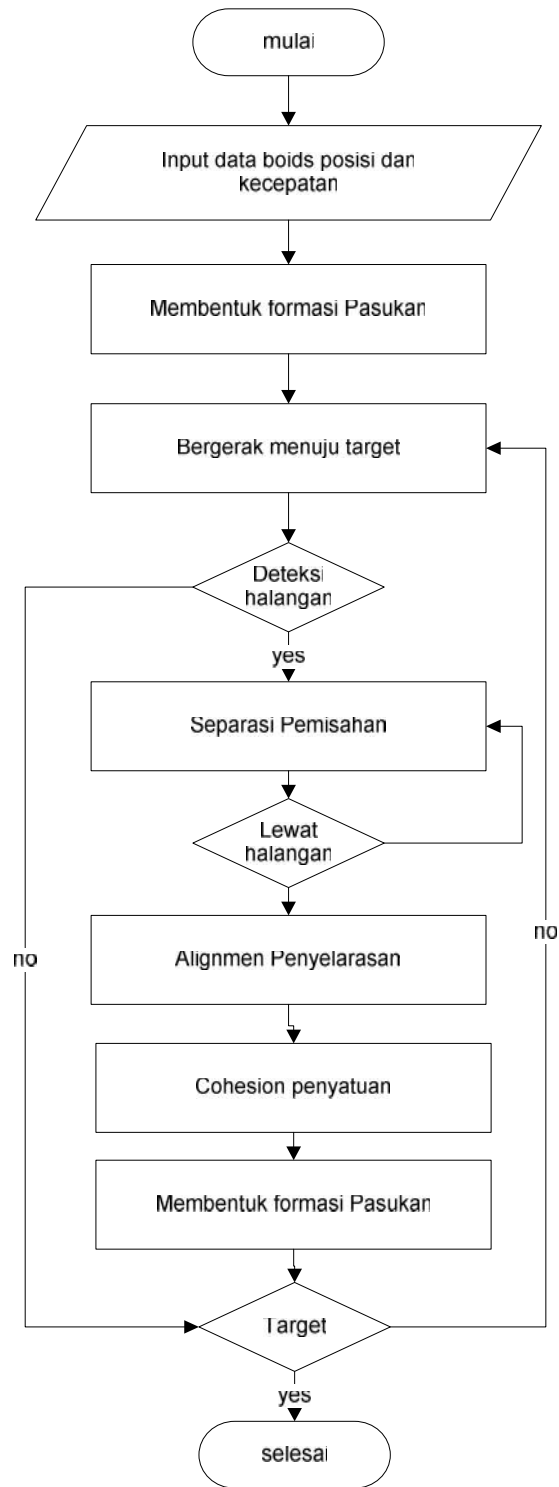


Gambar 3.8 Pemisahan Pasukan

Setelah melakukan melakukan pemisahan pasukan melakukan perilaku kendali penyelarasan dengan menjaga jarak antar pasukan agar dengan kecepatan yang sama dalam stiap agen pasukan. Pembentukan formasi kembali dilakukan dengan cara kohesi untuk mencapai konvergen pada saat pasukan bergerak menuju target seperti pada Gambar 3.9. Setiap agen pasukan melakukan pergerakan menuju ke posisi formasi sesuai dengan lokasi masing-masing agen pasukan dan jarak antar agen pasukan tetap dilakukan.



Gambar 3.9 Regrouping Formasi



Gambar 3.10 Pergerakan Pasukan dengan Model Boids

3.4. Penerapan Particle Swarm Optimization

Pemisahan atau separasi pasukan dilakukan saat terdapat halangan untuk menuju target seperti pada Gambar 3.3. Halangan tersebut juga dapat berupa halangan dinamis bergerak menghalangi pasukan untuk menuju target seperti pada Gambar 3.4. Particle Swarm Optimization digunakan pada bagian penyesuaian kecepatan atau *velocity matching* antar pasukan untuk mengoptimalkan pergerakan pasukan menuju target baik dengan halangan statis maupun dinamis dalam model boids yang terdapat pada Gambar 3.1.

Algoritma PSO pada penelitian ini, pasukan mewakili partikel. PSO adalah teknik optimasi stokastik yang kuat didasarkan pada gerakan dan kecerdasan kawanan. PSO menerapkan konsep interaksi sosial untuk memecahkan masalah. Menggunakan sejumlah agen (partikel) yang merupakan pergerakan secara bergerombol di dalam ruang pencarian mencari solusi terbaik. Setiap partikel diperlakukan sebagai titik dalam ruang N-dimensi yang berpindah sesuai dengan pengalaman berpindah sendiri serta pengalaman berpindah partikel lainnya.

Setiap partikel melacak koordinat dalam ruang solusi yang berhubungan dengan solusi terbaik (fitness) yang telah dicapai sejauh ini oleh partikel itu. Nilai ini disebut *personal best (pbes)*. Nilai terbaik lain yang dilacak oleh PSO adalah nilai terbaik yang diperoleh sejauh ini oleh partikel apapun di lingkungan partikel itu, nilai ini disebut *gbest*. Formula untuk menentukan nilai fitness sebagai berikut:

$$v = v + c1*rand()*(pbest-x) + c2*rand()*(gbest-x) \quad (3.1)$$

Keterangan:

v = velocity sebuah partikel.

$c1$ dan $c2$ = correction factor.

$pbest$ = posisi terbaik partikel.

$gbest$ = posisi partikel terbaik dari seluruh partikel yang ada di populasi.

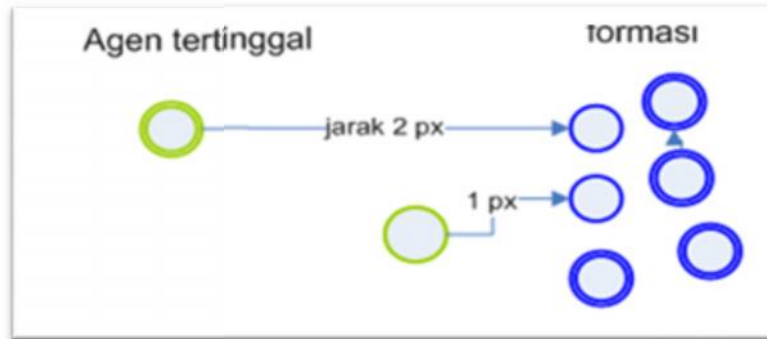
Konsep dasar PSO terletak dalam mempercepat setiap partikel ke arah *pbest* dan lokasi *gbest*, dengan bobot akselerasi acak pada setiap langkah *flow chart* PSO terdapat dalam Gambar 3.13.

Saat melewati halangan setiap agen dalam pasukan melakukan penyesuaian namun ada beberapa agen yang tertinggal karena melewati halangan seperti pada Gambar 3.11. Inisialisasi partikel dan kecepatan agen seperti pada pasukan boids dengan formasi kotak dan kecepatan rata-rata yang sama.

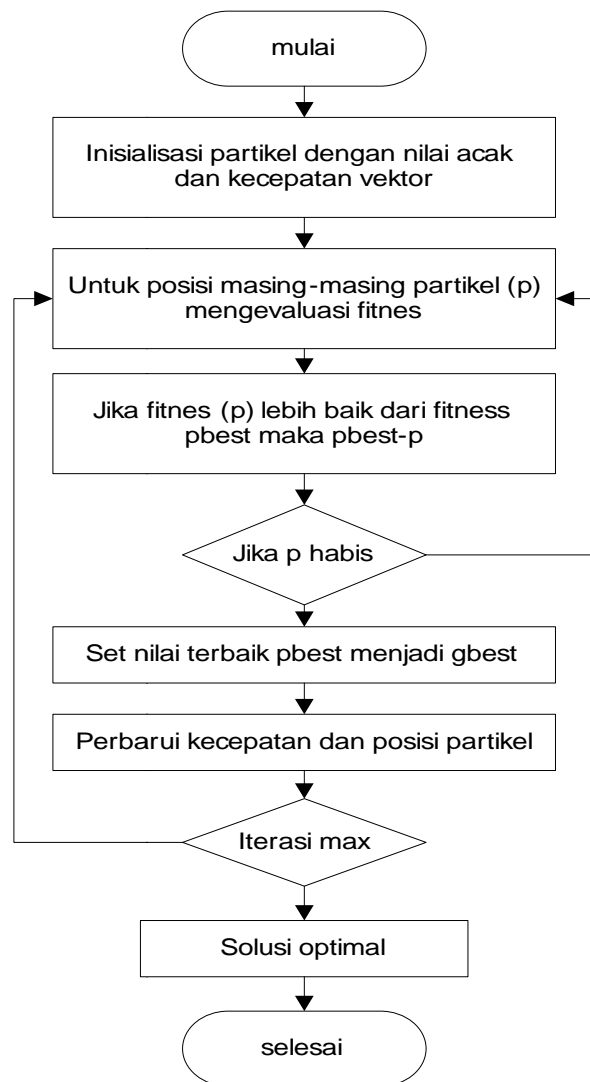


Gambar 3.11 Agen Terpisah dari Formasi

Pada Gambar 3.13 posisi setiap agen mengevaluasi posisi dan jarak setiap pasukan untuk formasi kotak. Jika jarak antar pasukan lebih jauh dari posisi formasi maka pasukan pada setiap agen melakukan percepatan sesuai dengan jarak agen ke posisi yang telah di tentukan dalam formasi seperti pada Gambar 3.12. Setiap agen melakukan percepatan sesuai dengan formula 3.1 dengan melakukan update posisi dan kecepatan atau disebut dengan *pbest*. Setelah setiap agen dapat melakukan percepatan untuk kembali ke formasi pasukan boids setiap *pbest* diset sebagai *gbest*. Semua pasukan menempati posisi formasi semula setelah melakukan update posisi dan kecepatan saat menuju target.



Gambar 3.12 Jarak Agen ke Formasi



Gambar 3.13 Flow Chart PSO

3.5. Skenario

Percobaan dalam simulasi pergerakan kerumunan pasukan menuju target dengan membandingkan algoritma boids tanpa PSO dan algoritma boids dengan PSO dalam melakukan pergerakan menghindari halangan dari segi waktu sampai menuju target.

s

halaman ini sengaja dikosongkan

BAB IV

PERCOBAAN DAN SIMULASI KOMPUTER

Pergerakan sekelompok agen yang telah diuraikan pada Bab 3 diimplementasikan dalam simulasi komputer dengan menggunakan Game Engine Unity3d. Untuk lebih memudahkan analisa, percobaan dan simulasi ini dibagi dalam 6 bagian yakni pergerakan kerumunan pasukan melewati satu halangan, melewati dua halangan, melewati tiga halangan, tiga halangan dalam lorong, empat halangan dinamis dalam lorong dan empat halangan dalam lorong dengan variasi populasi. Keadaan posisi awal pasukan dengan formasi kotak pada koordinat pada Tabel 4.1.

