TUGAS AKHIR – TE 091399

PENGATURAN PERILAKU PASUKAN *NON PLAYER CHARACTER* MENGGUNAKAN METODE *FLOCKING BEHAVIOR* BERBASIS *AGENT* PADA PERMAINAN *REAL TIME STRATEGY*

Priyodiva Robby Nugroho
NRP 2209100106

Dosen Pembimbing
Mochamad Hariadi, ST., M.Sc., Ph.D.
Christyowidiasmoro, ST., MT.

JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2014

FINAL PROJECT – TE 091399

BEHAVIOR ARRANGEMENTS OF NON PLAYER CHARACTER TROOPS USING AGENT-BASED FLOCKING BEHAVIOR ON REAL-TIME STRATEGY GAME

Priyodiva Robby Nugroho
NRP 2209100106

Advisors
Mochamad Hariadi, ST., M.Sc., Ph.D.
Christyowidiasmoro, ST., MT.

DEPARTMENT OF ELECTRICAL ENGINEERING
Faculty of Industrial Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2014

ABSTRAK

Pergerakan berkelompok sekumpulan pasukan, akan terlihat natural apabila dapat bergerak dengan memperhitungkan pergerakan pasukan lainnya. Pada permainan *real time strategy* Lume Wars, telah dirancang sistem perilaku pasukan yang menerapkan metode *flocking behavior* berbasis *agent*, guna memenuhi kebutuhan tersebut. Sistem perilaku pasukan yang dibuat harus diuji cobakan pada *device* yang berbeda-beda untuk memperoleh spesifikasi minimal untuk menjalankan sistem tersebut. Sistem yang dapat diterima adalah sistem yang dianggap sudah baik berdasarkan pendapat para pemain dan juga dari hasil pengujian pada masing-masing *device*. Dari hasil survei yang dilakukan, responden yang menganggap sistem perilaku ini sudah baik berjumlah tidak lebih dari 30%, sehingga dirasa perlu adanya pengembanangan lebih lanjut terkait perilaku yang diterapkan pada pasukan. Sedangkan dari pengujian device didapatkan spesifikasi minimal untuk dapat menjalankan permainan Lume Wars menggunakan 10 unit pasukan dengan FPS diatas nilai 60 adalah menggunakan *device* dengan sistem operasi android versi 4.1, CPU Dual Core 1,2 Ghz dan RAM 1GB.

Kata kunci : kecerdasan buatan, permainan *mobile,steering behavior, flocking behavior , real time strategy*

ABSTRACT

A group of troops movement, will look natural if can calculate movement of their neigbor troops. In the real time strategy games Lume Wars, we've designed system of troop behavior which will implement a agent-based flocking behavior method to fulfil that condition. The behavioral system of troops must be tried in different device to know the minimum specification for running this game. This behavioral system of troop will be accepted, if there are many of game tester says that the troop behavior is good enough and from the test result on every device. From the survei we retrieve, there are less than 30% tester that accept that system of troops behavior, so that we think the troop behavior system must be developed furher. From a device trial, we get the minimum specification that can run Lume Wars game with 10 unit of troops with minimum FPS above 60 is a device with operating system android version 4.1, CPU Dual Core 1,2 GHz, and RAM 1GB

Keyword: artificial intelligent, mobile game, steering behavior, flocking behavior, real time strategy

PENGATURAN PERILAKU PASUKAN *NON PLAYER CHARACTER* MENGGUNAKAN METODE *FLOCKING BEHAVIOR* BERBASIS *AGENT* PADA PERMAINAN *REAL TIME STRATEGY*

TUGAS AKHIR

Diajukan guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada
Bidang Studi Teknik Komputer dan Telematika
Jurusan Teknik Elektro
Institut Teknologi Sepuluh Nopember

Menyetujui:

Dosen Pembimbing I	Dosen Pembimbing II
Mochamad Hariadi, ST., M.Sc., Ph.D.	Christyowidiasmoro, ST., MT
NIP. 196912091997031002	NIP. 198301272009121004

SURABAYA
JANUARI, 2014

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT yang telah melimpahkan berkah, rahmat, serta hidayah-Nya sehingga penulis dapat menyelesaikan tugas akhir ini dengan judul: **Pengaturan Perilaku Pasukan *Non Player Character* menggunakan Metode *Flocking Behavior* berbasis *Agent* pada Permainan *Real Time Strategy*.**

Penelitian ini diusulkan dalam rangka menyelesaikan permainan Lume Wars yang merupakan permainan orisinil karya Teknik Komputer dan Telematika ITS. Selain itu, penelitian ini dilakukan sebagai prasyarat penulis untuk mendapatkan gelar Sarjana Teknik dari Jurusan Teknik Elektro ITS, bidang studi Teknik Komputer dan Telematika. Pengerjaan tugas akhir ini tak lepas dari bantuan berbagai pihak. Oleh karena itu penulis ingin menyampaikan terimakasih kepada:

1. Keluarga penulis, khususnya bapak dan ibu yang senantiasa memberikan dukungan baik dukungan material maupun spiritual.
2. Bapak Dr. Tri Arief Sardjono, S.T., M.T. selaku Ketua Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember.
3. Bapak Mochamad Hariadi ST., M.Sc., Ph.D. dan Bapak Christyowidiasmoro, ST., MT. selaku dosen pembimbing serta Bapak Dr. Supeno Mardi Susiki Nugroho, ST., MT. yang selalu memberikan saran, arahan dan bantuan dalam penyusunan tugas akhir ini. Penulis memohon maaf apabila ada kesalahan dari penulis selama proses pembelajaran ini.
4. Bapak ibu dosen pengajar Jurusan Teknik Elektro, khususnya bidang studi Teknik Komputer dan Telematika atas bimbingan, pengajaran, serta perhatiannya selama ini.
5. Tim pengerjaan permainan Lume Wars yang telah berupaya semaksimal mungkin sehingga permainan ini dapat selesai hingga sejauh ini
6. Seluruh teman-teman asisten bidang studi Teknik Komputer dan Telematika serta seluruh teman-teman angkatan e-49 Elektro ITS.

Kesempurnaan hanya milik Allah SWT, penyusunan tugas akhir ini tentu masih banyak kekurangan. Untuk itu penulis sangat

mengharapkan kritik dan saran yang membangun. Semoga penelitian ini dapat memberikan manfaat bagi kita semua.

Surabaya, Januari 2014

Penulis

DAFTAR ISI

ABSTRAK i
ABSTRACT iii
KATA PENGANTAR v
BAB 1 PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Permasalahan 3
1.3 Tujuan 3
1.4 Batasan Masalah 3
1.5 Sistematika Penulisan 3
BAB 2 TEORI PENUNJANG 5
2.1 *Real Time Strategy* (RTS) 5
2.2 Lume Wars 6
2.3 *Non-Player Character* (NPC) 7
2.4 *Finite State Machine* (FSM) 7
2.5 *Steering Behavior* 8
2.6 *Flocking Behavior* 10
2.7 GameMaker:Studio 13
BAB 3 DESAIN SISTEM DAN IMPLEMENTASI 15
3.1 Penentuan Perilaku Pasukan 16
3.2 Perancangan FSM Perilaku Pasukan 18
3.2.1 Pengecekan Kondisi Hero 18
3.2.2 Penentuan FSM Pasukan 20
3.3 Penerapan FSM Perilaku Pasukan 22
3.3.1 Perancangan Desain Animasi Pasukan 22
3.3.2 Penentuan Parameter dan Persamaan pada Perilaku Pasukan. 25
3.4 Penyatuan Bagian Permainan 29
BAB 4 PENGUJIAN DAN ANALISA 31
4.1 Pengujian *Device* 31
4.1.1 Hasil *Frame per Second* 33
4.1.2 Hasil CPU *Load* 36
4.1.3 Hasil Memory Usage 38
4.2 Pengujian Kesesuaian FSM Perilaku 41
4.3 Pengujian Survei Pengguna 43
4.2.1 Teknis Pelaksanaan Survei 43
4.2.2 Hasil Survei 44
BAB 5 PENUTUP 45
5.1 Kesimpulan 45

5.2 Saran 45
DAFTAR PUSTAKA 47
LAMPIRAN 49
BIOGRAFI PENULIS 71

DAFTAR GAMBAR

Gambar 1.1 Desain 5 karakter utama Lume Wars 1
Gambar 1.2 Pembagian tugas pengerjaan pada permainan Lume Wars 2
Gambar 2.1 Contoh permainan RTS *General Zero Hour*..... 5
Gambar 2.2 Tampilan awal permainan Lume Wars 6
Gambar 2.3 Model *finite state machine* sederhana..... 8
Gambar 2.4 *Separation* dalam *flocking* 11
Gambar 2.5 *Alignment* dalam *flocking* 12
Gambar 2.6 *Cohesion* dalam *Flocking* 12
Gambar 3.1 Diagram Blok Penelitian..... 15
Gambar 3.2 Diagram Blok Perancangan FSM Perilaku Pasukan 18
Gambar 3.3 FSM perilaku pasukan pada kondisi *hero alive* 21
Gambar 3.4 FSM perilaku pasukan pada kondisi *hero dead* 22
Gambar 3.5 Sprite pasukan ras Arian 23
Gambar 3.6 *Sprite* pasukan untuk ras Atlantis..... 23
Gambar 3.7 *Sprite* pasukan untuk ras Borneo 24
Gambar 3.8 Sprite pasukan untuk ras Maya 24
Gambar 3.9 Sprite pasukan untuk ras Outer 25
Gambar 3.10 Sprite animasi ledakan pada pasukan yang mati..... 25
Gambar 3.11 Ilustrasi Pasukan mengikuti follow point..... 26
Gambar 3.12 Langkah pencarian posisi *follow point*..... 27
Gambar 3.13 Illustrasi Pasukan menghindari pergerakan *hero* 28
Gambar 3.14 Animasi pasukan dalam Lume Wars..... 30
Gambar 4.1 Hasil FPS pada *state follow* 33
Gambar 4.2 Hasil FPS pada *State Attack*..... 34
Gambar 4.3 Hasil FPS pada *State Idle*..... 35
Gambar 4.4 Hasil CPU *load* pada *state follow* 36
Gambar 4.5 Hasil CPU *load* pada *state attack* 37
Gambar 4.6 Hasil CPU *load* pada *state idle* 38
Gambar 4.7 Hasil *memory usage* pada *state follow*..... 39
Gambar 4.8 Hasil *memory usage* pada *state attack* 40
Gambar 4.9 Hasil *Memory Usage* pada *state idle*..... 41

DAFTAR TABEL

Tabel 3.1 Daftar Parameter pada pasukan 26
Tabel 4.1 Spesifikasi *Smartphone* untuk pengujian..... 32
Tabel 4.2 Spesifikasi *Tablet* untuk pengujian..... 32
Tabel 4.3 Hasil Pengujian FSM Perilaku 42
Tabel 4.4 Daftar Pertanyaan pada kuisioner..... 43
Tabel 4.5 Hasil Jawaban Kuisioner 44

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Permainan *Real-Time Strategy* (RTS) merupakan salah satu genre permainan yang mensimulasikan pertempuran antara 1 atau lebih pemain. Dalam genre permainan ini, pada umumnya pemain diberikan kemampuan untuk mengendalikan pasukan atau unit tertentu dengan tujuan menaklukan suatu wilayah atau mengalahkan musuhnya. Misi yang dilakukan untuk mengalahkan musuh umumnya berbeda-beda. Pada permainan yang akan dibuat ini, misi utama adalah menghancurkan base milik musuh.

Gambar 1.1 Desain 5 karakter utama Lume Wars

Dalam pengerjaan tugas akhir ini, dibentuk tim yang akan mengembangkan permainan RTS berjudul Lume Wars. Lume Wars merupakan permainan dengan tema peperangan antara 5 ras yang memungkinkan pemain untuk memilih menjadi salah satu ras yang ada dan mengalahkan ras lainnya. Kelima ras yang ada memiliki karakter hero yang mempunyai kemampuan berbeda-beda. Desain awal karakter hero kelima ras seperti pada gambar 1.1, yang dibuat oleh mahasiswa pascasarjana *Gametech* Hadi Prayogo.

Pengembangan permainan Lume Wars ini juga dibagi menjadi beberapa bagian utama. Bagian pertama adalah pembuatan desain karakter dan lingkungan, juga desain skenario dan jalan cerita seluruh elemen dalam permainan. Dari skenario yang ada akan diimplementasikan pada karakter Hero dan pasukan dalam permainan agar bertindak sesuai dengan desain skenario di awal permainan.

Dalam permainan ini juga dibuat *intelligent agent* untuk gameplay permainan yang akan mengatur kondisi dalam permainan dengan parameter status dari permainan tersebut. Juga akan dibangun jaringan untuk menghubungkan para pemain dalam satu *server* agar dapat bermain secara *online* dan *real-rime.* Ilustrasi tiap bagian dalam pengembangan permainan Lume Wars diatas digambarkan seperti pada gambar 1.2.

Gambar 1.2 Pembagian tugas pengerjaan pada permainan Lume Wars

Salah satu bagian dalam pengembangan permainan ini adalah bagian implementasi skenario pada karakter *hero* dan pasukan yang ada. Bagian yang akan dikerjakan dalam tugas akhir ini adalah pengaturan perilaku pasukan NPC. Pasukan NPC yang ada dalam permainan ini memiliki penggunaan yang berbeda dalam permainan RTS lain yang dapat dikendalikan secara langsung oleh pemain, dalam permainan Lume Wars, pasukan yang ada merupakan *non player character* atau berarti pergerakannya tidak dapat dikendalikan secara langsung oleh pemain, melainkan perilakunya akan ditentukan dengan *trigger* dari lingkungan permainan dan juga *trigger* dari kondisi *hero*.

Pasukan NPC akan dirancang terlebih dahulu perilakunya menggunakan FSM, sehingga pasukan NPC tersebut akan memiliki perilaku dasar tanpa pengaruh kondisi lingkungan. Kemudian terdapat pula perilaku pasukan yang dipenngaruhi lingkungan dan karakter Hero, saat kondisi tersebut pasukan NPC akan menggunakan metode *flocking behavior.* Dengan flocking tersebut, dapat diterapkan 3 bagiannya dalam perilaku pasukan, yaitu bagian *cohesion, alignment,* dan *separation.*Sehingga pasukan dapat merespon kondisi dari lingkungan dan karakter utama sesuai dengan skenario yang dibuat.

1.2 Permasalahan

Dalam permbuatan permainan Lume Wars, dibutuhkan pasukan yang dapat bertindak tanpa perlu diberi input secara langsung oleh pemain atau disebut *non player character* (NPC). Perilaku pasukan yang akan diimplementasikan dalam permainan Lume Wars harus sesuai dengan kondisi-kondisi yang ada didalam permainan.

