



TESIS - TE142599  
**STRATEGI ADAPTIF KELOMPOK DI PERMAINAN  
TAKTIK MENGGUNAKAN GOAL-ORIENTED  
ACTION PLANNING**

RESTUADI STUDIawan  
07111150050002

DOSEN PEMBIMBING  
MOCHAMAD HARIADI, ST., M.SC., PH.D.  
DR. SURYA SUMPENO, ST., M.SC

PROGRAM MAGISTER  
BIDANG KEAHLIAN JARINGAN CERDAS MULTIMEDIA  
DEPARTEMEN TEKNIK ELEKTRO  
FAKULTAS TEKNOLOGI ELEKTRO  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
2018





TESIS - TE142599

**STRATEGI ADAPTIF KELOMPOK DI PERMAINAN  
TAKTIK MENGGUNAKAN GOAL-ORIENTED  
ACTION PLANNING**

RESTUADI STUDIAWAN  
07111150050002

DOSEN PEMBIMBING  
MOCHAMAD HARIADI, ST., M.SC., PH.D.  
DR. SURYA SUMPENO, ST., M.SC

PROGRAM MAGISTER  
BIDANG KEAHLIAN JARINGAN CERDAS MULTIMEDIA  
DEPARTEMEN TEKNIK ELEKTRO  
FAKULTAS TEKNOLOGI ELEKTRO  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
2018



**LEMBAR PENGESAHAN**

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

oleh:

Restuadi Studiawan  
NRP. 07111150050002


Tanggal Ujian : 14 Juni 2017  
Periode Wisuda : Maret 2018

Disetujui oleh:

  
1. Mochamad Hariadi, ST., M.Sc., Ph.D. (Pembimbing I)  
NIP: 19691209 199703 1 002

  
2. Dr. Surya Sumpeno, ST., M.Sc (Pembimbing II)  
NIP: 19690613 199702 1 003

  
3. Dr. Eko Mulyanto Yuniarno, S.T., M.T. (Penguji)  
NIP: 19680601 199512 1 009

  
4. Dr. Supeno Mardi Susiki Nugroho, ST., M.T. (Penguji)  
NIP: 19700313 199512 1 001

Dekan Fakultas Teknologi Elektro

  
Dr. Iri Arief Sardjono, S.T., M.T.  
NIP. 197002121995121001

*Halaman ini sengaja dikosongkan*

## **PERNYATAAN KEASLIAN TESIS**

Dengan ini saya menyatakan bahwa isi keseluruhan Tesis saya dengan judul “**STRATEGI ADAPTIF KELOMPOK DI PERMAINAN TAKTIK MENGGUNAKAN GOAL-ORIENTED ACTION PLANNING**” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Juni 2017

Restuadi Studiawan  
NRP. 07111150050002

*Halaman ini sengaja dikosongkan*



# **STRATEGI ADAPTIF KELOMPOK DI PERMAINAN TAKTIK MENGUNAKAN GOAL-ORIENTED ACTION PLANNING**

Nama mahasiswa : Restuadi Studiawan  
NRP : 07111150050002  
Pembimbing : 1. Mochamad Hariadi, ST., M.Sc., Ph.D.  
2. Dr. Surya Sumpeno, ST., M.Sc

## **ABSTRAK**

Meningkatnya kompleksitas permainan elektronik modern seiring dengan peningkatan kebutuhan akan agen cerdas yang dapat dibangun dengan mudah. Salah satu permainan elektronik yang membutuhkan agen cerdas tersebut adalah permainan *real-time tactics* (RTT). Dalam tipe permainan ini perencanaan aksi yang baik dapat membuat permainan yang menantang bagi pemain.

Penelitian ini mengeksplorasi kemungkinan penggunaan dari (GOAP) pada sebuah permainan RTT. Dengan menggunakan GOAP, dinamisme taktik dapat dibentuk dengan waktu penggunaan yang tidak terasa berat dalam permainan.

Kata kunci: *Goal-Oriented Action Planning*, kecerdasan buatan, perencanaan, permainan elektronik

*Halaman ini sengaja dikosongkan*

# **ADAPTIVE GROUP STRATEGY IN TACTICAL GAME USING GOAL-ORIENTED ACTION PLANNING**

By : Restuadi Studiawan  
Student Identity Number : 07111150050002  
Supervisor(s) : 1. Moch. Hariadi, ST. MSc. Ph.D.  
2. Dr. Surya Sumpeno, ST, M.Sc

## **ABSTRACT**

Along with improvement of modern electronic games, necessity of an intelligent agent that easily build is needed. One of electronic games that need good intelligent agent is realtime tactics. In this game type, good action planning is necessary to provide best experience to the player.

We explore usage possibility of Goal-Oriented Action Planning (GOAP) in tactical game. Using GOAP, tactic dinamism can be provided with reasonable amount of runtime.

Key words: Artificial intelligence, Electronic Games, Goal-Oriented Action Planning, Planning

*Halaman ini sengaja dikosongkan*

## KATA PENGANTAR

Alhamdulillah Rabbilalamin, Segala puji dan syukur hanya kepada Allah SWT atas rahmat dan karunia-Nya dan salawat beriring salam senantiasa kita hanturkan kepada pangkuan Nabi Besar kita umat Islam, Nabi Muhammad SAW, pada akhirnya penulis dapat menyelesaikan Tesis ini dengan judul *Strategi Adaptif Kelompok di Permainan Taktik Menggunakan Goal-Oriented Action Planning*.

Laporan Tesis ini penulis ajukan untuk memenuhi salah satu syarat guna mencapai gelar Magister Teknik (MT) pada Jurusan Teknik Elektro Bidang Keahlian Jaringan Cerdas Multimedia di Institut Teknologi Sepuluh November (ITS) Surabaya, Pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada :

1. Allah SWT, Atas limpahan rahmat dan karunia-Nya yang tak terhingga,
2. Kepada Bapak, Ibu, dan seluruh keluarga besar atas doa dan dukungannya,
3. Bapak Mochamad Hariadi, ST., M.Sc., Ph.D. dan Bapak Dr. Surya Sumpeno, ST., M.Sc selaku dosen pembimbing yang telah sangat luar biasa dalam memberikan semangat, arahan dan bantuan kepada penulis dalam menyelesaikan tesis ini,
4. Dan kepada semua pihak yang tidak bisa penulis sebut satu persatu.

Semoga Allah SWT senantiasa memberikan balasan kebaikan yang berlipat ganda. Harapan penulis, semoga Tesis ini dapat berguna dan bermanfaat, bagi siapa saja yang menggunakannya.

Penulis menyadari bahwa tugas tesis ini masih jauh dari sempurna, oleh karena itu penulis akan senantiasa menunggu dan menerima saran, serta kritik yang membangun untuk perkembangan dan kemajuan kita di masa yang akan datang.

Surabaya, 12 Juni 2017

Penulis

Restuadi Studiawan

*Halaman ini sengaja dikosongkan*

## DAFTAR ISI

LEMBAR PENGESAHAN.....	iii
PERNYATAAN KEASLIAN TESIS.....	v
ABSTRAK.....	vii
ABSTRACT.....	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL.....	xix
BAB 1 Pendahuluan.....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	2
1.5 Manfaat Penelitian.....	3
1.6 Sistematika Penulisan.....	3
BAB 2 Dasar Teori.....	5
2.1 Kajian Karya Terkait.....	5
2.2 Teori Penunjang.....	5
2.2.1 Real-Time Strategy.....	5
2.2.1.1 Navigasi 3D pada RTS di Ruang Angkasa.....	7
2.2.1.2 Kecerdasan Buatan dalam Real-Time Strategy.....	8
2.2.1.3 Real-Time Tactics.....	9
2.2.2 Fiksi Ilmiah yang Berlokasi di Luar Angkasa.....	10
2.2.2.1 Pergerakan Unit di Luar Angkasa.....	11
2.2.2.2 Pengorganisasian Unit di Luar Angkasa.....	12
2.2.2.3 Deteksi Unit di Luar Angkasa.....	12
2.2.2.4 Persenjataan di Luar Angkasa.....	13
2.2.3 Unity3D.....	14
2.2.4 Component Based Design.....	15
2.2.5 Goal Oriented Action Planning (GOAP).....	17

2.2.6 Blackboard System.....	21
BAB 3 Metode Penelitian.....	23
3.1 Sistem Permainan.....	23
3.1.1 Konsep Dasar Permainan.....	23
3.1.1.1 Kondisi Umum dalam Dunia Permainan.....	24
3.1.1.2 Konflik yang Diusung dalam Permainan.....	24
3.1.1.3 Teknologi yang Dimiliki dalam Dunia Permainan.....	25
3.1.1.4 Paradigma dalam Permainan.....	27
3.1.2 Alur Permainan.....	27
3.1.2.1 Observasi.....	28
3.1.2.2 Aksi.....	28
3.1.2.3 Realokasi.....	28
3.1.3 Desain Permainan.....	28
3.1.3.1 Parameter yang Dimiliki Unit dalam Permainan.....	29
3.1.3.2 Tipe Unit.....	30
3.1.3.3 Parameter Unit dalam Permainan.....	32
3.1.3.4 Komponen yang Dibuat Untuk Unit Permainan.....	34
3.1.4 Logika Unit.....	36
3.1.4.1 Dasar Logika Unit.....	36
3.1.4.2 Unit Pesawat.....	37
3.1.4.3 Unit Kapal Induk.....	37
3.1.5 Abstraksi Kontrol Unit.....	38
3.1.5.1 Navigasi.....	38
3.1.5.2 Serangan.....	39
3.1.5.3 Hangar.....	40
3.1.6 Fungsi Pendukung.....	41
3.2 Goal Oriented Action Planning.....	43
3.2.1 Strategi Umum.....	44
3.2.2 World State.....	44
3.2.3 Actions.....	45
3.2.4 Planner.....	47
3.3 <i>Rule-Based</i> AI sebagai Pembanding.....	48



BAB 4 Hasil dan Pembahasan.....	49
4.1 Spesifikasi Unit Pengujian.....	49
4.2 Hasil.....	49
4.2.1 Simbol Unit Dalam Permainan.....	49
4.2.2 Performa Produk.....	50
4.2.3 Pengujian dengan Kasus yang Ditentukan.....	50
4.2.3.1 Hanya ada Unit Kapal Induk di Dalam Dunia Permainan.....	51
4.2.3.2 Hanya ada Unit Pertahanan.....	52
4.2.3.3 Unit Kapal Induk dan Pertahanan.....	53
4.2.3.4 Hanya ada Unit Pesawat.....	54
4.2.3.5 Unit Kapal Induk dan Unit Pesawat.....	55
4.2.3.6 Ada Semua Unit di Stage Permainan.....	56
4.2.4 Pengujian Melawan Gelombang Unit.....	57
4.2.5 Pengujian Melawan <i>Rule-Based</i> AI.....	59
4.3 Analisis.....	61
BAB 5 Kesimpulan.....	63
5.1 Kesimpulan.....	63
5.2 Penelitian Selanjutnya.....	64
DAFTAR PUSTAKA.....	65
Biografi Penulis.....	67

*Halaman ini sengaja dikosongkan*

## DAFTAR GAMBAR

Gambar 2.1: Ilustrasi Navigasi RTS dalam Lingkungan Tiga Dimensi.....	7
Gambar 2.2: Pergerakan pesawat di dalam atmosfer (kiri) dan pergerakan pesawat di luar angkasa (kanan).....	11
Gambar 2.3: Struktur GameObject di Unity.....	14
Gambar 2.4: Komponen Dasar Roket.....	16
Gambar 2.5: Komponen Misil.....	16
Gambar 2.6: Perbandingan antara State Machine (kiri) dan GOAP (kanan).....	18
Gambar 2.7: Relationship class di GOAP.....	18
Gambar 2.8: Contoh Aksi yang Dihasilkan GOAP.....	20
Gambar 2.9: Komponen dasar dari Blackboard System.....	21
Gambar 3.1: Ilustrasi Rute Perjalanan Kapal Luar Angkasa dalam Permainan...	25
Gambar 3.2: Visualisasi Unit Kapal Induk.....	30
Gambar 3.3: Visualisasi Unit Pesawat Sebelum Diberi Perlengkapan.....	31
Gambar 3.4: Unit Pesawat membawa Paket Micro Missile.....	31
Gambar 3.5: Unit Pesawat membawa Paket Cruise Missile.....	32
Gambar 3.6: Unit Pesawat membawa Paket Torpedo.....	32
Gambar 3.7: Unit dalam Permainan.....	34
Gambar 3.8: Pergerakan Unit.....	38
Gambar 3.9: Pergerakan Misil.....	39
Gambar 3.10: FSM untuk Serangan Unit.....	39
Gambar 3.11: FSM untuk Hangar Kapal Induk (CV).....	40
Gambar 3.12: Flow Chart Kecerdasan Buatan Rule-Based AI.....	48
Gambar 4.1: Visualisasi Permainan.....	50
Gambar 4.2: Mengeluarkan Unit yang Membawa Torpedo Setelah Observasi....	51
Gambar 4.3: Melawan 5 Unit Kapal Induk.....	52
Gambar 4.4: Mengeluarkan Unit yang Membawa Cruise Missile.....	52
Gambar 4.5: Melawan 10 Unit Pertahanan.....	53
Gambar 4.6: Melawan 1 Unit Kapal Induk dan 4 Unit Pertahanan.....	54
Gambar 4.7: Lawan Memiliki 5 Unit Pesawat Dengan Muatan yang Berbeda....	55
Gambar 4.8: Lawan Memiliki 1 Unit Kapal Induk dan 5 Unit Pesawat Dengan	

Muatan yang Berbeda.....	56
Gambar 4.9: Lawan Memiliki 1 Unit Kapal Induk, 5 Unit Pesawat dan 4 Unit Pertahanan.....	57
Gambar 4.10: Pengujian Terhadap Gelombang Unit.....	58
Gambar 4.11: Saat GOAP Memenangkan Permainan.....	60
Gambar 4.12: Saat GOAP Kalah di Permainan.....	60

## DAFTAR TABEL

Tabel 3.1: Parameter Unit Permainan.....	33
--	----

*Halaman ini sengaja dikosongkan*

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Dalam permainan elektronik terdapat satu subgenre yang disebut sebagai *real time tactics* yang menuntut pemain untuk memfokuskan permainannya pada unit individu dan taktik dalam permainan daripada pengelolaan sumber daya dan membangun pangkalan. Bagaimana mengelola unit secara efektif untuk menyelesaikan obyektif misi menjadi tujuan utama dari tiap babak permainan.

Kata taktik sendiri berasal dari bahasa Yunani yang memiliki arti "seni penataan". Definisi tersebut mendasari munculnya subgenre ini dari genre *real time strategy* karena adanya pengkhususan dalam penataan unit-unit yang disediakan di awal misi. Permainan yang masuk dalam subgenre ini pada umumnya memiliki cakupan misi yang kecil dan unit yang terbatas. Kompleksitas dapat dicapai dengan memberi misi dimana unit yang ada di dalamnya dibatasi dan dapat diberi suplai yang memiliki batasan tertentu tergantung dari kondisi masing-masing permainan.

Agen cerdas yang digunakan dalam permainan dengan tipe ini dibuat dengan tujuan untuk menata penugasan unit-unit dalam kelompok dengan tujuan untuk menyelesaikan misi yang diberikan. Seiring dengan berkembangnya teknologi dalam dunia permainan elektronik, permintaan kecerdasan buatan yang menantang untuk agen cerdas di dalamnya juga meningkat. Perilaku agen yang dirancang oleh desainer kecerdasan buatan terkadang sampai pada saat dimana pemain dapat menemukan perilaku agen yang dapat dieksploitasi setelah memainkan permainan tersebut berulang kali.

Metode-metode untuk meningkatkan keinginan pemain dalam memainkan permainan tersebut kembali (*replayability*) telah banyak diimplementasikan di permainan elektronik yang beredar. Dari metode-metode yang digunakan pengembang aplikasi permainan, cukup banyak yang menggunakan pendekatan agen adaptif untuk meningkatkan *replayability* dari permainan yang dikembangkan. Dengan menggunakan agen yang adaptif terhadap kondisi permainan, eksploitasi pemain terhadap respon kecerdasan buatan rigid yang biasa

muncul dari agen cerdas dengan perencanaan yang dibuat permanen dapat dikurangi.

GOAP (*Goal Oriented Action Planning*) merupakan salah satu pendekatan agen adaptif yang mampu menyusun aksi dari pengamatan terhadap kondisi dunia permainan setelah mengetahui apa saja yang dapat dilakukan oleh unit-unit yang ada dibawah kendali agen cerdas. Hal ini sesuai dengan pola permainan taktik yang membutuhkan penataan ulang aksi dari unit-unit yang ada di dalamnya. Untuk itu, GOAP dipergunakan dalam sebagai penentu aksi kelompok.

Atas dasar itu, dipilihlah judul "***Strategi Adaptif Kelompok di Permainan Taktik menggunakan Goal Oriented Action Planning***".

## **1.2 Perumusan Masalah**

Penelitian berbasis GOAP yang ditemukan banyak berada pada domain agen individu dan strategi umum. Di dalam penelitian ini, kemampuan menyusun strategi kelompok dengan menggunakan GOAP diteliti.

## **1.3 Batasan Masalah**

Pembatasan masalah dalam penelitian ini adalah:

1. Menggunakan permainan taktik luar angkasa yang disusun bersamaan dengan penelitian dan dibangun menggunakan *game engine* Unity.
2. Aksi yang dijadikan dasar analisa merupakan aksi-aksi yang dapat dilakukan unit dalam permainan.
3. Hasil perencanaan agen merupakan daftar aksi yang bertujuan untuk menyesuaikan komposisi tim agen dengan komposisi tim lawan.

## **1.4 Tujuan Penelitian**

Berdasar pada latar belakang yang telah dijabarkan, maka tujuan penelitian yang ingin dicapai adalah untuk mendapatkan agen cerdas yang dapat menyesuaikan komposisinya terhadap komposisi unit lawan di dalam permainan *real time tactics*.



## **1.5 Manfaat Penelitian**

Adapun manfaat yang diharapkan dari penelitian ini adalah acuan membangun kecerdasan buatan menggunakan GOAP yang pada permainan taktik di Unity dapat menggunakan arsitektur yang didapatkan disini.

## **1.6 Sistematika Penulisan**

Sistematika penulisan buku tesis ini adalah sebagai berikut:

- Bab I : Pendahuluan  
Bagian ini menjelaskan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan dan manfaat penelitian.
- Bab II : Tinjauan Pustaka  
Dalam bagian ini akan dibahas mengenai teori-teori yang dapat menunjang dan menjadi acuan dalam perancangan penelitian. Bagian tersebut meliputi teori dasar mengenai Kecerdasan Buatan Adaptif, Goal Oriented Action Planning, arsitektur dalam Unity dan Blackboard System.
- Bab III : Metodologi Penelitian  
Pada bagian ini akan dibahas mengenai perancangan sistem dan implementasi. Yang di dalamnya terdapat blok diagram alur sistem secara keseluruhan dan penjelasan proses-proses dengan detail.
- Bab IV : Hasil dan Pembahasan  
Pada bagian ini akan dipaparkan hasil dari uji coba sistem dan analisisnya.
- BAB V : Penutup  
Pada bagian ini berisikan kesimpulan yang diperoleh berdasarkan hasil dan pembahasan. Selain itu pada bagian ini juga berisi penelitian selanjutnya yang akan dilakukan.

*Halaman ini sengaja dikosongkan*

## **BAB 2**

### **DASAR TEORI**

#### **2.1 Kajian Karya Terkait**

Beberapa penelitian telah dilakukan terkait penggunaan GOAP untuk menyusun perencanaan kecerdasan buatan dalam permainan elektronik. Dalam tesisnya, David Pittman meneliti tentang arsitektur implementasi GOAP dalam aksi agen individu [1]. Huiliang Zhang, Zhiki Shen, Chunyan Miao membahas tentang arsitektur berhierarki untuk mengantisipasi *big O* yang besar dari sifat dynamic programming yang dibawa oleh arsitektur ini [2].

#### **2.2 Teori Penunjang**

Bagian ini berisi teori-teori yang digunakan dalam penelitian ini. Tool yang digunakan dan paradigma yang digunakan dalam tool tersebut serta penjelasan dasar dari metode yang digunakan dalam penelitian ini diterangkan disini.

##### **2.2.1 Real-Time Strategy**

Permainan elektronik memiliki banyak genre. Salah satunya adalah genre strategi dimana pemain memiliki kuasa atas unit-unit permainan dan dituntut untuk menyusun strategi agar dapat memenangkan permainan berjalan. Pada awalnya permainan berjalan berbasis urutan (*turn-based*), dimana permainan akan berhenti menunggu pemain memilih aksi sebelum melanjutkan state permainan. Seiring dengan perkembangan kemampuan perangkat keras, permainan dapat menjalankan strategi secara *real time* sehingga pemain harus mengikut sertakan perencanaan waktu dalam strateginya.