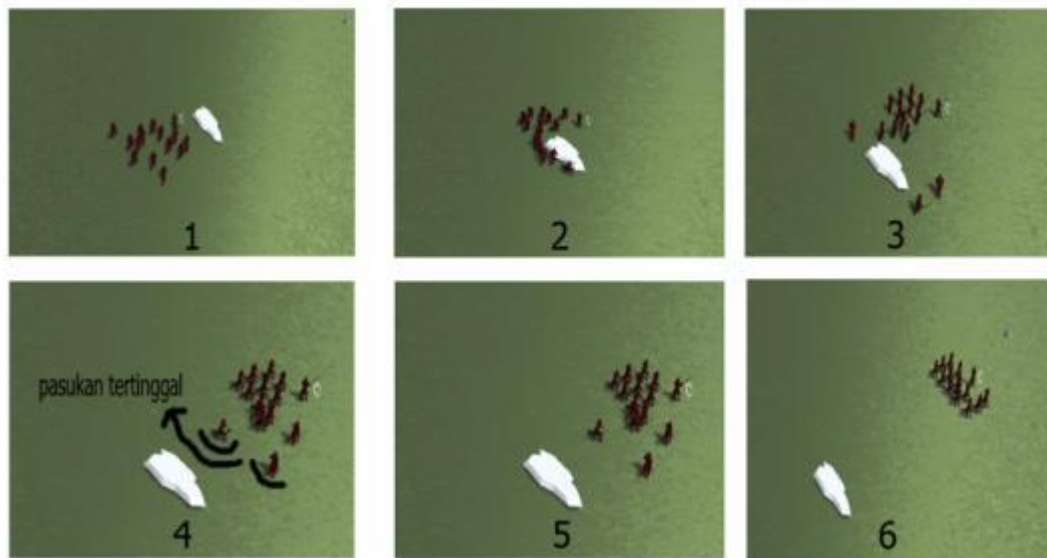
Tabel 4.1 Posisi Awal Percobaan

Agen	Posisi
leader	(310.3, 0.1, 143.7)
pasukan 1	(306.4, 0.1, 143.3)
pasukan 2	(308.3, 0.1, 143.3)
pasukan 3	(313.7, 0.1, 143.4)
pasukan 4	(306.4, 0.1, 141.7)
pasukan 5	(308.3, 0.1, 141.6)
pasukan 6	(310.0, 0.1, 141.7)
pasukan 7	(311.8, 0.1, 141.8)
pasukan 8	(313.9, 0.1, 141.8)
pasukan 9	(313.7, 0.1, 140.1)
pasukan 10	(311.8, 0.1, 140.1)
pasukan 11	(310.0, 0.1, 140.1)
pasukan 12	(308.4, 0.1, 140.1)
pasukan 13	(306.7, 0.1, 140.1)
pasukan 14	(312.4, 0.1, 143.8)

4.1. Pergerakan Pasukan dengan Satu Halangan

Pada skenario ini kerumunan pasukan membentuk formasi kotak dengan posisi awal pasukan pada Tabel 4.1. Pasukan bergerak dengan formasi kotak menuju target dengan melewati halangan dan berkumpul kembali pada formasi kotak kembali seperti pada Gambar 4.1. Pada skenario ini Algoritma boids digunakan

tanpa optimasi PSO yang menghasilkan pasukan konvergen pada saat melakukan kohesi dalam waktu rata-rata 8.6023998 detik.



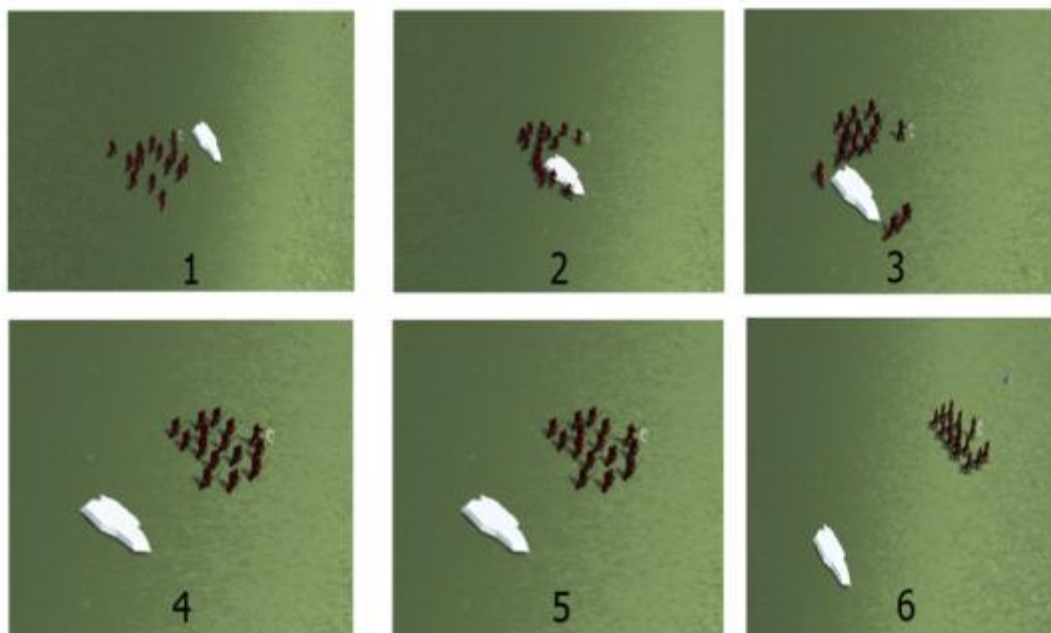
Gambar 4.1 Hasil Percobaan Pertama Boids

Pada Gambar 4.1 bagian pertama pasukan melakukan pergerakan menuju target dengan melewati satu halangan. Gambar bagian kedua dan ketiga pasukan melakukan separasi untuk menghindari halangan, pasukan pada gambar bagian keempat ada dua pasukan yang tertinggal sehingga pada gambar bagian kelima mulai mengejar untuk melakukan kohesi pada bagian keenam. Posisi pasukan saat melakukan kohesi dan kecepatan dari setiap pasukan terdapat pada Tabel 4.2. Pada skenario ini Algoritma boids digunakan dengan optimasi PSO yang menghasilkan pasukan konvergen pada saat melakukan kohesi dalam waktu rata-rata 6.1956 detik.

Pada Gambar 4.2 bagian pertama pasukan melakukan pergerakan menuju target dengan melewati satu halangan. Gambar bagian kedua dan ketiga pasukan melakukan separasi untuk menghindari halangan, pasukan pada gambar bagian keempat pasukan melakukan percepatan sehingga tidak tertinggal seperti pada Gambar 4.1 bagian keempat. Posisi pasukan saat melakukan kohesi dan kecepatan setiap pasukan terdapat pada Tabel 4.3.

Tabel 4.2 Hasil Percobaan Pertama Boids

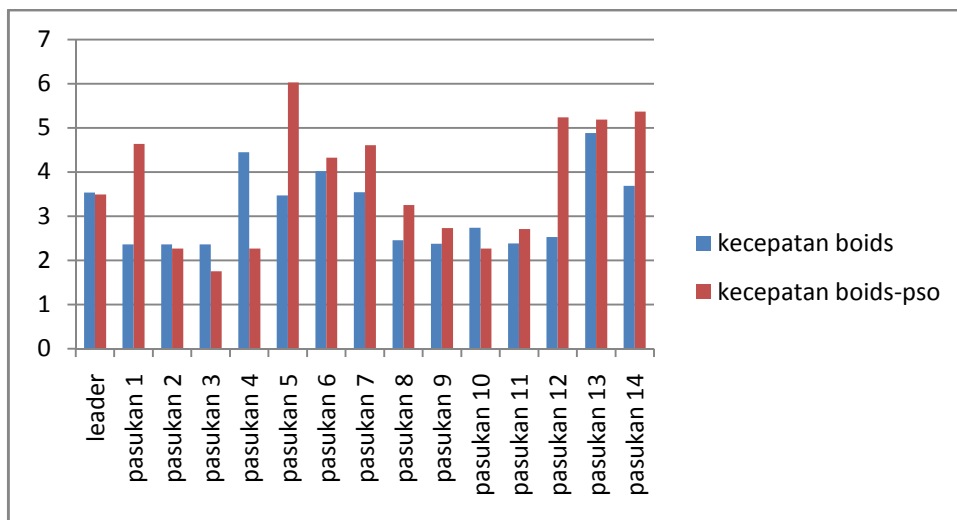
Agen	Posisi kohesi boids	Kecepatan boids
leader	(347.5, 0.1, 166.9)	3.538282
pasukan 1	(345.8, 0.1, 168.3)	2.358855
pasukan 2	(347.4, 0.1, 165.1)	2.358855
pasukan 3	(348.1, 0.1, 163.5)	2.358855
pasukan 4	(343.5, 0.1, 169.4)	4.451052
pasukan 5	(344.2, 0.1, 167.6)	3.469967
pasukan 6	(345.1, 0.1, 166.0)	4.018837
pasukan 7	(345.8, 0.1, 164.5)	3.544877
pasukan 8	(346.8, 0.1, 162.7)	2.454976
pasukan 9	(345.2, 0.1, 162.1)	2.373659
pasukan 10	(344.3, 0.1, 163.8)	2.73909
pasukan 11	(343.6, 0.1, 165.4)	2.383784
pasukan 12	(342.9, 0.1, 166.9)	2.526472
pasukan 13	(342.2, 0.1, 168.3)	4.881478
pasukan 14	(345.0, 0.1, 170.1)	3.687



Gambar 4.2 Hasil Percobaan Pertama Boids-PSO

Tabel 4.3 Hasil Percobaan Pertama Boids-PSO

Agen	Posisi kohesi	Kecepatan
leader	(347.5, 0.1, 166.9)	3.538282
pasukan 1	(345.8, 0.1, 168.3)	2.358855
pasukan 2	(347.4, 0.1, 165.1)	2.358855
pasukan 3	(348.1, 0.1, 163.5)	2.358855
pasukan 4	(343.5, 0.1, 169.4)	4.451052
pasukan 5	(344.2, 0.1, 167.6)	3.469967
pasukan 6	(345.1, 0.1, 166.0)	4.018837
pasukan 7	(345.8, 0.1, 164.5)	3.544877
pasukan 8	(346.8, 0.1, 162.7)	2.454976
pasukan 9	(345.2, 0.1, 162.1)	2.373659
pasukan 10	(344.3, 0.1, 163.8)	2.73909
pasukan 11	(343.6, 0.1, 165.4)	2.383784
pasukan 12	(342.9, 0.1, 166.9)	2.526472
pasukan 13	(342.2, 0.1, 168.3)	4.881478
pasukan 14	(345.0, 0.1, 170.1)	3.687

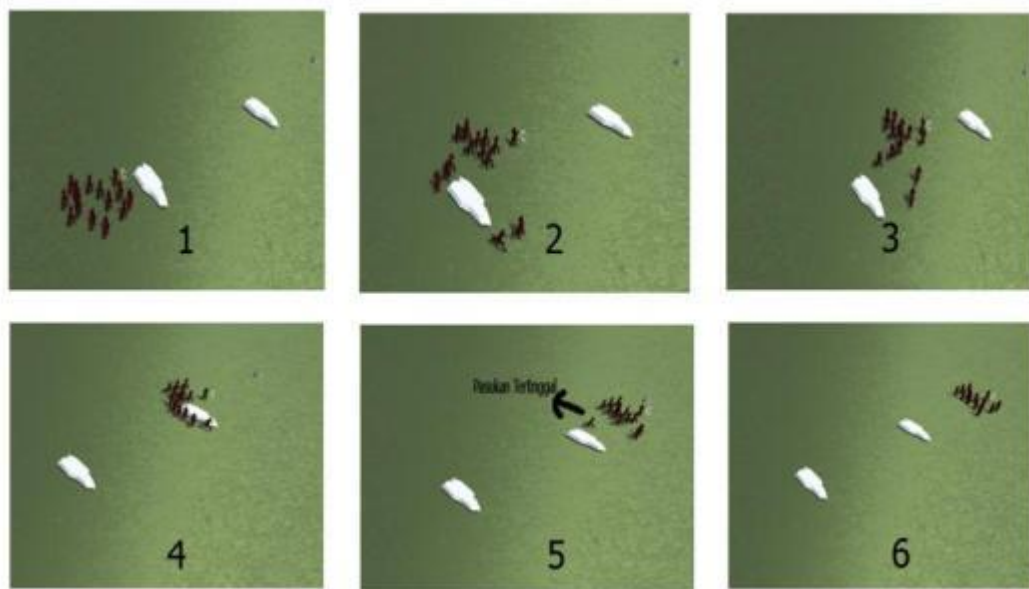


Gambar 4.3 Grafik Percobaan Pertama Boids-PSO

Kecepatan antar pasukan memiliki nilai yang berbeda untuk menyesuaikan kecepatan pasukan yang lain atau disebut dengan alignment. Pasukan boids dengan optimasi PSO dapat melakukan kohesi lebih cepat jika dibandingkan dengan pasukan boids tanpa optimasi PSO. Kecepatan pasukan boids cenderung meningkat untuk mengejar ketertinggalan pada saat melakukan kohesi seperti pada Gambar 4.3.

