



PROPOSAL TESIS

UAV SWARM CONTROL UNTUK MULTI-TARGET DINAMIS MENGGUNAKAN MULTI-AGENT REINFORCEMENT LEARNING

VINCENTIUS CHARLES MAYNAD
6022222024

DOSEN PEMBIMBING

Prof. Ir. H. Abdullah Alkaff, M.Sc., Ph.D.

PROGRAM MAGISTER

BIDANG KEAHLIAN TEKNIK SISTEM PENGATURAN

DEPARTEMEN TEKNIK ELEKTRO

FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

SURABAYA

2023

LEMBAR PENGESAHAN PROPOSAL TESIS

Judul : UAV Swarm Control untuk Multi-Target Dinamis Menggunakan Multi-Agent Reinforcement Learning
Oleh : Vincentius Charles Maynad
NRP : 6022222024

Telah diseminarkan pada

Hari : Rabu
Tanggal : 5 Juli 2023
Tempat : Ruang B104

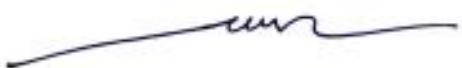
Mengetahui/menyetujui

Dosen Penguji:


Calon Dosen Pembimbing



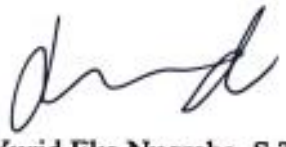
1. Prof. Dr. Ir. Achmad Jazidie, M.Eng.
NIP: 195902191986101001



1. Prof. Ir. H. Abdullah Alkaff, M.Sc.,
Ph.D.
NIP: 195501231980031002



2. Dr. Ir. Ari Santoso, DEA
NIP: 196602181991021001



3. Yurid Eka Nugraha, S.T., M.Eng., Ph.D.
NPP: 2023199511059

Halaman ini sengaja dikosongkan

UAV SWARM CONTROL UNTUK MULTI-TARGET DINAMIS MENGGUNAKAN MULTI-AGENT REINFORCEMENT LEARNING

Nama mahasiswa : Vincentius Charles Maynad
NRP : 6022222024
Pembimbing : Prof. Ir. H. Abdullah Alkaff, M.Sc., Ph.D.

ABSTRAK

Pada sebuah lingkungan dimana beberapa UAV (*pursuer*) ditugaskan untuk menghancurkan beberapa UAV lain (*evader*), permasalahan *tracking* multi-target menjadi salah satu masalah penting yang harus diselesaikan. Keseluruhan permasalahan dapat dibagi menjadi tiga, yakni 1) *Tracking*, 2) *Following* dan *Task Allocating*, dan 3) *Intercepting*.

Tujuan penelitian ini adalah merancang strategi kontrol berupa algoritma untuk mengatasi permasalahan multi-target pada sistem multi-UAV. Permasalahan akan dimodelkan sebagai model *Decentralized-Partially Observable Markov Decision Process* (Dec-POMDP) untuk membantu tiap *pursuer* memutuskan tindakan apa yang sebaiknya diambil selanjutnya. Kebijakan atau *policy* terbaik dari model Dec-POMDP akan dicari menggunakan salah satu algoritma *Multi-Agent Reinforcement Learning* (MARL), yaitu *actor-critic*. Adapun konsep *Centralized Training with Decentralized Execution* (CTDE) akan digunakan untuk merancang algoritma *actor-critic*. Setelah tugas dibagi diantara UAV *pursuer*, pencegahan dan penghancuran *evader* dilakukan menggunakan metode *Proportional Pursuit* (PP) dan *Proportional Navigation* (PN). Pengujian dilakukan berdasarkan simulasi dengan beberapa kombinasi jumlah *pursuer* dan *evader*.

Diharapkan dengan strategi kontrol dan algoritma yang dirancang, setiap tugas dapat dilaksanakan dengan baik.

Kata kunci: Sistem multi-agen, multi-target, *Markov Decision Process*, *Reinforcement Learning*