1.3 Tujuan

Tujuan penelitian ini adalah merancang dan menerapkan skenario perilaku pada sekumpulan pasukan yang nantinya akan digunakan dalam permainan Lume Wars.

1.4 Batasan Masalah

Batasan masalah dari Tugas Akhir ini adalah:

a. Skenario perilaku yang dibuat hanya digunakan pada bagian pasukan NPC saja.
b. Pasukan pada masing-masing ras berjumlah 10 unit.
c. Metode yang digunakan adalah *flocking behavior* yang akan dipengaruhi oleh suatu *agent* yang merupakan karakter *hero* dalam permainan.
d. Target *platform* permainan yang dibuat adalah *android device.*

1.5 Sistematika Penulisan

Laporan penelitian Tugas akhir ini tersusun dalam sistematika dan terstruktur sehingga lebih mudah dipahami dan dipelajari oleh pembaca maupun seseorang yang hendak melanjutkan penelitian ini.Alur sistematika penulisan laporan penelitian ini yaitu :

1. Bab I Pendahuluan
   Bab ini berisi uraian tentang latar belakang permasalahan, penegasan dan alasan pemilihan judul, tujuan penelitian, metodologi penelitian dan sistematika laporan.

2. Bab II Teori Penunjang
   Pada bab ini berisi tentang uraian secara sistematis teori-teori yang berhubungan dengan permasalahan yang dibahas pada penelitian ini. Teori-teori ini digunakan sebagai dasar dalam penelitian, yaitu informasi terkait *flocking behavior*, *steering behavior*, dan teori-teori penunjang lainya.

3. Bab III Desain Sistem dan Implementasi
   Bab ini berisi tentang penjelasan-penjelasan terkait sistem yang akan dibuat. Guna mendukung itu digunakanlah blok diagram agar sistem yang akan dibuat dapat terlihat dan mudah dibaca untuk diimplentasikan pada pembuatan sistem perilaku pasukan.

4. Bab IV Pengujian dan Analisis
   Bab ini menjelaskan tentang pengujian yang dilakukan terhadap sistem dalam penelitian ini dan menganalisa sistem. Spesifikasi *device* yang digunakan dalam pengujian juga disebutkan dalam bab ini. Sehingga ketika akan dikambangkan lebih jauh, spesifikasi perlengkapannya bisa dipenuhi dengan mudah tanpa harus melakukan ujicoba perangkat lunak maupun perangkat keras lagi.

5. Bab V Penutup
   Bab ini merupakan penutup yang berisi kesimpulan yang diambil dari penelitian dan pengujian yang telah dilakukan. Saran dan kritik yang membangun untuk pengembangkan lebih lanjut juga dituliskan pada bab ini.

BAB 2
TEORI PENUNJANG

2.1 *Real Time Strategy* (RTS)

Permainan *real time strategy* merupakan salah satu genre permainan yang mensimulasikan pertempuran antara 2 atau lebih kelompok. Setiap kelompok dapat dikendalikan oleh pemain atau komputer dengan menggunakan kecerdasan buatan. Pada genre permainan ini, setiap kelompok dapat membangun struktur bangunan, merekrut pasukan, memperebutkan area-area strategis dalam map, menghancurkan kekuatan lawan dan memenangkan permainan.

Dalam permainan RTS terdapat banyak jenis kekuatan yang dapat digunakan untuk menghancurkan musuh, contoh pada gambar 2.1 digunakan serangan roket untuk menghancurkan kubu lain. Ada banyak jenis kekuatan yang digunakan dalam permainan RTS tergantung dari permainan apa yang dimainkan.

Gambar 2.1 Contoh permainan RTS *General Zero Hour*

Dalam permainan ini, pemain juga harus memprediksi strategi apa yang akan digunakan lawan untuk bertahan, sehingga nantinya akan dapat menyerang lawan dengan memanfaatkan celah dalam strategi pertahanan lawan. Begitu pula tim yang dikendalikan oleh komputer harus dapat mengenali perilaku lawan, tujuan dan strategi apa yang

digunakan, sehingga akan menyerang berdasarkan pengetahuan tersebut.[1]

2.2 Lume Wars

Lume Wars merupakan permainan yang dikembangkan oleh tim beranggotakan 5 orang dari teknik komputer dan telematika. Permainan ini memiliki latar belakang terjadi pertempuran antara 5 ras yaitu Aria, Atlantic, Borneo, Maya, Outer. Dalam permainan, setiap ras akan menghancurkan markas lawan untuk dapat memenangkan permainan.

Lume Wars mengisahkan perebutan kekuasaan antara kelima ras yang ada, dimana tiap ras memiliki kekuatan masing-masing. Lume Wars merupakan peperangan antara kelima ras tersebut untuk memperebutkan merupakan sekumpulan cahaya yang mampu memberikan kekuatan pada ras yang berhasil memperoleh cahaya tersebut. Pada gambar 2.2 merupakan tampilan awal dari permainan Lume Wars, dimana terdapat pilihan menu, logo permainan, dan salah satu karakter dalam permainan.

Gambar 2.2 Tampilan awal permainan Lume Wars

Dalam pembuatan permainan Lume Wars dibagi menjadi 2 bagian pengerjaan yaitu desain dan programming. Bagian desain akan

mengerjakan desain karakter, bangunan, map, jalan cerita, antarmuka pengguna dan juga art book dari permainan lume wars. Bagian yang kedua adalah programming , dimana dalam pembuatan permainan ini dibutuhkan pemberian perilaku pada setiap karakter baik karakter Hero, Pasukan, atau Bangunan. Setiap perilaku pada tiap karakter ini perlu diprogram agar dapat bertindak sesuai skenario yang telah ditentukan.

Selain itu juga dibuat mekanik permainan yang akan memperhitungkan keseimbangan tiap karakter dan ras agar tidak terjadi ketidak seimbangan kekuatan yang menyebabkan permainan menjadi tidak menarik. Agar permainan ini dapat dimainkan oleh beberapa permain sekaligus perlu dibuat sistem jaringan yang dapat menghubungkan setiap pemain menggunakan jarinan *local area network* (LAN).

2.3 *Non-Player Character* (NPC)

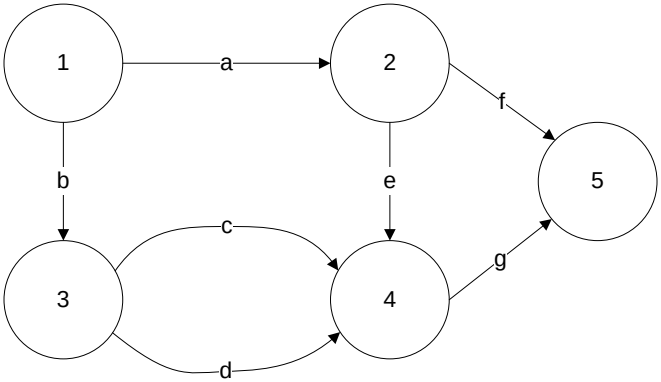
NPC dalam permainan komputer memiliki peran yang penting agar pemain tidak perlu mengendalikan seluruh karakter dalam permainan. NPC juga memiliki peran yang bervariasi, contohnya suatu NPC dapat dipergunakan sebagai pengamat (yang tidak terlibat langsung dalam pertempuran), sekutu, ataupun lawan. Suatu NPC pada umumnya ditentukan terlebih dahulu perilakunya menggunakan *finite-state machine* (FSM). Suatu aksi NPC dapat dipicu oleh suatu aksi karakter lain , atau masukan dari pemain.

Dalam pembuatan permainan, seorang desainer akan terlebih dahulu membuat skenario perilaku suatu karakter, dan kemudian dibuat FSM sesuai dengan skenario tersebut, karena FSM adalah cara yang cukup baik untuk membuat definisi awal dari perilaku sekuensial. Akan tetapi FSM akan menjadi cukup membosankan karena perilaku yang dihasilkan akan cenderung sama.

2.4 *Finite State Machine* (FSM)

Model *finite state machine* adalah suatu model matematika yang digunakan untuk mendefinisakan state state dalam membangun suatu program. FSM juga digunakan untuk menjelaskan bagaimana suatu objek dapat berubah dalam state suatu waktu.

Konstruksi model FSM secara sederhana yaitu terdapat susunan state dan fungsi transisional (*activity*). Dapat di gambarkan secara sederhana seperti berikut :



Gambar 2.3 Model *finite state machine* sederhana

Gambar 2.3, merupakan contoh finite state machine sederhana, dimana setiap lingkaran merepresentasikan *state* (1,2,3,4,5) yang nantinya akan mewakili setiap prilaku yang diinginkan. Fungsi transisi atau *activity* ditunjukkan pada gambar panah yang menghubungkan tiap-tiap lingkaran (a,b,c,d,e,f,g) menjadi jembatan antar *state* dan menjadi penghubung pada perilaku-perilaku yang dibuat.

2.5 *Steering Behavior*

Perilaku pergerakan yang dimaksud ini mengacu pada makalah Craig Reynold yang berjudul *steering behavior for autonomous character*[7], yang menjelaskan bagaimana suatu unit dapat bergerak dengan menggunakan komputasi yang dapat diterapkan pada simulasi atau permainan tanpa dikendalikan secara langsung. Perilaku bergerak suatu unit akan dibagi berdasarkan jumlah dan kesamaan perilakunya.

Sebagai contoh hewan rusa akan bergerak dengan 4 kaki dengan cara bergerak yang sama dengan hewan rusa lainnya, sehingga unit yang jenisnya sama akan memiliki perilaku yang sama. Selain itu terdapat pula pergerakan dengan jumlah banyak, sehingga perlu

digunakan komputasi yang lebih kompleks untuk mensimulasikan gerakan gerakan tersebut.

Berikut beberapa contoh perilaku *steering behavior* secara individu atau pasangan:

1. *Seek and Flee*
   Seek merupakan perilaku dimana unit bergerak menuju target yang diinginkan. Flee merupakan perilaku unit bergerak dari target yang telah ditentukan.

2. *Pursue and Evade*
   Perilaku *pursue* mirip dengan perilaku *seek*, hanya saja pada *pursue* target yang dituju memiliki kecepatan dan arah tersendiri, maka unit harus menghitung kira kira dimana posisi target setelah beberapa waktu tertentu, dan menuju ke lokasi tersebut.
   Evade merupakan perilaku menghindari unit lain yang sedang bergerak, dengan mengetahui arah dan kecepatan unit tersebut, maka apabila kedua unit akan bersimpangan di jalan, maka unit harus mengubah arah untuk menghindari kejadian tersebut.

3. *Wander*
   Wander merupakan pergerakan yang tidak memiliki tujuan, atau hanya berputara-putar pada suatu lokasi saja. Biasanya suatu unit akan diharuskan menuju arah tertentu yang memiliki nilai random, dan nilai tersebut selalu diubah setiap suatu waktu. Sehingga unit akan bergerak seolah kesana kemari tanpa tujuan.

4. *Arrival*
   Arrival merupakan perilaku menuju tujuan , berbeda dengan *seek* yang hanya akan bergerak statis, pergerakan pada *arrival* akan dipengaruhi gaya (*force*) yang dapat berubah ubah nilainya.

5. *Obstacle Avoidance*
   Perilaku ini akan terjadi apabila terdapat *obstacle* di jalan yang akan dilalui oleh suatu unit. Arah berjalan unit akan diberikan gaya tertentu yang akan membuat unit tersebut tidak menabrak *obstacle* tersebut.

Pada perilaku bergerak unit berkelompok atau berkerumun, perilaku yang tadi akan tetap digunakan , akan tetapi akan ditambahkan

dengan parameter tertentu agar bergeraknya tetap tampak seperti seharusnya. Contoh perilaku berkelompok dalam steering behavior adalah sebagai berikut :

1. *Crowd Path Following*
   Perilaku ini terjadi dengan menentukan *path* atau jalur bergerak sekelompok unit, dan unit tersebut akan mengikuti jalur tersebut sambil menghindari unit lainnya.

2. *Leader Following*
   Mengikuti *leader* merupakan metode yang umum digunakan, baik dalam implementasi permainan maupun robotika. Perilaku ini memungkinkan sekelompok unit menentukan *leader* terlebih dahulu kemudian bergerak sesuai dengan pergerakan *leader*.

3. *Flocking* (*combining: separation, alignment, cohesion*)
   Merupakan perilaku bergerak yang menyerupai pergerakan kawanan burung terbang. Dengan menerapkan 3 aturan dasar agar dapat bergerak searah dan tetap berada dalam kerumunan. Pada sub-bab berikutnya akan dibahas mengenai *flocking* lebih lanjut.

2.6 *Flocking Behavior*

Flocking merupakan istilah dari perilaku sekumpulan burung yang terbang secara berkelompok saat musim tertentu. Terbangnya sekelompok burung tersebut membentuk suatu pola yang beraturan, perilaku sekelompok burung yang memiliki pola bersama saat terbang ini kemudian disebut *Flocking*.

Contoh di alam yang menggambarkan suatu flocking adalah pergerakan burung yang sedang bermigrasi dari suatu daerah ke daerah lain ketika pergantian musim. Sebagai contoh adalah kelompok angsa yang terbang bersama-sama. Sekumpulan angsa tersebut terbang dengan menjaga jarak agar tidak terlalu dekat dan tidak terlalu jauh, dan memiliki arah yang sama.

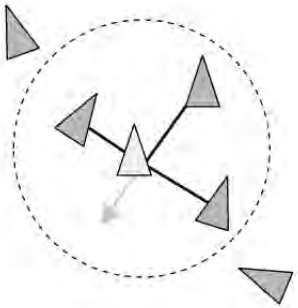
Dalam pergerakan karakter di dalam simulasi atau permainan, metode *flocking* ini juga dapat digunakan, sebagai contoh yaitu pada suatu permainan, metode ini cocok untuk diterapkan pada perilaku burung yang sedang terbang, sekumpulan ikan yang sedang berenang berkelompok, unit pasukan militer, dan juga pada kerumunan. Metode

ini juga pernah digunakan untuk mensimulasikan gerakan pasukan dalam permainan FPS *Unreal Turnament*[2]. *Flocking* juga dapat disimulasikan dan mendapatkan nilai yang baik ketika dikombinasikan dengan *Intelligent Moving* (IM), hasil yang didapat yaitu kemungkinan pasukan yang menerapkan *Flocking* dan IM akan memiliki kemungkinan memenangkan pertempuran daripada yang tidak menerapkannya[1].

Dalam makalah reynold mengenai perilaku flocking[6], setiap unit yang diberi perilaku flocking ini disebut *boid*. Tiap boid akan bergerak menyesuaikan pergerakan mereka dengan boid lain dengan mengikuti 3 aturan, yaitu: Separation, Cohesion, dan Alignment. Separation adalah gaya untuk menghindar, cohesion adalah gaya untuk berkumpul, dan alignment adalah gata untuk bergerak dengan arah dan kecepatan yang sama.