Permainan elektronik berbasis strategi yang berjalan secara real time mulai muncul di awal tahun 80an. Pengembang permainan elektronik pada era ini pada umumnya membuat permainan strategi berbasis *turn-based*, dan permainan *real-time* dipenuhi oleh permainan aksi. Beberapa permainan mulai mencoba menggabungkan elemen strategi dengan pola permainan *real-time* di masa ini. Permainan dengan komponen-komponen yang saat digabungkan akan menjadi dasar definisi permainan *real-time strategy (RTS)* mulai bermunculan di era ini.

Di akhir tahun 1980an dan awal tahun 1990an komponen-komponen tersebut sudah mulai digabungkan oleh pengembang permainan elektronik dalam membuat produk permainan strategi mereka. Beberapa permainan yang bisa dikatakan memiliki jiwa *RTS* muncul disini.

Dalam satu dekade perkembangannya, istilah *RTS* belum pernah digunakan sama sekali oleh pengembang ataupun dalam pemasaran permainan dengan tipe ini. Istilah ini pertama kali digunakan pada tahun 1992 saat produser dari permainan *Dune II* (Westwood Studios, 1992) mendeskripsikan jenis permainan elektronik yang dibuatnya [3]. Setelahnya, istilah *RTS* mulai banyak digunakan untuk mengategorikan permainan bertipe strategi dan berjalan secara *real time*.

Sejak *Dune II* diterbitkan, mulai banyak bermunculan permainan yang mengusung genre *RTS*. Beberapa seri populer seperti *Warcraft* (Blizzard Entertainment, 1994), *Starcraft* (Blizzard Entertainment, 1998), *Command and Conquer* (Westwood Studios, 1995) dan *Age of Empires* (Microsoft Studios, 1997) mampu bertahan hingga sekarang.

Komponen permainan yang pada umumnya ada dalam genre *RTS* adalah sebagai berikut:

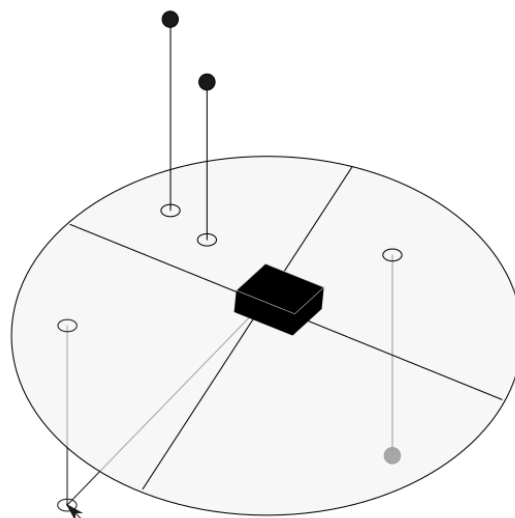
1. Berjalan secara *real-time* dimana permainan akan terus berjalan saat pemain menyusun strategi yang akan dia lakukan dalam permainan.
2. Memiliki abstraksi manajemen sumber daya.
3. Tuntutan menjalankan aksi secara *multi-tasking*. Jalannya permainan secara *real-time* memberikan peluang permainan untuk menjalankan banyak aksi sekaligus.
4. Memiliki *fog of war*. Tujuannya adalah untuk meningkatkan kompleksitas permainan dengan menyembunyikan beberapa atau seluruh atribut unit yang tidak dimiliki pemain dan berada di luar jangkauan unit-unit milik pemain.
5. Kompetisi dengan pemain lain atau kecerdasan buatan dengan *goal* tertentu yang ditentukan di awal permainan.

Walau masing-masing game di genre ini memiliki keunikan tersendiri, kelima komponen di atas pasti ada di dalam kelompok permainan *RTS*.

### 2.2.1.1 Navigasi 3D pada RTS di Ruang Angkasa

Selain permainan yang berlokasi di permukaan tanah, beberapa permainan *real-time strategy* mengambil tempat di luar angkasa seperti *Homeworld* (Relic Entertainment, 1999) dan *Sins of Solar Empire* (Stardock, 2008). *Homeworld* memasukkan unsur dimensi ketiga di dalam *gameplay*nya. Strategi yang digunakan dengan masuknya unsur dimensi ketiga ini akan berbeda dengan pola permainan *real-time strategy* pada umumnya, seperti arah datangnya serangan yang lebih bervariasi serta navigasi unit di *stage* permainan yang sedikit berbeda.

Cara menggerakkan unit yang umum digunakan dalam permainan *real-time strategy* adalah dengan memilih unit yang akan digerakkan, lalu mengarahkan kursor *mouse* ke lokasi atau objek yang diinginkan sebelum di-klik untuk memberi perintah bergerak atau aksi kepada unit yang bersangkutan. Banyak permainan *real-time strategy* yang mengambil lokasi di ruang angkasa menggunakan pola navigasi serupa dengan *real-time strategy* yang berlokasi di permukaan planet, dimana pergerakan unit dipetakan pada area dua dimensi. *Homeworld* mencoba meningkatkan batas pergerakan dari hanya berada di bidang dua dimensi menjadi pergerakan di bidang dua dimensi ditambah ketinggian dari bidang datar dasar dari lokasi permainan. Ilustrasi dari kontrol navigasi dalam permainan tersebut dapat dilihat pada Gambar 2.1.



Gambar 2.1: Ilustrasi Navigasi RTS dalam Lingkungan Tiga Dimensi

Gambar 2.1 menunjukkan ilustrasi antarmuka navigasi pada unit permainan. Sebuah bidang datar imajiner diletakkan pada posisi unit yang terpilih. Unit lain yang berada di sekitarnya akan di-*map* ke bidang datar tersebut untuk menunjukkan posisi pada bidang tersebut beserta penunjuk ketinggian dari bidang tersebut. Pergerakan unit diinstruksikan dengan melakukan *klik* pada objek yang dituju atau dengan melakukan klik pada area permainan. Cara kedua akan memerintahkan unit untuk bergerak pada bidang datar imajiner yang nampak pada antarmuka permainan. Untuk menaik/turunkan lokasi tujuan navigasi, permainan *Homeworld* menggunakan tombol [*SHIFT*] dan pergerakan *mouse* setelah menentukan lokasi di bidang datar imajiner yang diberikan.

Dengan adanya unsur pergerakan di ruang tiga dimensi ini, permainan *RTS* di ruang angkasa memiliki perbedaan mendasar dalam penyusunan strateginya.

### **2.2.1.2 Kecerdasan Buatan dalam Real-Time Strategy**

Domain kecerdasan buatan dalam permainan *RTS* memiliki lingkup yang luas. Hal ini disebabkan karena tingginya kompleksitas permainan, dimulai dari mengenali dan mengelola lingkungan permainan, analisa kemampuan lawan dari observasi yang diperoleh unit, menentukan unit apa yang perlu dibangun, *upgrade* yang dipilih sepanjang permainan, serta penyeimbangan pengelolaan sumber daya agar dapat mengantisipasi keadaan terkini dan perencanaan untuk memenangkan permainan sebagai tujuan jangka panjangnya.

Glen Robertson dan Ian Watson [4] melakukan review kecerdasan buatan pada *RTS* dengan memfokuskan pada penelitian yang menggunakan permainan *StarCraft*. Dalam review yang dibuat teknik-teknik yang digunakan dalam pembuatan kecerdasan buatan untuk *RTS* dibagi dalam tiga kelompok, yaitu:

1. *Tactical Decision Making*
2. *Strategic Decision Making*
3. *Plan Recognition and Learning*

Penelitian ini mencoba mengambil fokus pada teknik *state space planning* yang dikelompokkan dalam *strategic decision making*. *State space planning* merupakan teknik untuk melakukan perencanaan secara otomatis dari set keadaan yang ada dalam permainan. Dalam teknik ini, agen cerdas dihadapkan pada

kondisi awal (*start state*) dan diharapkan dapat mencapai kondisi akhir (*goal state*) dengan menyusun set aksi yang memiliki kondisi awal (*precondition*) dan efek yang dihasilkan jika aksi tersebut dilakukan (*effect*).

*State space planning* telah berhasil pada permainan *First Person Shooter (FPS)* dalam arsitektur yang diberi nama *Goal Oriented Action Planning (GOAP)*. Dalam arsitektur GOAP, agen cerdas dalam permainan dapat memunculkan perilaku yang bervariasi, kompleks dan menarik dengan kode yang terstruktur, dapat digunakan kembali dan mudah dikelola [5]. Tantangan yang dihadapi pengembang dalam penggunaan arsitektur ini adalah perlunya menentukan metode terbaik untuk mencari aksi yang dilakukan serta abstraksi lingkungan yang cukup sederhana untuk diolah GOAP perlu dipastikan terlebih dahulu.

Dari arsitektur yang ada, GOAP juga kurang mampu dalam mendeteksi apakah perencanaan yang sedang berjalan gagal atau ada perubahan goal state yang diakibatkan oleh perubahan lingkungan permainan. Selain itu juga, arsitektur ini pada umumnya hanya diterapkan pada satu *level* perencanaan aksi [6].

Untuk mengatasi tantangan yang telah disebutkan diatas, implementasi teknik *state space planning* di domain RTS memerlukan beberapa penyesuaian, seperti penggunaan *layer* dalam implementasi GOAP [7] dan penentuan fokus pada *goal* yang dipersempit secara dinamis [8]. Dengan dasar tersebut diharapkan penelitian ini dapat terlaksana.

### **2.2.1.3 Real-Time Tactics**

Dalam perkembangannya, beberapa permainan *real-time strategy* mencoba untuk meningkatkan kecepatan permainan. Porsi manajemen sumber daya dan riset teknologi dikurangi sehingga pemain dapat lebih fokus dalam mengorganisasi unit-unit permainan untuk menyelesaikan misi yang ditentukan. Kelompok permainan ini sering dikelompokkan dalam subgenre *real-time tactics (RTT)* karena walau masih membawa komponen *real-time strategy*, fokus pada perencanaan taktik jauh lebih besar daripada komponen strateginya.

Pola umum yang dimiliki permainan yang dikelompokkan pada subgenre ini adalah sebagai berikut:

1. Berjalan secara *real-time*.

2. Sumber daya yang sangat terbatas dan pada umumnya tidak dapat bertambah selama fase permainan.
3. Permainan menitik beratkan pada manajemen unit yang telah dimiliki saat permainan dimulai dan hampir tidak ada penambahan unit baru di dalam fase permainan.
4. Pemain lebih fokus pada unit secara individu dengan merencanakan taktik yang dapat dilakukan untuk memenangkan misi permainan.

Dalam industri permainan elektronik sendiri, istilah RTT masih bisa dibilang jarang digunakan. Hal ini disebabkan oleh tipisnya batas antara RTS dan RTT. Banyak permainan yang dipasarkan dengan membawa nama RTS walaupun memiliki unsur taktik yang kental. Beberapa RTS yang memiliki unsur taktik yang kental seperti *Company of Heroes* (THQ, 2006) dan *Warhammer 40000 : Dawn of War* (THQ, 2004) beserta sequelnya menggunakan titik di peta yang perlu dikuasai sebagai komponen strateginya. Dengan menguasai titik-titik strategis yang tersebar di arena permainan, pemain akan diberi poin yang dapat ditukar dengan unit permainan. Simplifikasi sumber daya ini membuat kedua permainan RTS tersebut dapat memfokuskan diri pada penyusunan taktik mereka.

Sementara permainan yang benar-benar dikategorikan dalam kategori RTT seperti seri *Total War* (Creative Assembly, sejak tahun 2000), *Armada 2526* (Ntronium Games, 2009) dan *Lord of Magic* (Impressions Games, 1997) memasukkan unsur strateginya dalam layer *turn-based* yang berjalan di antara fase-fase permainan *real-timenya*. Pada umumnya permainan dalam kategori RTT ini menggunakan kompleksitas lingkungan dan variasi unit sebagai fokus kompleksitas permainannya.

### **2.2.2 Fiksi Ilmiah yang Berlokasi di Luar Angkasa**

Perkembangan karya fiksi ilmiah yang mengambil tempat di luar angkasa telah dimulai sebelum permulaan abad ke-20. Namun perkembangannya tidak seberapa pesat hingga munculnya majalah-majalah *pulp fiction* yang memuat karya-karya fiksi ilmiah di akhir abad ke-19. Fiksi ilmiah yang mengambil tempat di luar angkasa sendiri baru mulai rutin muncul di majalah-majalah tersebut pada tahun 1920-an.

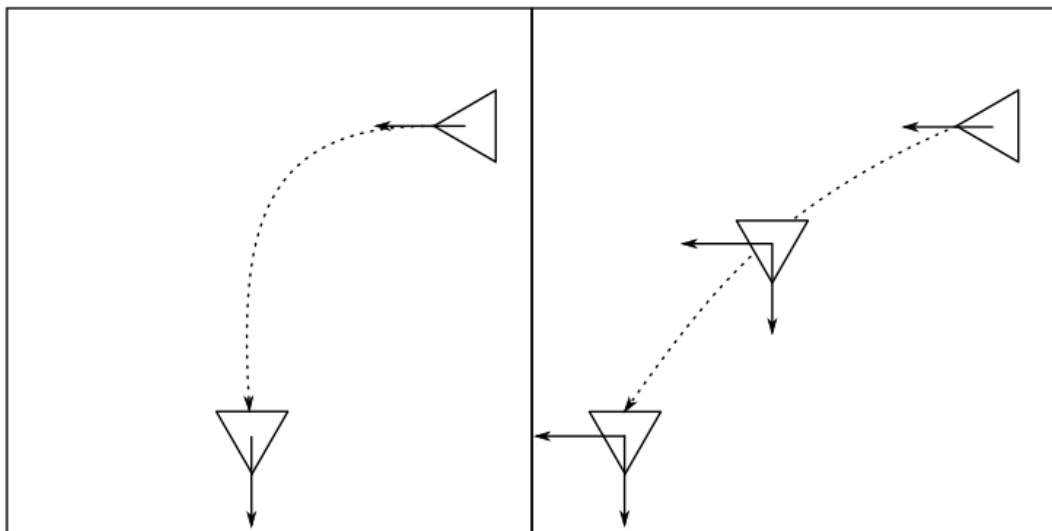


Dalam perkembangannya, belum adanya kondisi nyata yang berkaitan dengan konflik di luar angkasa menyebabkan penulis karya fiksi yang mengangkat tema tersebut membuat kondisi yang dapat dianalogikan pembaca pada dunia nyata saat karya fiksi tersebut dibuat.

### 2.2.2.1 Pergerakan Unit di Luar Angkasa

Pergerakan pesawat di seri film *Star Wars* (*The Walt Disney Company*) mengambil inspirasi dari pergerakan pesawat di film-film perang zaman perang dunia ke-dua. Unit-unit pesawat dalam film *Star Wars* melakukan pergerakan seperti layaknya pesawat yang bergerak di dalam atmosfer. Berbeda dengan pergerakan pesawat di dalam atmosfer, tidak adanya hambatan udara yang mempengaruhi pesawat mengakibatkan perbedaan cara mengubah arah pergerakan unit pesawat.

Ilustrasi pada Gambar 2.2, menunjukkan bahwa dengan mengubah haluan pesawat saja, unit dapat bergerak ke arah yang ingin dituju di dalam atmosfer. Sementara unit yang bergerak di luar angkasa memerlukan vektor pergerakan lain untuk menetralkan arah pergerakannya sebelumnya. Perbedaan ini berpengaruh pada penyusunan taktik unit dalam pengembangan ini.



Gambar 2.2: Pergerakan pesawat di dalam atmosfer (kiri) dan pergerakan pesawat di luar angkasa (kanan)

### **2.2.2.2 Pengorganisasian Unit di Luar Angkasa**

Selain pergerakan unit, pengorganisasian unit juga dianalogikan seperti kondisi dunia nyata. Penulis fiksi ilmiah pada umumnya menggunakan referensi organisasi angkatan laut sebagai dasarnya. Alasan dari penggunaan angkatan laut sebagai analogi dari pengorganisasian armada di ruang angkasa adalah kegiatan yang dilakukan armada militer di luar angkasa banyak yang mendekati kegiatan umum yang dilakukan oleh angkatan laut, seperti perawatan infrastruktur selama berada dalam sebuah misi, perencanaan pergerakan armada dan prosedur untuk meminta personel untuk memasuki kapal lain (*boarding*) [9]. Selain itu, kebutuhan untuk mengamankan wilayah [10], pengamanan dari sampah luar angkasa [11], pengawasan keamanan [12] dan pengawasan lalu lintas unit di luar angkasa [13] juga dapat menjadi dasar dari analogi pembentukan armada luar angkasa. Armada kecil untuk menjaga transportasi barang dipilih untuk dijadikan landasan pembuatan simulasi dalam penelitian ini.

### **2.2.2.3 Deteksi Unit di Luar Angkasa**

Teknik pendeteksian unit di luar angkasa juga menjadi dasar dari penyusunan taktik dan strategi dari permainan berbasis luar angkasa. Penggunaan *fog of war* di permainan RTS berbasis luar angkasa kurang sesuai dengan kondisi sesungguhnya. Kondisi luar angkasa yang memiliki ruang kosong yang sangat luas menyebabkan deteksi objek menjadi sangat mudah. Observasi unit di dalam atmosfer dapat terganggu oleh cuaca, batas horizon dan kepadatan udara. Sementara di ruang angkasa, batasan tersebut tidak ditemukan. Sebagai contoh, Voyager 1 yang saat ini berada sejauh 18.5 miliar kilometer dari bumi dapat dideteksi posisinya oleh radio teleskop dari permukaan bumi [14].

Dengan kondisi tersebut, penggunaan unit yang dapat menyembunyikan dirinya (*stealth unit*) menjadi tidak mungkin. Hal ini disebabkan karena bagaimanapun juga teknik yang diupayakan menyembunyikan unit di ruang angkasa, paling tidak jejak suhu dapat terdeteksi oleh sensor pasif yang membandingkan suhu latar belakang luar angkasa yang sangat dingin dengan objek-objek yang ada di depannya [15]. Dengan demikian, perencanaan taktik dan

strategi untuk permainan yang berlokasi di luar angkasa masih dapat dikembangkan lebih lanjut.

#### **2.2.2.4 Persenjataan di Luar Angkasa**

Permainan elektronik yang menggunakan konflik antar kelompok sebagai nilai jualnya pada umumnya memiliki persenjataan yang dapat digunakan oleh karakter atau unit di dalamnya. Begitu pula dengan permainan yang mengambil tempat di luar angkasa.

Persenjataan fiksi yang umum digunakan dalam karya yang mengambil tempat di luar angkasa pada umumnya dibagi dalam tiga kategori, yaitu:

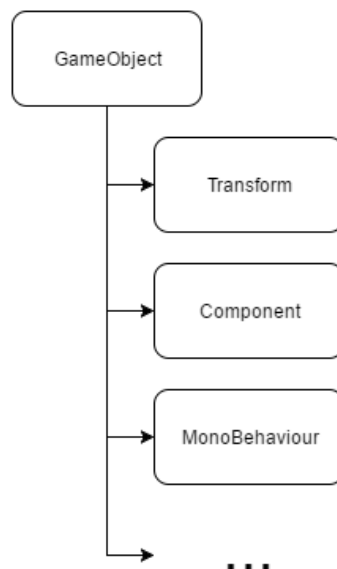
1. Senjata kinetik, merupakan senjata yang memanfaatkan energi kinetik sebagai daya hancurnya. Keuntungan dari senjata dengan tipe ini adalah memiliki daya rusak yang cukup besar dan dapat diluncurkan dengan teknologi minimal. Kelemahan yang dimiliki senjata jenis ini adalah kemungkinan korban tambahan jika proyektil yang diluncurkan tidak mengenai sasaran. Karena tidak adanya udara sebagai hambatan, proyektil akan terus meluncur sampai mengenai objek lain.
2. Senjata berbasis energi, merupakan senjata yang menggunakan energi non-kinetik sebagai daya hancurnya. Contoh dari senjata tipe ini adalah laser dan peluncur plasma. Keuntungan dari senjata ini adalah kecepatan yang jauh lebih tinggi dari senjata kinetik dan kemampuan untuk melemahkan energi yang dibawanya saat tidak mengenai sasaran. Sebaliknya, kelemahan dari senjata tipe ini adalah memerlukan teknologi tinggi untuk dapat meluncurkannya, daya hancur yang secara relatif lebih rendah dari senjata kinetik dan menurunnya daya hancur seiring bertambahnya jarak antara penembak dan sasaran.
3. Senjata cerdas, merupakan senjata yang dapat menyesuaikan arahnya terhadap lokasi sasaran. Misil dan torpedo adalah senjata yang berada dalam kelompok ini. Kecepatan yang jauh dibawah senjata tipe lainnya membuat senjata jenis ini dapat dengan mudah dihalau. Namun senjata ini dapat diberi muatan hulu ledak yang daya hancurnya bisa melebihi senjata tipe lainnya.