4.2. Pergerakan Pasukan dengan Dua Halangan

Pada skenario ini kerumunan pasukan membentuk formasi kotak dengan posisi awal pasukan pada Tabel 4.1. Pasukan bergerak dengan formasi kotak menuju target dengan melewati 2 halangan dan berkumpul kembali pada formasi kotak kembali seperti pada Gambar 4.4. Pada skenario ini Algoritma boids digunakan tanpa optimasi PSO yang menghasilkan pasukan konvergen pada saat melakukan kohesi dalam waktu rata-rata 10.5814 detik.

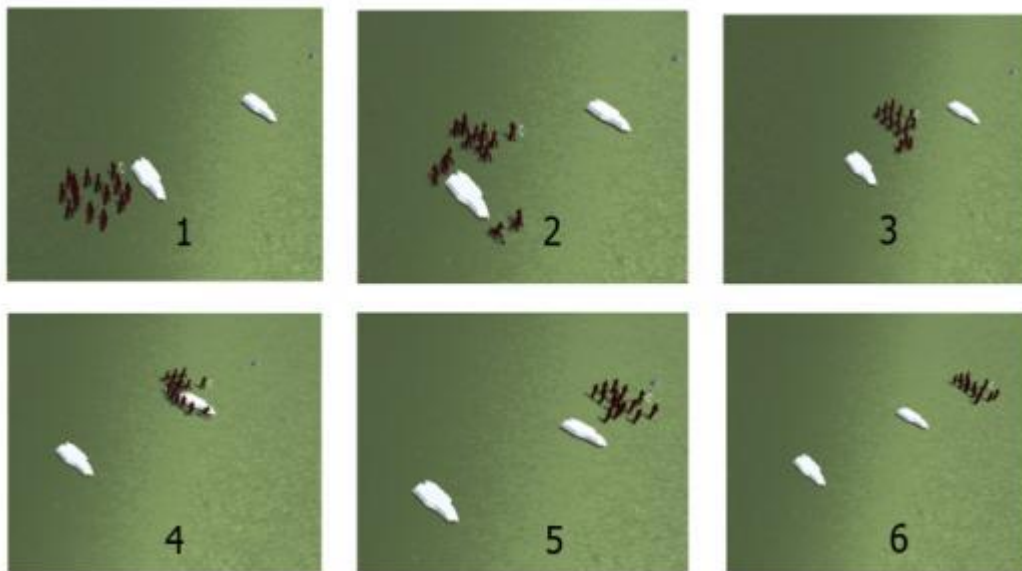


Gambar 4.4 Hasil Percobaan Kedua Boids

Pada Gambar 4.4 bagian pertama pasukan melakukan pergerakan menuju target dengan melewati dua halangan. Gambar bagian kedua pasukan melakukan separasi untuk menghindari halangan dan pada bagian ketiga mulai melakukan kohesi namun pada bagian gambar keempat terdapat halangan lagi sehingga pasukan melakukan separasi untuk melewati halangan. pasukan pada gambar bagian kelima ada 2 pasukan yang tertinggal pada saat melakukan kohesi karena terhambat oleh halangan. Posisi pasukan saat melakukan kohesi dan kecepatan setiap pasukan terdapat pada Tabel 4.4. Pada skenario ini Algoritma boids digunakan dengan optimasi PSO yang menghasilkan pasukan konvergen pada saat melakukan kohesi dalam waktu 8.27699984 detik.

Tabel 4.4 Hasil Percobaan Kedua Boids

Agen	Posisi kohesi boids	Kecepatan boids
leader	(355.5, 0.1, 171.7)	3.435248
pasukan 1	(352.0, 0.1, 171.5)	3.22689
pasukan 2	(355.3, 0.1, 169.9)	3.755778
pasukan 3	(355.8, 0.1, 168.2)	3.767144
pasukan 4	(351.7, 0.1, 174.5)	2.688454
pasukan 5	(352.3, 0.1, 172.6)	3.132564
pasukan 6	(353.0, 0.1, 171.1)	3.612861
pasukan 7	(353.7, 0.1, 169.4)	4.366056
pasukan 8	(354.4, 0.1, 167.4)	4.832808
pasukan 9	(352.7, 0.1, 167.0)	4.912731
pasukan 10	(352.1, 0.1, 168.9)	4.201026
pasukan 11	(351.5, 0.1, 170.5)	3.142995
pasukan 12	(350.9, 0.1, 172.0)	3.070807
pasukan 13	(350.3, 0.1, 173.7)	3.106869
pasukan 14	(353.2, 0.1, 175.0)	2.798033



Gambar 4.5 Hasil Percobaan Kedua Boids-PSO

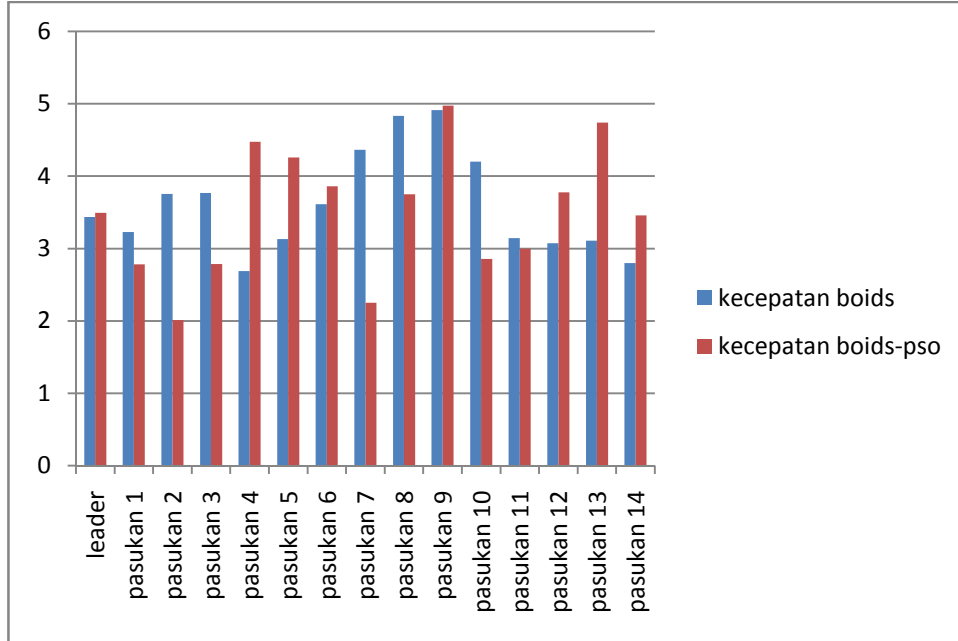
Pada Gambar 4.5 bagian pertama pasukan melakukan pergerakan menuju target dengan melewati dua halangan. Gambar bagian kedua pasukan melakukan separasi untuk menghindari halangan dan pada bagian ketiga mulai melakukan

kohesi yang terlihat lebih baik dari Gambar 4.4 bagian ketiga. Pasukan pada gambar bagian kelima ada 2 pasukan yang tertinggal pada saat melakukan kohesi namun melakukan percepatan sehingga lebih cepat untuk mendekat dengan pasukan yang lain bila dibandingkan Gambar 4.4. Posisi pasukan saat melakukan kohesi dan kecepatan setiap pasukan pada gambar bagian keenam terdapat pada Tabel 4.5.

Tabel 4.5 Hasil Percobaan Kedua Boids-PSO

Agen	Posisi kohesi	Kecepatan
leader	(348.6, 0.1, 169.3)	3.491987
pasukan 1	(347.3, 0.1, 171.0)	2.782513
pasukan 2	(348.4, 0.1, 167.6)	2.012664
pasukan 3	(349.1, 0.1, 165.9)	2.787558
pasukan 4	(345.1, 0.1, 172.3)	4.473285
pasukan 5	(345.4, 0.1, 170.4)	4.25968
pasukan 6	(346.1, 0.1, 168.8)	3.859287
pasukan 7	(346.9, 0.1, 167.1)	2.249158
pasukan 8	(347.6, 0.1, 165.2)	3.748672
pasukan 9	(345.6, 0.1, 164.7)	4.9743
pasukan 10	(345.3, 0.1, 166.6)	2.858054
pasukan 11	(344.6, 0.1, 168.2)	2.99648
pasukan 12	(344.1, 0.1, 169.8)	3.776763
pasukan 13	(343.5, 0.1, 171.5)	4.739386
pasukan 14	(346.7, 0.1, 172.8)	3.456755

Kecepatan antar pasukan memiliki nilai yang berbeda untuk menyesuaikan kecepatan pasukan yang lain atau disebut dengan alignment. Penyesuaian kecepatan pasukan boids dengan PSO pada Gambar 4.6 memiliki kecepatan tertinggi pada pasukan 9 yaitu 4.9743 dan kecepatan terendah pada pasukan 2 sedangkan pasukan boids tanpa PSO memiliki kecepatan tertinggi pada pasukan 9 yaitu 4.912731 dan kecepatan terendah pada pasukan 4 yaitu 2.688454. Perbandingan kecepatan saat melakukan kohesi antara pasukan boids dengan PSO dan pasukan pasukan boids tanpa PSO terdapat pada Gambar 4.6.

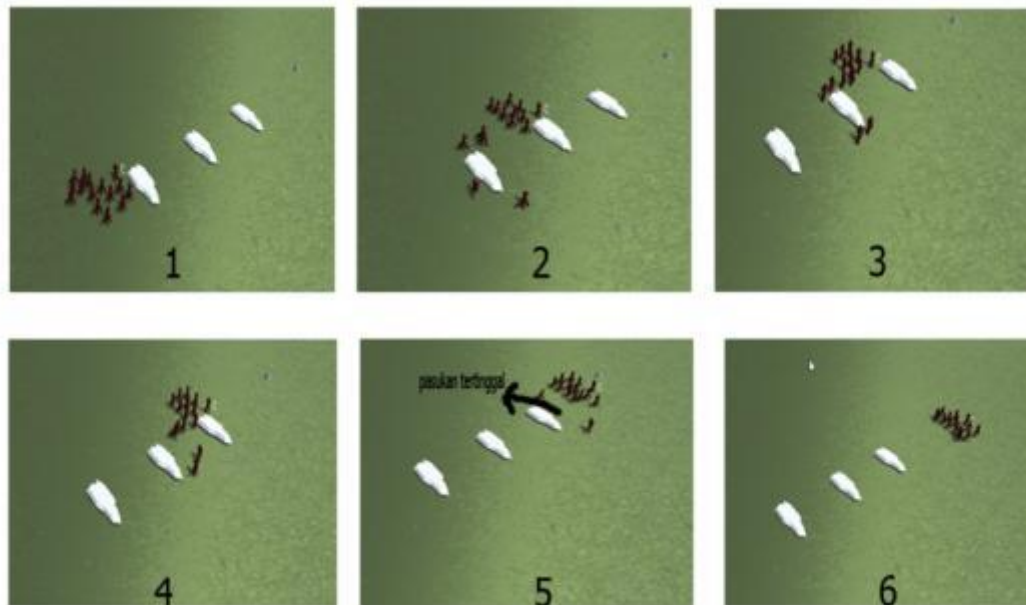


Gambar 4.6 Grafik Percobaan Kedua Boids-PSO

4.3. Pergerakan Pasukan dengan Tiga Halangan

Pada skenario ini kerumunan pasukan membentuk formasi kotak dengan posisi awal pasukan pada Tabel 4.1. Pasukan bergerak dengan formasi kotak menuju target dengan melewati tiga halangan dan berkumpul kembali pada formasi kotak kembali seperti pada Gambar 4.7. Pada skenario ini Algoritma boids digunakan tanpa optimasi PSO yang menghasilkan pasukan konvergen pada saat melakukan kohesi dalam waktu rata-rata 13.2762 detik.