Halaman ini sengaja dikosongkan

DAFTAR ISI

LEMBAR PENGESAHAN.....	iii
ABSTRAK.....	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	ix
DAFTAR TABEL.....	xi
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan.....	2
1.4 Batasan Masalah.....	2
1.5 Kontribusi.....	3
BAB 2 KAJIAN PUSTAKA.....	5
2.1 Kajian Penelitian Terkait.....	5
2.1.1 <i>Improving multi-target cooperative tracking guidance for UAV swarms using multi-agent reinforcement learning</i> [3].....	5
2.1.2 <i>Collaborative Decision-Making Method for Multi-UAV Based on Multiagent Reinforcement Learning</i> [10].....	10
2.1.3 UAV Formation Shape Control via Decentralized Markov Decision Processes [11].....	14
2.2 Teori Dasar.....	18
2.2.1 Model dinamika UAV [10], [12], [13].....	19
2.2.2 Decentralized Partially Observable Markov Decision Process (Dec-POMDP) [3], [10].....	20
2.2.3 Reinforcement Learning : Actor-Critic [9].....	21
2.2.4 Centralized Training Decentralized Execution (CTDE) [5], [10]...	22
2.2.5 Proportional Pursuit (PP) dan Proportional Navigation (PN) [12] .	24
BAB 3 METODOLOGI PENELITIAN.....	27
3.1 Pemodelan UAV.....	28
3.2 Pemodelan Dec-POMDP.....	29
3.3 Merancang algoritma MARL.....	31
3.4 Metode penghancuran target.....	32

3.5	Hipotesa Penelitian.....	32
3.6	Rencana Pengujian.....	32
3.7	Kriteria Pengujian	32
BAB 4 RENCANA DAN JADWAL KEGIATAN.....		33
4.1	Jadwal Kegiatan.....	33
DAFTAR PUSTAKA.....		35

DAFTAR GAMBAR

Gambar 2.1 Skenario multi-UAV melacak multi-target.....	5
Gambar 2.2 Ilustrasi jangkauan observasi dan jangkauan komunikasi dari tiap UAV.....	6
Gambar 2.3 Ilustrasi algoritma MAAC-R.....	8
Gambar 2.4 Simulasi MTTG menggunakan algoritma yang diajukan.....	9
Gambar 2.5 Model UAV yang digunakan dalam <i>earth-frame</i> dan <i>body-frame</i>	10
Gambar 2.6 Ilustrasi algoritma MATAC. Komponen utamanya adalah centralized critic, beberapa distributed actor, dan reply buffer.....	12
Gambar 2.7 Komponen penyusun critic dengan lebih rinci.....	12
Gambar 2.8 Komponen penyusun actor dengan lebih rinci.....	12
Gambar 2.9 Ilustrasi simulasi yang dilakukan pada [10].....	13
Gambar 2.10 Perbandingan rasio kemenangan MATAC dibanding algoritma-algoritma lain dari 300 percobaan.....	14
Gambar 2.11 Arsitektur kontrol UAV pada [11].....	16
Gambar 2.12 Simulasi yang dilakukan pada [11].....	17
Gambar 2.13 Simulasi dengan rintangan.....	17
Gambar 2.14 Perbandingan Tc dari kedua pendekatan.....	18
Gambar 2.15 Ilustrasi model drone fixed-wing dengan tiga derajat kebebasan.....	19
Gambar 2.16 Ilustrasi hubungan Dec-POMDP terhadap algoritma MDP lainnya.....	20
Gambar 2.17 Ilustrasi algoritma actor-critic.....	22
Gambar 2.18. Ilustrasi arsitektur CTDE.....	23
Gambar 2.19. Ilustrasi arah gerak, garis pandang, dan sudut deviasi.....	24
Gambar 3.1 Ilustrasi skenario penelitian yang akan dibuat.....	27
Gambar 3.2 Ilustrasi jangkauan sensor UAV yang terbatas.....	27
Gambar 3.3 Drone namikaze.....	28
Gambar 3.4 Diagram blok interaksi agen terhadap lingkungan.....	30
Gambar 3.5 Rincian pada tiap blok Agent dari gambar 3.4.....	31
Gambar 3.6 Gambaran skema pelatihan.....	31

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Tabel 2.1 Waktu rata-rata T_f	18
Tabel 4.1 Perencanaan Jadwal Kegiatan	33

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Dalam beberapa tahun terakhir, kontrol kooperatif sistem multi-*unmanned aerial vehicle* (multi-UAV) telah menjadi topik penelitian yang hangat di bidang kontrol penerbangan [1]. Kompleksitas kondisi lingkungan dan tugas yang semakin meningkat membuat sistem multi-UAV sangat dibutuhkan untuk menyelesaikan tugas tertentu [2], yang tidak bisa dilakukan oleh *single* UAV. Menggunakan kerja sama, sistem multi-UAV dapat menunjukkan koordinasi, kecerdasan, dan otonomi yang lebih unggul daripada kawanan UAV biasa [3]. Pemanfaatan sistem multi-UAV sudah banyak dibahas dan dikembangkan untuk berbagai bidang.