Penggabungan antara tiga jenis senjata yang dijabarkan diatas memerlukan perencanaan taktik dan strategi. Penelitian untuk mencari penggunaan optimal, kombinasi yang baik ataupun kombinasi yang dapat membuat pemain menyukai pengalaman bermain dapat dilakukan disini.

### 2.2.3 Unity3D

Unity merupakan *game engine* yang belakangan ini banyak digunakan pada pengembangan permainan elektronik dan simulasi. Unity dikembangkan sebagai engine 3D untuk membangun aplikasi permainan elektronik pada awal keberadaannya. Seiring dengan berjalannya waktu, banyak pengguna yang menggunakan *engine* ini untuk membuat simulasi dan juga permainan elektronik 2 dimensi. Dan saat laporan penelitian ini ditulis, Unity telah mendukung pengembangan aplikasi pada banyak perangkat, mulai dari smartphone, VR dan AR, desktop, konsol permainan, web sampai smart TV.

Dalam pengembangan permainan elektronik menggunakan Unity, pengembang perlu mengetahui struktur engine ini. Pada dasarnya, objek-objek yang dipergunakan dalam *stage* adalah aset yang telah disusun di *editor* yang disertakan dalam paket instalasi. Posisi, grafik, suara serta logika objek dipadukan di *editor* ini untuk membentuk *GameObject*. Gambar 2.3 merupakan ilustrasi dari struktur *GameObject* di Unity.



Gambar 2.3: Struktur *GameObject* di Unity

Representasi dari benda-benda yang dapat digunakan dalam menyusun objek permainan disebut *asset*. Sebuah *asset* dapat berasal dari aplikasi diluar Unity seperti grafik 2 dimensi dan 3 dimensi, berkas audio ataupun tipe-tipe lain yang didukung oleh Unity. Selain itu, ada juga *asset* yang dapat disusun oleh Unity seperti kontrol animasi, mixer audio dan texture untuk merender. Aset-aset inilah yang kemudian disusun menjadi *GameObject* menggunakan Unity editor.

Untuk mengimplementasikan logika dan kontrol terhadap aset dan properti yang ada dalam objek permainan, dibutuhkan *script*. Unity mendukung 3 bahasa pemrograman untuk membuat script, yaitu C#, JavaScript dan Boo (memiliki sintaks seperti *python*). Dengan *script* ini, logika untuk mengatur property dan jalannya permainan/simulasi dapat dibuat.

Dari dalam *script*, pengembang dapat mengontrol objek yang terkait. Sebuah *GameObject* dapat memiliki beberapa *script* sekaligus sehingga spesifikasi logika masing-masing *script* dapat dilakukan.

Setelah aset disusun di dalam *editor*, objek yang dikomposisi ini terkadang perlu disimpan dalam project. Istilah dalam Unity untuk tipe aset ini adalah *prefab*. Dengan menjadikan objek permainan sebagai *prefab*, objek tersebut dapat digunakan berulang kali. Untuk menggunakannya, prefab dapat ditarik langsung ke dalam stage atau dipanggil menggunakan fungsi `Instantiate()`.

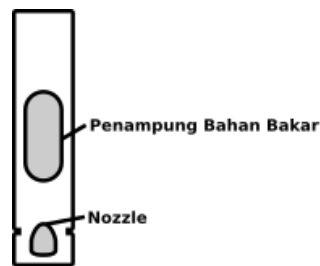
#### **2.2.4 Component Based Design**

Paradigma desain yang digunakan dalam menyusun *GameObject* di Unity adalah *Component Based Design*. Dalam paradigma ini, pemisahan perhatian pada komponen-komponen menjadi titik berat saat membuat desain aplikasi. Tujuannya adalah untuk membangun sistem yang tersusun atas komponen-komponen independen yang saling terkait sehingga fokus pengembangan dapat dispesifikasikan pada masing-masing komponen tergantung pada fungsinya. Dengan paradigma ini, *GameObject* dibuat dengan mengumpulkan komponen-komponen yang berkaitan dengan objek yang akan dibentuk.

Sebuah *component* dapat diibaratkan sebagai bagian kecil dari sistem yang besar. Masing-masingnya memiliki fungsi masing-masing dan pada umumnya mampu menyelesaikan fungsi tersebut secara independen. Selain itu, sebuah

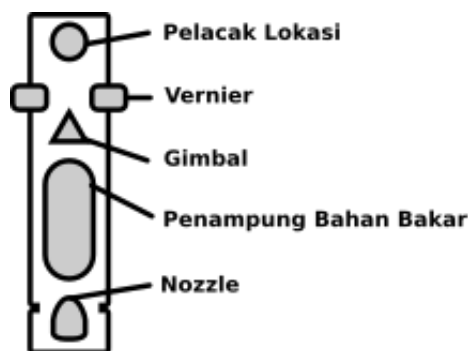
komponen jarang berdiri sendiri dan dapat digabungkan pada sebuah sistem dalam mencapai tujuan fungsionalnya, namun dapat menghasilkan hasil yang tidak perlu diketahui oleh komponen tersebut. [16]

Sebagai ilustrasi, sebuah roket memiliki banyak komponen yang berada di dalamnya. Roket memiliki komponen dasar berupa tempat menampung bahan bakar dan nozzle seperti terlihat pada Gambar 2.4. Selain itu, roket juga dapat memiliki *rocket engine*, alat stabilisasi arah seperti sirip, pengatur arah roket atau giroskop, dan sebuah struktur untuk menampung komponen-komponen tersebut.



Gambar 2.4: Komponen Dasar Roket

Dalam *component based design*, komponen-komponen tersebut dibuat dalam modul-modul terpisah dan kemudian digabungkan saat membentuk objek yang akan dipergunakan. Penggabungan komponen yang berbeda dapat menghasilkan objek yang fungsionalitas akhirnya berbeda juga. Seperti misal objek "roket" yang hanya berjalan lurus tanpa kontrol hanya membutuhkan bahan bakar, nozzle dan sirip. Sementara jika ingin membuat "misil" yang dapat mengikuti sasaran, selain komponen dasar diatas, perlu ditambahkan komponen pelacak lokasi dan pengatur arah roket seperti vernier thruster dan gimbal seperti yang terlihat pada Gambar 2.5.



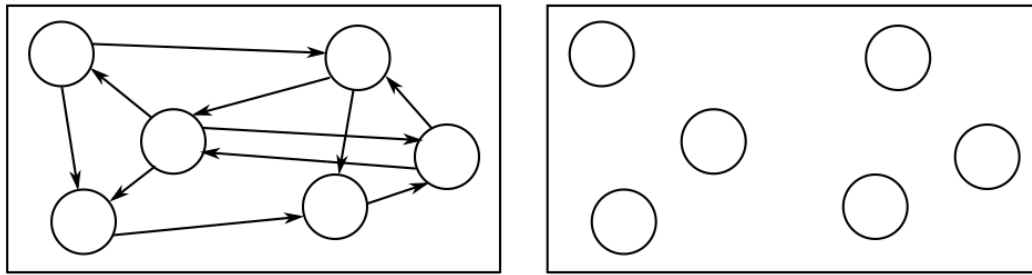
Gambar 2.5: Komponen Misil

Selain berdiri sebagai individu, komponen juga dapat disusun dari gabungan komponen-komponen lainnya. Seperti misal komponen dasar roket tersebut dapat disatukan dalam sebuah komponen yang mengusung fungsi navigasi. Dengan demikian, komponen baru yang terbentuk (navigasi) dapat digunakan kembali pada objek-objek lainnya seperti pesawat, satelit atau kapal luar angkasa.

### **2.2.5 Goal Oriented Action Planning (GOAP)**

GOAP merupakan pendekatan perencanaan aksi pada agen cerdas berbasis STRIPS (Stanford Research Institute Problem Solver) yang merupakan teknik problem solving yang dikembangkan di Stanford untuk membuat perencanaan otomatis dengan menyusun *problem-solving tree* dimana node yang dimilikinya merepresentasikan subproblem. Pembentukan tree ini dilakukan dengan membuat abstraksi keadaan dunia dan kemampuan agen. Abstraksi keadaan dunia yang berupa *first-order logic* kemudian diubah oleh operator kemampuan agen sampai keadaan dunia memenuhi tujuan agen [17]. Dengan membuat abstraksi terhadap kondisi dunia, divisi pengembangan kecerdasan buatan di permainan elektronik yang pada umumnya berasal dari disiplin ilmu berbeda dari divisi lainnya mampu membangun sistem keerdasan buatan dengan cara yang lebih intuitif dan ramah.

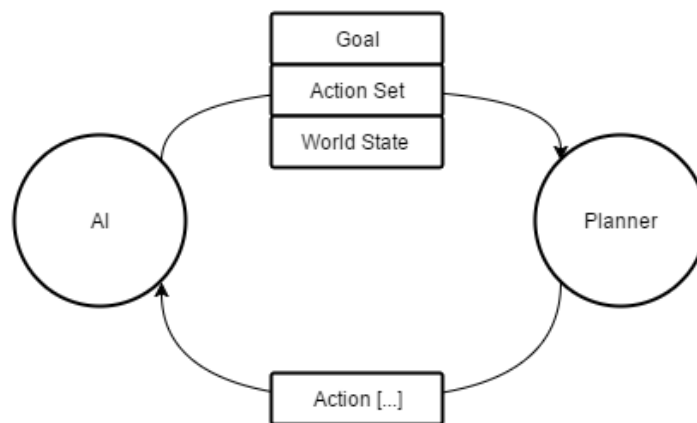
Tujuan dari pengembangan arsitektur ini adalah untuk menyederhanakan proses perencanaan kecerdasan buatan dalam permainan. Agen cerdas yang dibangun dalam permainan elektronik pada umumnya menggunakan *finite state machine* (FSM) [18] yang memerlukan perencanaan *state* dalam fase perencanaan permainan. Dalam sistem GOAP, logika *state transition* dipisahkan dari *state* sehingga perencanaan pembuatan logika agen diharapkan dapat menjadi lebih sederhana, seperti yang dapat dilihat pada Gambar 2.6. Dengan penyederhanaan ini, fleksibilitas dalam pengembangan kecerdasan buatan dalam permainan elektronik dapat ditingkatkan.



Gambar 2.6: Perbandingan antara State Machine (kiri) dan GOAP (kanan)

Sistem GOAP dibangun dengan tujuan untuk membuat keputusan yang menghasilkan rangkaian aksi (*plan*) untuk mencapai *goal state* yang telah ditentukan. Objek utama dalam GOAP adalah *goals* dan *actions*. *Goal* merupakan kondisi dunia yang ingin dicapai. Sementara *action* merupakan objek pengubah kondisi dunia yang memiliki syarat kondisi sebelum dapat dijalankan. *Plan* dari *action* dibuat oleh sebuah komponen yang disebut *planner*.

Setiap *action* memiliki *precondition* dan *effect* yang merupakan representasi dari *world state*. *Precondition* merupakan syarat *world state* untuk menjalankan sebuah *action* dan *effect* merupakan efek yang diberikan *action* tersebut terhadap *world* saat dijalankan. Gambar 2.7 merupakan ilustrasi hubungan antara objek-objek dalam arsitektur ini.



Gambar 2.7: Relationship class di GOAP

Input dari *planner* adalah kondisi saat ini, *goal* yang ditentukan serta himpunan *action* yang dapat dilakukan. Property dari *goal* merupakan bagian dari kondisi *world* yang akan membentuk sebuah kondisi yang dinamakan *goal world state*. *Plan* diformulasikan dengan melakukan pencarian pada *state space graph*.



Dalam GOAP, pencarian ini dilakukan menggunakan algoritma A-star (A\*) [19] untuk mendapatkan rangkaian *action* dengan nilai minimal yang dapat memproduksi *goal state* dari kondisi saat ini. State lain selain *goal state* pada kondisi yang dituju akan diabaikan di penelitian ini.

Sebagai contoh, untuk mengisi bahan bakar kendaraan di pompa swalayan, diperlukan interaksi antara pengemudi dan unit pompa pengisian bahan bakar. Kondisi yang dievaluasi pada kejadian ini adalah:

- ada pengemudi dalam mobil (*HasDriver*)
- jumlah bahan bakar dalam kendaraan penuh (*HasFuel*)
- ada yang mengisi bahan bakar (*HasPersonToRefuel*)

Sementara aksi yang dapat dilakukan dalam kejadian ini adalah:

- pengemudi bergerak ke pompa pengisian bahan bakar (*DriverToPump*)
  - precondition:
    - ada pengemudi dalam mobil
  - effect:
    - tidak ada pengemudi dalam mobil
    - ada yang mengisi bahan bakar
- pengemudi mengisi bahan bakar (*Refuel*)
  - precondition:
    - ada yang mengisi bahan bakar
  - effect:
    - jumlah bahan bakar dalam kendaraan penuh
- pengemudi kembali ke dalam kendaraan (*DriverToDriverSeat*)
  - precondition:
    - tak ada pengemudi dalam mobil
    - ada yang mengisi bahan bakar
  - effect:
    - ada pengemudi dalam mobil
    - tak ada yang mengisi bahan bakar

Dengan kondisi dan aksi tersebut, agen menganalisa *world state* dan mendapatkan hasil berikut:

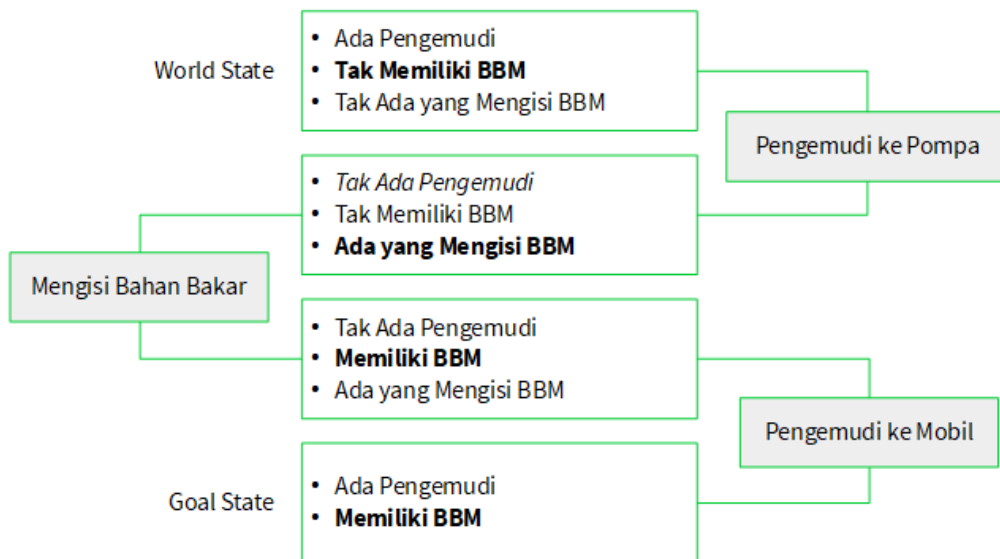
- Ada Pengemudi

- Tak Memiliki BBM
- Tak Ada yang Mengisi BBM

Dari *start condition* di atas, untuk dapat melanjutkan perjalanan kembali, agen membutuhkan *goal condition* sebagai berikut:

- Ada Pengemudi
- Memiliki BBM

*Plan* yang dibentuk oleh GOAP dalam contoh ini menghasilkan *set of actions* yang dapat dilihat pada Gambar 2.8.



Gambar 2.8: Contoh Aksi yang Dihasilkan GOAP

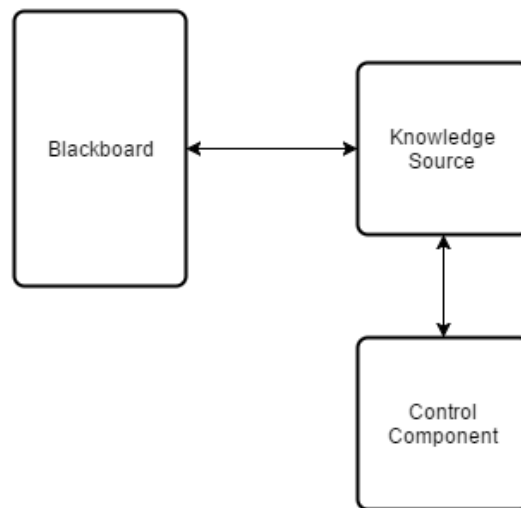
Keunggulan GOAP dalam penyederhanaan *state graph* memiliki beberapa kelemahan. Pertama, penyusunan *state graph* secara *online* membutuhkan sumber daya yang besar. Untuk mengantisipasinya, penggunaan layer untuk menyederhanakan pencarian dalam penyusunan *plan* [7] atau peningkatan efisiensi algoritma pencarian seperti yang dilakukan Ian Millington dan John Funge dengan menggunakan algoritma IDA\* [20] dapat dilakukan.

Kelemahan lainnya yang sering dihadapi adalah tidak mampunya sistem GOAP untuk mendeteksi apakah *plan* yang disusun berjalan sesuai rencana, berjalan diluar rencana ataupun telah gagal [6]. Dengan mengetahui kelemahan yang ada di atas, penelitian ini mencoba untuk mencari batasan dari penggunaan GOAP dengan menggunakan arsitektur ini pada permainan *real-time tactics*.

### 2.2.6 Blackboard System

Komunikasi antar agen cerdas dalam permainan membutuhkan sebuah sistem yang dapat menangani perpindahan informasi antar agen cerdas yang ada dalam permainan. Selain itu, perpindahan data di antara lapisan kecerdasan buatan yang dibangun juga memerlukan sebuah sistem yang dapat menangani perbedaan komposisi dan kebutuhan data.

Blackboard system dipilih untuk digunakan dalam penelitian ini dengan mengacu pada kemampuannya dalam menangani representasi *knowledge* yang beragam dan aktifitas *problem-solving* yang dinamis [21]. Komponen dasar dari sebuah sistem blackboard dapat dilihat di Gambar 2.9.



Gambar 2.9: Komponen dasar dari Blackboard System

Penjelasan dari komponen yang ada dalam sistem blackboard adalah sebagai berikut:

- *Knowledge Source*, merupakan sumber dari informasi yang disimpan di dalam *blackboard*.
- *Blackboard*, sebuah basis data yang berisi data observasi, hasil pengelolaan data yang masih parsial dan data lain yang digunakan dalam penyelesaian permasalahan.
- *Control component*, merupakan komponen yang membuat keputusan berdasarkan pada data yang ada pada *knowledge source* yang terkait.

Kelemahan GOAP yang berupa pendeteksian keberhasilan *plan* yang belum dapat ditangani diharapkan dapat diatasi dengan menggunakan sistem ini.

*Halaman ini sengaja dikosongkan*

## **BAB 3**

### **METODE PENELITIAN**

Porsi pengembangan dalam aplikasi yang dibangun pada penelitian ini adalah implementasi GOAP dalam permainan taktik yang dibangun diatas Engine Unity. Tujuan dari pengembangan adalah agar sistem yang digunakan dalam implementasi ini dapat dikembangkan lebih lanjut untuk menjadi produk yang dapat dilempar ke pasar. Platform diselesaikan dalam waktu 3 bulan termasuk arsitektur agen AI dan integrasi library GOAP pada aplikasi yang dibangun. Setelah itu, contoh kasus kondisi permainan dibuat untuk menguji adaptifitas agen cerdas yang dibangun.

#### **3.1 Sistem Permainan**

Permainan yang disusun sebagai landasan simulasi GOAP merupakan permainan *real-time tactics (RTT)* di luar angkasa. Alur permainan yang dibangun adalah observasi-aksi-relokasi. Dalam alur ini, prioritas permainan adalah mengobservasi semua lawan yang diketahui posisinya untuk mendapatkan detail data dari unit lawan tersebut. Setelah data unit lawan didapatkan, aksi yang sesuai diperintahkan pada unit yang ada di bawah kontrol tim.

Bagian-bagian sistem permainan yang dibangun agar dapat menjalankan alur tersebut dijabarkan dalam bagian ini.