Ketiga aturan dasar dari *flocking behavior* dijelaskan sebagai berikut :

a. *Separation*

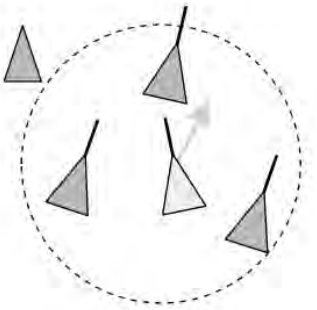


Gambar 2.4 *Separation* dalam *flocking* [4]

Separation mendefiniskan perilaku menjauh antara 1 unit dengan unti yang lain dalam 1 kelompok terlalu dekat akan bergerak menjauh untuk memberi jarak agar area tersebut tidak terlalu padat.

Pada gambar 2.4 digambarkan satu unit berada pada jarak yang terlalu dekat dengan unit lain, sehingga unit tersebut bergerak ke arah panah untuk menghindari daerah yang terlalu padat.

b. *Alignment*

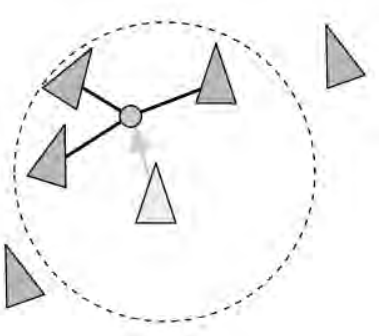


Gambar 2.5 *Alignment* dalam *flocking*[4]

Alignment mengatur agar unit menyesuaikan arah dalam pergerakan dari mayoritas unit lain. Dalam topik ini, pasukan akan menggunaka metode alingment dengan menyesuaikan dengan arah gerakan hero agent dalam permainan.

Pada gambar 2.5 diilustrasikan suatu unit menghadap arah yang berbeda dengan unit lainnya, sehingga unit tersebut menuju ke arah panah untuk menyesuaikan arah dengan unit lain.

c. *Cohesion.*



Gambar 2.6 *Cohesion* dalam *Flocking*[4]

Cohesion mengarahkan unit untuk bergerak menuju arah yang memiliki kepadatan unit rata-rata dari posisi sekelompok unit tersebut, sehingga nantinya unit-unit tersebut akan tampak berkumpul.

Pada gambar 2.6 diilustrasikan satu unit menuju ke arah panah untuk menuju daerah yang memiliki mayoritas unit untuk menjaga agar sekumpulan unit tersebut tetap berada dalam daerah satu kumpulannya. Dalam pembuatan permainan ini, unit pasukan akan bergerak menuju arah karakter utama untuk berkumpul di sekitar karakter tersebut.

2.7 GameMaker:Studio

GameMaker merupakan aplikasi game engine yang dikembangkan oleh yoyogames. GameMaker dirancang untuk membuat permainan dengan sistem drag and drop, sehingga para developer tidak perlu memahami tentang bahasa pemrograman. GameMaker membuat permainan berbasis gambar 2D, sehingga tidak diperlukan kemampuan untuk mengolah gambar 3D.

Versi terbaru dari GameMaker adalah GameMaker: Studio, yang mana sebelumnya versi terakhir yang di-release oleh yoyogames adalah versi 8.1. Dalam aplikasi game maker studio ini terdapat banyak fitur yang tidak terdapat pada versi versi sebelumnya, sebagai contoh fitur untuk build pada sistem operasi Android, iOS, OS X, HTML5, Ubuntu, Windows 8, Windows Phone 8.[3]

Pada GameMaker: Studio, setiap objek akan memiliki *event* dan *action* yang akan dijalankan dalam permainan. Event merupakan tempat dimana *action* akan dijalankan, contoh *event* pada GameMaker: Studio adalah *create*,*step*,*alarm*, dan *destroy*. Setiap *event* dapat memiliki banyak action. *Action* merupakan kondisi yang dijalankan pada suatu objek, seperti bergerak, menyerang, dan lain sebagainya. Pada GameMaker: Studio, *action* dapat dibuat baik menggunakan fitur *drag-and-drop* tanpa perlu pemrograman atau dengan script yang menggunakan bahasa pemrograman milik GameMaker sendiri yaitu *game maker language* (GML). Setiap *event* dan *action* yang dibuat, akan dijalankan sebanyak 30 kali setiap detik sesuai dengan jumlah *step* bawaan dari GameMaker: Studio.

BAB 3
DESAIN SISTEM DAN IMPLEMENTASI

Penelitian ini disusun untuk mendesain perilaku pasukan dan mengimplementasikannya pada setiap pasukan NPC dalam permainan Lume Wars. Skenario perilaku pasukan dibuat disesuaikan dengan kondisi-kondisi didalam permainan tersebut, sebagai contoh ketika terdapat musuh di sekitar pasukan, kondisi markas diserang, atau ketika hero mati setelah pertempuran. Contoh perilaku yang dirancang untuk pasukan adalah perilaku *spawn*, *idle*, *walk*, *attack*, dan perilaku lainnya. Perilaku bergerak pasukan secara berkelompok akan menerapkan 3 aturan *flocking behavior* yaitu *alignment*, *cohesion*, dan *separation* untuk membuat pasukan dapat memperhatikan kondisi pergerakan pasukan lain. Pada flocking behavior yang diterapkan pada pergerakan pasukan, akan terdapat satu unit yang akan menjadi acuan dalam kondisi pergerakan pasukan yakni karakter hero, yang akan menjadi leader bagi pasukan. Kondisi dan perilaku hero akan mempengaruhi pemilihan perilaku yang akan dilakukan pasukan didalam permainan.

Untuk dapat mengaplikasikan rancangan yang dimaksud penelitian ini disusun dalam beberapa tahap yang ditunjukkan pada gambar 3.1

Gambar 3.1 Diagram Blok Penelitian

3.1 Penentuan Perilaku Pasukan

Dalam permainan Lume Wars, pasukan akan memiliki perilaku atau kondisi yang masing-masing perilaku akan ditentukan *state* masing-masing. Perilaku-perilaku yang telah ditentukan adalah perilaku-perilaku berikut:

1. *Idle*

   Merupakan kondisi ketika pasukan tidak bergerak atau berada posisi yang tidak berubah.

2. *Walk*

   Kondisi dasar dari seluruh pergerakan pasukan, kondisi pasukan berjalan ini nantinya akan digunakan pada kondisi lainnya seperti *follow*, *evade*, *pursuit*, dan *escape.*

3. *Spawn & Respawn*

   Spawn dalam istilah permainan berarti munculnya monster atau karakter. Kondisi ini tidak dipengaruh oleh kondisi *Hero*. *Spawn* adalah kondisi munculnya unit pasukan ketika permainan baru saja dimulai, sedangkan *respawn* adalah kondisi muncul pasukan setelah mati didalam permainan. *Spawn* dan *respawn* pasukan telah ditetapkan waktunya yaitu setelah unit pasukan mati maka 3 detik kemudian pasukan akan hidup kembali. Baik kondisi *spawn* maupun *respawn* pasukan akan hidup di sekitar markas pasukan tersebut.

4. *Follow*

   Perilaku ini terjadi ketika terdapat *Hero* dalam arena permainan. Pada kondisi ini pasukan akan berjalan mengikuti pergerakan *Hero* dengan berada pada jarak tertentu dari *Hero*. Ketika melakukan kondisi ini maka pasukan akan menepapkan aturan *flocking* agar pergerakan tetap tampak natural.

5. *Evade*

   Pada perilaku *follow*, apabila terdapat kondisi dimana *hero* terdapat pada kerumunan pasukan, maka akan diberikan gaya evade pada pasukan untuk menghindari tabrakan dengan unit *hero*.

6. *Pursuit*

   Perilaku ini adalah kondisi dimana unit pasukan harus menuju target yang bergerak. Dalam permainan Lume Wars, perilaku ini terjadi ketika unit pasukan baru *respawn* setelah mati dan menuju kerumunan pasukan lainnya.

7. *Attack (Melee & Range)*

   Merupakan kondisi menyerang pasukan, pada permainan ini , terdapat 2 kondisi menyerang pasukan yaitu serangan jarak dekat (*Melee*) dan serangan jarak jauh (*Ranged*). Pasukan akan menyerang apabila terdapat musuh dalam jarak serangan pasukan. Pada permainan Lume Wars , pasukan *melee* memiliki jarak serangan 10, dan pasukan *ranged* memiliki jarak serangan 250.

8. *Back to Position*

   Perilaku ini dilakukan setelah pasukan melakukan penyerangan terhadap unit musuh. Kondisi ini mengharuskan unit pasukan untuk kembali ke posisi semula atau pada kerumunan pasukan lainnya.

9. *Escape*

   Merupakan kondisi pasukan harus kembali ke lokasi markas , karena *Hero* mati. Kondisi ini akan mengabaikan kondisi attack, sehingga apabila unit pasukan sedang menyerang atau tidak, unit tersebut akan tetap pergi menuju markas.

10. *Defend*

    Merupakan kondisi setelah pasukan escape dan telah mencapai markas, pasukan akan diam dan menghadap ke arah markas musuh, bila terdapa musuh pada area serangan pasukan maka pasukan akan menyerang musuh tersebut, sedangkan apabila *hero* hidup kembali ketika kondisi ini maka, kondisi *follow* akan diterpakan kembali.

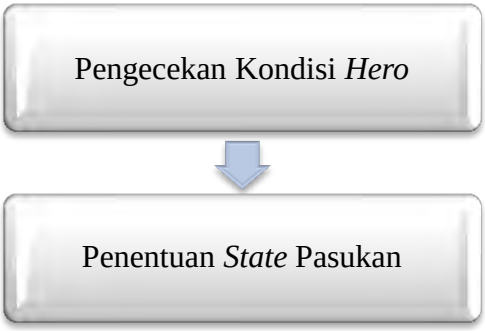
11. *Dead*

    Apabila health point unit pasukan mencapai nilai 0 maka masukan akan mati dan menghilang dari area permainan. Setelah kondisi ini pasukan akan *respawn* kembali setelah 3 detik.

3.2 Perancangan FSM Perilaku Pasukan

Pada perancangan FSM perilaku pasukan ini, akan terdapat 1 karakter dalam permainan berupa *agent* yang akan mempengaruhi perilaku pasukan. *Agent* tersebut akan bertindak sebagai *leader* dari pasukan, dimana karakter *agent* tersebut adalah karakter hero pada masing-masing ras di dalam permainan. Pasukan akan bergerak dan memiliki perilaku, dengan mengacu kepada kondisi hero dalam permainan. ketika kondisi *hero* hidup (*alive*), maka pasukan akan bergerak mengikuti *hero*, dan memiliki perilaku yang berubah berdasarkan perilaku *hero* apakah *hero* dalam kondisi *walk*, *attack*, atau *idle*. Ketika *hero* dalam kondisi mati (*dead*), maka pasukan akan bergerak dan memiliki perilaku tanpa dipengaruhi oleh pergerakan hero, sehingga hanya menggunakan rancangan FSM saja.

Perilaku perilaku apa saja yang akan dijalankan pada pasukan dan dalam kondisi apa akan dipengaruhi oleh perilaku dari *hero* dalam permainan, oleh karena itu dalam merancang FSM perilaku pasukan akan dilakukan berdasarkan diagram blok pada gambar 3.2 yaitu dilakukan perngecekan kondisi *hero* terlebih dahulu, kemudian ditentukan *state* apa yang akan diterapkan pada pasukan.



Gambar 3.2 Diagram Blok Perancangan FSM Perilaku Pasukan

3.2.1 Pengecekan Kondisi Hero

Kondisi *Hero* dalam permainan dibagi menjadi 2 kondisi yaitu ketika kondisi *Hero* hidup (*Alive*) dan mati (*Dead*). Pada masing-

masing kondisi *hero* tersebut pasukan akan memiliki perilaku yang berbeda.

Pembagian perilaku berdasarkan kondisi *hero* akan dibagi menjadi seperti berikut:

1. Kondisi *Hero* Hidup (*Alive*)

Kondisi ini merupakan kondisi ketika *Hero* berada dalam arena permainan. Perilaku *Hero* yang akan dijadikan acuan dalam menentukan perilaku pasukan ada 3 macam, yaitu :

- Kondisi saat *Hero* Diam (*Idle*)

  Ketika *hero* dalam keadaan diam/idle pasukan akan memiliki perilaku berikut :

  - *Idle*
  - *Evade*
  - *Attack*
  - *Back-to-Position*
  - *Pursuit*

  Perilaku pasukan tersebut akan dilakukan pada kondisi yang berbeda beda dalam permainan. Ketika *hero* diam ketika pasukan tidak berada dalam kerumunan maka kondisi pursuit akan dilakukan. *Evade* dilakukan ketika *hero* diam ketika berada di tengah-tengah kerumunan, dan *attack* dilakukan ketika ada musuh pada jarak serang pasukan.

- Kondisi saat *Hero* Berjalan (*Walk*)

  Ketika *hero* dalam keadaan *walk* pasukan akan memiliki perilaku berikut :

  - *Follow*
  - *Evade*
  - *Pursuit*

  Ketika *hero* berjalan pasukan akan memiliki perilaku yang sama yakni berjalan/walk. *Evade* dilakukan ketika *hero* memotong kerumunan.

- Kondisi saat *Hero Attack*
    - *Attack*
    - *Back-to-position*
    - *Pursuit*

    Ketika *hero* menyerang musuh pasukan akan memiliki perilaku *attack* pula. Ketika musuh yang diserang oleh pasukan telah mati pasukan akan melakukan perilaku *back-to-position*

2. Kondisi *Hero* Mati (*Dead*)

Ketika HP *hero* mencapai nilai 0 maka karakter tersebut akan menghilang dari arena permainan dan memasuki kondisi mati/dead. Perilaku yang mungkin dimiliki oleh pasukan pada kondisi ini adalah sebagai berikut :

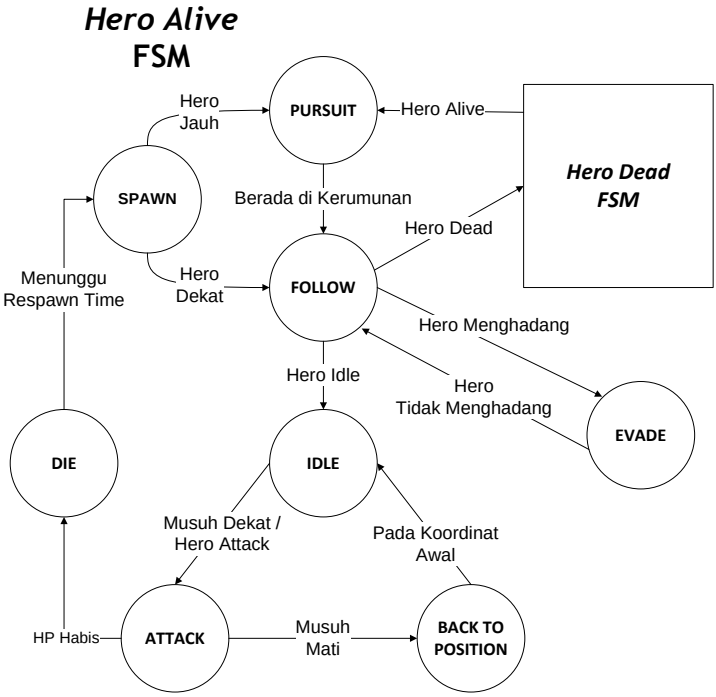
- *Escape*
- *Defend*
- *Attack*

Ketika *hero* mati, pasukan akan menuju ke sekitar markas untuk mempertahankan markas (*defense*) dari musuh yang datang. Ketika *hero* mati ketika pasukan berada disekitar musuh maka pasukan akan melarikan diri menuju markas (*escape)*.