Pada bidang militer yang melibatkan banyak pasukan, salah satu permasalahan yang cukup penting untuk diteliti adalah *Multi-Target Tracking Guidance* (MTTG) [3]. MTTG adalah kondisi dimana sekelompok UAV ditugaskan untuk melacak beberapa target yang tidak diketahui sebelumnya. Umumnya, UAV yang digunakan memiliki jangkauan sensor observasi dan komunikasi yang terbatas sehingga masing-masing UAV tidak bisa melihat lingkungan secara keseluruhan (*global*). Tiap UAV hanya bisa menerima informasi dari hasil pembacaan sensornya sendiri dan dari tetangganya melalui komunikasi.

Untuk menyelesaikan MTTG, umumnya permasalahan dimodelkan sebagai model *Markov Decision Process* (MDP). Sejatinya, MDP digunakan pada kondisi tanpa ketidakpastian model dan informasi *state* serta tindakan (*action*) dari semua UAV bisa didapatkan. Tentu, dengan komunikasi yang lebih luas, solusi yang lebih baik akan lebih mudah didapatkan [4]. Konsep ini diistilahkan sebagai konsep sentralisasi. Konsep sentralisasi memiliki beberapa kekurangan seperti kompleksitas dan waktu komputasi yang dibutuhkan [5]. Selain itu, asumsi tersebut kurang sesuai dengan deskripsi permasalahan yang akan dihadapi. Oleh karena itu, model yang lebih spesifik harus digunakan, yaitu *Decentralized Partially Observable Markov Decision Process* (Dec-POMDP). Konsep desentralisasi lebih cocok untuk menyelesaikan masalah walaupun sangat bergantung pada komunikasi antar UAV.

Multi Agent Reinforcement Learning (MARL) adalah metode yang menjanjikan untuk mencari solusi terbaik dari model Dec-POMDP. MARL memiliki banyak alternatif penyelesaian, [6], [7] menggunakan *actor-critic*, [8] menggunakan algoritma *co-evolutionary*, [9] menggunakan *Particle Swarm Optimization* (PSO). Diatas itu, paradigma *Centralized Training with Decentralized Execution* (CTDE) memungkinkan untuk melatih UAV-UAV secara terpusat sementara dalam eksekusinya, para UAV membuat keputusan hanya berdasarkan observasi lokal [10]. Paradigma ini secara signifikan menggeser kompleksitas ke pelatihan dan membuat eksekusi menjadi ringan.

Pada proposal ini, akan dijelaskan konsep permasalahan MTTG yang lebih luas dimana skenario yang dihadapi tidak berhenti hanya sampai pelacakan tetapi juga penyerangan. Kompleksitas permasalahan juga ditingkatkan karena target juga merupakan UAV dan bergerak dalam kelompok. Selain itu, target harus dihancurkan. Kemudian, diajukan metode penyelesaian menggunakan pendekatan Dec-POMDP dan MARL berdasarkan paradigma CTDE modifikasi.

1.2 Rumusan Masalah

Rumusan masalah dari penelitian ini adalah bagaimana cara mengontrol multi-UAV (disebut *pursuer*) untuk melacak, membagi tugas, dan menyerang beberapa target (multi-target dan disebut *evader*) UAV yang bergerak dalam kelompok.

1.3 Tujuan

Tujuan penelitian ini adalah merancang algoritma yang memanfaatkan pendekatan Dec-POMDP dan MARL untuk melakukan pelacakan, pembagian tugas, dan penyerangan terhadap multi-target pada sistem multi-UAV.

1.4 Batasan Masalah

Beberapa batasan masalah yang digunakan pada penelitian ini adalah:

1. Setiap UAV berupa *drone fixed-wing*.
2. Kecepatan UAV *pursuer* lebih cepat daripada UAV *evader*.

3. Penyerangan yang dilakukan oleh UAV *pursuer* adalah dengan menabrakan dirinya ke UAV *evader*.
4. Setiap UAV hanya bisa menerima informasi dari observasi lokal dan informasi dari sesama kelompok UAV (*pursuer* atau *evader*). Jangkauan komunikasi tiap UAV tidak terbatas.
5. Diasumsikan kalau komunikasi antar UAV adalah baik tanpa *delay*.

1.5 Kontribusi

Beberapa kontribusi yang diharapkan dari hasil penelitian ini adalah:

1. Membawa permasalahan ke ranah tiga dimensi yang lebih kompleks sesuai dengan target yang dihadapi.
2. Mengembangkan strategi kontrol dan asumsi yang berbeda dari sebelumnya sesuai dengan perluasan skenario masalah yang menjadi lebih lengkap (tiga permasalahan) dan kompleks (target yang berkumpul bersama).
3. Mengembangkan metode *training* MARL berbasis CTDE yang berbeda berupa kombinasi konsep sentralisasi dan desentralisasi.