Pada Gambar 4.7 bagian pertama pasukan melakukan pergerakan menuju target dengan melewati tiga halangan. Gambar bagian kedua dan ketiga pasukan melakukan separasi tanpa melakukan kohesi karena jarak halangan yang dekat. Pasukan pada gambar bagian kelima ada 2 pasukan yang tertinggal pada saat melakukan kohesi. Posisi pasukan saat melakukan kohesi dan kecepatan setiap pasukan pada gambar bagian keenam terdapat pada Tabel 4.6. Pada skenario ini Algoritma boids digunakan dengan optimasi PSO yang menghasilkan pasukan konvergen pada saat melakukan kohesi dalam waktu rata-rata 11.9452 detik.



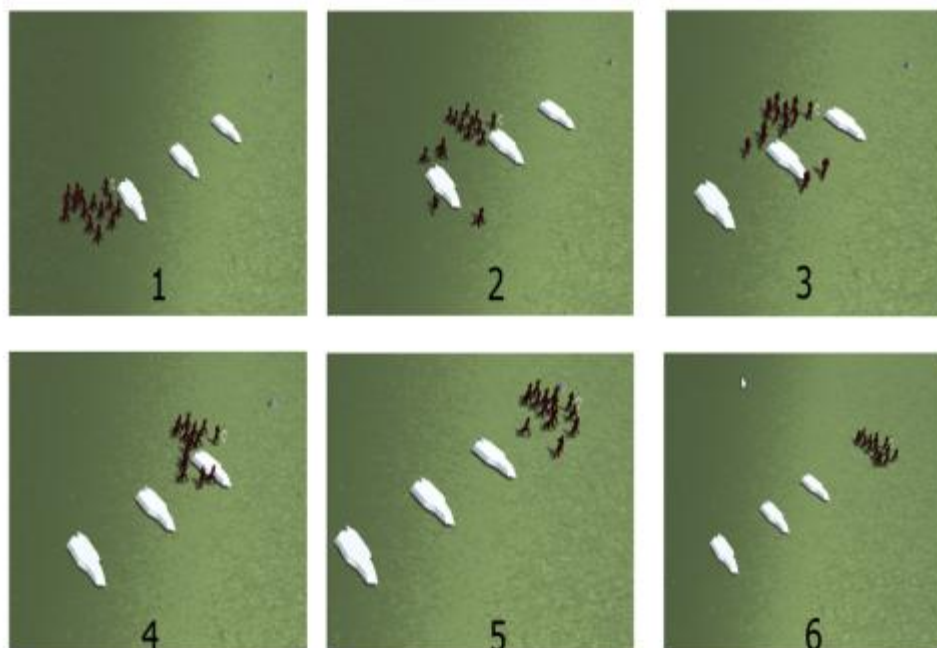
Gambar 4.7 Hasil Percobaan Ketiga Boids

Tabel 4.6 Hasil Percobaan Ketiga Boids

Agen	Posisi kohesi boids	Kecepatan boids
leader	(355.3, 0.1, 171.6)	3.49984
pasukan 1	(351.5, 0.1, 173.4)	3.103737
pasukan 2	(355.0, 0.1, 169.8)	3.11929
pasukan 3	(355.6, 0.1, 168.1)	3.172517
pasukan 4	(351.7, 0.1, 174.5)	3.157721
pasukan 5	(352.2, 0.1, 172.6)	3.94966
pasukan 6	(352.8, 0.1, 171.0)	3.630492
pasukan 7	(353.4, 0.1, 169.3)	3.597916
pasukan 8	(353.9, 0.1, 166.4)	4.534082
pasukan 9	(352.5, 0.1, 167.0)	3.981499
pasukan 10	(351.9, 0.1, 168.8)	3.58932
pasukan 11	(351.3, 0.1, 170.5)	3.595826
pasukan 12	(350.8, 0.1, 172.0)	3.892466
pasukan 13	(350.2, 0.1, 173.7)	3.846256
pasukan 14	(353.2, 0.1, 175.0)	3.322556

Pada Gambar 4.8 bagian pertama pasukan melakukan pergerakan menuju target dengan melewati tiga halangan. Gambar bagian kedua pasukan melakukan

separasi untuk menghindari halangan dan pada bagian ketiga mulai melakukan kohesi tetapi masih ada halangan lagi sehingga melakukan separasi lagi. Pasukan pada gambar bagian kelima ada 2 pasukan yang tertinggal pada saat melakukan kohesi namun tampak lebih dekat jika dibandingkan dengan Gambar 4.7. Posisi pasukan saat melakukan kohesi dan kecepatan setiap pasukan pada gambar bagian keenam terdapat pada Tabel 4.7.

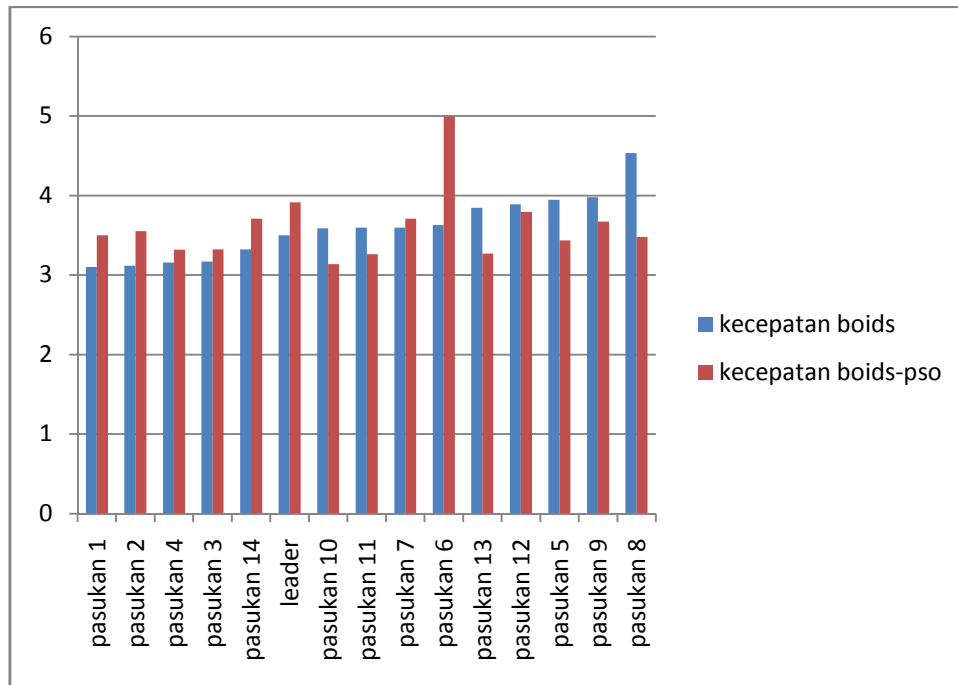


Gambar 4.8 Hasil Percobaan Ketiga Boids-PSO

Kecepatan antar pasukan memiliki nilai yang berbeda untuk menyesuaikan kecepatan pasukan yang lain atau disebut dengan alignment. Penyesuaian kecepatan pasukan boids dengan PSO pada Gambar 4.9 memiliki kecepatan tertinggi pada pasukan 9 yaitu 4.993078 dan kecepatan terendah pada pasukan 1 yaitu 3.271893 sedangkan pasukan boids tanpa PSO memiliki kecepatan tertinggi pada pasukan 8 yaitu 4.534082 dan kecepatan terendah pada pasukan 4 yaitu 3.157721. Perbandingan kecepatan saat melakukan kohesi antara pasukan boids dengan PSO dan pasukan boids tanpa PSO terdapat pada Gambar 4.9.

Tabel 4.7 Hasil Percobaan Ketiga Boids-PSO

Agen	Posisi kohesi	Kecepatan
leader	(354.3, 0.1, 171.2)	3.499961
pasukan 1	(350.8, 0.1, 171.2)	3.552114
pasukan 2	(354.0, 0.1, 169.5)	3.319453
pasukan 3	(354.6, 0.1, 167.7)	3.322851
pasukan 4	(350.6, 0.1, 174.2)	3.710238
pasukan 5	(351.2, 0.1, 172.3)	3.916127
pasukan 6	(351.8, 0.1, 170.7)	3.136941
pasukan 7	(352.4, 0.1, 169.0)	3.262586
pasukan 8	(353.1, 0.1, 167.1)	3.712251
pasukan 9	(351.1, 0.1, 166.2)	4.993078
pasukan 10	(350.8, 0.1, 168.5)	3.271893
pasukan 11	(350.3, 0.1, 170.1)	3.794619
pasukan 12	(349.7, 0.1, 171.7)	3.436315
pasukan 13	(349.2, 0.1, 173.4)	3.67487
pasukan 14	(350.5, 0.1, 173.0)	3.480518

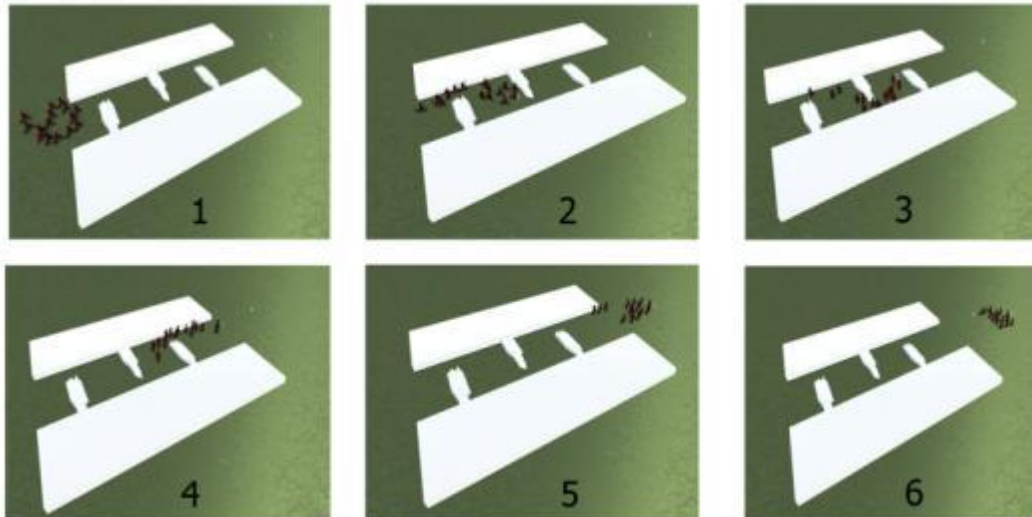


Gambar 4.9 Grafik Percobaan Ketiga Boids-PSO

4.4. Pergerakan Pasukan dengan 3 Halangan Dalam Lorong

Pada skenario ini kerumunan pasukan membentuk formasi kotak dengan posisi awal pasukan pada Tabel 4.1. Pasukan bergerak dengan formasi kotak menuju target dengan melewati tiga halangan dalam lorong dan berkumpul kembali pada formasi kotak kembali seperti pada Gambar 4.10. Pada skenario ini Algoritma boids digunakan tanpa optimasi PSO yang menghasilkan pasukan konvergen pada saat melakukan kohesi dalam waktu rata-rata 20.789 detik.

Pada Gambar 4.10 bagian pertama pasukan melakukan pergerakan menuju target dengan melewati tiga halangan dalam lorong. Gambar bagian kedua, ketiga dan keempat melakukan separasi dengan melewati tiga halangan dalam lorong. Pasukan pada gambar bagian kelima ada tiga pasukan yang tertinggal pada saat melakukan kohesi. Posisi pasukan saat melakukan kohesi dan kecepatan setiap pasukan pada gambar bagian keenam terdapat pada Tabel 4.8.