3.2.2 Penentuan FSM Pasukan

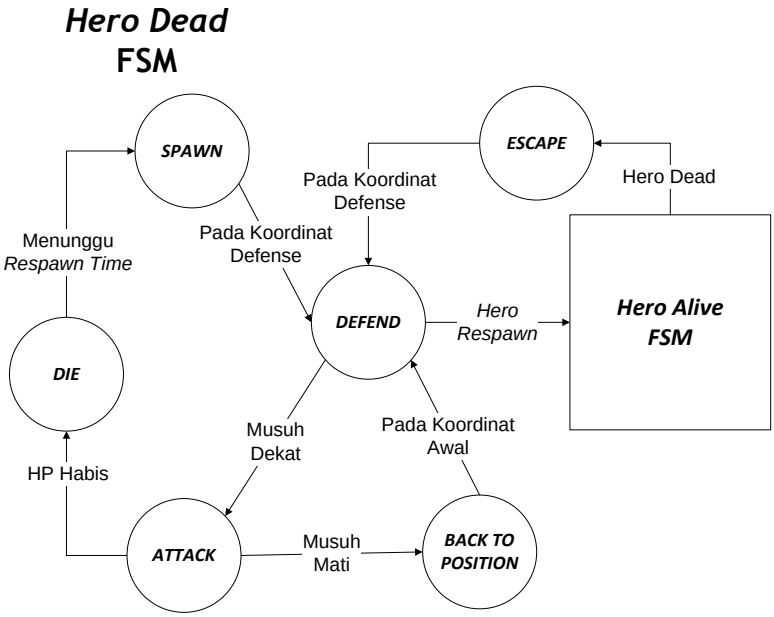
Setelah setiap kondisi pada pasukan ditentukan berdasarkan kondisi dari *hero*, diperoleh pada kondisi *hero alive*, terdapat 8 *state* perilaku yang akan diterapkan pada pasukan dan pada kondisi *hero dead*, terdapat 6 *state* perilaku yang akan diterapkan pada pasukan. Perilaku pada kondisi *hero alive* adalah *spawn, pursuit, follow, evade, idle, attack, back to position,* dan *die*, sedangkan perilaku pasukan pada kondisi *hero dead* adalah *spawn, defend, escape, attack, back to position,* dan *die.*

Dari kemungkinan *state* perilaku yang akan diterapkan pada pasukan, dirancang FSM berdasarkan kedua kondisi *hero*, yaitu kondisi *hero alive* dan *hero dead*. FSM Yang dibuat adalah pada gambar 3.2 untuk kondisi perilaku pasukan saat *hero alive* dan gambar 3.3 untuk kondisi perilaku pasukan saat *hero dead.*



Gambar 3.3 FSM perilaku pasukan pada kondisi *hero alive*

Pada kondisi *hero alive*, pasukan akan mengikut pergerakan hero ketika jarak *hero* dekat. Pada kondisi *follow*, pasukan akan memasuki kondisi *evade* ketika *hero* memotong kerumunan pasukan. Ketika menyerang unit musuh dan *hero* mati maka pasukan akan melarikan diri dari pertempuran dan menggunakan FSM perilaku ketika *hero* tidak ada dalam arena permainan seperti pada Gambar 3.4.



Gambar 3.4 FSM perilaku pasukan pada kondisi *hero dead*

Pada kondisi *hero dead*, pasukan harus mempertahankan markas menggunakan kondisi *defend*. Ketika terdapat musuh pada jarak serang pasukan, maka pasukan akan memasuki kondisi *attack*. Apabila *hero respawn* maka pasukan akan kembali menggunakan FSM perilaku pada kondisi *hero alive* seperti pada gambar 3.2

3.3 Penerapan FSM Perilaku Pasukan

Pasukan dalam permainan akan memiliki perilaku yang dapat bertidak secara autonomous tanpa perlu dikendalikan secara langsung oleh pemain.

3.3.1 Perancangan Desain Animasi Pasukan

Dalam pengaplikasian perilaku yang telah dibuat, akan dibuat terlebih dahulu desain animasi karakter-karakter pasukan yang akan

digunakan dalam permainan. Pada bagian ini akan dikerjakan oleh anggota tim Lume Wars yaitu Widi Sarinastiti. Terdapat 3 perilaku yang akan dirancang animasinya menggunakan gambar dimensi dua, perilaku tersebut adalah *idle*, *walk*, dan *attack*. Selain itu terdapat pula sprite untuk menampilkan animasi ledakan ketika pasukan mati.

Sprite yang digunakan adalah sebagai berikut.

1. Animasi Ras Arian

Berdasarkan desain konsep *art* permainan Lume Wars, dibuat sprite untuk diterapkan pada pasukan untuk ras Arian. *Sprite* yang digunakan pada pasukan ras Arian adalah seperti pada gambar 3.5.

Gambar 3.5 Sprite pasukan ras Arian

2. Animasi Ras Atlantis

Berdasarkan desain konsep *art* permainan Lume Wars, dibuat sprite untuk diterapkan pada pasukan untuk ras Atlantis. *Sprite* yang digunakan pada pasukan ras Atlantis adalah seperti pada gambar 3.6.

Gambar 3.6 *Sprite* pasukan untuk ras Atlantis

3. Animasi Ras Borneo

Berdasarkan desain konsep *art* permainan Lume Wars, dibuat sprite untuk diterapkan pada pasukan untuk ras Borneo. *Sprite* yang digunakan pada pasukan ras Borneo adalah seperti pada Gambar 3.7.

Gambar 3.7 *Sprite* pasukan untuk ras Borneo

4. Animasi Ras Maya

Berdasarkan desain konsep *art* permainan Lume Wars, dibuat sprite untuk diterapkan pada pasukan untuk ras Maya. Sprite yang digunakan pada pasukan ras Maya adalah seperti pada gambar 3.8.

Gambar 3.8 Sprite pasukan untuk ras Maya

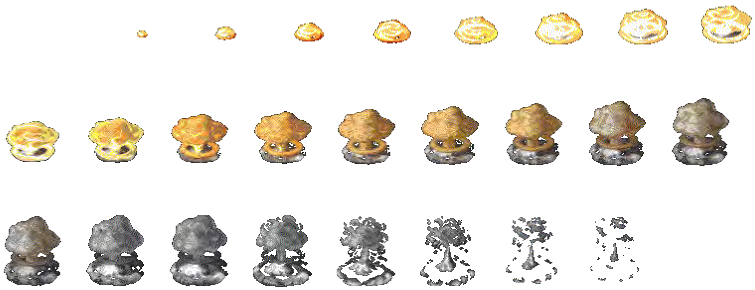
5. Animasi Ras Outer

Berdasarkan desain konsep *art* permainan Lume Wars, dibuat sprite untuk diterapkan pada pasukan untuk ras Outer. Sprite yang digunakan pada pasukan ras Outer adalah seperti pada Gambar 3.9.

Gambar 3.9 Sprite pasukan untuk ras Outer

6. Animasi Ledakan

Berdasarkan desain konsep *art* permainan Lume Wars, Selain sprite yang digunakan pada pergerakan pasukan, terdapat pula animasi ledakan ketika pasukan dalam kondisi *dead*. Sprite yang digunakan ketika kondisi pasukan *dead* adalah pada gambar 3.10.



Gambar 3.10 Sprite animasi ledakan pada pasukan yang mati

3.3.2 Penentuan Parameter dan Persamaan pada Perilaku Pasukan

Perilaku yang dimiliki pasukan akan dijalankan dengan menggunakan beberapa persamaan. Perilaku yang memili persaaman adalah perilaku berikut.Pada pergerakan pasukan digunakan parameter pada tabel 3.1

Tabel 3.1 Daftar Parameter pada pasukan

No	Parameter	Keterangan
1	*Health Point (HP)*	Pasukan memiliki health point (HP) untuk menentukan nilai nyawa yang dimiliki pasukan
2	*Damage*	Nilai damage pada setiap serangan
3	*Attack Range*	Jarak Serangan
4	*Attack Radius*	Jarak minimal untuk menyerang musuh
5	X	Koordinat terhadap sumbu X
6	Y	Koordinat terhadap sumbu Y
7	*Speed X*	Kecepatan terhadap sumbu X
8	*Speed Y*	Kecepatan terhadap sumbu X
9	*Max Speed*	Kecepatan maksimal
10	*Max Force*	Gaya maksimal yang mungkin diberikan
11	*Arrival Force*	Gaya bergerak menuju target
12	*Separation Force*	Gaya menghindari unit pasukan lain
13	*Force Evade*	Gaya untuk menghindari hero

Selain parameter, perilaku pasukan juga menerapkan beberapa persamaan yang digunakan pada kondisi kondisi tertentu. Perilaku yang menerapkan fungsi persamaan adalah perilaku berikut

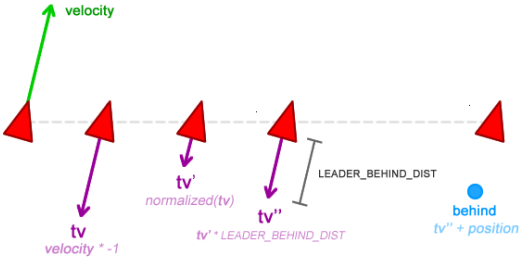
1. Follow

Pada perilaku *follow*, pasukan akan mengikuti pergerakan *hero* dalam permainan. Pada Gambar 3.11, terlihat 10 pasukan mengikuti *hero* dengan cara mengikuti titik merah yang merupakan *follow point.*

Gambar 3.11 Ilustrasi Pasukan mengikuti follow point

Untuk mendapatkan *follow point*, yang digunakan 2 parameter yaitu *distance behind leader* atau jarak *hero* dengan pasukan dan *hero movement speed* atau kecepatan bergerak *hero*. Langkah pencarian nilai *follow point* adalah seperti pada Gambar 3.12, dimana gambar segitiga merah menggambarkan *hero*.

Setelah diketahui kecepatan dan arah, nilai tersebut dikali negatif 1 untuk mendapat kecepatan yang sama dengan arah yang berlawanan. Nilai yang diperoleh kemudian di-normalisasi dan dikali dengan *distance behind hero*, untuk mendapatkan koordinat yang diinginkan.



Gambar 3.12 Langkah pencarian posisi *follow point*

Kemudian pasukan harus bergerak secara berkelompok dalam permainan, akan diterapkan aturan *flocking* yang menerapakan 3 gaya yaitu *separation, cohesion,* dan *alignment. Separation* adalah gaya yang digunakan ketika berada di jarak kurang dari jarak minimal antar pasukan. Jarak minimal antar pasukan disebut *separation distance* dimana nilai yang digunakan dalam permainan ini adalah 80 piksel. *Cohesion* adalah gaya untuk tetap berada di dalam kerumunan, dan *alignment* adalah gaya untuk mengikuti arah dari kerumunan. Pada implementasi pasukan ini, gaya *cohesion* dan *alignment* dijadikan 1 untuk menjadi lebih sederhana menjadi gaya *arrival.*

a. Separation

$$fs = \frac{1}{n}\sum_{m=1}^{n}\left(neighbor_{position_m} - unit_{position}\right) \quad (3.1)$$

$$fs = normalize\ (fs) \times max\_fs \quad (3.2)$$

Fungsi persamaan 3.1 dan persamaan 3.2 digunakan untuk mencari nilai *fs* yang merupakan *separation force* atau gaya pasukan saling berpisah satu sama lain. Ketika terdapat unit pasukan yang berada dalam jarak minimal antar pasukan (*separation radius*), unit pasukan lain yang berada pada *separation radius* disebut *neighbor*. *n* merupakan jumlah *neighbor*, *m* adalah index urutan *neighbor*, dan *max_fs* adalah gaya maksimal yang mungkin diterapkan pada *separation force*. Persamaan 3.1 digunakan untuk mendapatkan total *fs* berdasarkan jarak pasukan dan seluruh neighbor yang ada. Persamaan 3.2 mencari nilai normalisasi dari *fs* untuk kemudian dikali dengan nilai maksimal dari *fs*.

b. Arrival (Cohesion + Alignment)

$$fa = normalize\ (hero\_position - unit\_position) \times max\_fa \quad (3.3)$$

Fungsi persamaan 3.3 digunakan untuk mengetahui nilai *fa* atau *arrival force*. Gaya *arrival force* ini digunakan untuk mengikuti target sambil tetap berdekatan dengan pasukan lainya. dimana *max_fa* merupakan gaya maksimal yang mungkin diterapkan pada gaya *arrival*.

Target yang dituju merupakan *follow point* atau *hero* masing masing ras pasukan.

2 Evade

Pada perilaku *evade*, pasukan akan menghindari pergerakan *hero* agar tidak terjadi tabrakan. Perilaku *evade* ini diilustrasikan seperti pada Gambar 3.13.

Gambar 3.13 Illustrasi Pasukan menghindari pergerakan *hero*

Pada Gambar 3.13, 2 titik biru merupakan *evade point* dan lingkaran coklat merupakan *evade region*, dimana ketika terdapat pasukan berada didalam *evade region*, pasukan tersebut akan diberi gaya *evade* untuk menghindari pergerakan *hero*.

Pada kotak putih terdapat 2 pasukan yang mendapat gaya *evade* untuk bergerak menuju arah panah coklat karena berada di dalam *evade region*, sedangkan pada kotak ungu terdapat 8 pasukan yang tidak terkena gaya *evade* dan tetap pada kondisi *follow*.

$$fe = normalize\left(unit\_position - hero\_position\right) \times max\_speed \quad (3.3)$$

$$fe = \left(fe - velocity\right) \times max\_fe \quad (3.4)$$

Untuk melakukan gaya *evade*, digunakan persamaan 3.3 dan 3.4. *fe* merupakan *evading force* yang dicari, *max_speed* adalah kecepatan maksimal pada pasukan, *velocity* adalah kecepatan pasukan saat ini, dan *max_fe* adalah gaya *evade* maksimal yang mungkin diberikan.