Halaman ini sengaja dikosongkan

BAB 2

KAJIAN PUSTAKA

Bab ini akan membahas materi yang berhubungan dengan penelitian yang akan dikerjakan, antara lain kajian penelitian terkait dan teori dasar. Penelitian yang akan dilakukan adalah perancangan algoritma pelacakan, pembagian tugas, dan penyerangan target untuk sistem *multi-UAV*. Beberapa pustaka terdahulu mengenai penelitian terkait kendali sistem *multi-UAV* akan dijadikan sebagai bahan literatur. Teori-teori dasar untuk menunjang metode yang diajukan juga akan dijelaskan.

2.1 Kajian Penelitian Terkait

Sub bab berikut akan menjelaskan penelitian-penelitian mengenai metode kontrol multi-UAV untuk melaksanakan berbagai tugas kompleks seperti pelacakan *multi-target* dan pembentukan formasi yang pernah dilakukan.

2.1.1 *Improving multi-target cooperative tracking guidance for UAV swarms using multi-agent reinforcement learning* [3]

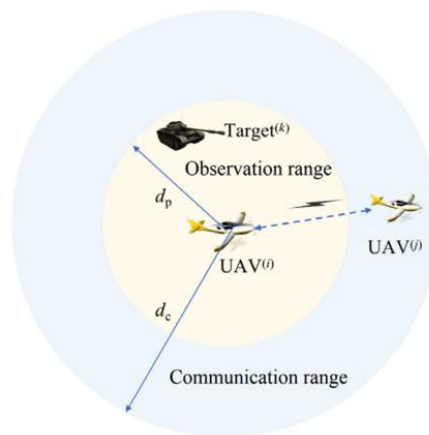


Gambar 2.1 Skenario multi-UAV melacak multi-target

Paper yang ditulis oleh Wenhong Zhou, Jie Li, Zhihong Liu, dan Lincheng Shen ini membahas mengenai solusi permasalahan *Multi-Target Tracking Guidance* (MTTG) dimana sistem dengan multi-UAV harus melacak multi-target yang bergerak. MTTG akan menjadi permasalahan yang kompleks bila dilakukan pada lingkungan yang tidak diketahui (mengacu pada lingkungan dan target yang dicari). Selain itu, diasumsikan kalau tiap UAV memiliki sifat desentralisasi. Sifat desentralisasi berarti

masing-masing UAV hanya bisa memanfaatkan informasi dari lingkungan atau UAV tetangga disekitarnya. Mereka tidak bisa mengakses informasi *global* atau informasi untuk keseluruhan lingkungan. Karena jangkauan observasi yang terbatas seperti terlihat pada gambar 2.2 dan gerakan target yang tidak terkontrol, UAV bisa saja kehilangan target yang dilacak. Selain itu, karena tidak ada pembagian target yang jelas maka sebuah UAV bisa saja melacak beberapa target dalam waktu yang bersamaan atau sebaliknya. Oleh karenanya, UAV harus selalu menjaga target tetap dalam radius pengelihatannya dan bekerja sama dengan sesama UAV lain untuk melacak target sebanyak mungkin. Apalagi, mereka harus memastikan tidak terjadi tabrakan antar satu sama lain dan terbang melewati batasan.

Salah satu metode yang cukup populer untuk menyelesaikan permasalahan seperti itu adalah *Multi-Agent Deep Reinforcement Learning* (MADRL). MADRL bisa digunakan untuk melatih kerja sama antar UAV. Di MADRL, masing-masing UAV belajar bagaimana cara berperilaku untuk memaksimalkan *reward* yang mereka dapatkan melalui interaksi berulang dengan lingkungan dan agen lain dan mempelajari hubungan koordinasi potensial antara agen. Namun, MADRL sendiri masih mengalami kendala bila diterapkan pada kondisi desentralisasi, salah satunya karena kompleksitas komputasi yang dibutuhkan.