##### **3.1.1 Konsep Dasar Permainan**

Permainan yang dibangun sebagai dasar uji coba arsitektur GOAP berbasis pada genre *real time tactics* di luar angkasa. Konsep dunia yang digunakan didesain dengan kondisi teknologi yang mendekati kondisi dunia saat ini. Dengan mengacu pada kondisi tersebut, teknologi yang digunakan mengambil pendekatan *hard science fiction*, dimana teknologi yang digunakan berkembang dengan dasar pengetahuan yang dimiliki oleh kondisi ilmu dan teknologi saat karya fiksi ilmiah tersebut dibuat [22]. Teknologi fiksi yang memiliki persyaratan teknologi yang belum dimiliki oleh teknologi saat ini seperti *warp* atau *exotic matter* tidak dimasukkan dalam pembuatan dunia yang mendasari permainan.

### **3.1.1.1 Kondisi Umum dalam Dunia Permainan**

Permainan memiliki setting tidak terlalu jauh di masa depan. Di masa tersebut, manusia sudah mulai mengambil sumber daya yang ada di planet Jupiter untuk kebutuhan energi mereka. Aktifitas pertambangan disana membutuhkan pemukiman bagi pekerja tambang dan keluarga serta layanan penunjang kehidupan mereka. Suplai dari Bumi masih dapat mencukupi di awal perkembangannya. Seiring berjalannya waktu dan bertambahnya populasi, kebutuhan serta keinginan untuk menopang kehidupan disana dengan kemampuan sendiri juga meningkat, dan kota-kota baru yang berasal dari kumpulan pemukiman awal mulai bermunculan.

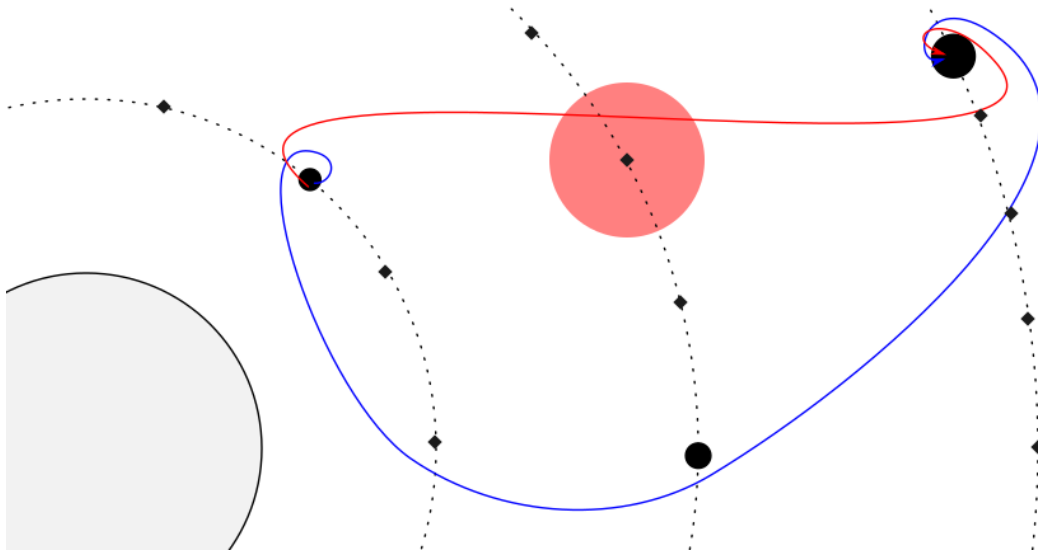
Perdagangan komoditas dari dan ke orbit Jupiter membutuhkan adanya jalur navigasi kapal luar angkasa. Untuk menunjang perjalanan dan menjaga keamanan, dibutuhkan juga pos-pos navigasi yang dibangun di antara orbit Bumi dan orbit Jupiter. Pos-pos navigasi ini dibangun dengan pola menyebar rata. Saat kapal atau armada ingin bergerak, rute dibangun dari poin satu ke poin lainnya sampai tiba di tujuan. Pola rute yang digunakan dapat diminta sesuai dengan kebutuhan perjalanan, seperti rute tercepat, rute yang memiliki pos-pos peristirahatan ataupun rute yang aman dari gangguan benda luar angkasa lain.

Banyaknya pos navigasi yang dibangun dan standar pola penggunaan rute yang rutin digunakan untuk navigasi mengakibatkan adanya pos-pos navigasi yang jarang dilalui lalu lintas kapal luar angkasa. Di tempat-tempat inilah konflik yang diusung permainan berlangsung.

### **3.1.1.2 Konflik yang Diusung dalam Permainan**

Pos navigasi yang jarang dilalui oleh lalu lintas kapal luar angkasa memberi kesempatan pada orang-orang yang tidak ingin berada di bawah aturan tertentu untuk tinggal disekitarnya. Walau patroli sering dilakukan untuk mencegah terjadinya hal-hal yang tidak diinginkan, jumlah pos navigasi yang sangat banyak menyebabkan jaminan keamanan tidak dapat dicapai dengan sempurna. Pada akhirnya, perencanaan rute perjalanan menjadi terbiasa untuk membuat rute yang dapat dijamin keamanannya, walaupun bukan rute tercepat untuk mencapai tujuan.

Walaupun terjamin keamanannya, rute umum yang ramai digunakan menyebabkan pergerakan kapal disana menjadi lebih lambat daripada rute yang tidak terlalu ramai. Pergerakan komoditas yang membutuhkan kecepatan pengiriman terkadang harus melewati lokasi yang memiliki jaminan keamanan yang lebih rendah daripada jalur lainnya. Kondisi ini memberi kesempatan pada mereka yang ingin memanfaatkannya untuk kepentingan sendiri. Gambar 3.1 merupakan ilustrasi dari rute dan sebaran pos navigasi dalam dunia permainan.



*Gambar 3.1: Ilustrasi Rute Perjalanan Kapal Luar Angkasa dalam Permainan*

Di Gambar 3.1, ilustrasi rute dimulai dari planet Bumi dan berakhir di planet Jupiter. Rute berwarna biru merupakan rute yang aman namun memiliki waktu tempuh yang lebih besar dari rute berwarna merah. Rute berwarna merah melalui pos navigasi yang memiliki tingkat keamanan yang rendah sehingga kemungkinan adanya tindakan yang tidak diinginkan lebih besar daripada dengan melalui rute berwarna biru. Area dengan tingkat keamanan rendah ini umum digunakan oleh pelaku pembajakan untuk melakukan aksinya. Permainan yang dibangun menggunakan kejadian pembajakan sebagai skenario utamanya.

### **3.1.1.3 Teknologi yang Dimiliki dalam Dunia Permainan**

Dunia fiksi yang dibangun sebagai latar belakang permainan berada tidak terlalu jauh di masa depan. Teknologi yang ada disana masih menggunakan

teknologi yang telah berada dalam kajian ilmiah saat ini. Kondisi teknologi yang menjadi dasar penyusunan strategi dalam permainan dijabarkan di bagian ini.

Manusia membutuhkan kondisi lingkungan yang mendukung aktifitas yang mereka lakukan. Aktifitas di luar angkasa memerlukan perlindungan bagi mereka yang melakukannya. Medan radiasi planet Jupiter yang sangat kuat [23] memaksa mereka yang beraktifitas disana harus berada di balik perlindungan dari radiasi.

Untuk melakukan aktifitas yang memerlukan manipulasi langsung, masyarakat pada zaman tersebut menggunakan *spacecraft* yang diberi *manipulator* untuk memegang dan memanipulasi objek-objek di luar angkasa. Para pekerja tetap berada di dalam ruangan yang dapat memberi perlindungan secukupnya dari terpaan radiasi yang dimiliki oleh planet Jupiter. Sementara pekerjaan yang perlu penanganan langsung oleh tangan manusia dibawa ke dalam area dengan perlindungan radiasi.

Gaya sentripetal digunakan sebagai satu-satunya teknik mensimulasikan gravitasi bumi di luar angkasa ataupun di objek luar angkasa dengan gaya gravitasi yang sangat lemah. Ukuran modul simulasi gravitasi untuk menjamin kenyamanan saat mensimulasikan gravitasi sebesar bumi sangat besar. Hanya kapal dengan jarak tempuh yang jauh, stasiun luar angkasa yang membutuhkan pekerjaannya untuk tinggal dalam waktu yang lama serta konstruksi pemukiman saja yang memilikinya. Unit-unit kapal lokal serta kendaraan kecil tidak memiliki kemampuan untuk membuat simulasi gravitasi sendiri. Kondisi ini menyebabkan kehidupan umum pada dunia yang dibangun berputar di sekitar konstruksi pemukiman atau stasiun luar angkasa saja, kecuali ada kondisi khusus yang menyertainya.

Transportasi komoditas di luar angkasa membutuhkan kendaraan. Tenaga penggerak yang digunakan disana disusun dengan anggapan tidak ada teknologi baru yang ditemukan selain yang telah mulai dikembangkan saat ini. Tidak ada perbedaan yang terlalu besar selain efisiensi dan daya yang lebih besar dari teknologi pendorong yang ada saat ini [24].

Untuk menjamin keselamatan perjalanan, sistem pendataan keberadaan kapal di luar angkasa juga telah dikembangkan di dunia yang dibangun. Kapal yang akan melakukan perjalanan mendaftarkan rutenya ke sistem umum, dan akan



terus melaporkan posisinya dalam periode tertentu kepada pos navigasi yang dilewatinya. Penanganan khusus dilakukan saat terjadi kondisi khusus seperti kapal yang keluar jalur ataupun hilangnya laporan posisi. Masing-masing pos navigasi juga memiliki sensor pelacakan untuk memastikan lancarnya perjalanan kapal-kapal yang melewatinya.

#### **3.1.1.4 Paradigma dalam Permainan**

Dari kajian yang dilakukan di atas, paradigma yang digunakan untuk membangun dunia permainan adalah:

1. Semua objek terlihat. Teknologi fiksi yang digunakan sebagai basis dari dunia permainan tidak memiliki kemampuan untuk menyembunyikan diri mereka. Secara teknis, paradigma ini diberikan untuk membuat pace permainan yang cepat karena tidak membutuhkan unit yang selalu bergerak mencari unit lawan yang tersembunyi.
2. Perlengkapan yang dimiliki objek permainan hanya dapat diketahui setelah melakukan observasi terhadapnya. Setiap tim akan mendapat seluruh data perlengkapan dari unit yang ada di dalamnya, namun untuk mendapatkan data perlengkapan unit lawan, perlu mengadakan observasi lebih lanjut.
3. Jumlah unit terbatas. Jumlah maksimum unit yang ada di dunia permainan merupakan kapasitas komando dari unit utama dan dapat ditentukan pada masing-masing skenario.
4. Unit dapat mengganti perlengkapannya. Sebagai pendukung kompleksitas permainan dan mengatasi terbatasnya jumlah unit, kemampuan untuk mengganti perlengkapan unit di kapal induk diberikan.

#### **3.1.2 Alur Permainan**

Alur observasi-serang-relokasi dipilih untuk menonjolkan adaptifitas agen cerdas yang akan dibangun. Di awal permainan, pemain diberikan sebuah kapal induk dengan kapasitas komando yang telah ditentukan dan dapat mengeluarkan unit pesawat untuk melakukan lebih banyak aksi terhadap dunia permainan. Pemain hanya dapat mengetahui detail dari armadanya sendiri dan armada yang dimiliki oleh kawan. Armada musuh sudah langsung terlihat oleh pemain, namun

perlengkapan apa yang dibawanya dan bagaimana menghadapinya harus diamati lebih dalam lagi. Penjelasan lebih lanjut tentang alur ini dibahas dalam bagian ini.

#### **3.1.2.1 Observasi**

Fase ini merupakan fase dimana unit musuh masih memiliki parameter yang belum diketahui. Jarak yang jauh dari unit lawan menyebabkan kaburnya informasi tentang perlengkapan apa yang dibawa oleh unit tersebut. Untuk memperoleh informasi tersebut, permainan mengarahkan untuk mengirimkan unit untuk mendeteksi perlengkapan yang dimiliki oleh unit lawan.

#### **3.1.2.2 Aksi**

Dalam fase ini, aksi untuk melemahkan atau menghancurkan unit musuh dilakukan. Unit kawan diperintahkan untuk melakukan aksi agresif terhadap sasaran yang ditentukan oleh pimpinan tim (pemain atau agen cerdas). Dalam bagian ini, pilihan unit apa yang dikeluarkan dan alokasi sasaran unit yang bersangkutan ditentukan.

#### **3.1.2.3 Realokasi**

Perubahan kondisi dalam dunia permainan membutuhkan alokasi ulang dari perlengkapan unit yang ada di dalam dunia permainan. Fase realokasi merupakan fase untuk mengalihkan penugasan agen-agen yang ada di bawah kendali agen cerdas dan mengganti perlengkapannya jika dibutuhkan. Perintah mengganti target serangan, navigasi sampai pergantian perlengkapan unit permainan ada di dalam alur ini.

### **3.1.3 Desain Permainan**

Untuk dapat mengeksposisi alur yang dijabarkan pada bagian sebelum ini, permainan dibangun dengan memberi batasan jumlah unit yang dapat berada di dalam dunia permainan dan memiliki kemampuan untuk mengganti perlengkapan yang dibawa. Masing-masing kelompok permainan memiliki satu unit kapal induk yang dapat menangani 4 unit pertahanan dan 5 unit penyerang yang perlengkapannya dapat diganti. Selain itu, jumlah perlengkapan yang ada dalam kapal induk juga dibatasi.

### 3.1.3.1 Parameter yang Dimiliki Unit dalam Permainan

Dasar dari nilai yang digunakan untuk implementasi logika ada pada parameter yang ada pada unit permainan. Parameter ini dibuat unik sesuai dengan unit dan perlengkapan masing-masing untuk menonjolkan dinamisme permainan. Masing-masing unit permainan memiliki parameter-parameter dasar berikut.

#### **Health**

Nilai representasi konsistensi struktur objek permainan. Memiliki nilai maksimum yang merupakan kondisi optimal objek, dan semakin rendah nilainya, semakin buruk kondisi objek permainan tersebut. Jika *health* bernilai 0, maka objek permainan dinyatakan hancur dan tidak dapat digunakan lagi dalam babak permainan yang berjalan.

#### **Attack**

Nilai kemampuan serangan yang dibawa oleh unit permainan. *Attack* merupakan nilai yang memiliki kemungkinan mengurangi *health* dari sasaran setelah perintah serang dieksekusi. Dalam permainan ini, nilai ini berasal dari total daya serang seluruh misil yang dibawa oleh unit yang memiliki parameter ini.

#### **Defense**

Untuk menghalau serangan yang dilakukan oleh lawan, dibutuhkan pertahanan. Nilai dari pertahanan masing-masing unit permainan masuk dalam nilai *defense* ini. Serangan laser yang dapat menghancurkan misil yang datang adalah dasar dari penghitungan nilai *defense*.

#### **Agility**

Selain *defense*, *agility* juga dapat mempengaruhi nilai serangan yang masuk ke unit permainan. Bedanya, kapabilitas yang ada disini merupakan nilai dari kecepatan unit permainan dan kemampuannya memutar badan unit permainan. Dengan gabungan kedua parameter tersebut, nilai *agility* ada untuk menjadi dasar perhitungan logika untuk menarik nilai kemampuan mencapai sasaran dan menghindar dari serangan musuh.

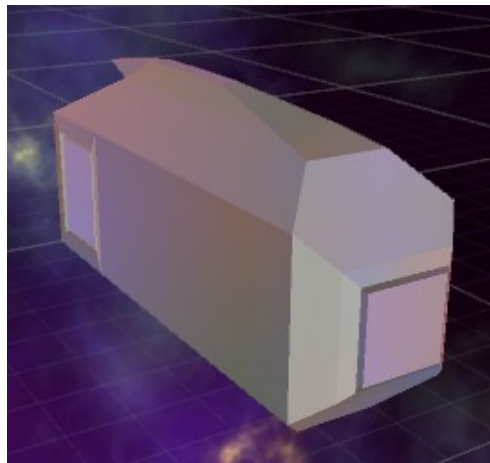
### 3.1.3.2 Tipe Unit

Unit yang ada dalam permainan dibangun dengan didasari 4 parameter yang ada diatas. Agar karakteristik masing-masing unit terlihat jelas, kemampuan yang dimiliki dibuat berbeda. Unit-unit yang dibangun untuk permainan yang dikembangkan adalah sebagai berikut.

#### **Kapal Induk / *Carrier Ship* (CS)**

Merupakan unit yang memiliki kemampuan untuk menyimpan pesawat dan memasang atau mengganti perlengkapan pesawat yang masuk di dalamnya. Selain itu, nilai *health* dari pesawat yang masuk di dalam unit ini juga dikembalikan menjadi nilai maksimal. Permainan akan berakhir jika kapal ini berhasil dihancurkan. Oleh karenanya, strategi dan taktik yang disusun merupakan perpaduan antara usaha untuk menghancurkan kapal induk lawan tanpa melupakan perlindungan terhadap kapal induk milik sendiri.

Unit ini memiliki nilai *health* terbesar dan memiliki komponen pertahanan dari serangan misil. Kapal induk ini bukanlah kapal yang digunakan untuk menyerang. Tidak ada komponen penyerang khusus yang dipasang sebagai perlengkapannya. Visualisasi unit kapal induk terlihat pada Gambar 3.2.



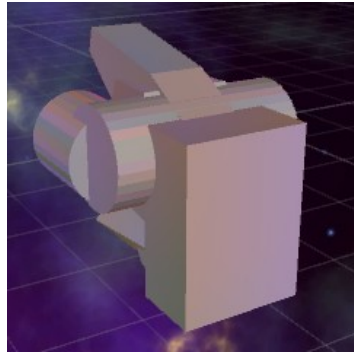
Gambar 3.2: Visualisasi Unit Kapal Induk

#### **Unit Pesawat / *Variable Pod* (VP)**

Merupakan unit dengan *agility* terbesar di dalam permainan. Unit yang kecil dan dapat disimpan dalam kapal induk. Memiliki kemampuan mengganti

perlengkapan untuk melakukan serangan terhadap lawan. Perlengkapan yang dapat dipasang pada unit ini disimpan dalam kapal induk.

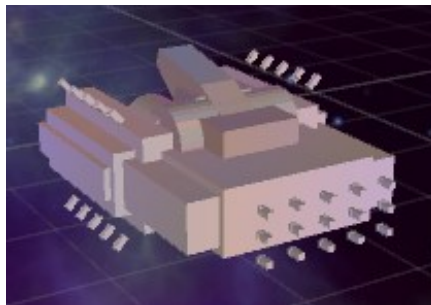
Unit ini merupakan unit serbaguna yang pada dasarnya merupakan unit untuk membawa kotak kontainer dan memiliki manipulator untuk memegang objek-objek di ruang angkasa. Dalam keadaan darurat, unit ini dapat diberi perlengkapan untuk mendukung misi yang diberikan oleh kapal induk. Visualisasi unit pesawat terlihat pada Gambar 3.3.



*Gambar 3.3: Visualisasi Unit Pesawat Sebelum Diberi Perlengkapan*

Dalam simulasi ini, perlengkapan yang dapat digunakan oleh unit pesawat adalah sebagai berikut:

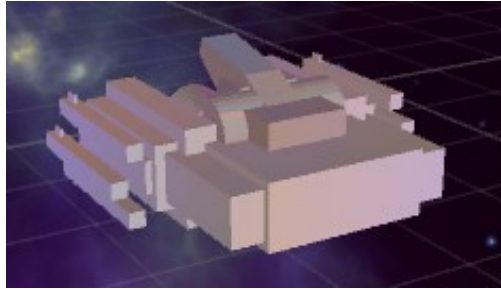
- *Micro Missile (M)* merupakan misil dengan daya hancur minimal, *agility* yang tinggi namun memiliki *health* terkecil. Digunakan untuk melakukan serangan terhadap unit pesawat lain. Visualisasi unit pesawat yang diberi perlengkapan yang membawa *micro missile* terlihat pada Gambar 3.4.



*Gambar 3.4: Unit Pesawat membawa Paket Micro Missile*

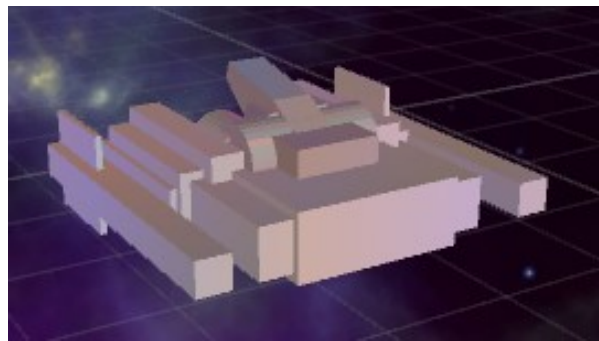
- *Cruise Missile (C)* merupakan misil dengan daya hancur sedang, *health* terbesar. Nilai *attack* yang dapat dilakukannya sama dengan *micro missile*

dan dapat digunakan untuk menembus pertahanan yang dikeluarkan unit lawan. Visualisasi unit pesawat yang diberi perlengkapan yang membawa *cruise missile* dapat dilihat pada Gambar 3.5.