Langkah pertama adalah mengurangi koordinat pasukan dan koordinat *hero*, untuk mendapat gaya sebesar jarak antara kedua koordinat tersebut dan memiliki arah yaitu koordinat pasukan ke arah yang berlawanan dengan koordinat *hero*. Kemudian nilai tersebut di-normalisasi dan dikali dengan *max_speed* untuk mendapat gaya *evade* sebesar *max_speed* dengan arah yang didapat. Gaya tersebut dikurangi dengan *velocity* atau kecepatan saat ini dan dikali dengan *max_fe* untuk mendapatkan gaya *evade* yang diinginkan.

3.4 Penyatuan Bagian Permainan

Penyatuan bagian pengerjaan permainan Lume Wars ini dilakukan di dalam IDE Game Maker : Studio. Dalam IDE Game Maker : Studio, objek pasukan yang telah dibuat memiliki *events* dan *actions* yang akan menjalankan seluruh program yang telah dibuat. *Events* yang dimiliki objek pasukan adalah *create, alarm0, step, end step* dan *draw*. Pada *action step*, terdapat *action* berupa *script* atau program yang ditulis dalam bahasa permrograman GML dimana terdapat perilaku pasukan yang akan dijalankan di dalam permainan Lume Wars.

Pasukan yang dianimasikan pada permainan Lume Wars berjumlah 10 unit. Pasukan dalam permainan akan bergerak mengikuti hero dalam kondisi *follow*. Animasi pasukan dalam permainan ditunjukkan pada gambar 3.14

Gambar 3.14 Animasi pasukan dalam Lume Wars

BAB 4
PENGUJIAN DAN ANALISA

Dalam bab ini akan dibahas mengenai pengujian dari sistem yang telah direalisasikan dengan tujuan untuk mengetahui apakah fungsi dari sistem yang direncanakan telah berfungsi sesuai harapan.

Pengujian pada penelitian ini dilakukan dengan beberapa metode antara lain melakukan pengujian seberapa jauh device dapat menjalankan sistem yang dibuat untuk dapat menentukan minimal spesifikasi dalam memainkan permainan Lume Wars, melakukan survei untuk mengetahui hasil secara kuantitatif seberapa besar kesesuaian perilaku yang dibuat terhadap permainan Lume Wars.

4.1 Pengujian *Device*

Pengujian ini dilakukan pada *device* Android. *Device* akan dilakukan menggunakan beberapa macam *device* yang berbeda, *device* yang digunakan akan dibagi menjadi 2 jenis *device* utama, yaitu *Smartphone* dan *Tablet*. Pada masing-masing jenis *device*, akan digunakan 3 *device*, sehingga total digunakan 6 *device* Android.

Terdapat 3 perilaku yang akan dijadikan *state* pengujian, yaitu perilaku *Follow*, *Idle*, dan *Attack*. Pasukan yang akan diujikan berjumlah 10 sampai dengan 100 unit setiap kelipatan 10 unit. Nilai yang akan diperoleh adalah nilai *frame per second* (FPS), CPU *load*, dan *memory usage* saat menjalankan state-state pengujian permainan. Aplikasi yang akan digunakan adalah file Lume Wars Beta.APK yang memiliki ukuran sebelum instalasi sebesar 32,93 MB dan ukuran setelah instalasi sebesar 37,47 MB.

Langkah pengambilan data pada device ini yaitu, setelah permainan dimulai, dicatat nilai FPS pada setiap *state* sebanyak 3 kali, untuk kemudian diambil nilai rata-rata FPS pada *state* tersebut. Kemudian diambil nilai CPU *load* dan *memory usage*, dengan pengambilan data dilakukan sebanyak 3 kali, sehingga menghasilkan data saat pengambilan pertama, kedua, dan ketiga. Satuan yang digunakan pada data CPU *load* adalah persentase (%) penggunaan CPU, dan pada data memory usage adalah satuan *Kilobyte(KB).*

Spesifikasi device smartphone yang digunakan adalah seperti pada tabel 4.1 dan spesifikasi device tablet yang digunakan adalah seperti pada tabel 4.2.

Tabel 4.1 Spesifikasi *Smartphone* untuk pengujian

	Device 1	***Device 2***	***Device 3***
Product Name	Lenovo S880	Sony Xperia M	Samsung Galaxy S III I9300
Ope*rating* System	Android 4.0.1 Ice Cream Sandwich	Android 4.1 Jelly Bean	Android 4.1.2 Jelly Bean
Processor	Cortex-A9 1 GHz Single Core	Krait 1 GHz Dual Core	Cortex-A9 1.4 GHz Quad Core
GPU	PowerVR SGX531	Adreno 305	Mali-400MP
Memory	512 MB	1 GB	1 GB
Display Size	5.0 inches	4.0 inches	4.8 inches
Disp. Resolution	480 x 800 pixels	480 x 854 pixels	720 x 1280 pixels
Storage	Int : 4 GB	Int : 4 GB	Int : 16 GB
	Ext : *NA*	Ext : *NA*	Ext : *NA*

Tabel 4.2 Spesifikasi *Tablet* untuk pengujian

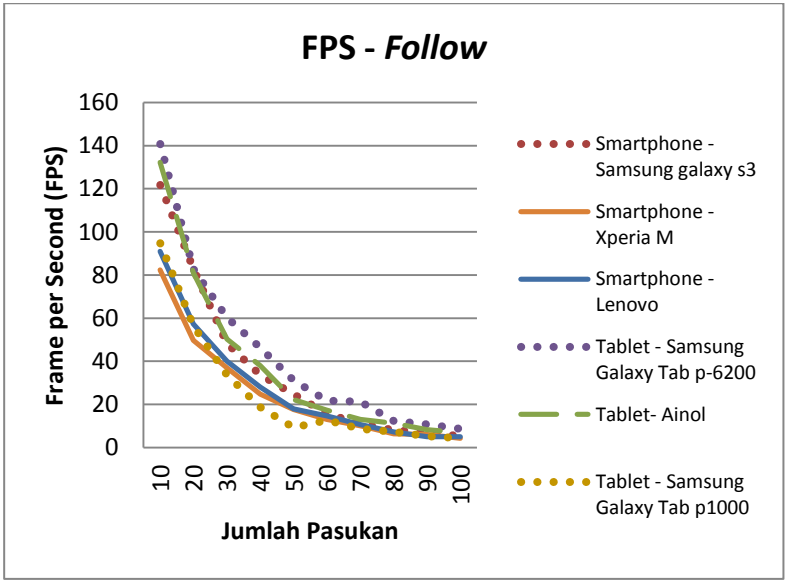
	Device 4	***Device 5***	***Device 6***
Product Name	Galaxy Tab GT-P1000	Galaxy Tab 7.0 Plus GT-P6200	Novo 7 Numy AX1
Ope*rating* System	Android 2.3.6 Gingerbread, Rooted	Android 4.2 Ice Cream Sandwich	Android 4.2 Jelly Bean, Rooted
Processor	Cortex-A8 1GHz Single Core	1.2 GHz Dual Core	MTK8389 Quad Core 1.2 GHz
GPU	PowerVR SGX 540	Mali-400MP	PowerVR SGX 544MP
Memory	512 MB	1 GB	DDR3 1 GB
Display Size	7.0 inches	7.0 inches	7.0 inches
Disp. Resolution	600 x 1024 pixels	600 x 1024 pixels	600 x 1024 pixels
Storage	Int : 16 GB	Int : 16 GB	Int : 8 GB
	Ext : *NA*	Ext : *NA*	Ext : 2 GB Micro SD

Dari pengujian yang dilakukan pada masing masing *smartphone*, diperoleh hasil sebagai berikut

4.1.1 Hasil Frame per Second

1. *State Follow*

Pengujian ini dilakukan ketika pasukan dalam kondisi menjalankan perilaku *attack*,untuk memperoleh nilai FPS pada *device*. Hasil yang diperoleh berupa grafik seperti pada gambar 4.1.



Gambar 4.1 Hasil FPS pada *state follow*

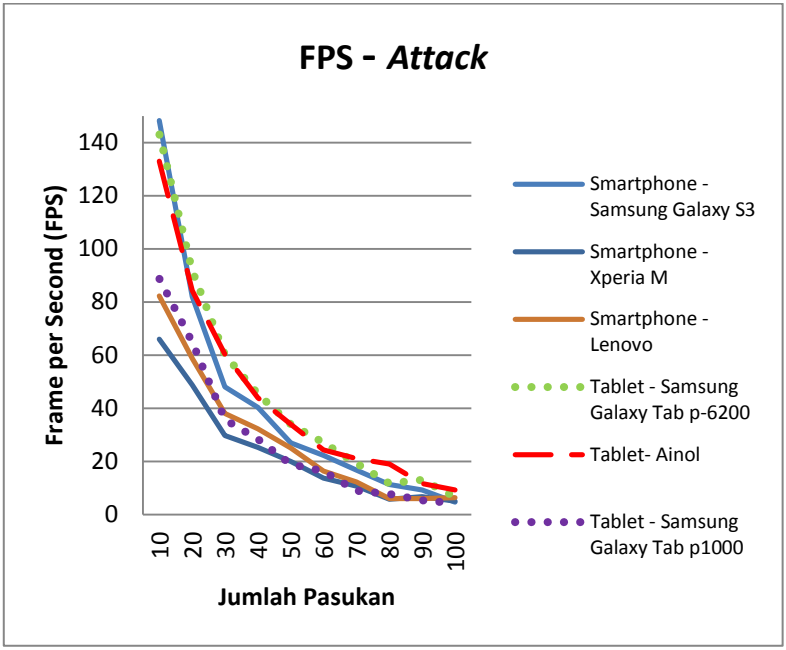
Hasil yang didapat pada bagian *follow*, FPS tertinggi adalah pada *device* 5 saat jumlah pasukan 10 dengan FPS rata-rata sebesar 140,7 dan FPS terendah pada *device* 4 saat jumlah pasukan 100 dengan FPS rata-rata sebesar 4,3. Animasi *state follow* ini dianggap cukup baik untuk dilakukan dengan jumlah pasukan maksimal sebanyak 40 unit dengan FPS berkisar antara 20 sampai 50.

Setelah diambil nilai rata-rata tiap *device* dari seluruh percobaan pasukan 10 sampai pasukan 100, diperoleh nilai rata-rata paling baik adalah *device* 5 dengan nilai FPS rata-rata sebesar 43,6.

Nilai rata-rata FPS paling rendah adalah dari *device* 2 dan *device* 4, dengan nilai sama sebesar 25,1.

2. State Attack

Pengujian ini dilakukan ketika pasukan dalam kondisi menjalankan perilaku *attack*,untuk memperoleh nilai FPS pada *device.* Hasil yang diperoleh berupa grafik seperti pada gambar 4.2



Gambar 4.2 Hasil FPS pada *State Attack*

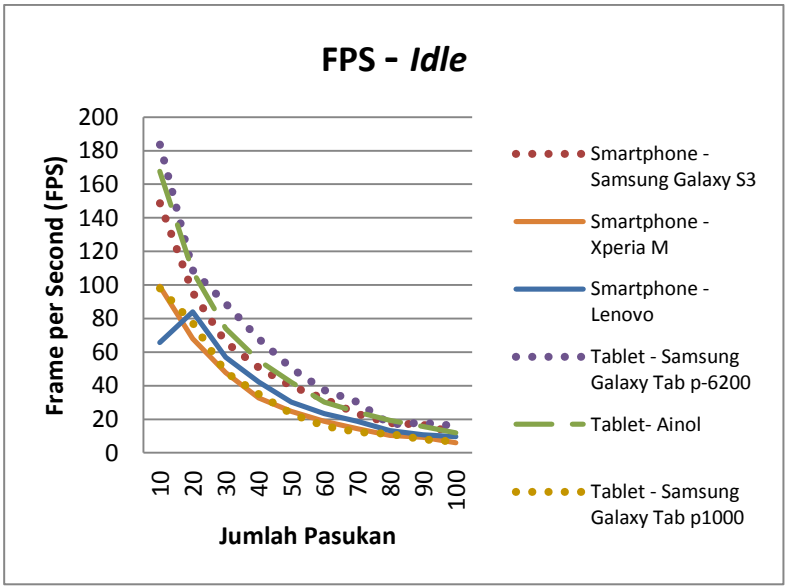
Hasil yang didapat pada *state attack*, FPS tertinggi adalah pada *device* 3 pada jumlah pasukan 10 dengan FPS rata-rata sebanyak 148.3, dan fps terendah adalah pada *device* 4 dengan FPS rata-rata sebesar 4.3. *State attack* ini dianggap cukup baik untuk dijalankan sampai jumlah maksimal pasukan sebanyak 40 unit, dengan nilai FPS sebesar 25 sampai 45.

Setelah diambil nilai rata-rata tiap *device* dari seluruh percobaan dengan pasukan berjumlah 10 unit sampai dengan pasukan

berjumlah 100 unit, nilai rata-rata FPS paling baik yaitu *device* dengan nilai tertinggi, yaitu milik *device* 5 dengan nilai FPS 45.2, sedangkan nilai rata-rata FPS dengan nilai buruk yaitu device dengan nilai terendah, yaitu milik *device* 2 dengan FPS 23.2.

3. State Idle

Pengujian ini dilakukan ketika pasukan dalam kondisi menjalankan perilaku *idle*, untuk memperoleh nilai FPS pada *device*. Hasil yang diperoleh berupa grafik seperti pada gambar 4.3.



Gambar 4.3 Hasil FPS pada *State Idle*

Hasil yang didapat pada *state idle*, FPS tertinggi adalah pada device 5 dengan FPS rata-rata sebesar 183.7 saat jumlah pasukan sebesar 10 unit, dan FPS terendah adalah pada device 2 pada jumlah pasukan 100 unit dengan fps rata-rata sebesar 6. *State idle* ini dianggap dapat berjalan dengan cukup baik sampai pada jumlah pasukan sebanyak 60 unit.