Gambar 2.2 Ilustrasi jangkauan observasi dan jangkauan komunikasi dari tiap UAV

Untuk itu, masalah MTTG dimodelkan sebagai model *Decentralized Partially Observable Markov Decision Process* (Dec-POMDP) yang didefinisikan sebagai

$$(N, S, A, P, R, O, Z, \gamma) \quad (2.1)$$

dimana N adalah himpunan n UAV, S adalah *state space* dari state s , A adalah *joint action space* dari tiap UAV, probabilitas $P(s'|s, a) \rightarrow [0,1]$ menotasikan model probabilitas transisi dari *state* s ke *state* selanjutnya s' , setelah mengeksekusi *joint action* a , R adalah fungsi *joint reward* yang didapatkan setelah mengeksekusi a pada *state* s . O adalah *joint observation space* dari semua UAV, Z adalah model observasi individu dari tiap UAV berdasarkan *state* s . Sedangkan γ adalah konstanta *discount factor*. Sedangkan untuk menggambarkan gerak UAV, digunakan persamaan kinematik UAV

$$\begin{cases} x_{U,t+1}^{(i)} = x_{U,t}^{(i)} + \Delta t v_U^{(i)} \cos \theta_{U,t}^{(i)}, & 0 \leq x_{U,t}^{(i)} \leq x_{\max} \\ y_{U,t+1}^{(i)} = y_{U,t}^{(i)} + \Delta t v_U^{(i)} \sin \theta_{U,t}^{(i)}, & 0 \leq y_{U,t}^{(i)} \leq y_{\max} \\ \theta_{U,t+1}^{(i)} = \theta_{U,t}^{(i)} + \Delta t a_t^{(i)}, & -\dot{\theta}_{\max} \leq a_t^{(i)} \leq \dot{\theta}_{\max} \end{cases} \quad (2.2)$$

dimana x , y , dan θ adalah posisi UAV pada sumbu x , posisi UAV pada sumbu y , dan arah UAV, secara berurutan. Δt adalah *time step* simulasi, v_U adalah kecepatan UAV yang bernilai konstan, dan $a_t^{(i)}$ adalah tindakan UAV i .

Untuk menyelesaikan model Dec-POMDP diatas, penulis mengajukan metode *decentralized MADRL* yang dilengkapi sistem *rewarding* modifikasi, *maximum reciprocal reward*, agar proses pembelajaran kebijakan pelacakan kooperatif menjadi lebih baik.

$$r_c^{(i)} = \frac{1}{d^{(i)}} \sum_{j \in \mathcal{N}^{(i)}} d^{(ij)} r_e^{(j)} \quad (2.3)$$

Dimana $\mathcal{N}^{(i)}$ adalah himpunan UAV tetangga dari UAV i , $d^{(ij)}$ adalah ketergantungan UAV i dan UAV j , sedang $d^{(i)} = \sum_{j \in \mathcal{N}^{(i)}} d^{(ij)}$, dan $\frac{1}{d^{(i)}}$ adalah koefisien normalisasi. Konsep *reciprocal reward* berarti tindakan

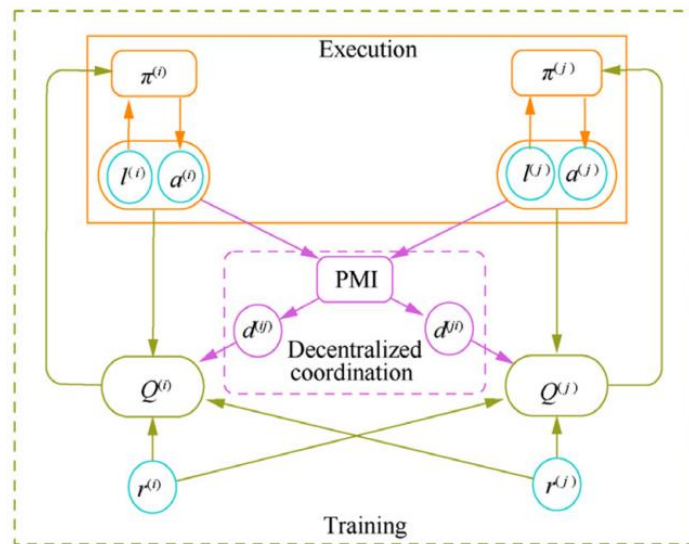
yang diambil oleh satu UAV tidak hanya mempengaruhi *reward* UAV itu sendiri, namun juga UAV yang lain. Modifikasi ini membentuk ulang *reward* yang diterima masing-masing UAV dengan regularisasi yang didefinisikan sebagai *dot product* dari vektor *reward* semua UAV tetangganya dan vektor ketergantungan yang sesuai antara UAV dan tetangganya. Vektor ketergantungan atau dependensi bisa didapatkan menggunakan *Pointwise Mutual Information (PMI) neural network*. Didefinisikan pasangan variabel acak diskrit (X, Y) maka distribusi gabungan mereka adalah $P(X, Y)$, distribusi marginalnya adalah $P(X)$ dan $P(Y)$ secara berurutan, serta $P(X)P(Y)$ adalah hasil kali dari distribusi marginal. PMI digunakan untuk menghitung ketergantungan dari sebuah pasangan (X, Y) .