*Gambar 3.5: Unit Pesawat membawa Paket Cruise Missile*

- *Torpedo (T)* misil dengan nilai *attack* terbesar namun memiliki *agility* yang jauh lebih kecil dari *micro missile* walaupun masih lebih besar dari *cruise missile*. Digunakan untuk menyerang kapal induk lawan. Visualisasi unit pesawat yang diberi perlengkapan yang membawa *torpedo* dapat dilihat pada Gambar 3.6.



*Gambar 3.6: Unit Pesawat membawa Paket Torpedo*

### **Unit Pertahanan / *Weapon Emplacement (WE)***

Merupakan unit yang ada untuk meningkatkan pertahanan di di sekitar kapal induk. Nilai serangan laser yang dapat ditembus oleh misil dengan tipe *cruise* menyebabkan unit ini menjadi sasaran pesawat yang membawa *cruise missile*.

#### **3.1.3.3 Parameter Unit dalam Permainan**

Unit-unit permainan yang dibuat membawa perlengkapan yang berbeda-beda. Parameter yang dibawa oleh masing-masing konfigurasi akan memberikan

variasi terhadap unit permainan pada setiap konfigurasi. Selain parameter dasar yang ada pada bagian 3.1.3.1, parameter lain yang dimiliki unit dari konfigurasi adalah sebagai berikut:

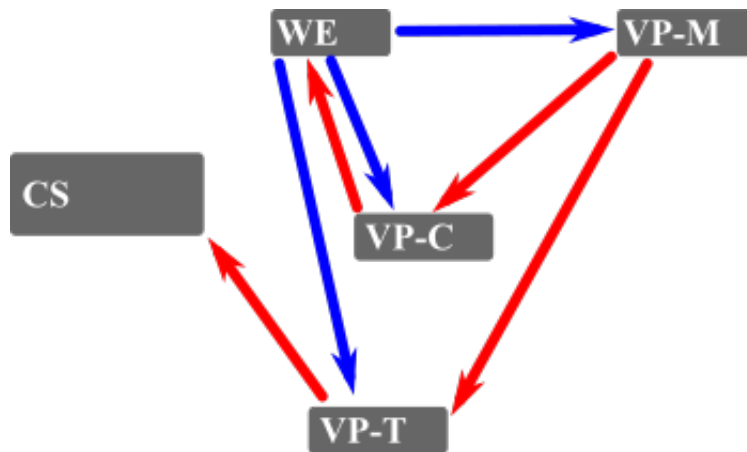
- *Hangar*, jumlah komponen yang dapat meluncurkan unit pesawat.
- Nilai serang per misil, merupakan nilai serangan sebuah misil yang dibawa oleh unit. Memiliki nilai nol jika unit tidak membawa misil.
- Ketahanan misil, besaran *health* yang dimiliki misil. Nilai ini berpengaruh pada kemampuan misil untuk menembus pertahanan yang dimiliki sasaran serangnya.
- Jumlah misil, jumlah misil yang dibawa oleh unit dengan parameter ini.
- Nilai serang per laser, merupakan nilai pertahanan yang dimiliki oleh unit. Masing-masing unit dapat membawa lebih dari satu komponen laser. Nilai serangan per komponen disimpan pada parameter ini.
- Jumlah laser, merupakan jumlah komponen laser yang dibawa oleh unit. Masing-masing unit membawa lebih dari satu komponen laser untuk mempersiapkan diri dari rentetan serangan misil.

Data di Tabel 3.1 merupakan nilai dari parameter yang dimiliki oleh masing-masing konfigurasi unit.

		Health	Hangar	Nilai Serang Per Misil	Ketahanan Misil	Jumlah Misil	Nilai Serang Per Laser	Jumlah Laser
Unit Pesawat	Micro	100	0	10	0.005	500	0.001	10
	Cruise			10	1	6		
	Torpedo			2000	0.1	2		
Unit Pertahanan		10	0	0	0	0	0.005	50
Unit Kapal Induk		10000	1	0	0	0	0.005	60

*Tabel 3.1: Parameter Unit Permainan*

Parameter unit-unit diatas disusun untuk membentuk komposisi gunting-batu-kertas seperti yang terlihat pada Gambar 3.7.



Gambar 3.7: Unit dalam Permainan

Panah yang tertera pada Gambar 3.7 merupakan representasi efektivitas unit terhadap unit lainnya. Panah merah memiliki arti unit tersebut baik digunakan untuk melawan arah panah, sedangkan warna biru merepresentasikan unit tersebut cocok untuk menghadang serangan dari unit yang dituju oleh panah.

### 3.1.3.4 Komponen yang Dibuat Untuk Unit Permainan

Dalam tesis ini, penulis mencoba untuk membuat unit permainan dengan komponen *generic* yang dapat dipasangkan pada unit yang berbeda-beda. Variasi unit didapatkan dari kombinasi komponen dan parameter yang dimilikinya. Komponen-komponen yang dibuat untuk unit dalam penelitian ini adalah sebagai berikut.

#### Komponen Observasi

Komponen ini mensimulasikan sistem sensor dari unit permainan. Seperti nama yang dimilikinya, tujuan dari komponen ini adalah untuk mengobservasi unit yang ada di dunia permainan.

Unit yang ada diluar jarak efektif komponen ini hanya diketahui keberadaannya saja, namun belum dapat diketahui komponen apa yang dia miliki. Setelah observasi dilakukan, perubahan informasi dari unit yang terdeteksi akan disimpan oleh komponen yang menangani logika kelompok unit untuk diteruskan ke kontrol pemain ataupun agen cerdas.



## **Komponen Penggerak**

Untuk dapat melakukan observasi, unit harus memiliki kemampuan untuk bergerak. Komponen ini dibangun dengan memanfaatkan pergerakan dinamik yang dimiliki *physics engine* Unity3D. Hal ini disebabkan karena dalam iterasi pertama, pergerakan secara kinetik dengan mengubah langsung posisi unit memiliki akurasi yang lebih rendah saat melakukan pergerakan dengan kecepatan tinggi.

Adapun tipe komponen yang dibuat untuk pergerakan adalah sebagai berikut:

1. Komponen pendorong, merupakan komponen yang dibuat untuk menggerakkan unit ke arah tertentu. Dalam permainan ini, arah dorongan komponen hanya ditujukan ke arah depan unit.
2. Komponen perotasi, dibuat untuk mengubah arah unit permainan. Dengan komponen ini, unit permainan dapat merubah arah hadapnya sehingga pergerakan unit dapat dikendalikan lebih baik lagi.

## **Komponen Serangan**

Komponen serangan merupakan komponen yang dipergunakan untuk meluncurkan serangan untuk mengalahkan unit lawan. Dalam permainan ini, tipe serangan yang digunakan adalah serangan dengan menggunakan misil.

Komponen ini akan menunggu perintah untuk meluncurkan misil pada sasaran yang ditentukan.

## **Komponen Pertahanan**

Komponen pertahanan dibuat untuk menanggulangi serangan lawan. Di komponen ini, serangan bertipe laser dengan nilai serang yang sangat kecil jika dibandingkan dengan jumlah nyawa dari unit permainan, namun cukup untuk menghancurkan misil digunakan.

Komponen ini akan secara otomatis menjadikan unit dan misil musuh yang ada di dalam area observasinya sebagai sasaran, dengan prioritas berupa objek terdekat yang akan dijadikan sasaran utama.

## **Komponen Logika Unit**

Agar komponen-komponen sebelumnya dapat berjalan, perlu diterapkan logika per unitnya. Tujuan dari komponen ini adalah agar perintah sederhana yang diberikan pada unit mampu dilaksanakan. Penjelasan lebih lanjut tentang komponen ini akan dibahas pada bagian berikutnya.

### **3.1.4 Logika Unit**

Komponen logika unit merupakan set dasar untuk mensinkronkan komponen-komponen unit dalam melakukan tugasnya. Logika dasar yang dimiliki oleh seluruh tipe unit akan dikumpulkan pada sebuah abstraksi agar dapat diturunkan untuk dikembangkan lebih lanjut per unitnya.

Komunikasi antar komponen juga dikelola di komponen logika ini. Hasil dari observasi dan parameter yang dikirimkan ke masing-masing komponen melalui sistem blackboard dilakukan disini.

#### **3.1.4.1 Dasar Logika Unit**

Dalam menggunakan komponen-komponennya, unit dalam permainan memiliki logika dasar yang dilakukan oleh semuanya. Untuk menghindari kompleksitas, logika dasar tersebut disatukan dalam sebuah abstraksi yang akan diturunkan pada masing-masing implementasi logika.

Logika dasar dari seluruh unit permainan adalah sebagai berikut:

1. Memperbarui observasi, dengan mengambil data observasi terkini yang dimiliki oleh tim. Unit lawan diurutkan berdasarkan pada jarak, nilai serangan yang dimiliki dan ada berapa unit kawan yang menjadikan unit lawan yang bersangkutan sebagai sasaran.
2. Melakukan analisa pertahanan, seluruh unit lawan termasuk misil yang ada di dalam jarak observasi dialokasikan pada komponen pertahanan yang dimiliki unit.
3. Melakukan analisa serangan, unit lawan yang dijadikan sasaran dicek apakah sudah dalam jarak serang optimal. Setelah diuji, aksi yang dilakukan ada di antara:
  1. Mendekati sasaran jika posisi unit berada di luar jarak serang optimal.

2. Meluncurkan serangan saat memasuki jarak serang optimal.

Pengujian jarak serang optimal dilakukan dengan menggunakan fungsi berikut:

$$d = (v_0 \times (\frac{msHlt}{enDef})) + (msAcl \times (\frac{msHlt}{enDef})^2) \quad (3.1)$$

$d$  : jarak serang optimal misil

$v_0$  : kecepatan relatif awal terhadap sasaran saat misil diluncurkan

$msHlt$  : *health* yang dimiliki misil

$msAcl$  : besaran akselerasi yang dimiliki misil

$enDef$  : nilai pertahanan lawan yang diserang

#### 3.1.4.2 Unit Pesawat

Unit pesawat memperluas logika dasar yang dijabarkan di atas. Pada logika untuk unit pesawat ditambahkan parameter agresivitas. Ada tiga tipe agresivitas yang dimasukkan pada unit permainan yaitu:

1. Aktif. Di tingkat agresivitas ini, unit akan secara aktif mencari sasaran terdekat saat sasaran yang ditentukan pemain atau agen cerdas telah berhasil dikalahkan.
2. Pasif. Disini, saat sasaran yang ditentukan berhasil dikalahkan, unit pesawat akan kembali ke kapal induk sampai diberikan perintah baru.
3. Kembali ke hangar, unit kembali ke kapal induk walaupun sasaran yang ditentukan masih aktif. Tingkat agresivitas ini digunakan untuk mengembalikan unit yang jenis senjatanya tidak cocok atau nilai *health*nya kecil.

#### 3.1.4.3 Unit Kapal Induk

Tidak ada perbedaan mendasar antara logika unit kapal induk dengan logika dasar unit permainan. Disini hanya menambahkan logika untuk meluncurkan unit yang dipilih oleh pemain atau agen cerdas dan mengarahkan sasaran ke kapal induk lawan atau unit terdekat jika semua unit yang dimiliki sudah habis. Ini dilakukan untuk mempercepat jalannya permainan.

### 3.1.5 Abstraksi Kontrol Unit

Unit-unit yang dibuat pada permainan ini memiliki aksi yang berbeda-beda tergantung tipenya. Agar aksi dapat dipanggil oleh kontrol permainan seperti agen cerdas ataupun pemain, sebuah standar untuk memanggil aksi-aksi unit dibuat.

Alur dasar dari abstraksi kontrol unit adalah dengan mengumpulkan komponen-komponen yang akan digunakan untuk diberikan perintah sesuai dengan tujuan kontrol. Aksi yang dibuat adalah sebagai berikut:

#### 3.1.5.1 Navigasi

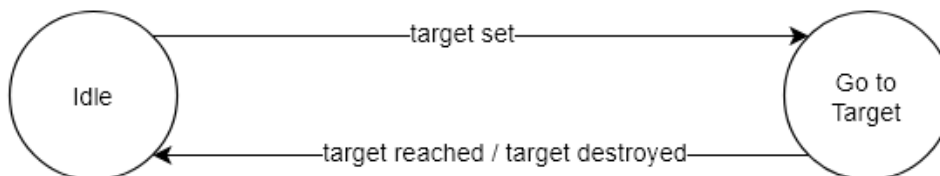
Merupakan aksi dasar yang dimiliki setiap unit dalam permainan. Aksi ini mengatur pergerakan unit dalam permainan. Sebelum aksi ini dieksekusi, pemain atau agen cerdas perlu mengalokasikan sasaran pergerakan. Setelahnya, tipe gerakan pendekatan dipilih. Disini ada dua tipe pergerakan yang dapat dilakukan terhadap serangan, yaitu:

1. Mendekati sasaran, unit akan mendekati sasaran secara langsung dengan arah lurus.
2. Mengitari sasaran, unit akan mendekati sasaran sambil mengitarinya.

Setelah tipe pergerakan dipilih, jarak dari tujuan navigasi ditetapkan. Unit akan berhenti dari sasaran sejauh jarak yang ditentukan jika pilihan tipe pergerakan adalah mendekati, dan menggunakan parameter ini sebagai jari-jari orbit jika tipe gerakan yang dipilih adalah mengitari sasaran.

Selain itu, jika sasaran merupakan hangar dari kapal induk kawan, saat unit sudah dalam jarak tertentu, unit akan dimasukkan dalam hangar dari kapal induk.

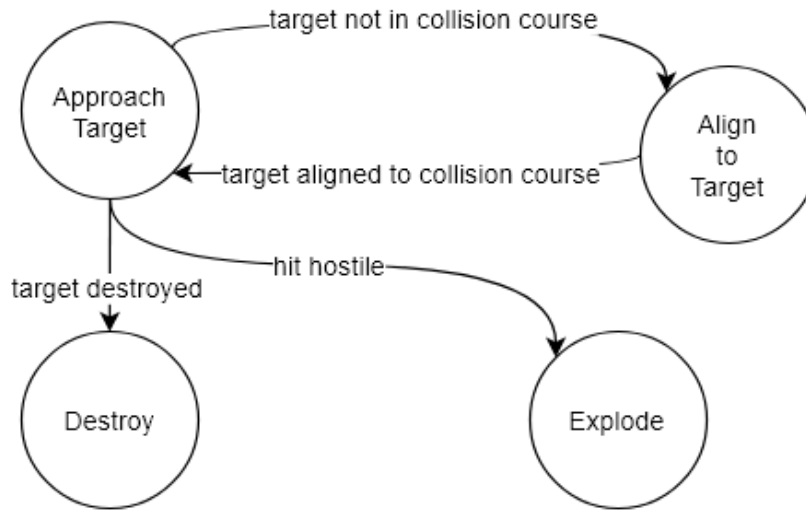
Gambar 3.8 merupakan FSM dari logika pergerakan unit.



Gambar 3.8: Pergerakan Unit

Misil memiliki dasar pergerakan yang hampir sama dengan unit kapal biasa. Perbedaannya terletak pada tujuan dari pergerakannya. Jika unit kapal bergerak

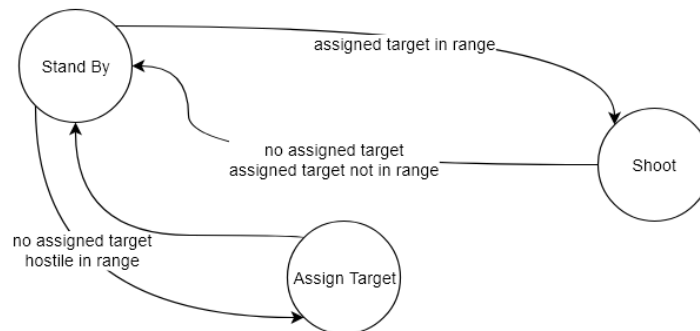
untuk mencapai sasaran dan melakukan aksi terhadap sasaran navigasi, misil memiliki tujuan untuk menabrakkan dirinya pada objek yang menjadi sasaran dari pergerakannya. FSM dari logika pergerakan misil dapat dilihat pada Gambar 3.9.



Gambar 3.9: Pergerakan Misil

### 3.1.5.2 Serangan

Merupakan aksi yang dimiliki unit yang memiliki modul serangan misil. Hampir serupa dengan aksi navigasi, dalam serangan ini pemain atau agen perlu menentukan sasaran aksi terlebih dahulu. Jika sasaran telah ditentukan, modul serangan akan menguji apakah sasaran ada dalam jarak serang optimalnya. Modul navigasi akan aktif saat sasaran berada diluar jarak tembak untuk mendekatinya. FSM dari modul serangan terlihat pada Gambar 3.10.



Gambar 3.10: FSM untuk Serangan Unit

### 3.1.5.3 Hangar

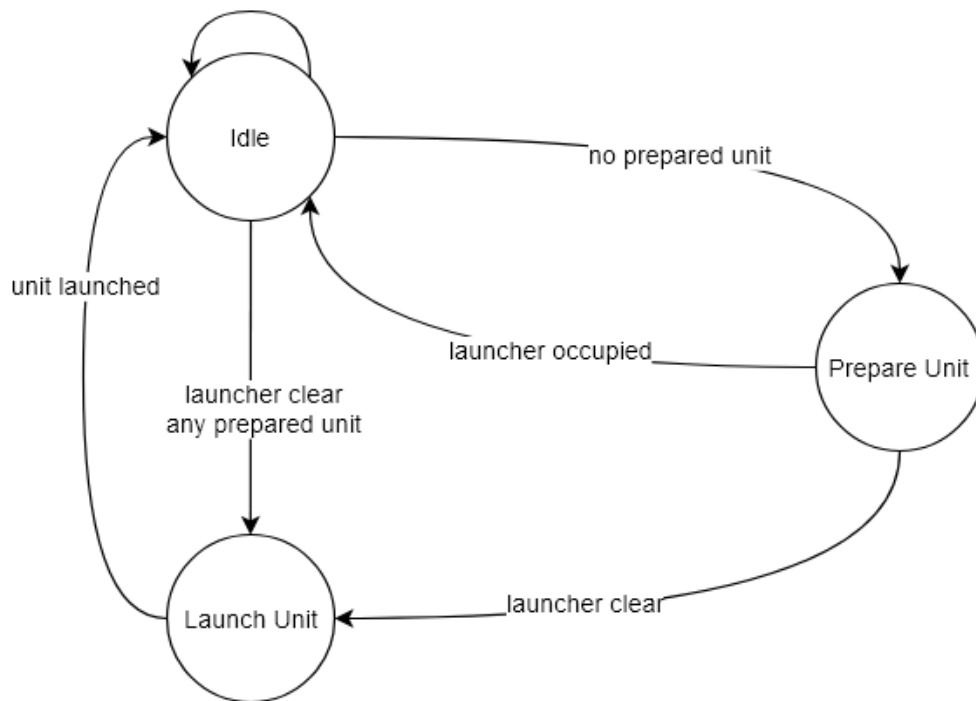
Unit pesawat diluncurkan berdasarkan perintah yang diberi oleh pemain atau agen cerdas dari hangar yang dimiliki oleh kapal induk. Perintah untuk meluncurkan unit membutuhkan parameter

Untuk meluncurkan unit pesawat dari hangar dibutuhkan kondisi berikut:

1. Ketersediaan unit inti pesawat.
2. Ketersediaan perlengkapan pesawat yang sesuai dengan pilihan pemain atau agen cerdas.

Saat kondisi di atas terpenuhi dan eksekusi aksi dilakukan, jika masih ada antrian peluncuran unit, unit yang baru saja disusun akan dimasukkan dalam antrian peluncuran.

FSM untuk aksi yang dapat dilakukan bagian ini dapat dilihat pada Gambar 3.11.



Gambar 3.11: FSM untuk Hangar Kapal Induk (CV)

### 3.1.6 Fungsi Pendukung

Standar pembobotan unit yang sama ditentukan agar agen cerdas yang dibangun dapat dibandingkan. Untuk itu fungsi-fungsi independen dibuat untuk mendukung agen cerdas yang diimplementasikan pada sistem permainan.

Fungsi yang dibuat untuk mendukung implementasi logika adalah sebagai berikut.