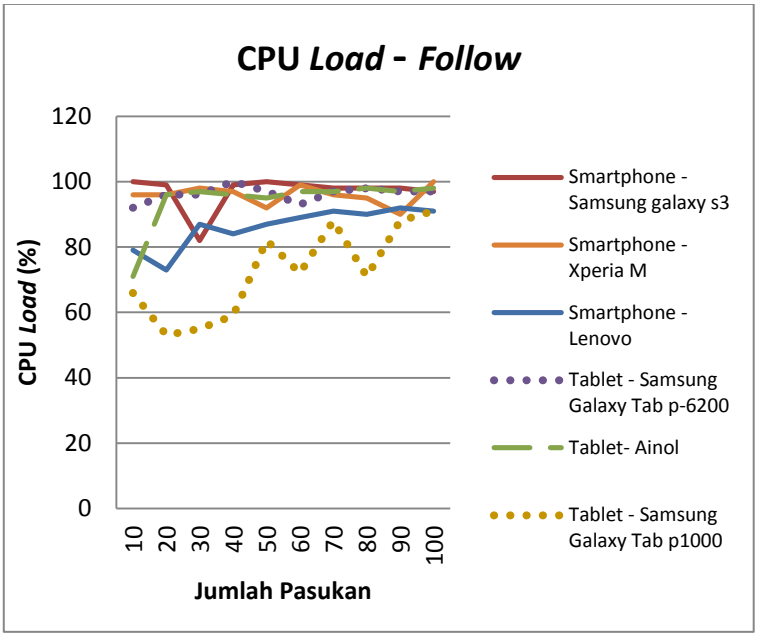
Setelah diambil nilai rata-rata tiap *device* dari seluruh percobaan dengan pasukan berjumlah 10 unit sampai dengan pasukan

berjumlah 100 unit, nilai FPS paling baik pada *state idle* adalah nilai dari device 5 dengan nilai FPS 61.8. nilai FPS paling rendah milik device 2 dengan FPS 33.1.

4.1.2 Hasil CPU *Load*

1. State Follow

Pengujian ini dilakukan ketika pasukan dalam kondisi menjalankan perilaku *follow*, untuk memperoleh nilai CPU *load* pada *device.* Hasil yang diperoleh berupa grafik seperti pada gambar 4.4.

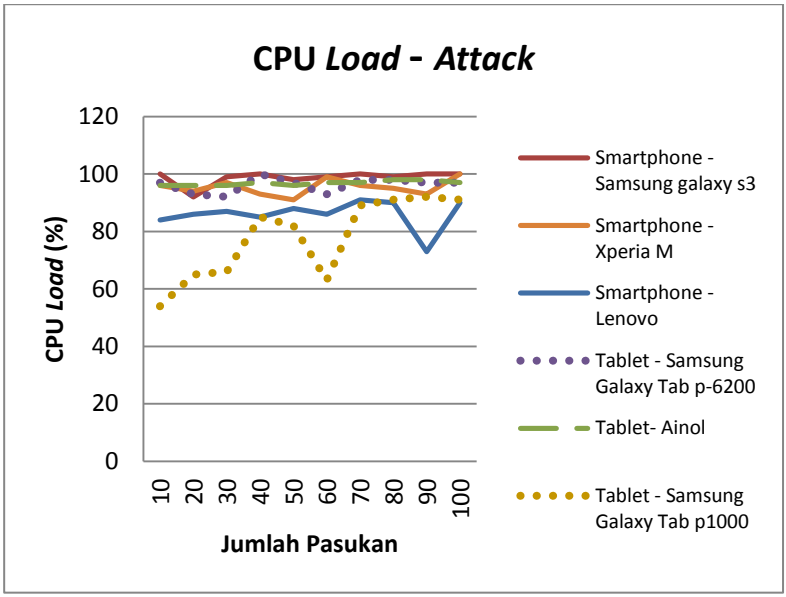


Gambar 4.4 Hasil CPU *load* pada *state follow*

Hasil yang didapat pada *state follow*, CPU *load* tertinggi diperoleh pada *device* 3 saat jumlah pasukan sebanyak 10 unit dan 50 unit, pada *device* 5 saat jumlah pasukan 40 unit, pada *device* 2 saat jumlah pasukan 10 unit, dengan CPU *Load* sebesar 100%. CPU *Load* terendah diperoleh pada device 4 saat jumlah pasukan sebanyak 20 unit dengan *CPU load* sebesar 53%.

2. *State Attack*

Pengujian ini dilakukan ketika pasukan dalam kondisi menjalankan perilaku *attack*, untuk memperoleh nilai CPU *load* pada *device*. Hasil yang diperoleh berupa grafik, yaitu pada gambar 4.5.

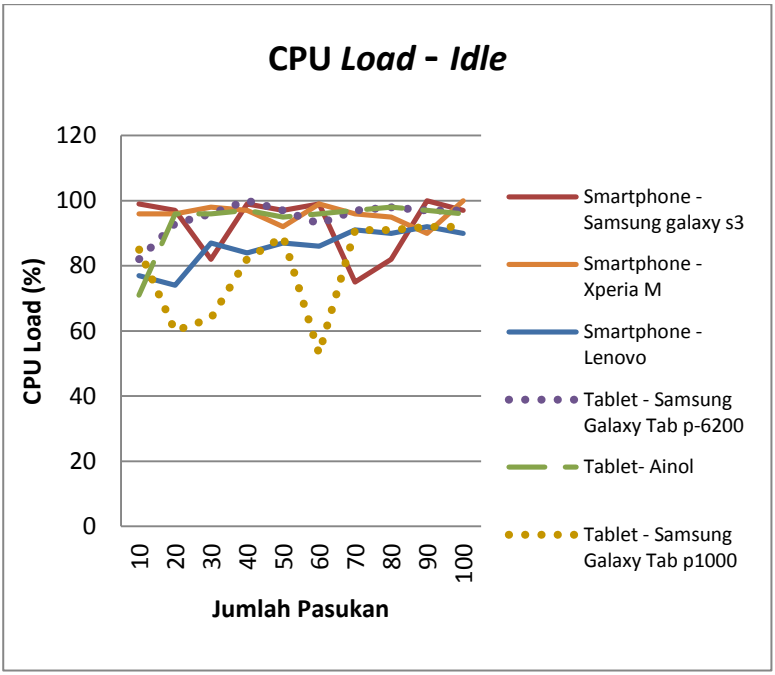


Gambar 4.5 Hasil CPU *load* pada *state attack*

Hasil yang didapat pada *state attack*, CPU *load* tertinggi adalah pada device 5 saat jumlah pasukan sebanyak 30 unit dan 60 unit, pada *device* 3 saat jumlah pasukan sebanyak 60 unit dengan nilai CPU *load* sebesar 99%. CPU *load* terendah yang diperoleh adalah pada *device* 4 saat jumlah pasukan sebanyak 30 unit dengan nilai CPU *load* sebesar 68%.

3. *State Idle*

Pengujian ini dilakukan ketika pasukan dalam kondisi menjalankan perilaku *idle*, untuk memperoleh nilai CPU *load* pada *device*. Hasil yang diperoleh berupa grafik, yaitu pada gambar 4.6.



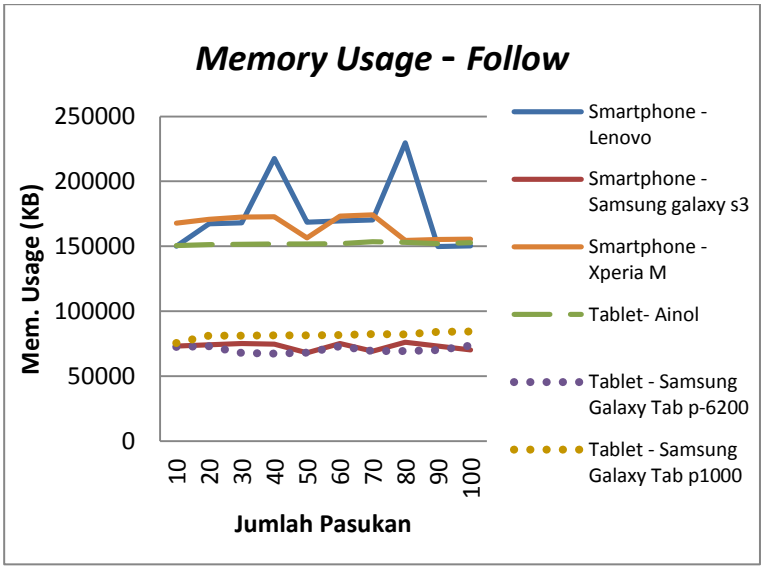
Gambar 4.6 Hasil CPU *load* pada *state idle*

Hasil yang didapat pada *state idle* yaitu, CPU *load* tertinggi pada device 3 saat jumlah pasukan sebanyak 10, 20, dan 30 unit dengan nilai CPU *load* sebesar 100%. CPU *load* terendah adalah pada *device* 4 saat jumlah pasukan berjumlah 30 unit dengan nilai CPU *load* sebesar 59%.

4.1.3 Hasil Memory Usage

1. State Follow

Pengujian ini dilakukan ketika pasukan dalam kondisi menjalankan perilaku *follow*, untuk memperoleh nilai *memory usage* pada *device*. Hasil yang diperoleh berupa grafik, yaitu pada gambar 4.7.



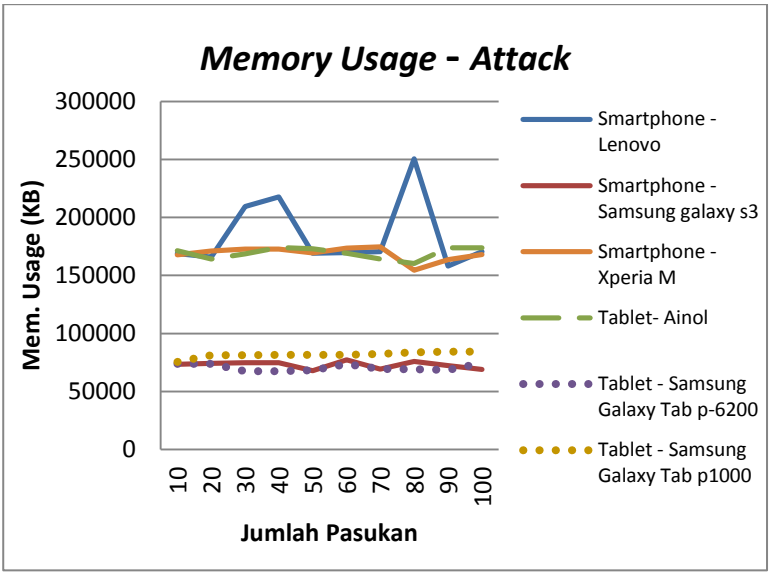
Gambar 4.7 Hasil *memory usage* pada *state follow*

Hasil nilai *memory usage* yang didapat pada *state follow* yaitu, *memory usage* tertinggi pada *device* 1 saat jumlah pasukan sebanyak 80 unit dengan nilai *memory usage* 229785KB. Sedangkan nilai *memory usage* terendah adalah pada device 5 saat pasukan berjumlah 40 unit dengan nilai *memory usage* sebesar 67268KB.

Nilai *memory usage* paling stabil pada *device* 6 dengan nilai minimal 150656KB dan nilai maksimal 153471KB, dengan nilai jarak sebesar 2815KB. Sedangkan nilai *memory usage* paling tidak stabil adalah pada *device* 1 dengan nilai minimal 149713KB dan nilai maksimal 229785KB, dengan nilai jarak minimal dan maksimal sebesar 80072KB.

2. State Attack

Pengujian ini dilakukan ketika pasukan dalam kondisi menjalankan perilaku *attack*, untuk memperoleh nilai *memory usage* pada *device.* Hasil yang diperoleh berupa grafik, yaitu pada gambar 4.8.



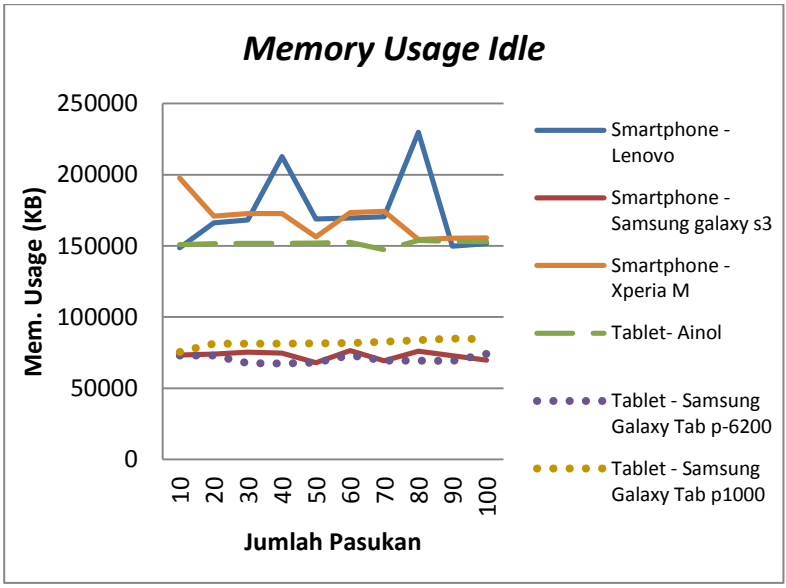
Gambar 4.8 Hasil *memory usage* pada *state attack*

Hasil *memory usage* yang diperoleh pada *state attack* yaitu, *memory usage* tertinggi pada *device* 1 saat pasukan sebanyak 80 unit dengan nilai *memory usage* sebesar 250529KB. Nilai *memory usage* terendah adalah pada device 5 saat jumlah pasukan berjumlah 40 unit dengan nilai *memory usage* sebesar 67392KB.

Nilai *memory usage* yang dinilai paling stabil adalah pada *device* 5 dengan nilai perbedaan nilai *memory usage* minimal sebesar 67392KB dan nilai maksimal 74487KB, dengan nilai jarak minimal dan maksimal sebesar 7095KB. Sedangkan nilai *memory usage* paling tidak stabil adalah pada *device* 1 dengan nilai minimal 157988KB dan nilai maksimal 250529KB, dengan nilai jarak minimal dan maksimal sebesar 92541KB.

3. State Idle

Pengujian ini dilakukan ketika pasukan dalam kondisi menjalankan perilaku *idle*, untuk memperoleh nilai *memory usage* pada *device*. Hasil yang diperoleh berupa grafik, yaitu pada gambar 4.9.



Gambar 4.9 Hasil *Memory Usage* pada *state idle*

Hasil *memory usage* yang diperoleh pada *state attack* yaitu, *memory usage* tertinggi pada device 1 saat pasukan sebanyak 90 unit dengan nilai sebesar 229837KB. Nilai *memory usage* terendah adalah pada *device* 5 saat pasukan berjumlah 40 unit dengan nilai *memory usage* sebesar 67280KB.

Nilai *memory usage* paling stabil adalah pada *device* 6 dengan nilai *memory usage* minimal sebesar 147253KB dan nilai maksimal sebesar 153718KB, dengan nilai jarak sebesar 6465KB.

4.2 Pengujian Kesesuaian FSM Perilaku

Pengujian ini dilakukan guna mengetahui apakah perilaku yang telah diimplementasikan pada pasukan telah sesuai dengan rancangan FSM perilaku pasukan. Pengujian ini dilakukan dengan mencoba seluruh perilaku dari FSM dalam permaian, kemudian akan didapatkan kemungkinan-kemungkinan perilaku yang mungkin terjadi dalam permainan dan apakah perilaku pada FSM benar terlaksana atau tidak. Dari pengujian,terdapat beberapa perilaku yang tidak berjalan sesuai dengan skenario pada awal perancangan FSM perilaku. Hal tersebut

terjadi karena adanya pengaruh dari perilaku dan kondisi lain pada pasukan.