$$PMI(X; Y) = \log \frac{P(X, Y)}{P(X)P(Y)} = \log \frac{P(X|Y)}{P(X)} = \log \frac{P(Y|X)}{P(Y)} \quad (2.4)$$

sehingga

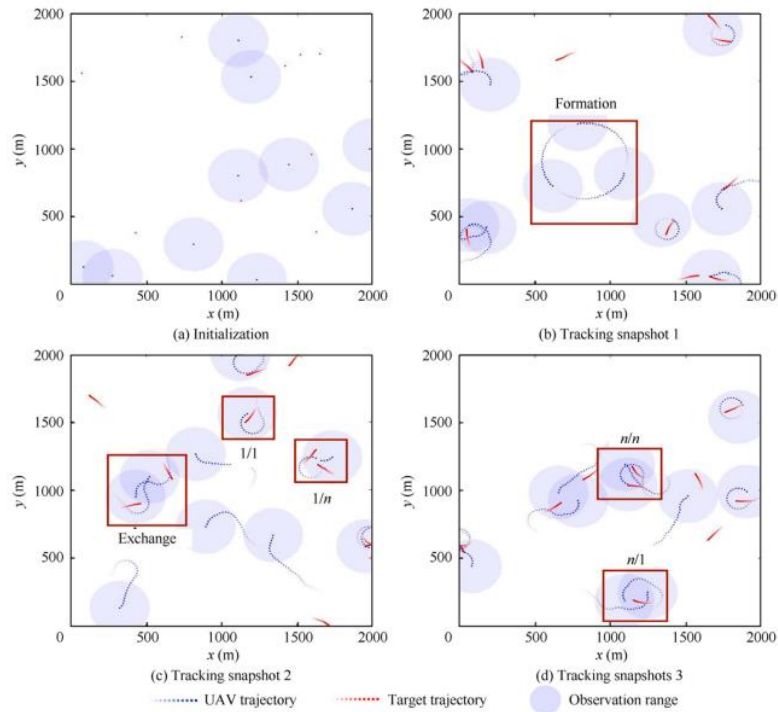
$$d^{(ij)} = PMI(l^{(i)}, a^{(i)}; l^{(j)}, a^{(j)}) \quad (2.5)$$

Kemudian, kebijakan kooperatif yang akan digunakan oleh seluruh UAV dibuat menggunakan algoritma *Reciprocal Reward Multi-Agent Actor-Critic (MAAC-R)*.



Gambar 2.3 Ilustrasi algoritma MAAC-R

Pada algoritma ini, pengalaman yang dirasakan tiap UAV akan dikumpulkan untuk melatih jaringan actor-critic bersama dan jaringan PMI. Lalu, jaringan yang sudah dilatih akan dibagikan ke semua UAV untuk dijalankan dengan cara desentralisasi.

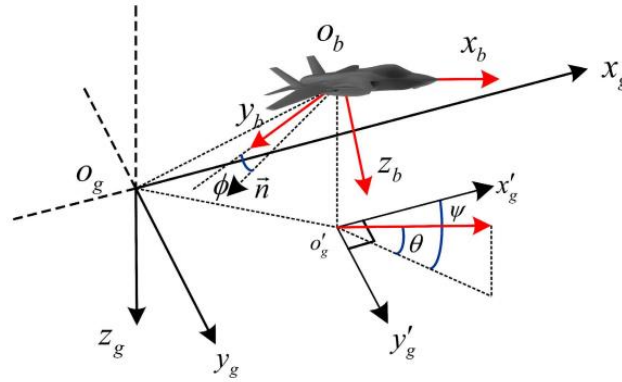


Gambar 2.4 Simulasi MTTG menggunakan algoritma yang diajukan Gambar 2.4 menunjukkan kalau algoritma serta upaya-upaya yang dilakukan sebelumnya membuahkan hasil yang baik. Di gambar 2.4.d terlihat kalau semua target sudah berhasil dilacak.

Paper ini membahas permasalahan MTTG pada lingkungan yang tidak diketahui secara komprehensif. Dimulai dari pemodelan hingga pembuatan algoritma untuk mencari solusi. Hanya saja, target yang dipilih masih merupakan sistem yang bergerak di luasan planar. Belum mempertimbangkan target yang bergerak dalam ruang tiga dimensi. Selain itu, permasalahan dibatasi hanya sampai seluruh target bisa dilacak oleh kelompok UAV.