#### GetWeight

Fungsi untuk mengambil bobot dari unit permainan terhadap sasaran yang ditentukan. Bobot yang dikeluarkan oleh fungsi ini adalah nilai perbandingan antara kapasitas serangan unit terhadap pertahanan yang dimiliki sasaran. Fungsi ini memiliki notasi sebagai berikut:

$$w = \begin{cases} 0 & dpm - wf > 1 \\ 1 & dpm - wf \leq 1 \end{cases} \times \begin{cases} (W_A \times atkDist) + dpm & pdu - wf > 1 \\ atkDist + (W_B \times dpm) & pdu - wf \leq 1 \end{cases} \quad (3.2)$$

- $w$  : bobot unit permainan terhadap sasaran yang ditentukan.
- $dpm$  : *Damage Per Missile*, perbandingan antara nilai serangan dari tiap misil yang dimiliki unit terhadap sasaran serangan. Jika nilai serangan misil sama dengan sisa *health* yang dimiliki sasaran serang, variabel ini akan bernilai satu.
- $wf$  : faktor dasar pembobotan unit. Faktor ini digunakan untuk mencegah penggunaan misil yang nilai serangannya jauh di atas sisa *health* dari sasaran serangan.
- $W_A$  : konstanta bobot untuk unit lawan yang dapat dikalahkan dengan seluruh potensial serangan yang dimiliki unit yang menyerang.
- $W_B$  : konstanta bobot untuk unit lawan yang tidak dapat dikalahkan dengan seluruh potensial serangan yang dimiliki unit yang menyerang.
- $atkDist$  : *Attack Distance*, jarak serang maksimum setelah mempertimbangkan pertahanan yang dimiliki sasaran.
- $pdu$  : *Potential Damage Unit*, nilai potensial serangan terhadap sasaran. Merupakan total nilai serangan yang dapat dilakukan oleh unit terhadap sasaran. Jika total serangan yang dimiliki unit sama dengan

sisanya *health* yang dimiliki sasaran serang, variabel ini akan bernilai satu.

### GetAttackDistance

Mengambil jarak serangan. Nilai yang dikeluarkan oleh fungsi merupakan perbandingan antara jarak serangan maksimum dan jarak optimal untuk menyerang setelah memperhitungkan nilai pertahanan serangan. Fungsi ini memiliki notasi sebagai berikut:

$$atkDist = \min\{1, (mslHlt \div enDef) \times mslSpd \div rdrRng\} \quad (3.3)$$

*atkDist* : jarak serang maksimum yang dapat dilakukan, bernilai satu jika jarak serang sama dengan jangkauan radar.

*mslHlt* : nilai *health* yang dimiliki misil.

*enDef* : nilai pertahanan yang dimiliki unit lawan.

*mslSpd* : kecepatan misil

*rdrRng* : jarak jangkauan radar yang dimiliki unit.

### FindAppropriateTarget

Fungsi untuk mencari target yang cocok bagi unit lawan yang dimasukkan dalam parameter. Seluruh unit lawan yang berada dalam jangkauan serangan dimasukkan dalam fungsi ini dan unit dengan bobot tertinggi akan dijadikan sasaran serang. Didapatkan dengan fungsi berikut:

$$target = \max \left\{ w + \begin{cases} 3 & targetCheck = VP-T \\ 2 & targetCheck = VP-M \\ 1 & targetCheck = VP-C \end{cases} \right\} \quad (3.4)$$

*target* : unit lawan terpilih

*w* : bobot unit permainan terhadap sasaran yang sedang dicek

*targetCheck* : unit lawan yang sedang dicek

### ArrangeHostile

Mengurutkan unit lawan dengan prioritas yang disusun dengan urutan sebagai berikut:

1. Unit Pesawat, diurutkan berdasarkan misil yang dibawanya, yaitu:



- a. Membawa Torpedo, unit lawan yang membawa torpedo memiliki prioritas tertinggi untuk diserang karena tanpa kapal induk, pengkonfigurasi ulang unit tidak dapat dilakukan.
  - b. Membawa Cruise Missile, unit ini merupakan prioritas serangan berikutnya karena unit ini menyerang unit pertahanan. Tanpa adanya unit tersebut, serangan lawan akan lebih mudah untuk mencapai kapal induk.
  - c. Membawa Micro Missile, unit ini menyerang unit pesawat lain. Oleh karenanya, unit yang membawa micro missile harus dihilangkan dari dunia permainan agar unit penyerang dapat menjalankan misinya.
2. Unit Pertahanan, perlu dihancurkan agar serangan terhadap kapal induk dapat dilakukan dari jarak aman.
  3. Unit Kapal Induk, diletakkan sebagai prioritas terakhir karena serangan langsung ke kapal induk dengan mengabaikan pertahanan yang dimilikinya dapat membalikkan keadaan permainan dengan cepat.

### **FindBestObserver**

Mencari unit di stage yang cocok untuk melakukan observasi. Mengambil unit yang memiliki nilai serangan terendah di *stage*.

### **3.2 Goal Oriented Action Planning**

Dengan berdasarkan pada desain sistem permainan yang dijelaskan pada bagian 3.1, GOAP dibuat dengan hierarki dan level yang ditangani oleh agen cerdas berbasis GOAP ada pada taktik umum dari permainan. *Goal* dari perencanaan disini adalah untuk memaksimalkan alokasi unit pada musuh yang terobservasi. Saat ada musuh yang belum teramati, kirim *scout unit* terlebih dahulu. Perencanaan level dua adalah tentang bagaimana mengalokasikan unit yang ada di stage dan apakah perlu mengeluarkan unit baru dari hangar atau mengganti perlengkapan dari *core unit* yang ada di dalam permainan.

### 3.2.1 Strategi Umum

Aksi yang disusun oleh GOAP disusun dengan mengan menggunakan fungsi perbandingan yang melihat nilai *attack*, *defense* dan *agility* dari unit-unit yang ada di dalam permainan. Aksi yang dilakukan terbagi atas:

1. Meluncurkan unit, dengan melakukan konfigurasi unit pesawat dan meluncurkannya.
2. Melakukan observasi. Mengalokasikan unit yang berada di dunia permainan untuk mendeteksi perlengkapan yang dibawa oleh unit lawan.
3. Mengalokasikan unit. Melakukan alokasi terhadap unit yang ada di dalam dunia permainan.
4. Menarik kembali unit. Saat unit yang ada di dunia permainan memiliki nilai potensial serangan yang lebih rendah daripada melakukan konfigurasi ulang, agen GOAP akan menarik unit yang memiliki nilai potensial rendah kembali ke kapal induk untuk dikonfigurasi ulang.

### 3.2.2 World State

World state merupakan representasi nilai dari kondisi dunia permainan yang akan dievaluasi untuk membuat rencana aksi. World state yang digunakan pada penelitian ini adalah sebagai berikut:

- *HostileUnobserved*, merepresentasikan jumlah unit musuh yang belum diobservasi.
- *HostileObservedAssigned*, merupakan jumlah unit musuh yang sudah terobservasi dan ada unit kawan yang dialokasikan padanya.
- *HostileObservedUnassigned*, jumlah unit musuh yang sudah terobservasi dan tidak ada unit kawan yang dialokasikan padanya.
- *FriendlyVPHangar*, jumlah unit yang ada dalam hangar.
- *FriendlyVPStageAssigned*, jumlah unit yang ada di stage permainan dan sudah dialokasikan pada unit lain.
- *FriendlyVPStageUnassigned*, jumlah unit yang ada di stage permainan dan belum dialokasikan pada unit lain.

### 3.2.3 Actions

*Actions* yang dikembangkan dalam penelitian ini memiliki tujuan untuk memaksimalkan alokasi unit-unit dalam kelompok terhadap unit lawan. Masing-masing unit kawan yang ada di *stage* permainan dievaluasi dan ditentukan sasarannya sesuai dengan kapabilitas masing-masing.

*Action* yang dikembangkan adalah sebagai berikut:

1. Observasi Lawan, action yang mengambil satu unit di stage untuk melakukan observasi terhadap lawan yang belum terobservasi.
  - *Precondition*
    1. Ada unit lawan yang belum diobservasi di dunia permainan.
    2. Ada unit pesawat kawan yang belum dialokasikan.
  - *Effect*
    1. Unit lawan terobservasi.
    2. Unit pesawat kawan yang belum dialokasikan berkurang satu.
    3. Unit pesawat kawan yang telah dialokasikan bertambah satu.
2. Alokasi Unit, action yang mengambil unit yang belum memiliki sasaran untuk diberi sasaran aksi.
  - *Precondition*
    1. Ada unit pesawat kawan yang belum dialokasikan.
    2. Ada unit lawan yang telah diobservasi dan belum dijadikan sasaran serang unit pesawat kawan.
  - *Effect*
    1. Jumlah pesawat kawan yang telah dialokasikan bertambah. Jumlahnya adalah nilai minimum antara jumlah pesawat kawan yang belum dialokasikan atau jumlah lawan yang telah diobservasi dan belum dijadikan sasaran serang.
    2. Jumlah pesawat kawan yang belum dialokasikan berkurang. Jumlahnya adalah nilai minimum antara jumlah pesawat kawan yang belum dialokasikan atau jumlah lawan yang telah diobservasi dan belum dijadikan sasaran serang.
    3. Jumlah pesawat lawan yang telah dijadikan sasaran serang

- bertambah sebesar nilai minimum antara jumlah pesawat kawan yang belum dialokasikan atau jumlah lawan yang telah diobservasi dan belum dijadikan sasaran serang.
4. Jumlah pesawat lawan yang belum dijadikan sasaran serang berkurang sebesar nilai minimum antara jumlah pesawat kawan yang belum dialokasikan atau jumlah lawan yang telah diobservasi dan belum dijadikan sasaran serang.
3. Dealokasi Unit, action untuk membebaskan unit dari alokasi sasaran jika ada unit lawan yang lebih perlu dialokasikan.
    - *Precondition*
      1. Ada unit pesawat kawan yang telah dialokasikan.
      2. Ada unit lawan yang belum dijadikan sasaran serang.
      3. Tidak ada unit kawan yang ada di dalam hanggar.
      4. Tidak ada unit kawan yang belum dialokasikan dalam dunia permainan.
    - *Effect*
      1. Jumlah unit kawan yang tidak teralokasikan bertambah sejumlah nilai minimal antara jumlah unit kawan yang teralokasikan atau jumlah unit kawan yang belum dijadikan sasaran serangan.
      2. Jumlah unit kawan yang teralokasikan berkurang sejumlah nilai minimal antara jumlah unit kawan yang teralokasikan atau jumlah unit kawan yang belum dijadikan sasaran serangan.
  4. Luncurkan Unit Penyerang, meluncurkan unit penyerang agar dapat dialokasikan ke lawan.
    - *Precondition*
      1. Ada unit kawan yang berada di hanggar.
      2. Ada unit lawan di dunia permainan.
    - *Effect*
      1. Jumlah unit pesawat kawan yang ada di dalam hanggar berkurang sejumlah nilai minimal antara jumlah unit kawan yang ada dalam hanggar atau total unit lawan yang ada dalam dunia

permainan.

2. Jumlah unit kawan yang belum dialokasikan bertambah sejumlah nilai minimal antara jumlah unit kawan yang ada dalam hanggar atau total unit lawan yang ada dalam dunia permainan.
5. Luncurkan Unit Observasi, meluncurkan unit untuk melakukan observasi.
- *Precondition*
    1. Ada unit yang berada di dalam hanggar.
    2. Tidak ada unit kawan yang berada di dunia permainan.
  - *Effect*
    1. Jumlah unit kawan yang berada dalam hanggar berkurang satu.
    2. Jumlah unit kawan yang belum dialokasikan di dunia permainan bertambah satu.

#### **3.2.4 Planner**

Planner yang digunakan dalam penelitian ini mengambil dari library yang telah ada. Planner ini mampu mengakomodasi kebutuhan implementasi GOAP dalam penelitian ini. Alur penggunaannya adalah sebagai berikut:

##### **Buat Planner**

Disini *actions* yang digunakan pada GOAP dideklarasikan. Selain itu fungsi komparasi dan node search dideklarasikan disini juga.

##### **Ambil World State Saat Ini**

Setelah *actions*, fungsi komparasi dan fungsi search dideklarasikan, ambil kondisi *world* saat ini untuk dijadikan dasar penyusunan rencana.

##### **Tentukan Goal State**

Dalam bagian ini goal state dari perencanaan dideklarasikan. Untuk penelitian ini, goal state yang ditentukan adalah sebagai berikut:

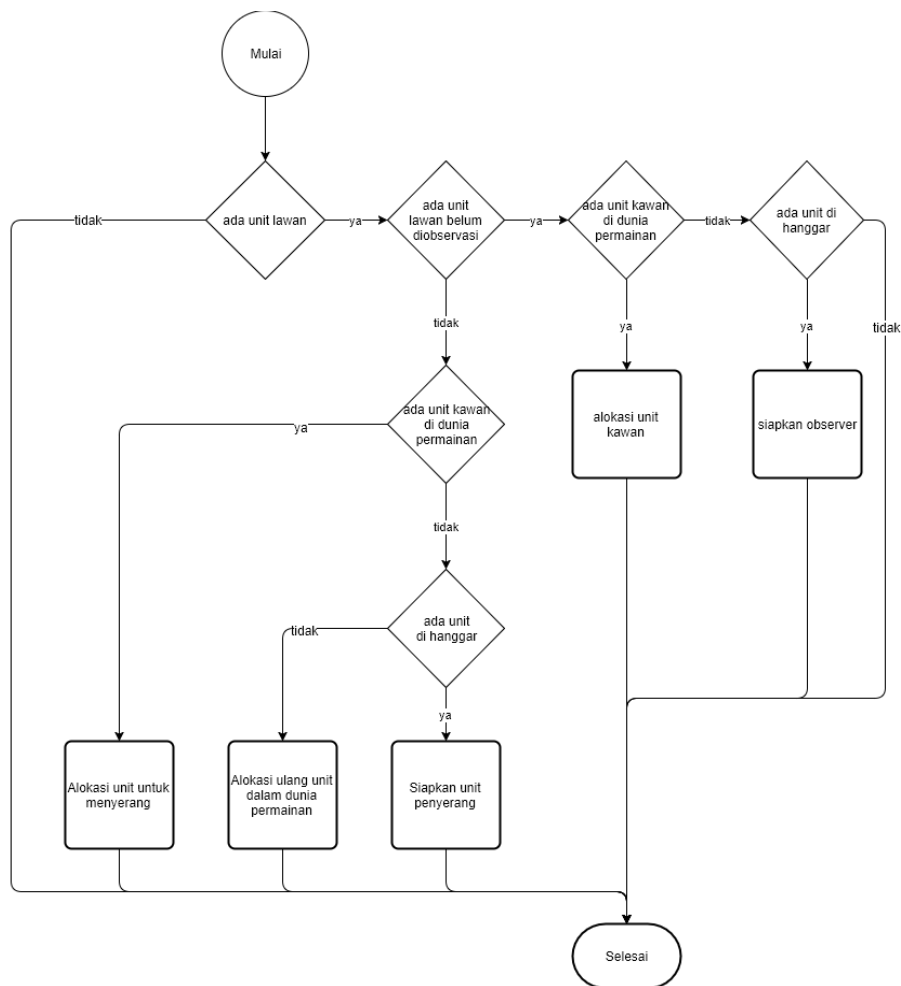
1. Tidak ada unit lawan yang belum terobservasi.
2. Tidak ada unit lawan yang terobservasi.

### Jalankan Planner untuk Mendapatkan Plan

Dengan menggunakan goal state diatas, perencanaan dilakukan. Saat menjalankan planner ini, proses penataan rencana oleh GOAP dilakukan.

### 3.3 Rule-Based AI sebagai Pembanding

Untuk menguji kemampuan agen GOAP yang dibangun, sebuah agen dengan *Rule-Based* AI dibangun. Agen ini dibangun dengan menggunakan fungsi pembobotan yang sama dengan yang digunakan oleh agen GOAP. Dalam *Rule-Based* AI, fungsi-fungsi pembobotan yang ada dihubungkan dengan logika sederhana yang disusun seperti *flow chart* yang dapat dilihat pada Gambar 3.12.



Gambar 3.12: Flow Chart Kecerdasan Buatan Rule-Based AI

## **BAB 4**

### **HASIL DAN PEMBAHASAN**

Metodologi yang dijabarkan pada bab 3 diujicobakan pada bab ini. Ujicoba dilakukan dengan melakukan *running* program dengan skenario yang ditentukan. Parameter yang dievaluasi adalah komposisi dari unit-unit yang ada di dalam dunia permainan.

#### **4.1 Spesifikasi Unit Pengujian**

Spesifikasi hardware dan software untuk menjalankan penelitian ini adalah sebagai berikut :

- Processor : Intel(R) Core i5-2430M CPU @2.40Ghz
- Memory : 8GB
- Harddisk : TOSHIBA MK6476GSX - 596.17 GB
- Sistem Operasi : Windows 10 Pro 64-bit

#### **4.2 Hasil**

Hasil pengujian dalam penelitian ini dilakukan dengan menjalankan permainan yang dibangun dalam pengerjaan penelitian ini. Visualisasi dari permainan tersebut nampak pada Gambar 4.1.

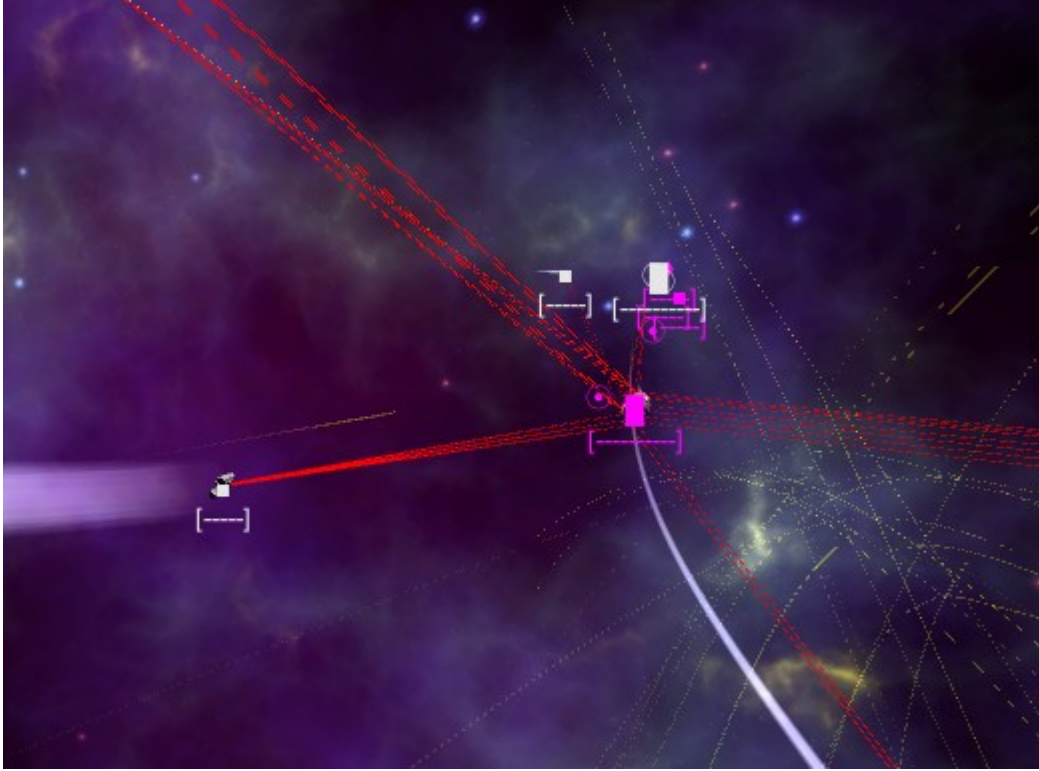
##### **4.2.1 Simbol Unit Dalam Permainan**

Untuk mempermudah pengamatan terhadap unit yang ada di *stage*, simbol diberikan kepada masing-masing unit. Simbol yang diberikan adalah sebagai berikut:

1. ■ Kapal Induk
2. ▲ Pesawat yang membawa Torpedo
3. ■ Pesawat yang membawa Cruise Missile
4. ◆ Pesawat yang membawa Micro Missile
5. ● Unit Pertahanan

Sementara warna yang diberikan adalah sebagai berikut:

1. Putih, untuk unit yang memiliki tim yang sama dengan fokus observasi.
2. Merah, untuk unit yang memiliki tim berbeda dengan fokus observasi.



Gambar 4.1: Visualisasi Permainan

#### 4.2.2 Performa Produk

Dari uji coba yang dilakukan, *planner* membuat rencana dengan mata rantai sejumlah tiga sampai lima aksi dengan waktu eksekusi pada umumnya dibawah 0.01 detik. Hal ini menunjukkan bahwa GOAP cukup cepat untuk diimplementasikan pada permainan taktik yang menggunakan lapisan penyederhanaan aksi dalam implementasinya.