Contoh penyebab yang diketahui mengakibatkan tidak terlaksananya suatu perilaku adalah adanya gaya atau pengaruh dari perilaku lain. Hasil yang didapatkan dari pengujian FSM Perilaku ini terdapat pada Tabel 4.3

Tabel 4.3 Hasil Pengujian FSM Perilaku

No	*State*	Kemungkinan	Perilaku Terlaksana
1	*Spawn*	-	Ya
2	*Pursuit*	1. *Hero* dalam kondisi *Attack/Idle*	Ya
		2. *Hero* dalam kondisi *Walk*	Ya
3	*Follow*	1. Pasukan tidak dalam kondisi menyerang	Ya
		2. Pasukan dalam kondisi *Attack* dan jarak dengan *hero* kurang dari 200 *pixel*	Tidak
4	*Idle*	1. Tidak semua pasukan berada pada *Idle Radius*	Tidak
		2. Seluruh pasukan ada pada *Idle Radius*	Ya
		3. *Hero* berada ditengah kerumunan pasukan	Tidak
5	*Attack*	1. Musuh berada disebelah unit pasukan	Ya
		2. Terdapat *Hero* diantara Pasukan dan Musuh	Tidak
6	*Back to Position*	1. Terdapat musuh disekitar pasukan	Tidak
		2. Tidak terdapat musuh lain disekitar pasukan	Ya
7	*Escape*	-	Ya
8	*Evade*	-	Ya
9	*Defend*	1. Tidak semua Pasukan berada pada *Defense Region*	Tidak
		2. Seluruh pasukan berada pada *Defense Region*	Ya
10	*Die*	-	Ya

Dari hasil pengujian kesesuaian perilaku pada tabel 4.3, terdapat 17 kemungkinan perilaku yang terjadi, dari 10 perilaku yang telah dirancang pada FSM. Diperoleh 11 kemungkinan perilaku yang terlaksana, dan 6 kemungkinan perilaku yang tidak terlaksana. Perilaku yang tidak terlaksana dapat disebabkan karena pengaruh dari gaya dari perilaku lain, contoh gaya *evade*, gaya *follow*, dan lain sebagainya.

4.3 Pengujian Survei Pengguna

Pengujian ini dilakukan untuk mengetahui apakah perilaku yang dibuat sudah cocok dengan permainan Lume Wars. Pengujian dilakukan dengan melakukan survei. Survei dilakukan terhadap responden yang akan mencoba permainan lume wars, untuk mengetahui pendapat mereka terhadap kesesuaian perilaku yang telah dibuat dengan permainan Lume Wars secara keseluruhan.

Tingkat kesesuaian perilaku yang dibuat akan ditentukan oleh seberapa besar beta tester memberikan pendapat bahwa perilaku yang dibuat dinilai sangat baik.

4.1.3 Teknis Pelaksanaan Survei

Pelaksanaan survei dilakukan dengan memberikan file apk permainan Lume Wars untuk kemudian dimainkan. Setelah beta tester melakukan percobaan permainan, beta tester akan diminta untuk mengisi survei berupa kuisioner yang terdiri dari beberapa pertanyaan sederhana. Jawaban dapat diisi dengan pilihan Sangat Baik, Cukup Baik, Tidak Baik, Sangat Tidak Baik. Perilaku dianggap sudah sesuai apabila jawaban berupa Sangat Baik, dan dinilai tidak cukup sesuai bila jawaban berupa Sangat Tidak Baik. Daftar pertanyaan yang dibuat seperti pada Tabel 4.4.

Tabel 4.4 Daftar Pertanyaan pada kuisioner

1	Apakah menurut anda perilaku *FOLLOW* pasukan sudah baik
2	Apakah menurut anda perilaku *EVADE* pasukan dalam permainan ini sudah baik
3	Apakah menurut anda perilaku *PURSUIT* pasukan dalam permainan ini sudah baik
4	Apakah menurut anda perilaku *ATTACK* pasukan dalam permainan ini sudah baik
5	Apakah menurut anda, *DISTANCE LEADER* dan pasukan dalam permainan ini sudah tepat
6	Apakah menurut anda, *DISTANCE* antar pasukan sudah tepat

Dalam pertanyaan yang diajukan, terdapat 4 perilaku yang dijadikan pertanyaan, yaitu perilaku *follow*, *evade*, *pursuit*, dan *attack*. Selain itu terdapat pada nomor 5 dan 6 diajukan pertanyaan mengenai

nilai parameter dari pasukan yaitu nilai jarak antara pasukan dan hero pada *distance leader,* dan jarak antara sesama pasukan pada *distance.*

4.1.4 Hasil Survei

Survei yang dilakukan mendapatkan 32 orang beta tester yang telah mencoba memainkan permaianan Lume Wars dan mengisi kuisioner. Hasil survei secara rinci terdapat pada tabel 4.5.

Perilaku yang dibuat dinilai telah sesuai dengan permainan apabila terdapat 50% beta tester yang cukup baik atau sangat baik

Tabel 4.5 Hasil Jawaban Kuisioner

	Sangat tidak baik	tidak baik	cukup baik	sangat baik
Pertanyaan 1	0%	25%	65.625%	9.375%
Pertanyaan 2	3.125%	9.375%	71.875%	15.625%
Pertanyaan 3	3.125%	12.5%	56.25%	28.125%
Pertanyaan 4	6.25%	21.875%	59.375%	12.5%
Pertanyaan 5	6.25%	21.875%	43.75%	28.125%
Pertanyaan 6	3.125%	21.875%	50%	25%

Pada Tabel 4.5, ditunjukkan hasil yang diperoleh dari 6 pertanyaan yang diajukan dalam survei. Pada Pertanyaan 1, diperoleh hasil 9.375% sangat baik, 65.625% cukup baik, dan 25% kurang baik. Pada Pertanyaan 2, diperoleh hasil 15.625% sangat baik, 71.875% cukup baik, 9.375% tidak baik, 3.125% sangat tidak baik. Pada Pertanyaan 3, diperoleh hasil 28.125% sangat baik, 56.25% cukup baik, 12.5% tidak baik, dan 3.125% sangat tidak baik. Pada Pertanyaan 4, diperoleh hasil 12.5% sangat baik, 59.375% cukup baik, 21.875% tidak baik, 6.25% sangat tidak baik.

Selain itu survei pada parameter yang digunakan yaitu parameter distance leader yang berarti jarak antara Hero dengan pasukan pada Pertanyaaan 5, yang diperoleh hasil 28.125% sangat baik, 43.75% cukup baik, 21.875% tidak baik dan 6.25% sangat tidak baik. Pada hasil mengenai parameter unit distance yang berarti jarak antara unit pasukan satu dengan lainnya di dalam kerumunan pada Pertanyaan 6, diperoleh hasil 25% sangat baik, 50% cukup baik, 21.875% tidak baik, dan 3.125% sangat tidak baik.

BAB 5
PENUTUP

5.1 Kesimpulan

Dari hasil perancangan, implementasi dan pengujian seluruh sistem yang dikerjakan dalam tugas akhir ini, dapat ditarik beberapa kesimpulan:

1. Nilai FPS berbanding terbalik dengan jumlah pasukan yang diujikan. Sedangkan nilai CPU *load* dan *memory usage* tidak dipengaruhi oleh banyak sedikitnya jumlah pasukan tetapi berdasarkan arsitektur dan spesifikasi pada tiap-tiap *device*.
2. Nilai FPS pada tiap *device* rata-rata diatas 25 FPS pada pengujian dengan jumlah pasukan sebanyak 40 unit pasukan. Pada pengujian menggunakan lebih dari 40 unit pasukan dinilai *device* berjalan dengan kondisi berat dan/atau *slow*.
3. Dari hasil kuisioner, jawaban Sangat Baik tidak mencapai nilai 30%, sehingga dinilai bahwa perilaku yang dibuat masih perlu dikembangkan lebih lanjut.
4. Dari hasil pengujian FSM, terdapat 17 kondisi yang mungkin terjadi dari total 10 perilaku yang dibuat. Sebanyak 35.29% kondisi perilaku tidak terlaksana disebabkan adanya perngaruh dari gaya pada perilaku lain. Berdasarkan jumlah perilaku yang tidak terlaksana, perilaku yang telah diimplementasikan dirasa cukup sesuai dengan perilaku awal yang dibuat.

5.2 Saran

Untuk pengembangan lebih lanjut mengenai tugas akhir ini, maka penulis menyarankan :

1. Perilaku attack pada pasukan perlu ditambahkan metode lebih lanjut, sebagai contoh *opponent selection.*
2. Animasi yang berjalan pada permainan ini dirasa kurang baik karena sedikitnya sprite yang digunakan, minimal sprite yang digunakan sebaiknya sebanyak 8 gambar setiap *state*.

DAFTAR PUSTAKA

[1] Danielsiek. Holger, “Intelligent Moving of Groups in Real-Time Strategy Games”, IEEE Symposium on Computational Intelligence and Games pp 71-78, 2008

[2] Z. Shen and S. Zhou, “Behavior representation and simulation formilitary operations on urbanized terrain,” Simulation, vol. 82, no. 9,pp. 593–607, 2006.

[3] “About YoYo Games” <URL: https://www.yoyogames.com/company/about> Diakses 7 Januari 2014

[4] Davison, Andrew “Killer Game Programming In Java”,O’REILLY,Ch 22, 2005

[5] Gu. Dongbing, “Leader Following Flocking :Algorithms and Experiments” IEEE Transactions on Control Systems Technology, vol. 17, no. 5, september 2009

[6] Reynolds. Craig W, ”Flocks, Herds, and Schools: A Distributed Behavioral Model”, ACM SIGGRAPH '87 Conference Proceedings, pp. 25-34, California, July 1987

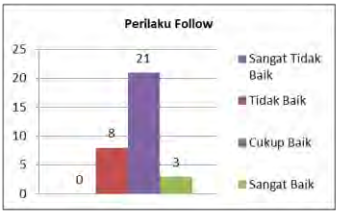
[7] Reynolds. Craig W, “Steering Behavior for Autonomous Character” Reynolds, C. W. “Steering Behaviors For Autonomous Characters” Game Developers Conference, pp 763-782, San Jose, California, 1999

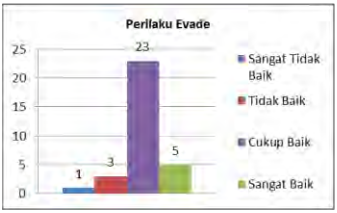
BIOGRAFI PENULIS

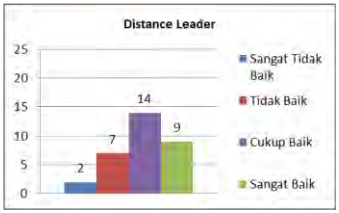
Priyodiva Robby Nugroho atau biasa dipanggil Robby dilahirkan di bandung pada tanggal 13 september 1991. Penulis menempuh pendidikan Sekolah Dasar di SD Hang Tuah 10 Juanda. Rasa penasaran terhadap hal baru membawa penulis pergi merantau selama 6 tahun di kota solo untuk menempuh pendidikan sekolah menengah pertama dan sekolah menengah atas di MTS dan SMA PPMI Assalaam Surakarta. Setelah lulus dari pendidikan SMA, penulis meneruskan pendidikan di Kampus Institut Teknologi Sepuluh Nopember tepatnya di Jurusan Teknik Elektro bidang studi Teknik Komputer dan Telematika.Saat menjadi mahasiswa ikut aktif dalam kegiatan kemahasiswaan seperti himpunan mahasiswa di jurusan teknik elektro. Selain itu untuk menyalurkan hobi terhadap bidang robotik, penulis juga bergabung dalam divisi workshop untuk mengikuti kegiatan lomba di bidang line follower nasional. Setelah bergabung sebagai asisten pada laboratorium teknik elektro B201, penulis mencoba mempelajari hal-hal baru yang terkait ilmu komputer. Saat ini penulis menaruh minat besar dalam penelitian teknologi *game* dan *mobile*.

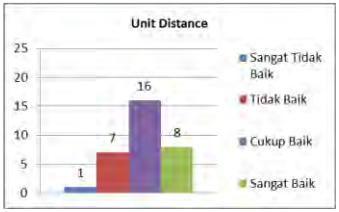
LAMPIRAN

1. Hasil kuisioner terhadap responden









2. Tampilan *in-game* pasukan pada *state idle*

3. Tampilan *in-game* pasukan pada *state follow*

4. Tampilan *in-game* pasukan pada *state attack*

5. Tampilan *in-game* pasukan pada *state evade*

2. Data hasil pengujian pada *device*

a. Data pengujian perilaku *follow* pada *device* 1

Jumlah Pasukan	Data ke-	*Follow*		
		FPS	CPU (%)	RAM (KB)
10	1	92	79	150031
	2	87	57	167169
	3	94	81	167539
20	1	51	73	167193
	2	61	58	168239
	3	60	77	207922
30	1	39	87	168022
	2	40	87	209158
	3	41	85	156569
40	1	28	84	217563
	2	29	72	212670
	3	27	86	212881
50	1	16	87	168592
	2	18	87	169260
	3	20	87	164273
60	1	14	89	169586
	2	15	89	210720
	3	15	87	205841
70	1	9	91	170232
	2	11	88	170705
	3	12	89	165879
80	1	7	90	229785
	2	8	82	248792
	3	7	90	248407
90	1	6	92	149713
	2	2	91	169942
	3	7	91	164883
100	1	5	91	150176
	2	4	93	170556
	3	6	91	165697

b. Data pengujian perilaku *idle* pada *device* 1

Jumlah Pasukan	Data ke-	*Idle*		
		FPS	CPU (%)	RAM (KB)
10	1	64	77	148735
	2	32	77	167536
	3	101	66	167793
20	1	82	74	166160
	2	87	64	168702
	3	83	41	208145
30	1	60	87	168164
	2	51	71	208959
	3	60	85	158723
40	1	43	84	212630
	2	42	60	212674
	3	42	86	212929
50	1	32	87	168841
	2	29	87	169324
	3	30	88	164309
60	1	24	86	169646
	2	23	87	205617
	3	23	87	205936
70	1	18	91	170513
	2	18	88	170765
	3	20	89	165891
80	1	14	90	229837
	2	12	82	248804
	3	14	88	247383
90	1	12	92	149773
	2	8	91	170002
	3	12	91	164894
100	1	10	90	151679
	2	9	90	170612
	3	10	90	165721