Gambar 4.10 Hasil Percobaan Keempat Boids

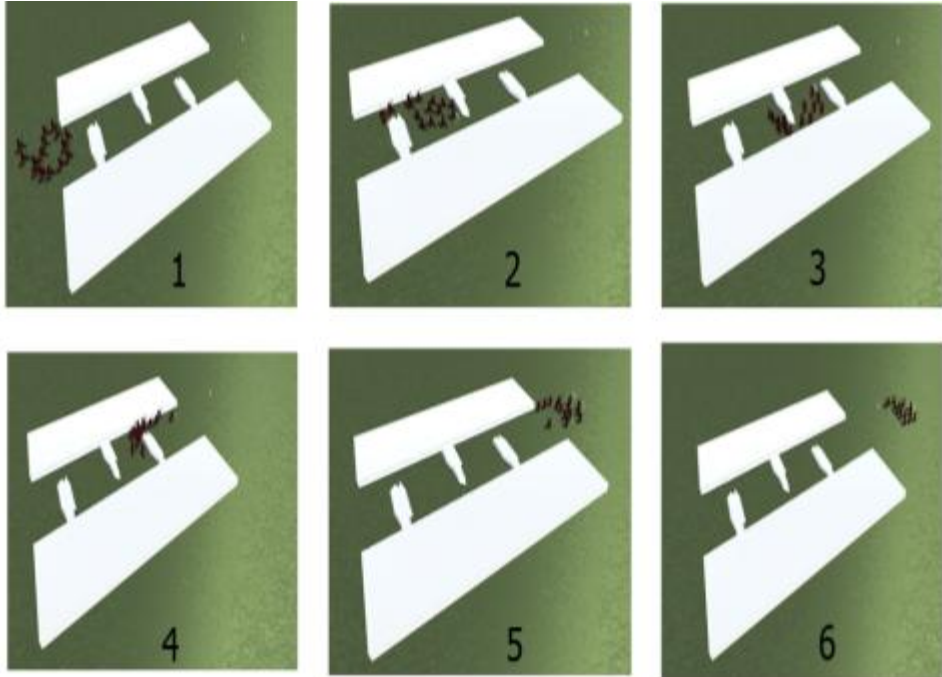
Pada skenario ini Algoritma boids digunakan dengan optimasi PSO yang menghasilkan pasukan konvergen pada saat melakukan kohesi dalam waktu rata-rata 17.445 detik. Pada Gambar 4.11 bagian pertama pasukan melakukan pergerakan menuju target dengan melewati tiga halangan dalam lorong. Gambar bagian kedua,

ketiga dan keempat melakukan separasi dengan melewati tiga halangan dalam lorong dan tempat lebih baik jika dibandingkan pada Gambar 4.10. Pasukan pada gambar bagian kelima ada 3 pasukan yang tertinggal sudah tampak lebih rapat untuk melakukan kohesi. Posisi pasukan saat melakukan kohesi dan kecepatan setiap pasukan pada gambar bagian keenam terdapat pada tabel 4.9.

Tabel 4.8 Hasil percobaan Keempat Boids

Agen	Posisi kohesi	Kecepatan
leader	(362.7, 0.1, 174.0)	2.787423
pasukan 1	(361.2, 0.1, 175.6)	4.728606
pasukan 2	(358.1, 0.1, 171.3)	3.019634
pasukan 3	(363.2, 0.1, 170.6)	4.886824
pasukan 4	(359.0, 0.1, 176.9)	4.313242
pasukan 5	(359.6, 0.1, 175.1)	4.493374
pasukan 6	(360.1, 0.1, 173.4)	3.272341
pasukan 7	(361.0, 0.1, 171.8)	4.633302
pasukan 8	(361.7, 0.1, 169.7)	2.718994
pasukan 9	(359.9, 0.1, 169.5)	2.813086
pasukan 10	(359.4, 0.1, 171.2)	3.773388
pasukan 11	(358.7, 0.1, 172.6)	3.053958
pasukan 12	(358.1, 0.1, 174.3)	2.554683
pasukan 13	(357.3, 0.1, 176.0)	4.16323
pasukan 14	(358.1, 0.1, 176.6)	4.135425

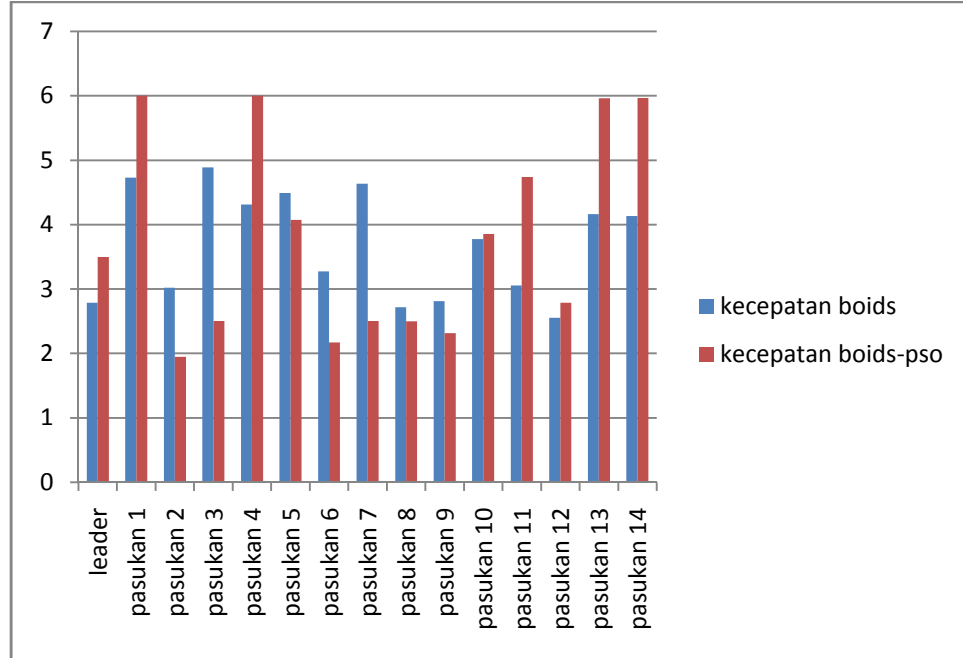
Kecepatan antar pasukan memiliki nilai yang berbeda untuk menyesuaikan kecepatan pasukan yang lain atau disebut dengan alignment. Penyesuaian kecepatan pasukan boids dengan PSO pada Gambar 4.11 memiliki kecepatan tertinggi pada pasukan 4 yaitu 6.000211 dan kecepatan terendah pada pasukan 2 yaitu 1.945068 sedangkan pasukan boids tanpa PSO memiliki kecepatan tertinggi pada pasukan 3 yaitu 4.886824 dan kecepatan terendah pada pasukan 12 yaitu 2.554683. perbandingan kecepatan saat melakukan kohesi antara pasukan boids dengan PSO dan pasukan pasukan boids tanpa PSO terdapat pada gambar 4.12.



Gambar 4.11 Hasil Percobaan Keempat Boids-PSO

Tabel 4.9 Hasil Percobaan Keempat Boids-PSO

Agen	Posisi kohesi	Kecepatan
leader	(362.7, 0.1, 174.0)	2.787423
pasukan 1	(361.2, 0.1, 175.6)	4.728606
pasukan 2	(358.1, 0.1, 171.3)	3.019634
pasukan 3	(363.2, 0.1, 170.6)	4.886824
pasukan 4	(359.0, 0.1, 176.9)	4.313242
pasukan 5	(359.6, 0.1, 175.1)	4.493374
pasukan 6	(360.1, 0.1, 173.4)	3.272341
pasukan 7	(361.0, 0.1, 171.8)	4.633302
pasukan 8	(361.7, 0.1, 169.7)	2.718994
pasukan 9	(359.9, 0.1, 169.5)	2.813086
pasukan 10	(359.4, 0.1, 171.2)	3.773388
pasukan 11	(358.7, 0.1, 172.6)	3.053958
pasukan 12	(358.1, 0.1, 174.3)	2.554683
pasukan 13	(357.3, 0.1, 176.0)	4.16323
pasukan 14	(358.1, 0.1, 176.6)	4.135425



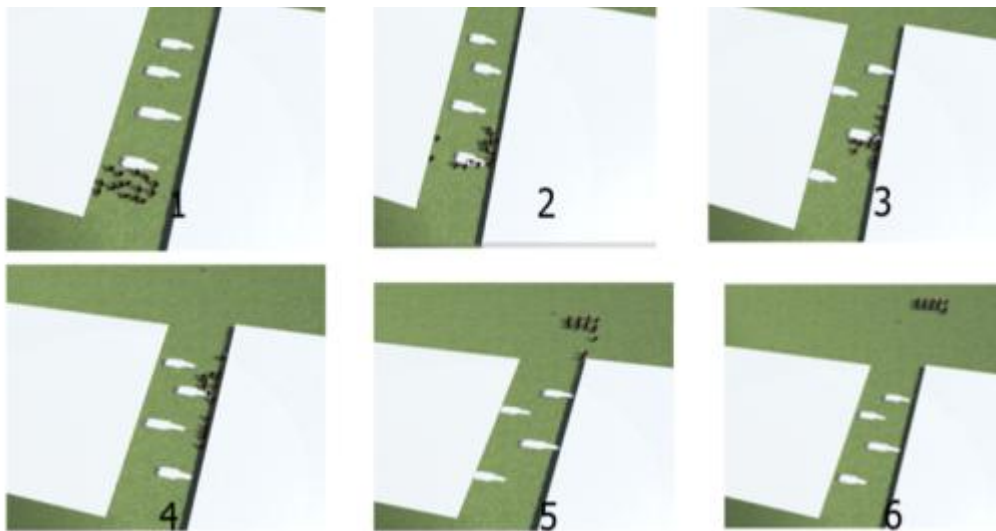
Gambar 4.12 Grafik Percobaan Keempat Boids-PSO

4.5. Pergerakan Pasukan dengan 4 Halangan Dinamis Dalam Lorong

Pada skenario ini kerumunan pasukan membentuk formasi kotak dengan posisi awal pasukan pada Tabel 4.1. Pasukan bergerak dengan formasi kotak menuju target dengan melewati tiga halangan dalam lorong dan berkumpul kembali pada formasi kotak kembali seperti pada Gambar 4.13. Pada skenario ini Algoritma boids digunakan tanpa optimasi PSO yang menghasilkan pasukan konvergen pada saat melakukan kohesi dalam waktu rata-rata 25.3048 detik.

Pada Gambar 4.13 bagian pertama pasukan melakukan pergerakan menuju target dengan melewati empat halangan dinamis dalam lorong. Gambar bagian kedua, ketiga dan keempat melakukan separasi dengan melewati empat halangan dinamis dalam lorong. Saat melewati lorong dengan halangan dinamis pasukan tertabrak dan terhalang dari pergerakan halangan sehingga pergerakan menuju target menjadi lebih lama. Pasukan pada gambar bagian kelima ada tiga pasukan yang tertinggal pada saat melakukan kohesi. Posisi pasukan saat melakukan kohesi dan kecepatan setiap pasukan pada gambar bagian keenam terdapat pada Tabel 4.10.

Pada skenario ini Algoritma boids digunakan dengan optimasi PSO yang menghasilkan pasukan konvergen pada saat melakukan kohesi dalam waktu rata-rata 24.5104 detik. Pada Gambar 4.14 bagian pertama pasukan melakukan pergerakan menuju target dengan melewati empat halangan dinamis dalam lorong. Gambar bagian kedua, ketiga dan keempat melakukan separasi dengan melewati empat halangan dinamis dalam lorong. Saat melewati lorong dengan halangan dinamis pasukan tertabrak dan terhalang dari pergerakan halangan sehingga pergerakan menuju target tetap tertinggal tapi tampak lebih rapat dari pada Gambar 4.13. Posisi pasukan saat melakukan kohesi dan kecepatan setiap pasukan pada gambar bagian keenam terdapat pada Tabel 4.11.