2.1.2 Collaborative Decision-Making Method for Multi-UAV Based on Multiagent Reinforcement Learning [10]

Kondep dari metode yang digunakan pada *paper* yang ditulis oleh Shaowei Li, Yuhong Jia, Fan Yang, Qingyang Qin, Hui Gao, dan Yaoming Zhou ini sebenarnya bisa dikatakan hampir samadengan *paper* sebelumnya, yaitu model MDP yang diselesaikan dengan algoritma *actor-critic*. Namun, penulis mengimplementasikannya pada skenario yang sama sekali berbeda serta memperkenalkan *gate recurring unit* pada *actor* sehingga UAV bisa membuat keputusan berdasarkan riwayat keputusan sebelumnya. Selain itu, diperkenalkan juga *transformer* pada *critic* agar UAV bisa mengetahui dampak dari perubahan lingkungan dengan lebih baik.



Gambar 2.5 Model UAV yang digunakan dalam *earth-frame* dan *body-frame*

Untuk memodelkan UAV, penulis menggunakan persamaan gerak UAV sederhana pada *earth-frame*.

$$\begin{cases} \dot{x} = v \cos \theta \cos \psi \\ \dot{y} = v \cos \theta \sin \psi \\ \dot{z} = -v \sin \theta \end{cases} \quad (2.6)$$

dimana \dot{x} , \dot{y} , dan \dot{z} adalah perubahan posisi UAV pada sumbu x , y , dan z , secara berurutan. v adalah kecepatan UAV, θ adalah sudut *pitch*, dan ψ adalah sudut *yaw*. Perubahan v , θ , dan ψ sendiri dapat ditulis sebagai

$$\begin{cases} \dot{v} = g(n_x - \sin \theta) \\ \dot{\theta} = \frac{g}{v}(n_z \cos \phi - \cos \theta) \\ \dot{\psi} = \frac{gn_z \sin \phi}{v \cos \theta} \end{cases} \quad (2.7)$$

dengan \dot{v} , $\dot{\theta}$, dan $\dot{\psi}$ adalah percepatan, *pitch rate*, dan *yaw rate*, secara berurutan. n_x adalah *tangential overload*, n_z adalah *normal overload*, g adalah percepatan gravitasi, sedangkan ϕ adalah sudut *roll*. Hal yang menarik adalah UAV bisa menyerang UAV lain. Daerah penyerangannya berbentuk kerucut dan probabilitas keberhasilan serangan ditentukan oleh jarak dan posisi relatif UAV terhadap target.

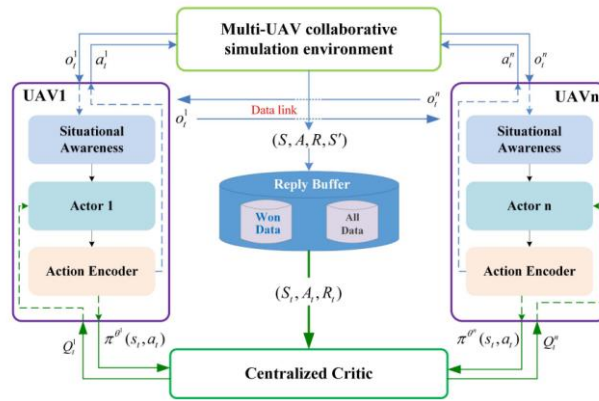
Pertama-tama, permasalahan akan dimodelkan dalam bentuk Dec-POMDP. Fungsi dari kondisi unggul dapat ditulis sebagai

$$A_\pi(s, a) = Q_\pi(s, a) - V_\pi(s) \quad (2.8)$$

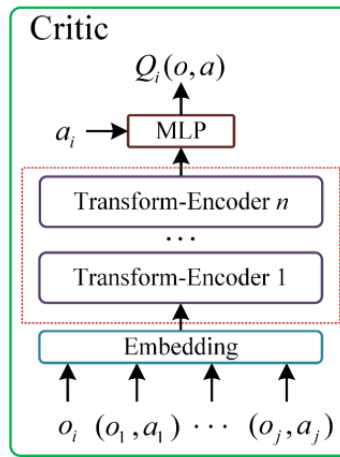
dan setiap UAV akan memiliki fungsi *reward* yang sama. Mereka berkerja sama untuk mencapai *reward* maksimum yang dirumuskan sebagai

$$J(\pi) \triangleq \mathbb{E}_{a_t, s_t} [\sum_t \gamma^t R(s_t, a_t)] \quad (2.9)$$