c. Data pengujian perilaku *attack* pada *device* 1

Jumlah Pasukan	Data ke-	*Attack*		
		FPS	CPU (%)	RAM (KB)
10	1	55	84	168899
	2	92	79	166872
	3	100	75	211325
20	1	61	86	165931
	2	53	78	168123
	3	62	70	208144
30	1	38	87	209363
	2	37	82	208688
	3	39	87	177636
40	1	32	85	217619
	2	29	89	212740
	3	36	86	212860
50	1	24	88	169089
	2	26	87	164216
	3	25	87	164353
60	1	16	86	169770
	2	17	80	205669
	3	16	87	243844
70	1	10	91	170612
	2	14	89	170937
	3	13	89	165935
80	1	7	90	250529
	2	5	81	248289
	3	6	91	247445
90	1	9	73	157988
	2	7	92	170061
	3	2	92	164902
100	1	8	90	170417
	2	3	92	170670
	3	8	90	165873

d. Data pengujian perilaku *follow* pada *device* 2

Jumlah Pasukan	Data ke-	*Follow*		
		FPS	CPU (%)	RAM (KB)
10	1	78	96	167769
	2	80	98	169074
	3	89	93	169698
20	1	48	96	170870
	2	49	94	171279
	3	52	95	171695
30	1	38	98	172452
	2	36	96	172576
	3	37	93	172720
40	1	25	97	172656
	2	26	96	172913
	3	23	97	193456
50	1	18	92	156485
	2	19	93	171406
	3	16	95	233906
60	1	13	99	173268
	2	14	93	213319
	3	12	95	213556
70	1	10	96	174218
	2	11	95	211558
	3	9	97	215973
80	1	6	95	154380
	2	7	92	210832
	3	6	93	211203
90	1	5	90	155275
	2	7	91	176103
	3	6	90	211901
100	1	4	100	155458
	2	5	89	175090
	3	4	97	204222

e. Data pengujian perilaku *idle* pada *device* 2

Jumlah Pasukan	Data ke-	*Idle*		
		FPS	CPU (%)	RAM (KB)
10	1	100	82	197605
	2	96	98	169162
	3	102	93	169680
20	1	68	91	170879
	2	67	94	171343
	3	69	95	171747
30	1	47	98	172687
	2	48	96	172636
	3	48	93	172780
40	1	34	97	172805
	2	30	96	172966
	3	34	97	193512
50	1	26	92	156375
	2	23	93	171466
	3	25	95	234022
60	1	20	99	173455
	2	19	93	213376
	3	17	91	213717
70	1	15	96	174352
	2	13	93	211622
	3	15	98	216033
80	1	10	95	154504
	2	11	92	210953
	3	10	93	211311
90	1	9	90	155395
	2	9	91	176163
	3	10	90	212185
100	1	7	100	155686
	2	5	89	175214
	3	6	97	212590

f. Data pengujian perilaku *attack* pada *device* 2

Jumlah Pasukan	Data ke-	*Attack*		
		FPS	CPU (%)	RAM (KB)
10	1	70	96	167635
	2	55	98	169390
	3	73	92	169756
20	1	46	94	170943
	2	49	91	171403
	3	51	92	171803
30	1	25	97	172749
	2	33	96	172812
	3	31	92	172944
40	1	27	93	172846
	2	24	96	173078
	3	25	97	201908
50	1	20	91	169370
	2	20	91	171578
	3	20	96	242362
60	1	14	99	173571
	2	13	93	213440
	3	14	94	213601
70	1	11	96	174588
	2	10	93	215741
	3	11	98	216349
80	1	5	95	154564
	2	6	92	211013
	3	6	93	211539
90	1	7	93	163643
	2	6	91	176279
	3	7	89	216501
100	1	6	100	168087
	2	4	89	175278
	3	5	98	212702

g. Data pengujian perilaku *follow* pada *device* 3

Jumlah Pasukan	Data ke-	*Follow*		
		FPS	CPU (%)	RAM (KB)
10	1	125	100	73051
	2	110	50	69707
	3	130	50	75493
20	1	84	99	74126
	2	76	99	74876
	3	88	75	69934
30	1	49	82	75123
	2	46	100	69474
	3	50	95	68515
40	1	38	99	74673
	2	34	66	69985
	3	29	97	67639
50	1	21	100	67935
	2	26	98	67981
	3	27	99	67860
60	1	12	99	75123
	2	18	100	67981
	3	17	97	69829
70	1	10	98	69238
	2	10	100	67881
	3	13	97	67034
80	1	7	98	76060
	2	7	98	65086
	3	10	100	64112
90	1	6	98	73138
	2	7	75	62164
	3	8	82	61190
100	1	5	97	70216
	2	7	100	69242
	3	4	98	68268

h. Data pengujian perilaku *idle* pada *device* 3

Jumlah Pasukan	Data ke-	*Idle*		
		FPS	CPU (%)	RAM (KB)
10	1	168	99	73376
	2	140	100	69592
	3	138	100	74602
20	1	101	97	74004
	2	96	100	74873
	3	89	75	69792
30	1	60	82	75347
	2	68	100	68299
	3	70	95	68632
40	1	52	99	74725
	2	48	98	67528
	3	53	100	67737
50	1	38	97	67999
	2	41	99	67838
	3	40	100	67910
60	1	37	99	76387
	2	31	75	67979
	3	29	100	69974
70	1	24	75	69334
	2	25	99	67893
	3	21	98	66986
80	1	20	82	75945
	2	19	97	64905
	3	14	97	63864
90	1	16	100	72824
	2	15	98	61783
	3	19	100	60743
100	1	13	97	69702
	2	12	99	68662
	3	10	98	67621

i. Data pengujian perilaku *attack* pada *device* 3

Jumlah Pasukan	Data ke-	*Attack*		
		FPS	CPU (%)	RAM (KB)
10	1	141	100	73532
	2	152	97	69357
	3	152	100	74740
20	1	79	92	74159
	2	86	98	74924
	3	81	98	69829
30	1	50	99	74864
	2	48	92	69238
	3	46	97	67924
40	1	29	100	74840
	2	44	98	69391
	3	48	100	67745
50	1	21	98	67807
	2	34	75	67988
	3	26	99	67931
60	1	22	99	77398
	2	21	99	68248
	3	24	97	70119
70	1	18	100	69347
	2	15	98	67899
	3	17	92	66901
80	1	14	99	75791
	2	12	98	64681
	3	8	75	63571
90	1	9	100	72461
	2	11	92	61351
	3	8	100	60241
100	1	6	97	69131
	2	3	75	68021
	3	5	100	66911

j. Data pengujian perilaku *follow* pada *device* 4

Jumlah Pasukan	Data ke-	*Follow*		
		FPS	CPU (%)	RAM (KB)
10	1	89	66	75512
	2	90	61	178400
	3	105	61	171851
20	1	55	53	81159
	2	56	55	81823
	3	58	55	172863
30	1	38	55	81207
	2	34	87	81611
	3	29	87	172919
40	1	22	59	81327
	2	16	85	81823
	3	18	88	172919
50	1	15	82	81415
	2	8	52	81943
	3	5	80	172855
60	1	13	72	81531
	2	14	72	81773
	3	11	78	173486
70	1	8	88	82436
	2	11	91	83271
	3	7	90	178346
80	1	6	71	82213
	2	8	91	172919
	3	8	90	179322
90	1	3	88	84132
	2	6	90	178631
	3	7	91	179331
100	1	3	91	84414
	2	5	91	191862
	3	5	92	192421

k. Data pengujian perilaku *idle* pada *device* 4

Jumlah Pasukan	Data ke-	*Idle*		
		FPS	CPU (%)	RAM (KB)
10	1	103	85	75512
	2	90	81	178400
	3	101	50	171191
20	1	76	60	81223
	2	82	75	82079
	3	74	73	172991
30	1	45	64	81207
	2	52	59	81675
	3	47	57	172459
40	1	35	82	81391
	2	36	74	81951
	3	34	78	172671
50	1	22	89	81479
	2	25	87	82007
	3	23	90	172919
60	1	19	53	81697
	2	11	91	82313
	3	18	90	173812
70	1	14	91	82717
	2	13	87	81812
	3	10	87	178921
80	1	11	91	83798
	2	12	89	172919
	3	11	89	179318
90	1	8	92	84878
	2	8	91	178317
	3	8	91	179651
100	1	7	92	84414
	2	6	91	199819
	3	7	91	198159

l. Data pengujian perilaku *attack* pada *device* 4

Jumlah Pasukan	Data ke-	*Attack*		
		FPS	CPU (%)	RAM (KB)
10	1	80	54	75512
	2	90	87	175424
	3	96	83	171851
20	1	56	65	81351
	2	69	89	172607
	3	70	69	247339
30	1	38	66	81271
	2	36	68	172203
	3	33	49	172523
40	1	30	85	81519
	2	27	78	172479
	3	28	47	255143
50	1	17	82	81607
	2	21	73	172471
	3	19	64	172983
60	1	16	63	81689
	2	16	78	172607
	3	17	85	203457
70	1	13	89	82411
	2	12	88	177652
	3	2	84	179445
80	1	10	91	83721
	2	3	89	172459
	3	10	89	179423
90	1	3	92	84411
	2	7	92	178781
	3	6	91	179552
100	1	6	91	84321
	2	3	91	199112
	3	4	92	199331

m. Data pengujian perilaku *follow* pada *device* 5

Jumlah Pasukan	Data ke-	*Follow*		
		FPS	CPU (%)	RAM (KB)
10	1	156	92	72607
	2	146	92	68554
	3	120	89	68286
20	1	94	96	72907
	2	80	95	72987
	3	74	95	69529
30	1	61	96	67868
	2	58	99	72812
	3	62	98	73670
40	1	49	100	67268
	2	43	97	67246
	3	47	98	67233
50	1	34	97	67969
	2	35	97	67712
	3	24	96	67964
60	1	20	93	73082
	2	23	89	73518
	3	22	96	73824
70	1	21	97	69221
	2	22	97	69382
	3	21	97	69277
80	1	12	98	69394
	2	10	98	69520
	3	15	98	69733
90	1	9	97	70020
	2	10	97	68349
	3	13	97	68581
100	1	7	97	73586
	2	10	97	69218
	3	9	97	69094

n. Data pengujian perilaku *idle* pada *device* 5

Jumlah Pasukan	Data ke-	*Idle*		
		FPS	CPU (%)	RAM (KB)
10	1	198	82	72867
	2	180	82	68643
	3	173	89	68332
20	1	125	93	73032
	2	111	95	72943
	3	90	95	69555
30	1	90	96	67610
	2	86	99	73074
	3	91	98	73781
40	1	68	100	67280
	2	70	97	67266
	3	66	100	67257
50	1	52	97	67996
	2	50	97	67597
	3	49	96	67597
60	1	37	93	73204
	2	36	89	73630
	3	39	96	73876
70	1	30	97	69439
	2	31	97	69234
	3	30	97	69198
80	1	16	98	69386
	2	17	98	69268
	3	18	97	69697
90	1	19	97	69088
	2	17	97	68396
	3	18	97	68515
100	1	15	97	74088
	2	16	97	69443
	3	17	97	69258

o. Data pengujian perilaku *attack* pada *device* 5

Jumlah Pasukan	Data ke-	*Attack*		
		FPS	CPU (%)	RAM (KB)
10	1	162	97	73722
	2	138	96	68506
	3	129	89	68026
20	1	86	93	73782
	2	98	95	73429
	3	92	98	67429
30	1	55	92	67616
	2	62	99	73370
	3	63	90	73717
40	1	48	100	67392
	2	41	97	67225
	3	47	98	68001
50	1	36	97	68208
	2	34	93	67829
	3	32	98	68039
60	1	28	93	73441
	2	26	99	73696
	3	27	96	74280
70	1	19	98	69315
	2	18	97	69161
	3	21	97	69309
80	1	11	98	69285
	2	13	98	69324
	3	11	98	69576
90	1	14	97	68683
	2	13	97	68319
	3	12	97	68600
100	1	10	97	74487
	2	1	97	69143
	3	9	97	69250

p. Data pengujian perilaku *follow* pada *device* 6

Jumlah Pasukan	Data ke-	*Follow*		
		FPS	CPU (%)	RAM (KB)
10	1	128	71	150656
	2	135	93	385242
	3	134	97	425698
20	1	79	96	151402
	2	81	96	276542
	3	83	96	312858
30	1	52	97	151562
	2	48	96	447402
	3	51	96	649581
40	1	38	96	151653
	2	37	97	403986
	3	39	96	444236
50	1	26	95	151839
	2	21	95	570693
	3	20	95	610793
60	1	21	97	152123
	2	15	96	579544
	3	16	97	620641
70	1	12	97	153471
	2	10	97	457096
	3	17	97	497843
80	1	10	98	153066
	2	11	98	338042
	3	13	98	379707
90	1	7	97	152084
	2	8	96	509709
	3	10	97	551636
100	1	6	98	152704
	2	7	97	493313
	3	8	98	534593

q. Data pengujian perilaku *idle* pada *device* 6

Jumlah Pasukan	Data ke-	*Idle*		
		FPS	CPU (%)	RAM (KB)
10	1	167	71	150832
	2	165	97	385402
	3	171	97	425797
20	1	116	96	151627
	2	110	96	287438
	3	98	96	312958
30	1	78	96	151716
	2	75	97	447502
	3	69	96	649813
40	1	57	97	151746
	2	58	97	404094
	3	50	97	444358
50	1	42	95	151951
	2	43	95	570874
	3	41	95	610856
60	1	30	96	152577
	2	31	96	579780
	3	30	95	620064
70	1	24	97	147253
	2	25	97	457327
	3	24	97	497991
80	1	20	98	153718
	2	19	97	338434
	3	19	98	380055
90	1	16	97	153140
	2	15	96	509799
	3	16	94	550416
100	1	13	96	152819
	2	12	97	498518
	3	11	98	534323

r. Data pengujian perilaku *attack* pada *device* 6

Jumlah Pasukan	Data ke-	*Attack*		
		FPS	CPU (%)	RAM (KB)
10	1	125	96	171428
	2	134	97	397718
	3	140	97	438230
20	1	82	96	164142
	2	84	97	288369
	3	87	96	333533
30	1	61	96	168610
	2	58	95	467999
	3	62	95	656436
40	1	46	97	174161
	2	45	97	424751
	3	41	96	454729
50	1	34	96	173048
	2	33	96	591764
	3	35	95	619225
60	1	26	97	169161
	2	24	96	600538
	3	23	95	638589
70	1	22	97	164286
	2	21	95	472004
	3	20	97	513151
80	1	18	98	160230
	2	16	97	359105
	3	23	98	400673
90	1	14	98	173792
	2	13	97	522264
	3	8	97	571072
100	1	9	97	173763
	2	11	97	514298
	3	8	97	552270

Halaman ini sengaja dikosongkan