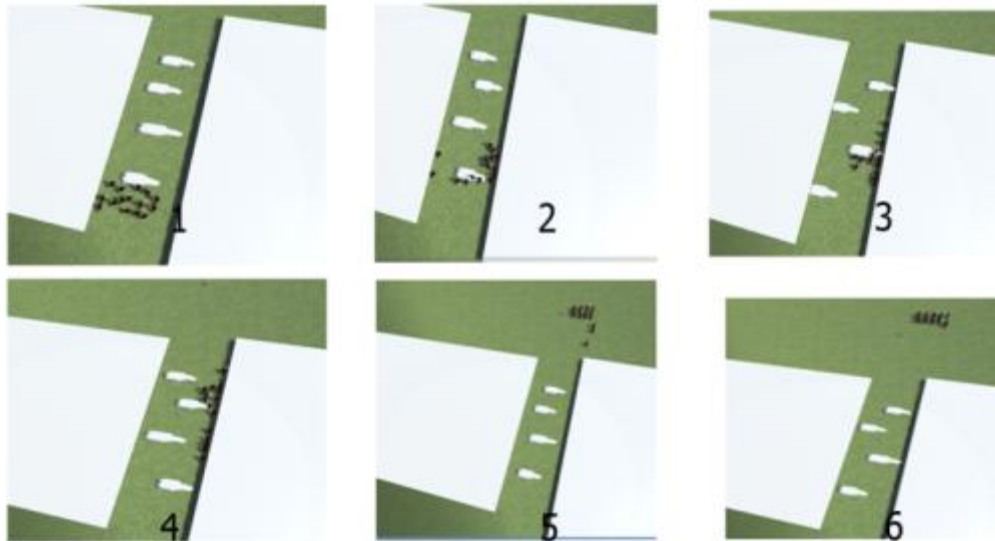
Gambar 4.13 Hasil Percobaan Kelima Boids

Kecepatan antar pasukan memiliki nilai yang berbeda untuk menyesuaikan kecepatan pasukan yang lain atau disebut dengan alignment. Penyesuaian kecepatan pasukan boids dengan PSO pada gambar 4.14 memiliki kecepatan tertinggi pada pasukan 8 yaitu 6.014611 dan kecepatan terendah pada pasukan 2 yaitu 1.897806 sedangkan pasukan boids tanpa PSO memiliki kecepatan tertinggi pada pasukan 9 yaitu 5.968565 dan kecepatan terendah pada leader yaitu 1.485733. Perbandingan

kecepatan saat melakukan kohesi antara pasukan boids dengan PSO dan pasukan pasukan boids tanpa PSO terdapat pada Gambar 4.14.

Tabel 4.10 Hasil Percobaan Kelima Boids

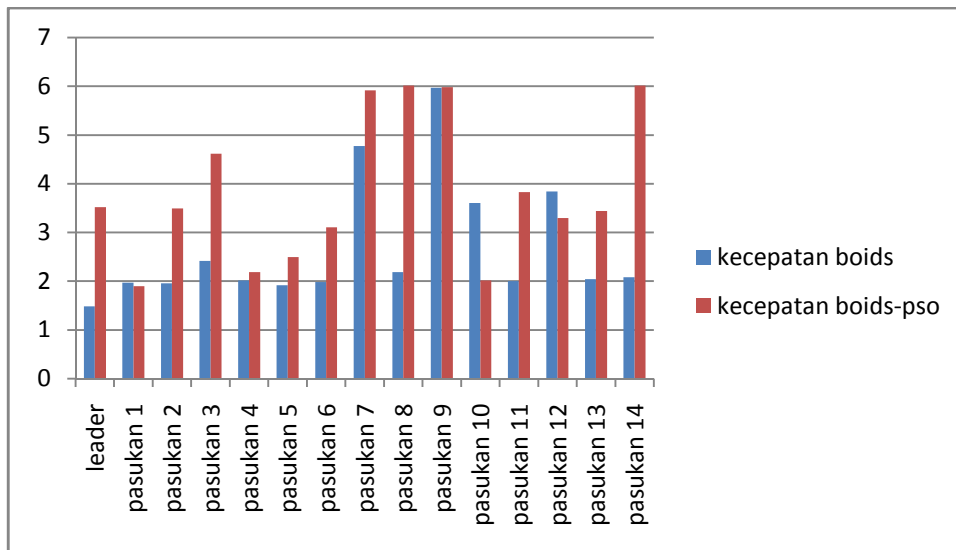
Agen	Posisi kohesi	Kecepatan
leader	(377.4, 0.1, 180.6)	1.485733
pasukan 1	(375.7, 0.1, 181.8)	1.972505
pasukan 2	(377.8, 0.1, 178.9)	1.957831
pasukan 3	(378.7, 0.1, 177.4)	2.418117
pasukan 4	(373.3, 0.1, 182.6)	2.012879
pasukan 5	(374.2, 0.1, 180.9)	1.919418
pasukan 6	(375.3, 0.1, 179.5)	1.980715
pasukan 7	(376.0, 0.1, 176.9)	4.772812
pasukan 8	(377.6, 0.1, 176.4)	2.189946
pasukan 9	(373.8, 0.1, 174.2)	5.968565
pasukan 10	(375.0, 0.1, 177.2)	3.605612
pasukan 11	(373.9, 0.1, 178.6)	2.000782
pasukan 12	(373.1, 0.1, 180.0)	3.839236
pasukan 13	(372.1, 0.1, 181.4)	2.041762
pasukan 14	(374.6, 0.1, 183.4)	2.080778



Gambar 4.14 Hasil Percobaan Kelima Boids-PSO

Tabel 4.11 Hasil Percobaan Kelima Boids-PSO

Agen	Posisi kohesi	Kecepatan
leader	(374.9, 0.1, 178.9)	3.518407
pasukan 1	(373.1, 0.1, 180.1)	1.897806
pasukan 2	(372.7, 0.1, 176.6)	3.494596
pasukan 3	(375.9, 0.1, 175.5)	4.616558
pasukan 4	(370.7, 0.1, 180.9)	2.185771
pasukan 5	(371.8, 0.1, 179.2)	2.494859
pasukan 6	(372.7, 0.1, 177.8)	3.103135
pasukan 7	(373.7, 0.1, 176.3)	5.917296
pasukan 8	(373.6, 0.1, 173.8)	6.014611
pasukan 9	(371.7, 0.1, 172.8)	5.983851
pasukan 10	(372.4, 0.1, 175.4)	2.01553
pasukan 11	(371.3, 0.1, 176.9)	3.830299
pasukan 12	(370.5, 0.1, 178.2)	3.294562
pasukan 13	(369.5, 0.1, 179.7)	3.440741
pasukan 14	(372.0, 0.1, 181.7)	6.013259

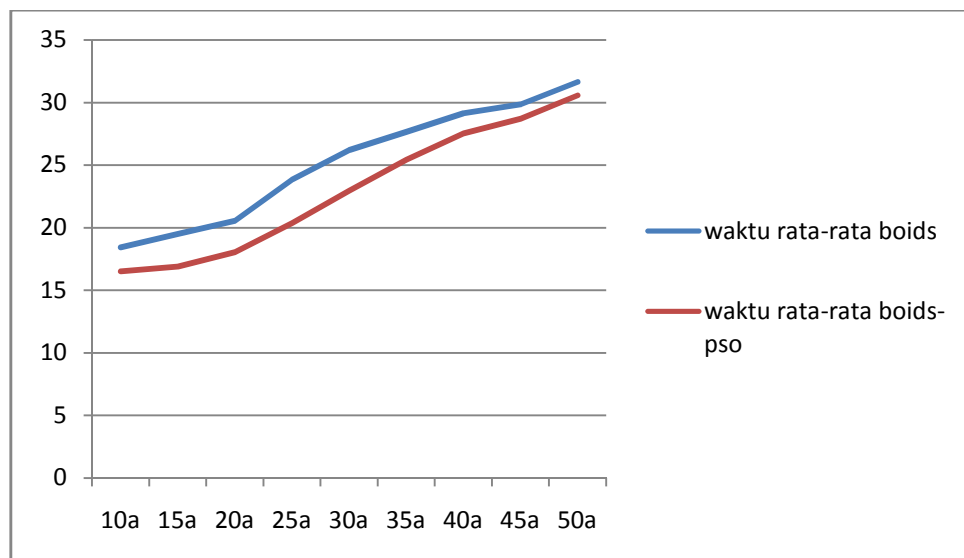


Gambar 4.15 Grafik Percobaan Kelima Boids-PSO

4.6. Pergerakan Pasukan dengan Variasi Populasi

Percobaan pada skenario keenam ini ditujukan untuk mengetahui berapa waktu dihitung saat pasukan memulai pergerakan, melewati halangan dan membentuk formasi kembali. Selain itu juga akan dihitung kompleksitas program percobaan ini dengan kata lain akan dicari rumus Big O dari program untuk penelitian ini. Sebuah algoritma perlu dihitung tingkat kompleksitasnya karena kompleksitas penerapan algoritma juga mempengaruhi pemakaian CPU dan *memory*. Semakin kompleks suatu algoritma maka akan semakin besar kebutuhan pemakaian CPU dan *memory* serta ruang pada *disk*. Program untuk penelitian ini dijalankan pada laptop dengan CPU Intel Core i5/1.6GHz 2.3 GHz, RAM 8 GB DDR3 dan VGA NVIDIA GeForce 720.

Pada skenario ini pengujian waktu sampai tujuan pasukan ke target dengan variasi populasi antara 10 pasukan sampai 50 pasukan dengan selisih 5 pasukan. Pergerakan pasukan seperti pada Gambar 4.10 dengan melewati lorong dan tiga halangan statis. Waktu dihitung saat pasukan memulai pergerakan, melewati halangan, membentuk formasi kembali dan menuju target yang telah ditentukan. Posisi awal pasukan terdapat pada Tabel 4.1 dan target untuk posisi leader pada koordinat (377.4, 0.1, 180.6).



Gambar 4.16 Waktu Rata-rata Populasi

Pada Tabel 4.12 setiap jumlah pasukan memiliki lama waktu menuju target yang berbeda. Semakin banyak jumlah pasukan waktu yang dibutuhkan menuju target semakin lama. Pasukan menuju target menggunakan boids dengan optimasi PSO dapat menempuh waktu yang lebih cepat dari pada pasukan menuju target menggunakan boids tanpa PSO grafik perbandingan kecepatannya terdapat pada Gambar 4.16.

Tabel 4.12 Hasil Percobaan Keenam Boids-PSO

jumlah pasukan	percobaan ke	waktu boid	waktu rata-rata boids	waktu boids- pso	waktu rata-rata boids- pso
10a	1	18.52	18.42666667	16.76	16.50666667
	2	18.58		16.36	
	3	18.18		16.4	
15a	1	19.52	19.50666667	17.42	16.88
	2	19.32		16.94	
	3	19.68		16.28	
20a	1	20.74	20.55333333	17.8	18.02666667
	2	20.34		18.1	
	3	20.58		18.18	
25a	1	23.38	23.84666667	19.22	20.37333333
	2	23.4		21.3	
	3	24.76		20.6	
30a	1	25.66	26.19333333	22.36	22.93333333
	2	26.54		22.7	
	3	26.38		23.74	
35a	1	28.42	27.64	26.56	25.41333333
	2	27.02		24.34	
	3	27.48		25.34	
40a	1	28.72	29.13333333	27.94	27.51333333
	2	29.72		26.88	
	3	28.96		27.72	
45a	1	29.28	29.85333333	29.26	28.69333333
	2	29.68		27.92	
	3	30.6		28.9	
50a	1	31.68	31.63333333	30.86	30.55333333
	2	31.56		30.44	
	3	31.66		30.36	

Untuk menghitung kompleksitas pada penelitian ini waktu rata-rata (detik) pergerakan dalam sekenario melewati tiga halangan dalam lorong pada Tabel 4.13 dengan 100 iterasi dan 200 iterasi menggunakan boids menunjukkan peningkatan waktu saat jumlah pasukan ditambah.