#### 4.2.3 Pengujian dengan Kasus yang Ditentukan

Beberapa Skenario digunakan untuk menguji kemampuan dari kecerdasan buatan yang dibangun, dan dapat dilihat pada bagian ini. Masing-masing skenario diuji coba sebanyak sepuluh kali dan jumlah unit yang terhitung dalam menjalankan skenario tersebut dirata-rata untuk mendapatkan grafik jumlah unit di dunia permainan.

Untuk mempermudah pembacaan grafik dalam pengujian ini, unit dalam permainan diberi singkatan sebagai berikut:

- Kapal Induk / *Carrier Ship* (CS)



- Pesawat yang membawa Torpedo / *Variable Pod Torpedo* (VP\_TPD)
- Pesawat yang membawa Cruise Missile / *Variable Pod Cruise* (VP\_CRS)
- Pesawat yang membawa Micro Missile / *Variable Pod Micro* (VP\_MCR)
- Unit Pertahanan / *Energy Relay* (ER)

Informasi formasi unit yang berada dalam dunia permainan diajikan dengan menggunakan grafik yang berisi data jumlah unit dari kedua tim. Sumbu  $x$  dari grafik merepresentasikan perubahan kondisi dunia permainan seiring berjalannya waktu permainan. Setiap *event* yang merubah jumlah unit dicatat dan ditambahkan dalam grafik ini. Sementara sumbu  $y$  dari grafik merepresentasikan jumlah unit yang berada dalam dunia permainan.

#### 4.2.3.1 Hanya ada Unit Kapal Induk di Dalam Dunia Permainan

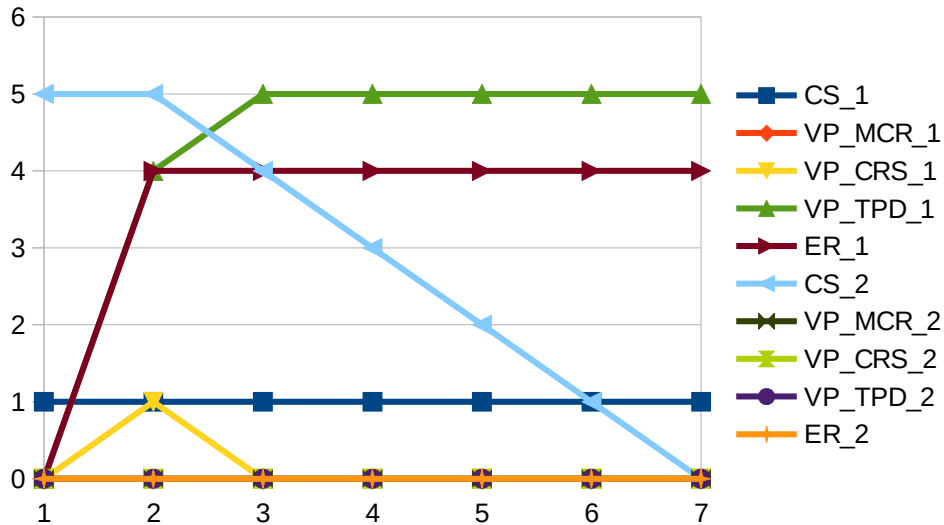
Dalam skenario ini, ada lima unit kapal induk yang disiapkan saat permainan dimulai. Perilaku yang diharapkan adalah agen cerdas dapat mengeluarkan pesawat yang membawa torpedo untuk menghancurkannya. *Screenshot* saat pengujian berlangsung dapat dilihat pada Gambar 4.2.



Gambar 4.2: Mengeluarkan Unit yang Membawa Torpedo Setelah Observasi

Dari ujicoba yang dilakukan, agen GOAP melakukan observasi dengan mengeluarkan sebuah unit pesawat yang membawa cruise missile. Setelah semua unit lawan yang berada di dunia permainan berhasil diobservasi, agen GOAP mengembalikan unit pesawat yang membawa cruise missile untuk diganti perlengkapannya sembari mengeluarkan unit pesawat yang membawa misil dengan nilai serang terbesar, yaitu unit pesawat yang membawa torpedo.

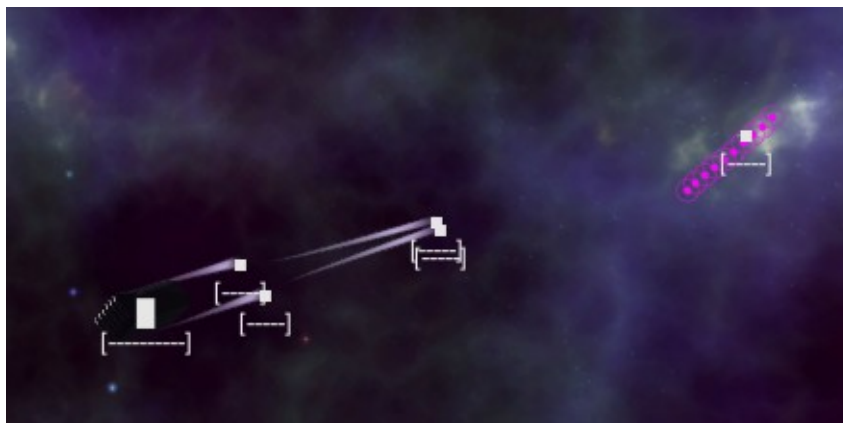
Grafik jumlah unit yang berada di dunia permainan saat pengujian berlangsung dapat dilihat pada Gambar 4.3.



Gambar 4.3: Melawan 5 Unit Kapal Induk

#### 4.2.3.2 Hanya ada Unit Pertahanan

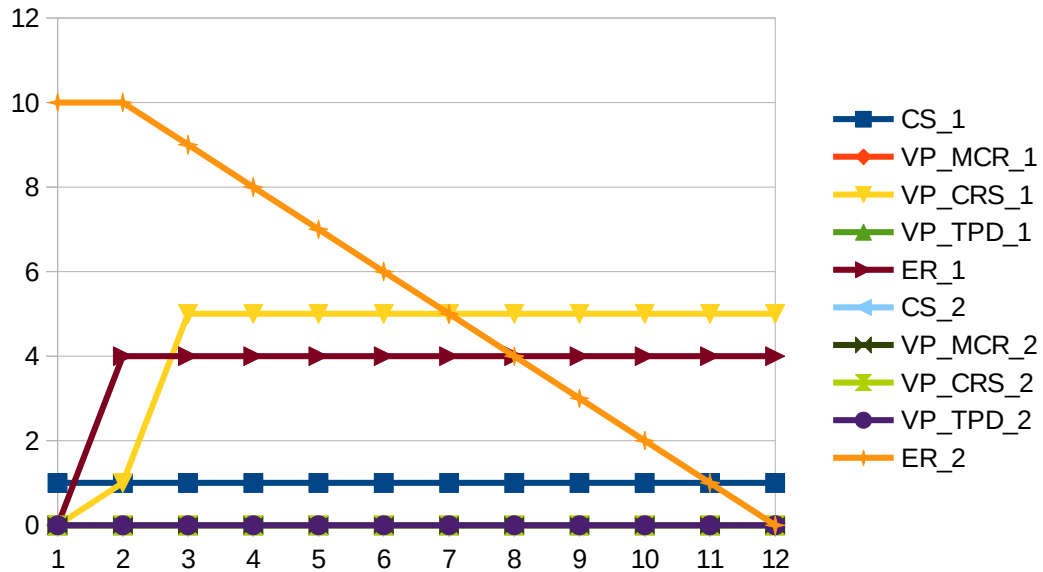
Dalam skenario ini 10 unit pertahanan dijadikan lawan dari agen GOAP. Perilaku yang diharapkan adalah kapal induk mengeluarkan unit yang membawa cruise missile untuk menghancurkan unit pertahanan lawan. *Screenshot* saat pengujian berlangsung dapat dilihat pada Gambar 4.4.



Gambar 4.4: Mengeluarkan Unit yang Membawa Cruise Missile

Setelah mengeluarkan unit untuk melakukan observasi, agen GOAP memutuskan untuk menambah jumlah pesawat yang membawa cruise missile untuk menembus pertahanan yang dimiliki unit pertahanan lawan.

Grafik jumlah unit yang berada di dunia permainan saat pengujian berlangsung dapat dilihat pada Gambar 4.5.



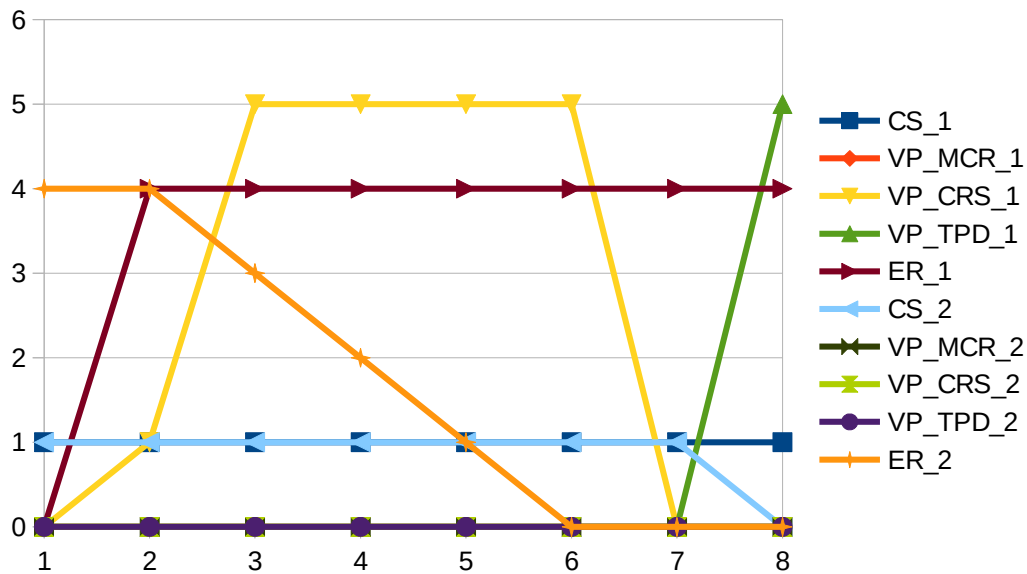
Gambar 4.5: Melawan 10 Unit Pertahanan

#### 4.2.3.3 Unit Kapal Induk dan Pertahanan

Dalam skenario ini lawan diberi unit kapal induk yang kosong dan 4 unit pertahanan. Perilaku yang diharapkan adalah kapal induk mengeluarkan unit yang membawa cruise missile untuk menghabiskan unit pertahanan dan setelahnya mengeluarkan unit yang membawa torpedo untuk menghancurkan kapal induk lawan.

Ujicoba yang dilakukan memberikan hasil yang sesuai dengan harapan dimana agen GOAP mengeluarkan unit untuk menghancurkan unit pertahanan lawan terlebih dahulu. Setelah unit pertahanan lawan berhasil dihabiskan, agen GOAP menarik seluruh unit yang membawa cruise missile kembali ke hangar kapal induk. Disini, agen GOAP memutuskan untuk mengeluarkan satu pesawat yang membawa torpedo untuk menghancurkan unit kapal induk lawan yang tersisa.

Grafik jumlah unit yang berada di dunia permainan saat pengujian berlangsung dapat dilihat pada Gambar 4.6.



Gambar 4.6: Melawan 1 Unit Kapal Induk dan 4 Unit Pertahanan

#### 4.2.3.4 Hanya ada Unit Pesawat

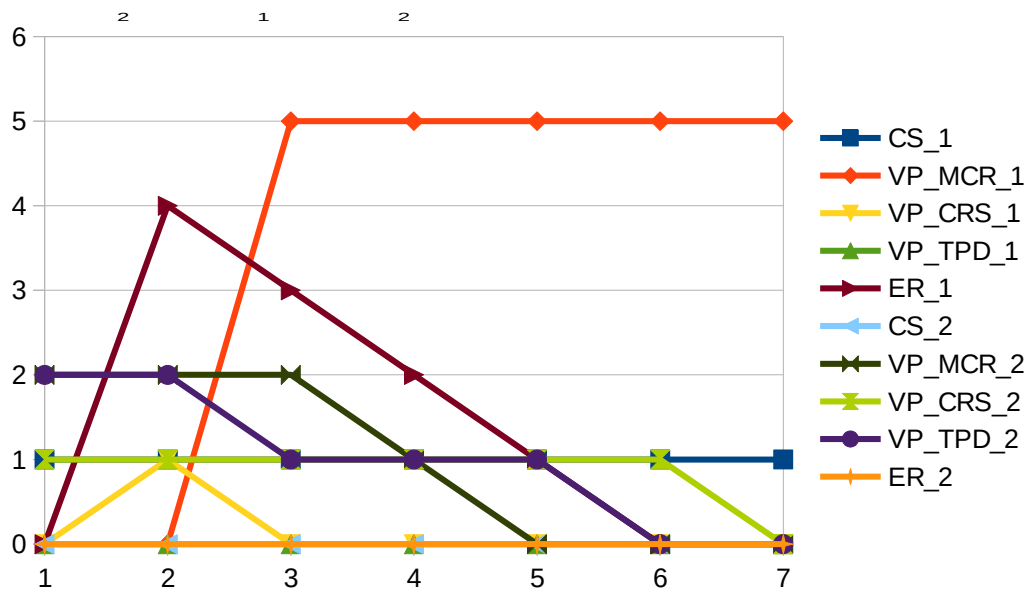
Dalam skenario ini, lawan diberi 5 unit pesawat dengan komposisi:

- 2 unit membawa torpedo
- 1 unit membawa *cruise missile*
- 2 unit membawa *micro missile*

Hasil yang diharapkan adalah kapal induk mengeluarkan unit *micro missile* lalu melawan unit dengan urutan pilihan berupa torpedo, *cruise* dan terakhir *micro missile*.

Grafik jumlah unit yang berada di dunia permainan saat pengujian berlangsung dapat dilihat pada Gambar 4.7. Dalam pengujian ini, agen GOAP melakukan observasi terlebih dahulu terhadap semua unit lawan yang ada. Agen GOAP kemudian memutuskan untuk mengeluarkan lima unit pesawat yang membawa *micro missile* untuk menghadapi lima unit pesawat yang terdeteksi.

Dalam pengalokasiannya, agen GOAP memberi prioritas terhadap unit lawan yang membawa *micro missile* terlebih dahulu, setelah itu mengurangi unit lawan yang membawa *torpedo* dan terakhir menghabiskan unit lawan yang membawa *cruise missile*.



Gambar 4.7: Lawan Memiliki 5 Unit Pesawat Dengan Muatan yang Berbeda

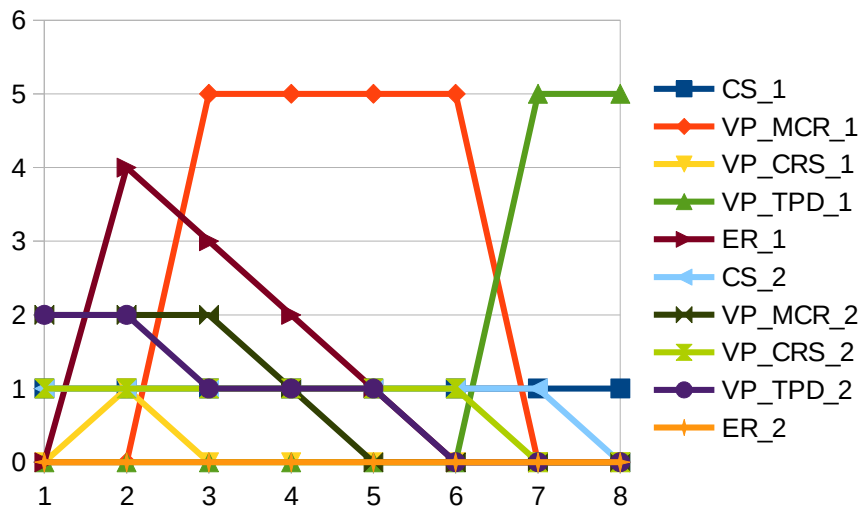
#### 4.2.3.5 Unit Kapal Induk dan Unit Pesawat

Dalam ujicoba ini lawan yang dihadapkan pada agen GOAP adalah 1 unit kapal induk yang tidak memiliki hangar dan 5 unit pesawat dengan tipe muatan yang sama dengan bagian sebelumnya. Hasil yang diharapkan adalah unit pesawat lawan dihabiskan terlebih dahulu sebelum unit kapal induk dihancurkan.

Grafik jumlah unit yang berada di dunia permainan saat pengujian berlangsung dapat dilihat pada Gambar 4.8. Dalam pengujian ini, agen GOAP berusaha untuk menghabiskan unit pesawat terlebih dahulu sebelum menyiapkan perlengkapan untuk menghancurkan kapal induk lawan.

Unit pesawat dihancurkan dengan prioritas dimulai dari unit yang membawa perlengkapan *micro missile*, setelahnya menghancurkan unit yang membawa *torpedo* sebelum melawan unit yang membawa *cruise missile*.

Agen GOAP menarik kembali semua unit pesawat yang membawa *micro missile* untuk diganti perlengkapannya setelah semua unit pesawat lawan telah habis. Unit yang dikeluarkan setelah semua unit pesawat berhasil ditarik adalah unit yang membawa *torpedo*. Dalam pengujian ini, semua unit pesawat yang dimiliki diluncurkan kembali dengan *torpedo* untuk menghancurkan unit kapal induk yang dimiliki lawan.



Gambar 4.8: Lawan Memiliki 1 Unit Kapal Induk dan 5 Unit Pesawat Dengan Muatan yang Berbeda

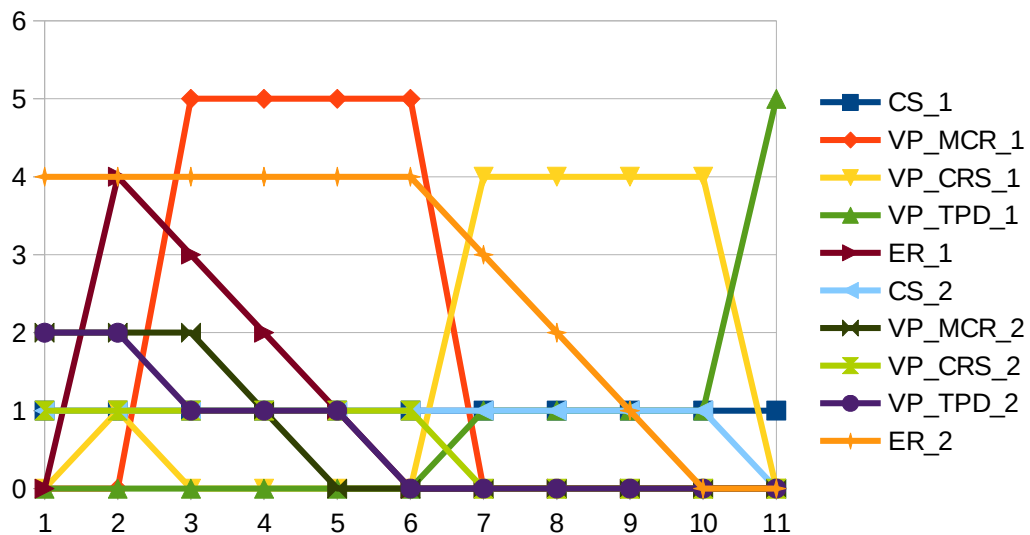
#### 4.2.3.6 Ada Semua Unit di Stage Permainan

Dalam skenario ini, semua tipe unit dimunculkan di dunia permainan dengan komposisi:

- 1 unit kapal induk
- 2 unit membawa torpedo
- 1 unit membawa *cruise missile*
- 2 unit membawa *micro missile*
- 4 unit pertahanan

Hasil yang diharapkan adalah setelah setelah unit pesawat lawan dikalahkan semua, unit pertahanan dihabiskan terlebih dahulu sebelum unit kapal induk dihancurkan.

Grafik jumlah unit yang berada di dunia permainan saat pengujian berlangsung dapat dilihat pada Gambar 4.9. Dalam pengujian ini, agen GOAP memberi prioritas awal untuk mengalahkan seluruh unit pesawat yang dimiliki lawan. Unit-unit pertahanan lawan menjadi sasaran serangan setelah unit pesawat yang dimiliki oleh lawan telah habis. Kapal induk lawan merupakan unit yang dijadikan prioritas serangan terakhir setelah semua unit lawan lainnya berhasil dikalahkan.