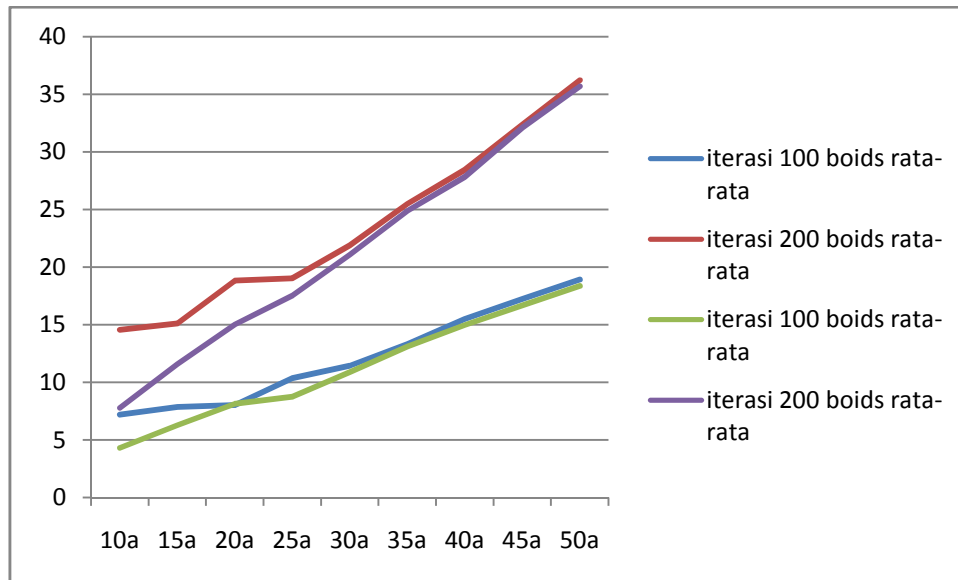
Tabel 4.13 Kompleksitas Boids

jumlah pasukan	percobaan ke	Iterasi 100 boids	iterasi 100 boids rata-rata	Iterasi 200 boids	iterasi 200 boids rata-rata
10	1	7.14	7.206666667	14.44	14.54
	2	7.14		14.4	
	3	7.34		14.78	
15	1	8.24	7.853333333	15.44	15.09333
	2	8.04		15.32	
	3	7.28		14.52	
20	1	7.72	8.02	22.32	18.82
	2	7.88		14.64	
	3	8.46		19.5	
25	1	9.46	10.37333333	18.12	19.01333
	2	10.26		18.82	
	3	11.4		20.1	
30	1	10.76	11.43333333	21.14	21.86
	2	13.28		23.78	
	3	10.26		20.66	
35	1	12.06	13.32666667	24.2	25.48667
	2	12.76		24.92	
	3	15.16		27.34	
40	1	15.76	15.48666667	28.68	28.42667
	2	15.8		28.5	
	3	14.9		28.1	
45	1	17.7	17.21333333	32.49	32.33333
	2	17.24		32.17	
	3	16.7		32.34	
50	1	20.12	18.90666667	37.46	36.2
	2	17.8		35.04	
	3	18.8		36.1	

Pada Tabel 4.14 pergerakan pasukan melewati tiga halangan dalam lorong dengan 100 iterasi dan 200 iterasi menggunakan boids dengan PSO menunjukkan peningkatan waktu (detik) saat jumlah pasukan ditambah dan lebih cepat jika dibandingkan dengan Tabel 4.13 seperti pada Gambar 4.17.

Tabel 4.14 Kompleksitas Boids-PSO

jumlah pasukan	percobaan ke	waktu Iterasi 100 boids-pso	iterasi 100 boids-pso rata-rata	Iterasi 200 boids-pso	iterasi 200 boids-pso rata-rata
10	1	4.52	4.3	7.72	7.766667
	2	4.18		7.78	
	3	4.2		7.8	
15	1	7.2	6.26	11.1	11.56
	2	5.82		12.52	
	3	5.76		11.06	
20	1	8.5	8.12	15.3	15.02667
	2	7.16		14.18	
	3	8.7		15.6	
25	1	9.059999	8.733333	17.72	17.51333
	2	8.52		17.56	
	3	8.62		17.26	
30	1	11.82	10.86667	22.04	21.07333
	2	10.44		20.64	
	3	10.34		20.54	
35	1	14.16	13.10667	25.3	24.87333
	2	12.14		24.2	
	3	13.02		25.12	
40	1	15.12	14.96667	27.94	27.84
	2	14.9		27.74	
	3	14.88		27.84	
45	1	16.62	16.64	32.18	32.12
	2	16.36		32.14	
	3	16.94		32.04	
50	1	18.04	18.34667	35.34	35.69333
	2	18.32		35.72	
	3	18.68		36.02	



Gambar 4.17 Grafik Kompleksitas Boids dan Boids-PSO

Untuk membandingkan kompleksitas program penelitian ini perlu ditentukan persamaan Big O program ini. Gambar 4.18 adalah gambar *loop* dari program penelitian. Dimana N adalah iterasi yang dilakukan sebanyak 50 atau 100 kali dengan algoritma boids dan PSO.

```

for i=1:N++
    %statemen
    If(%statemen ){
        Separation
    }
    If(%statemen ){
        Aligment
    }
    If(%statemen ){
        Kohesi
    }
    If(%statemen ){
        PSO
    }
end

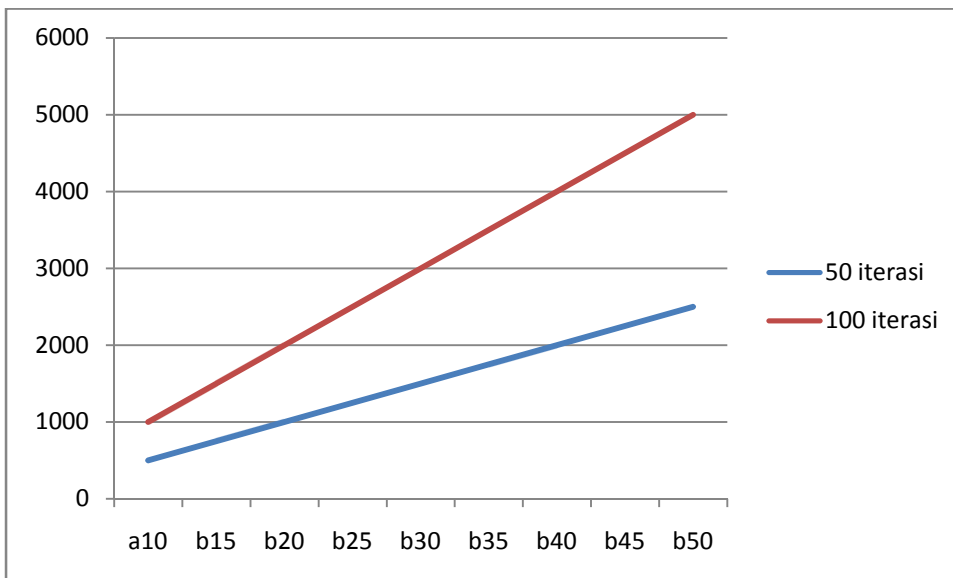
```

Gambar 4.18 *Loop* Program Peneletian

Tabel 4.15 Kompleksitas Big O

populasi	Kompleksitas O(N)	
	50 iterasi	100 iterasi
a10	500	1000
b15	750	1500
b20	1000	2000
b25	1250	2500
b30	1500	3000
b35	1750	3500
b40	2000	4000
b45	2250	4500
b50	2500	5000

Tabel 4.13 dan Tabel 4.14 menunjukkan bahwa setiap agen kompleksitas yang dibutuhkan untuk menyelesaikan 50 atau 100 iterasi adalah 1,5 kali lebih banyak. Sedangkan pada tabel 4.15 setiap penambahan 10 agen akan menambah nilai kompleksitas program sebanyak 1,2 kali. pada Gambar 4.17 dan Gambar 4.18 menunjukkan kesamaan pola yang menunjukkan bahwa kompleksitas dalam program penelitian ini dengan kompleksitas O(N).



Gambar 4.18 Grafik Kompleksitas

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan yang didapat pada percobaan pergerakan menggunakan boids dengan optimasi PSO dan pergerakan boids tanpa PSO berhasil menuju target dengan melewati halangan yang ada dan berhasil membentuk formasi kembali. Waktu rata-rata untuk membentuk formasi kembali setelah melewati halangan pada saat menggunakan boids dengan PSO lebih cepat 13.16 % dari boids tanpa PSO. Waktu rata-rata yang dibutuhkan untuk melewati halangan dan membentuk formasi kembali dengan variasi jumlah pasukan 10 sampai 50 pasukan dengan selisih 5 pasukan dibutuhkan pasukan 6.5 % lebih lama saat formasi ditambah 10 % pasukan dengan menggunakan boids dan 7.4 % lebih lama saat menggunakan boids dengan PSO.

5.2 Saran

Pergerakan pasukan menggunakan boids dapat mengendalikan kerumunan menuju target dengan melewati halangan yang ada dengan membuat formasi. Pada penelitian selanjutnya diharapkan dapat mengkombinasikan boids dengan optimasi yang lain dan skenario boids dengan optimasi PSO dilakukan pada saat pertempuran dalam permainan. Pada penelitian ini menggunakan perubahan arah 2 dimensi dan untuk penelitian selanjutnya diharapkan dapat menggunakan perubahan arah 3 dimensi dengan optimasi PSO.

halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- [1] Berg, Sindre S. *Artificial Intelligence Techniques in Real-Time Strategy Games -Architecture and Combat Behavior*, NTNU Norwegia, 2006
- [2] <http://clashofclans.wikia.com/wiki>, diakses maret 2015.
- [3] Reynolds C. “ Flocks, Herds, and Schools: A Distributed Behavioural Model”, *Computer Graphics*, 1987.
- [4] M. Dewi, M. Hariadi, and M.H. Purnomo, “Simulating the movement of the crowd in an environment using flocking”, *IEEE International conference on Instrumentaion,Communication, Information Technology and Biomedical Engineering* 2011.
- [5] Chen, Ying-Piing., Lin. Ying-Yin., “Controlling the movement of crowds in computer graphics by using the mechanism of particle swarm optimization”, *Aplied Soft Computing Elsevier*. 2009.
- [6] Cui Z., Shi Z. “Boid particle swarm optimisation”, *International Journal of Innovative Computing and Applications*, 2009.
- [7] Alaliyat S., Yndestad H. , Sanfilippo F., *Optimisation of Boids Swarm Model Based on Genetic Algorithm and Particle Swarm Optimisation Algorithm (Comparative Study)*, *Proceedings of the 28th European Conference on Modelling and Simulation (ECMS)*, Brescia, Italy; 2014
- [8] Ahmed, Hazem dan Glasgow, Janice, “Swarm Intelligence : Concepts, Models and Applications”, *School of Computing Queen’s University Kingston, Ontario, Canada K7L3N6*, 2012-585.
- [9] Dervis Karaboga, Beyza Gorkemli, Celal Ozturk, Nurhan Karaboga, “A Comprehensive Survey : Artificial Bee Colony (ABC) Algotirhm and Applications”, *Springer Science + Business Media B. V*, 11 March 2012
- [10] Darren Doherty, Colm O’Riordan, “The Design and Implementation of AI in Modern Computer Games”, *Department of Information Technology National University of Ireland Galway*
- [11] Franklin, Stan, “Autonomous Agents as Embodied AI”, *Cybernetics and System*, 28 : 6 (1997) 499-520.
- [12] Thalmann, D., Musse, S.R., Kallmann, M. “Virtual Human’ Behaviour : Individuals, Groups, and Crowd”, *Proceeding of Digital Media Futures*. 1999.
- [13] Reynolds C. *Steering behaviors for autonomous characters*. In *Game Developers Conference*, Miller Freeman Game Group, San Francisco, CA, USA, 1999.
- [14] J. Kennedy and R. C. Eberhart. *Particle swarm optimization*. In *Proceedings of the 1995 IEEE International Conference on Neural Networks*. IEEE Service Center, Piscataway, 1995.
- [15] Y.ShiandR.C.Eberhart.*Parametersselection in particle swarm optimization*. In V. W. Porto, N. Saravanan, D. Waagen, and A. Eibe, editors,*Proceedings of the Seventh Annual Conference on Evolutionary Programming*., Springer-Verlag, 1998.