Gambar 4.9: Lawan Memiliki 1 Unit Kapal Induk, 5 Unit Pesawat dan 4 Unit Pertahanan

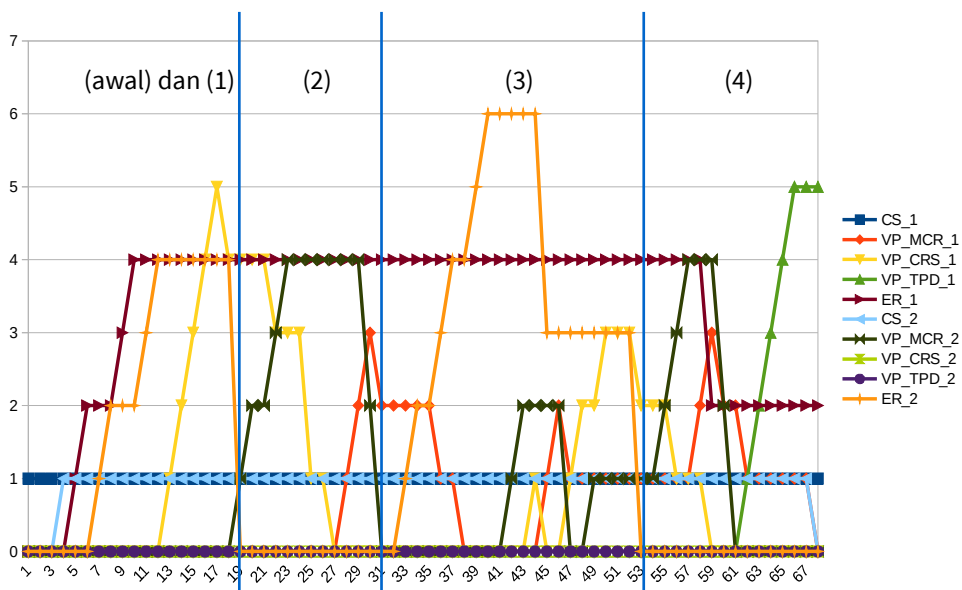
#### 4.2.4 Pengujian Melawan Gelombang Unit

Pengujian ini bertujuan untuk memastikan bahwa agen cerdas dapat beradaptasi dengan menata komposisi unit yang berada di dunia permainan terhadap kondisi yang telah didefinisikan di sebelum permainan dimulai.

Skenario yang digunakan dalam pengujian ini adalah sebagai berikut:

- Kondisi awal : satu unit kapal induk.
- Gelombang pertama : Menambah empat Unit Pertahanan.
- Gelombang kedua : Menambah empat Unit Pesawat setelah unit dari gelombang sebelumnya habis.
- Gelombang ketiga : Menambah enam Unit Pertahanan dan dua Unit Pesawat setelah unit dari gelombang sebelumnya habis.
- Gelombang keempat : Menambah empat Unit Pesawat setelah unit dari gelombang sebelumnya habis.

Kelima langkah tersebut dilakukan dalam satu sesi permainan. Pengujian dilakukan sebanyak sepuluh kali dan jumlah unit yang dikeluarkan oleh agen GOAP dari masing-masing pengujian dirata-rata sehingga menghasilkan grafik seperti yang terlihat pada Gambar 4.10.



Gambar 4.10: Pengujian Terhadap Gelombang Unit

Grafik yang ditunjukkan pada Gambar 4.10 dibagi dalam empat bagian berdasarkan segmen skenario yang telah ditentukan. Kondisi awal permainan dan komposisi berikutnya tercakup di bagian awal dari grafik. Kondisi-kondisi berikutnya ditunjukkan per bagian. Penjelasan dari masing-masing bagian grafik adalah sebagai berikut:

- Kondisi awal dan gelombang pertama. Agen GOAP mengeluarkan unit observasi dan memutuskan untuk mengeluarkan lima unit yang membawa *cruise missile* untuk menghancurkan seluruh unit pertahanan yang dikeluarkan oleh gelombang pertama.
- Gelombang kedua. Untuk melawan unit pesawat lawan yang membawa *micro missile*, dibutuhkan unit yang membawa *micro missile* juga. Karena batas unit pesawat yang ada di dunia permainan sudah penuh, agen GOAP menarik seluruh unit pesawat yang tidak membawa *micro missile*. Dalam hal ini, unit yang ada di dunia permainan hanya membawa *cruise missile* sehingga agen GOAP menarik seluruh unit yang ada. Setelah berhasil menarik seluruh unit, agen GOAP mengganti perlengkapan pesawat dengan *micro missile* untuk mengalahkan unit lawan yang berada di dunia permainan.



- Gelombang ketiga. Dari unit pertahanan dan pesawat yang dikeluarkan lawan, agen GOAP memberi prioritas untuk melawan unit pesawat lawan terlebih dahulu. Disini, unit yang berada di fase sebelumnya ditarik kembali seluruhnya karena unit yang dikeluarkan lawan belum terobservasi. Setelah berhasil mengobservasi unit lawan, pesawat yang membawa *micro missile* kembali dikeluarkan untuk menghancurkan unit pesawat lawan sebelum mengganti perlengkapannya dengan *cruise missile* untuk menghancurkan seluruh unit pertahanan lawan yang dikeluarkan.
- Gelombang keempat. Dalam gelombang ini, agen GOAP kembali melakukan observasi lingkungan sebelum mengeluarkan unit yang membawa *micro missile* untuk menghadapi unit pesawat yang dikeluarkan. Setelah unit pesawat yang dimiliki lawan telah habis, semua unit yang tidak membawa *torpedo* ditarik kembali untuk diganti perlengkapannya. Lima unit yang membawa *torpedo* menjadi komposisi terakhir sebelum permainan berakhir.

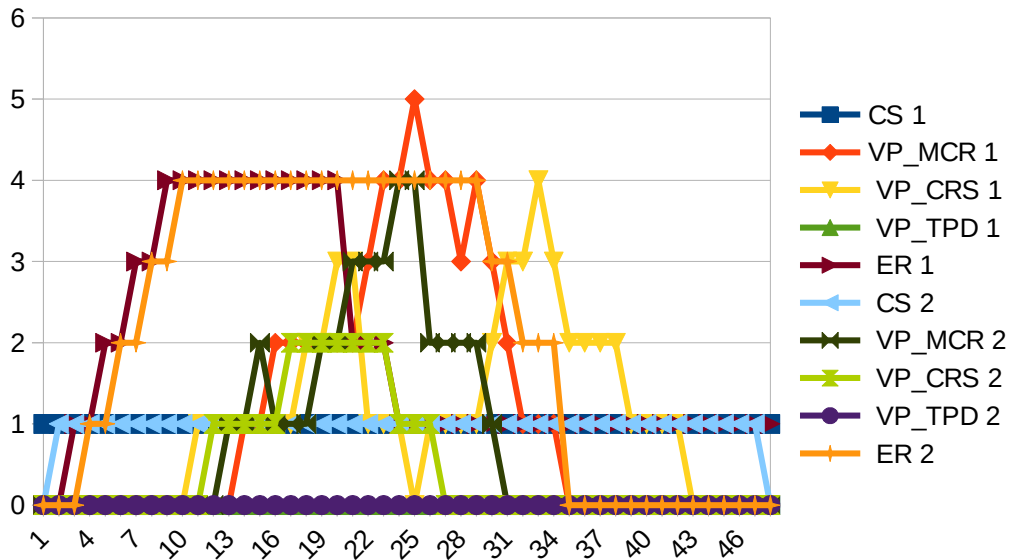
Dari pengujian ini, terlihat bahwa agen cerdas berbasis GOAP mampu mengadaptasikan komposisinya terhadap kondisi lawan.

#### **4.2.5 Pengujian Melawan *Rule-Based AI***

Pengujian dengan skenario yang telah ditentukan memberikan hasil yang mendekati harapan. *Rule-Based AI* yang dibangun dengan menggunakan dasar logika yang sama dengan implementasi arsitektur GOAP diuji pada bagian ini untuk mendapatkan perbandingan kecepatan respon terhadap perubahan lingkungan. Pengujian dilakukan dengan mengadu kedua agen sebanyak lima puluh kali. Unit yang dapat digunakan oleh kedua agen merupakan unit yang serupa dan perlengkapan yang dapat digunakan oleh unit pesawat diberi jumlah yang sama.

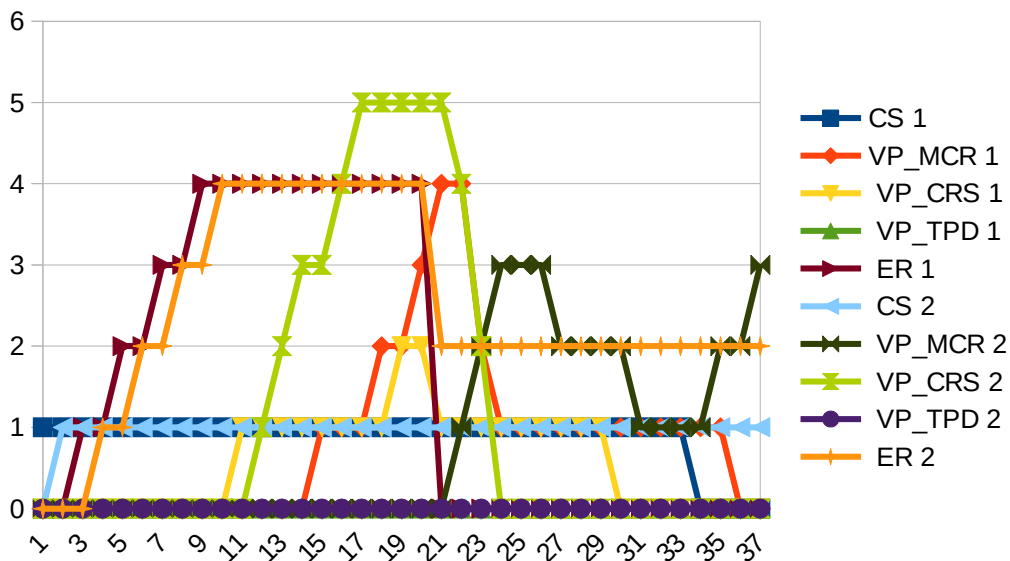
Setelah pengujian dilakukan, hasil yang didapatkan menunjukkan bahwa arsitektur GOAP mendapatkan kemenangan sebesar empat puluh persen dari seluruh sesi permainan.

Grafik yang ditunjukkan oleh Gambar 4.11 merupakan komposisi unit yang diambil dari salah satu sesi permainan saat agen GOAP memenangkan sesi permainan tersebut.



Gambar 4.11: Saat GOAP Memenangkan Permainan

Grafik yang ditunjukkan oleh Gambar 4.12 merupakan komposisi unit yang diambil dari salah satu sesi permainan saat agen GOAP kalah di sesi permainan tersebut.



Gambar 4.12: Saat GOAP Kalah di Permainan

### 4.3 Analisis

Pengujian yang dilakukan menunjukkan kemampuan GOAP menyusun komposisi unit dalam permainan RTT untuk mengantisipasi kondisi lawan. Skenario-skenario yang diberikan dapat diselesaikan oleh agen walaupun memiliki sedikit perbedaan dengan pola penyelesaian yang diharapkan.

Kemampuan agen untuk menarik kembali unit yang tidak sesuai dengan kondisi dunia dapat diamati dalam pengujian ini. Hal ini menunjukkan kemampuan adaptasi agen terhadap kondisi dunia permainan.

Kondisi kemenangan sebesar empat puluh persen saat melawan *Rule-Based* AI disebabkan oleh respon yang lebih cepat dari agen GOAP. Hal ini terjadi karena *Rule-Based* AI tidak memerlukan fase perencanaan sebelum menjalankan logika permainannya.

*Halaman ini sengaja dikosongkan*

## **BAB 5**

### **KESIMPULAN**

#### **5.1 Kesimpulan**

Dalam proses pengembangan yang dilakukan, arsitektur GOAP sangat membantu dalam menurunkan kompleksitas desain kecerdasan buatan dalam permainan yang disusun. Kondisi pengambilan keputusan yang berulang dapat muncul dalam pembuatan agen menggunakan *Rule-Based AI*. Dengan sifat GOAP yang hanya membutuhkan parameter *precondition*, kemungkinan redundansi kode dapat ditekan.

Pengujian yang dilakukan menunjukkan kemampuan agen GOAP untuk menyelesaikan skenario yang diberikan. Ada sedikit perbedaan antara harapan penyelesaian skenario dengan hasil nyata yang diberikan oleh agen GOAP. Hal ini disebabkan oleh kurangnya akurasi dalam proses memberikan bobot terhadap unit-unit yang ada dalam dunia permainan.

Selama pengujian ditemukan juga kelemahan dari agen GOAP yang dibangun. Proses penyusunan *plan* yang dilakukan dalam fase yang berbeda dengan pelaksanaannya menyebabkan tertundanya pelaksanaan aksi. Agen yang menggunakan *Rule-Based AI* dapat melaksanakan aksinya langsung setelah menganalisa lingkungan permainan. Oleh karenanya, hasil pengujian saat agen GOAP diadu dengan agen *Rule-Based AI* memberikan hasil yang condong pada kemenangan agen *Rule-Based AI*.

Di sisi lain, desainer kecerdasan buatan diharuskan untuk mengevaluasi semua aksi yang berkaitan dengan parameter unit saat ada perubahan jenis parameter dalam fase pengembangan. Hal ini dapat diminimalisir dengan memastikan seluruh parameter yang digunakan dalam permainan telah final sebelum memulai desain agen cerdas.

Dari penelitian ini dapat ditarik kesimpulan bahwa agen GOAP masih memiliki kesempatan untuk digunakan dalam permainan *real-time tactics* karena kompleksitas desain yang lebih rendah dari *real-time strategy*. Kelemahan yang muncul menunjukkan bahwa persiapan parameter yang matang diperlukan sebelum memulai desain kecerdasan buatan dengan arsitektur GOAP ini.

## 5.2 Penelitian Selanjutnya

Penelitian lebih lanjut dalam bidang yang menggunakan arsitektur GOAP dapat dilakukan dengan memperbanyak definisi taktik permainan untuk menguji batas kemampuan agen dengan variasi perencanaan yang lebih banyak. Selain itu, pengaturan tingkat kesulitan dengan memberi variabel yang dapat dikontrol pada proses perencanaan aksi dimungkinkan untuk diteliti juga.

Jika waktu eksekusi menurun akibat banyaknya node yang dibuat saat perencanaan aksi, taktik yang dibuat dapat dipecah lagi per aksi unit dengan menggunakan hierarki. Selain itu, penghitungan bobot unit permainan dapat dibuat menjadi lebih adaptif.

Agen GOAP yang dibuat pada penelitian ini masih *rigid* terhadap unit yang sudah ditentukan di awal. Penelitian yang ditujukan untuk membuat agen dapat beradaptasi terhadap unit tambahan yang dimasukkan setelah permainan dipublikasikan juga dapat dilakukan sebagai pengembangan dari penelitian ini.

## DAFTAR PUSTAKA

- [1] D. Pittman, “Practical Development of Goal-Oriented Action Planning AI,” 2008.
- [2] H. Zhang, Z. Shen, and C. Miao, “Enabling goal oriented action planning with Goal Net,” *Proc. - 2009 IEEE/WIC/ACM Int. Conf. Intell. Agent Technol. IAT 2009*, vol. 2, no. January, pp. 271–274, 2009.
- [3] R. Moss, “Build, gather, brawl, repeat: The history of real-time strategy games,” 2017. [Online]. Available: <https://arstechnica.com/gaming/2017/09/build-gather-brawl-repeat-the-history-of-real-time-strategy-games/>. [Accessed: 06-Oct-2017].
- [4] G. Robertson and I. Watson, “A Review of Real-Time Strategy Game AI,” *AI Mag.* 35.4, pp. 75–104, 2014.
- [5] J. Orkin, “Applying Goal-Oriented Action Planning to Games,” *AI Game Program. Wisdom* 2, no. December, pp. 217–227, 2003.
- [6] B. G. Weber, M. Mateas, and A. Jhala, “Applying Goal-Driven Autonomy to StarCraft,” *Artif. Intell.*, no. Orkin, pp. 101–106, 2010.
- [7] G. Maggiore, C. Santos, D. Dini, F. Peters, H. Bouwknecht, and P. Spronck, “LGOAP: Adaptive layered planning for real-time videogames,” *IEEE Conf. Comput. Intell. Games, CIG*, 2013.
- [8] H. Chan, A. Fern, S. Ray, N. Wilson, and C. Ventura, “Online Planning for Resource Production in Real-Time Strategy Games,” *Icaps*, pp. 65–72, 2007.
- [9] M. Hipple, “Against An Air Force Space Corps: Space Belongs to the Navy!,” 2017. [Online]. Available: <http://nationalinterest.org/blog/the-buzz/against-air-force-space-corps-space-belongs-the-navy-21350>. [Accessed: 15-Dec-2017].
- [10] R. Robinson, “The Space Authority and the Orbital Patrol,” 2015. [Online]. Available: <http://www.rocketpunk-manifesto.com/2015/07/the-space-authority-and-orbital-patrol.html>. [Accessed: 11-Oct-2017].
- [11] L. David, “How to Clean Up Space Junk: DARPA’s Orbital Catcher’s Mitt,” 2011. [Online]. Available: <https://www.space.com/11657-space-junk-orbital-debris-cleanup-darpa.html>. [Accessed: 11-Oct-2017].
- [12] R. Garitta, “The Other Fleet,” 2016. [Online]. Available:

- <http://twilightgm.blogspot.co.id/2016/08/the-other-fleet.html>. [Accessed: 15-Dec-2015].
- [13] G. Anderson, “Creating a mission control for the commercial spaceflight industry,” 2016. [Online]. Available: <http://www.thespacereview.com/article/2989/1>. [Accessed: 08-Sep-2017].
- [14] E. Landau, “Voyager Signal Spotted By Earth Radio Telescopes,” 2013. [Online]. Available: [https://www.nasa.gov/mission\\_pages/voyager/multimedia/pia17047.html](https://www.nasa.gov/mission_pages/voyager/multimedia/pia17047.html). [Accessed: 02-Jul-2017].
- [15] W. Chung, “Detection in Space Warfare,” 2017. [Online]. Available: [http://www.projectrho.com/public\\_html/rocket/spacewardetect.php](http://www.projectrho.com/public_html/rocket/spacewardetect.php). [Accessed: 21-Dec-2017].
- [16] Porter, “Unity: Now You’re Thinking With Components,” 2013. [Online]. Available: <https://gamedevelopment.tutsplus.com/articles/unity-now-youre-thinking-with-components--gamedev-12492>. [Accessed: 21-Sep-2017].
- [17] R. E. Fikes and N. J. Nilsson, “STRIPS: A new approach to the application of theorem proving to problem solving,” *Artif. Intell.*, vol. 2, no. October, pp. 189–208, 1972.
- [18] D. Fu and R. Houlette, “The Ultimate Guide to FSMs in Games.” pp. 283–302, 2004.
- [19] P. E. Hart, N. J. Nilsson, and B. Raphael, “A Formal Basis for the Heuristic Determination of Minimum Cost Paths,” *ACM SIGART Bull.*, no. 37, pp. 28–29, 1972.
- [20] I. Millington and J. Funge, *Artificial Intelligence for Games, Second Edition*. 2009.
- [21] I. D. Craig, “Blackboard systems,” *Artif. Intell. Rev.*, vol. 2, no. 2, pp. 103–118, 1988.
- [22] P. S. Miller, “Astounding Science Fiction,” Feb-1957.
- [23] F. Bagenal, “The magnetosphere of Jupiter: Coupling the equator to the poles,” *J. Atmos. Solar-Terrestrial Phys.*, vol. 69, no. 3, pp. 387–402, 2007.
- [24] Martin J. L. Turner, *Rocket and Spacecraft Propulsion: Principles, Practice and New Developments*. Praxis, 2001.



## BIOGRAFI PENULIS



Penulis, Restuadi Studiawan, lahir di Ambon pada tanggal 1 Desember 1987. Penulis adalah putra kedua dari Ibut Priono Leksono dan Dwi Sulistiyani. Penulis menyelesaikan pendidikan Sekolah dasar di Ambon, Maluku dan melanjutkan jenjang Sekolah Menengah Pertama sampai Sekolah Menengah Atas di Sidoarjo, Jawa Timur. Setelah lulus dari Sekolah Menengah Atas, penulis menempuh pendidikan S-1 Teknik Informatika di Institut Teknologi Sepuluh Nopember Surabaya.

Penulis menempuh studi S-1 selama 6 tahun. Setelah lulus, penulis turut memberi kontribusi pada sebuah studio sebagai pengembang permainan elektronik pada tahun 2010. Sembari berkarya, penulis mendaftar di Jurusan Teknik Elektro ITS untuk memperdalam ilmu pengembangan permainan elektronik. Penulis resmi selesai menempuh studi S-2 dan diwisuda pada Maret 2018.