LAMPIRAN

Percobaan Pertama (waktu/detik)			Percobaan Kedua (waktu/detik)	
PERCOBAAN KE	BOIDS	BOIDS-PSO	BOIDS	BOIDS-PSO
1	8.639999	6.32	9.74	8.28
2	9.139999	6.34	10.08	7.96
3	9.24	6.4	9.7	8.02
4	8.639999	6.3	9.74	8.08
5	8.86	6.36	9.8	8.099999
6	8.2	6.32	10.3	8.139999
7	8.26	5.98	10.36	8.2
8	9.26	6.04	11.62	7.6
9	8.84	6.1	11.9	7.66
10	8.3	6.34	9.96	8.46
11	8.36	6.4	10.2	8.62
12	8.34	6.92	10.26	8.72
13	8.98	6.98	11.14	8.42
14	9.059999	6.74	10.78	8.46
15	7.94	6.8	10.84	8.52
16	8.599999	6.3	10.88	8.52
17	9.26	6.36	9.74	8.86
18	9.32	6.86	9.8	8.179999
19	7.9	6.06	10.3	8.24
20	7.52	6.1	10.36	8.28
21	8.639999	6.06	11.62	8.46
22	9.139999	6.12	11.9	8.62
23	9.24	5.68	9.96	8.72
24	8.639999	5.74	10.2	8.42
25	8.86	5.98	10.26	8.46
26	8.2	6.04	11.14	7.96
27	8.26	6.02	10.78	8.02
28	9.26	5.94	10.36	8.08
29	8.84	6.7	11.62	8.099999
30	8.3	5.8	11.9	8.139999
31	8.36	5.86	9.96	8.2
32	8.34	5.88	10.2	7.6
33	8.98	5.94	10.26	7.66
34	9.059999	6.22	11.14	8.46

35	7.94	6.28	10.78	8.62
36	8.599999	6.24	10.84	8.72
37	9.26	6.28	10.88	8.42
38	9.32	6.4	9.74	8.46
39	7.9	6.5	9.8	8.52
40	7.52	6.28	10.3	8.52
41	9.26	6.34	10.36	8.86
42	8.84	6.38	11.62	8.179999
43	8.3	6.92	11.9	8.24
44	8.36	6.98	9.96	8.28
45	8.34	6.74	10.2	8.139999
46	8.98	6.8	11.9	8.2
47	9.059999	6.3	9.96	7.6
48	7.94	6.36	10.2	7.66
49	8.599999	6.86	10.26	8.46
50	9.26	6.06	11.14	8.62
51	9.32	6.1	10.78	8.72
52	7.9	6.06	10.36	8.42
53	7.52	6.12	11.62	8.46
54	8.98	5.68	11.9	8.52
55	9.059999	5.74	9.96	8.02
56	7.94	5.98	10.2	8.08
57	8.599999	6.04	10.26	8.099999
58	9.26	6.02	11.14	8.139999
59	9.32	5.94	10.78	8.2
60	7.9	6.7	10.84	7.6
61	7.52	5.8	10.88	7.66
62	8.84	5.86	9.74	8.46
63	8.3	5.88	9.96	8.62
64	8.36	5.94	10.2	8.72
65	8.34	6.22	11.9	8.42
66	8.98	5.74	9.96	8.46
67	9.059999	5.98	10.2	8.52
68	7.94	6.04	10.26	8.52
69	8.599999	6.02	11.14	8.86
70	9.26	5.94	10.78	8.179999
71	9.32	6.7	10.36	8.24
72	7.9	5.8	11.62	8.28
73	7.52	5.86	11.9	8.139999

74	8.98	5.88	9.96	8.2
75	9.059999	5.94	10.2	7.6
76	7.94	6.22	10.26	7.66
77	8.599999	6.28	9.96	8.46
78	9.26	6.24	10.2	8.62
79	9.32	6.28	10.26	8.72
80	7.94	6.4	11.14	8.179999
81	7.94	6.5	10.78	8.24
82	8.599999	6.3	10.84	8.28
83	9.26	6.36	10.88	8.139999
84	9.32	6.86	9.74	8.2
85	7.9	6.06	9.96	8.62
86	7.52	6.1	10.2	8.72
87	8.98	6.06	11.9	8.179999
88	7.94	6.12	9.96	8.24
89	8.599999	5.68	10.2	8.28
90	9.26	5.74	10.78	8.139999
91	9.32	5.98	10.84	8.2
92	7.9	6.04	10.88	8.62
93	7.52	6.02	9.74	8.72
94	8.98	6.86	9.96	8.179999
95	9.26	6.06	10.2	8.24
96	9.32	6.1	11.9	8.28
97	7.9	6.06	9.96	8.2
98	7.52	6.12	10.2	7.6
99	9.26	5.68	11.9	7.66
100	8.84	5.74	9.96	8.46

Percobaan Ketiga (waktu/detik)			Percobaan Keempat (waktu/detik)	
PERCOBAAN KE	BOIDS	BOIDS-PSO	BOIDS	BOIDS-PSO
1	13.8	11.62	21.94	16.7
2	13.72	11.82	19.42	16.78
3	13.78	11.88	19.52	16.82
4	13.82	11.86	22.46	16.88
5	13.3	12.7	21.38	17.54
6	13.38	11.32	22.22	17.64
7	13.42	11.38	19.42	17.48
8	14.66	11.72	19.5	17.56
9	15.2	11.78	21.82	17.6
10	12.5	11.84	19.4	17.66
11	12.56	11.88	19.48	17.14
12	15.02	11.94	22.16	17.14
13	15.08	12	18.52	17.2
14	13.2	11.8	22.98	17.26
15	13.36	12.84	22.4	17.32
16	13.42	11.88	22.3	17.28
17	10.84	11.86	19.04	17.34
18	10.84	12.7	21.24	17.4
19	13.1	11.32	21.94	18.04
20	13.68	11.38	19.42	18.14
21	13.74	11.72	19.52	17.7
22	12.9	11.78	22.46	17.28
23	12.96	11.84	21.38	18.08
24	12.82	11.88	22.22	18.14
25	12.88	11.94	19.42	18.2
26	12.94	12	19.5	17.38
27	14.66	11.88	21.82	17.44
28	15.2	11.94	19.4	17.5
29	12.5	12	19.48	17.92
30	12.56	11.8	22.16	17.6
31	15.02	12.84	18.52	17.66
32	15.08	11.88	22.98	17.72
33	13.2	11.86	22.4	17.26
34	13.36	12.7	22.3	16.72
35	13.42	11.32	19.5	17.02

36	10.84	11.8	21.82	17.08
37	10.84	12.84	19.4	17.12
38	13.1	11.88	19.48	17.18
39	13.68	11.86	22.16	17.68
40	13.74	12.7	18.52	17.78
41	12.9	11.32	22.4	17.64
42	12.96	11.38	22.3	17.7
43	12.82	11.72	19.04	17.76
44	12.9	11.78	21.24	17.82
45	12.96	11.84	21.94	17.38
46	12.82	11.88	19.42	17.44
47	12.88	11.94	19.52	17.5
48	12.94	12	22.46	17.92
49	14.66	11.88	21.38	17.6
50	15.2	11.94	22.22	17.66
51	12.5	12	19.42	17.72
52	12.56	11.8	19.5	17.26
53	15.02	11.94	21.82	16.72
54	15.08	12	19.4	17.02
55	13.2	11.88	19.48	17.08
56	13.36	11.94	21.24	17.12
57	13.42	12	21.94	17.18
58	10.84	11.8	19.42	17.68
59	13.1	12.84	19.52	17.78
60	13.68	11.88	22.46	17.64
61	13.74	11.86	22.3	17.7
62	12.9	12.7	19.04	17.76
63	12.96	11.32	21.24	16.82
64	12.82	11.8	21.94	16.88
65	12.9	12.84	19.42	17.54
66	12.96	11.88	19.52	17.64
67	12.82	11.86	22.46	17.48
68	12.88	12.7	21.38	17.56
69	12.94	11.32	22.22	17.6
70	14.66	11.38	19.42	17.66
71	15.2	11.72	19.5	17.14
72	12.5	11.84	21.82	17.14
73	12.5	11.88	22.22	17.2
74	12.56	11.94	19.42	17.44

75	15.02	12	19.5	17.5
76	15.08	11.88	21.82	17.92
77	13.2	11.94	19.4	17.6
78	13.36	12	19.48	17.66
79	13.42	11.8	21.24	17.72
80	10.84	11.94	21.94	17.26
81	13.1	12	19.42	16.72
82	13.68	11.88	19.52	17.02
83	13.74	11.94	22.46	17.08
84	12.9	12	22.3	17.12
85	12.96	11.8	19.5	17.18
86	12.82	12.84	21.82	17.68
87	12.9	11.88	19.4	17.78
88	12.96	11.86	19.48	17.64
89	12.82	12.7	22.16	17.7
90	12.88	11.32	18.52	17.76
91	12.94	11.88	22.4	16.82
92	13.1	11.94	22.3	16.88
93	13.68	12	19.04	17.54
94	13.74	11.8	21.24	17.64
95	12.9	11.94	21.94	17.48
96	12.96	12	19.42	17.56
97	12.82	11.88	19.52	17.5
98	12.9	11.8	22.46	17.92
99	13.74	12.84	21.38	17.6
100	12.9	11.88	22.22	17.66

Percobaan Ketiga (waktu/detik)		
PERCOBAAN KE	BOIDS	BOIDS-PSO
1	25.96	24.72
2	25.86	24.56
3	25.32	24.62
4	25.34	24.68
5	25.2	24.72
6	25.26	25.14
7	23.8	25.2
8	25.52	25.26
9	25.58	25.06
10	25.52	23.88
11	25.58	23.94
12	24.92	24
13	25.04	24.7
14	25.04	24.74
15	25.1	25.54
16	25.16	25.6
17	25.22	25.66
18	25.1	24.26
19	26.1	24.32
20	27.76	20.36
21	27.82	24.74
22	24.8	24.8
23	22.52	24.86
24	22.58	25.02
25	25.56	25.16
26	25.04	23.34
27	25.04	24.72
28	25.1	24.56
29	25.16	24.62
30	25.22	24.68
31	25.16	24.72
32	26.1	25.14
33	27.76	25.2
34	27.82	25.06
35	24.8	23.88

36	22.52	23.94
37	22.58	24
38	25.56	24.7
39	25.34	24.74
40	25.2	25.54
41	25.26	25.6
42	23.8	25.66
43	25.52	24.26
44	25.58	24.32
45	25.52	20.36
46	25.58	24.74
47	24.92	24.8
48	25.04	24.86
49	25.04	25.02
50	25.1	25.16
51	25.16	23.34
52	25.22	24.72
53	24.92	24.56
54	25.04	24.62
55	25.1	24.68
56	25.16	24.72
57	25.22	25.14
58	25.04	23.88
59	26.1	23.94
60	27.76	24
61	27.82	24.7
62	24.8	24.74
63	22.52	25.54
64	22.58	25.6
65	25.56	25.66
66	25.04	24.26
67	25.04	24.32
68	25.1	20.36
69	24.92	24.74
70	25.04	24.8
71	25.04	25.66
72	25.1	24.26
73	25.16	24.32
74	25.22	20.36

75	24.92	24.74
76	25.04	24.8
77	25.1	24.86
78	25.16	25.02
79	25.22	25.16
80	25.16	23.34
81	25.22	24.72
82	24.92	24.56
83	25.04	24.62
84	25.1	24.68
85	25.16	25.14
86	25.22	23.88
87	25.04	23.94
88	26.1	24
89	27.76	24.7
90	27.82	24.74
91	24.8	25.54
92	22.52	25.6
93	22.58	25.66
94	25.22	24.32
95	25.04	20.36
96	26.1	24.74
97	27.76	24.8
98	27.82	24.86
99	27.76	25.02
100	27.82	25.16

BIOGRAFI



Syahri Mu'min, Lahir di Mojokerto 16 Juni 1988, Menyeam pendidikan dasar di Madrasah Ibtidaiyah Pekukuhan Kabupaten Mojokerto, lalu melanjutkan ke SMPN 2 Mojosari kabupaten Mojokerto dan MAN Mojosari Kabupaten Mojokerto. Tahun 2006 Menempuh Pendidikan S1 UIN Maulana Malik Ibrahim Malang jurusan teknik Informatika. Penulis aktif dalam kewirausahaan dan kegiatan Relawan Palang Merah Indonesia di Wilayah Mojokerto dan Kota Malang.