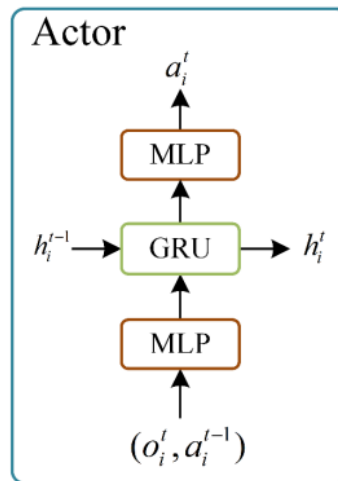
Sebelum masuk ke algoritma, ada konsep penting yang dipakai, yaitu *Centralized Training Decentralized Execution* (CTDE). Pada CTDE, saat proses *training* dilakukan, terdapat *critic* tersentralisasi yang menerima *state-action* gabungan sebagai *input* dan mengeluarkan estimasi *reward* yang didapatkan. Dengan begitu, tiap UAV bisa menerima informasi dari UAV-UAV lain secara *global*. Sedangkan saat eksekusi, masing-masing UAV akan bertindak secara desentralisasi, hanya bergantung pada *state* lokal. Kemudian konsep ini diadopsi oleh algoritma utama untuk menyelesaikan persoalan Dec-POMDP, *Multi-Agent Transformer-Based Actor-Critic* (MATAC).



Gambar 2.6 Ilustrasi algoritma MATAc. Komponen utamanya adalah *centralized critic*, beberapa *distributed actor*, dan *reply buffer*.

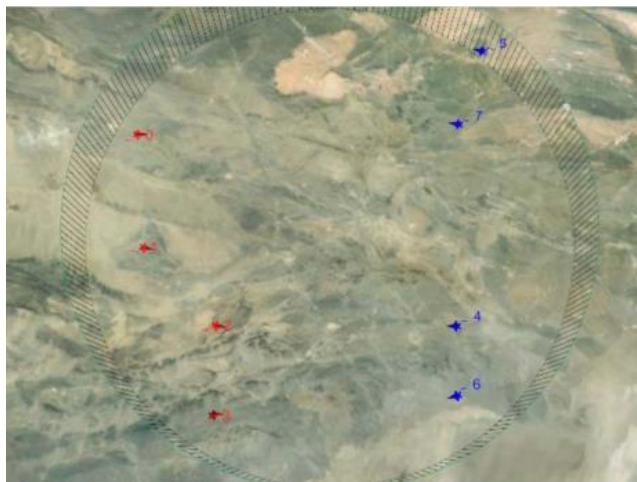


Gambar 2.7 Komponen penyusun *critic* dengan lebih rinci



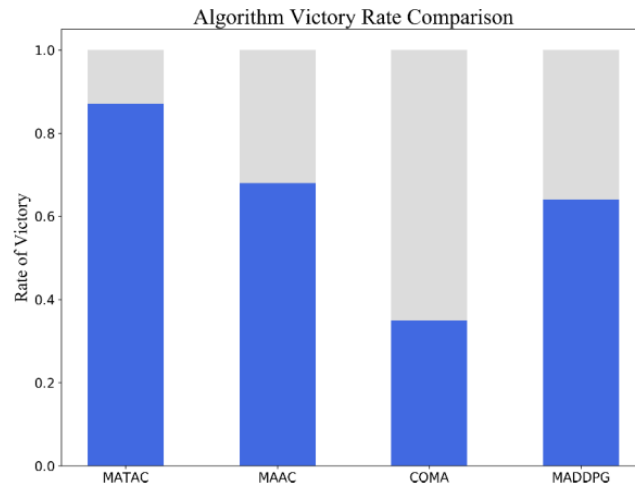
Gambar 2.8 Komponen penyusun *actor* dengan lebih rinci

Pada gambar 2.6, algoritma MATAC secara keseluruhan ditunjukkan. Proses *training* dibagi menjadi dua, yaitu pengumpulan data dan pembaruan algoritma. Proses pengumpulan data adalah saat setiap UAV menghasilkan data dari interaksinya dengan lingkungan, yang dianalisis dan disimpan dalam *buffer replay*. Data kemenangan disalin ke *won data*. Ketika data mencapai jumlah yang ditentukan, pembaruan strategi UAV dimulai, saat ini kumpulan data diambil dari *buffer replay* untuk memperbarui *critic* dan *actor* masing-masing sesuai dengan formula pembaruan. Saat eksekusi, setiap UAV memperoleh informasi situasi saat ini berdasarkan sensor dan pertukaran informasi lokal. Prosesnya sama dengan proses pengumpulan data, hanya saja tidak ada operasi penyimpanan data.



Gambar 2.9 Ilustrasi simulasi yang dilakukan pada [10]

Simulasi dilakukan dengan skenario 4 UAV merah melawan 4 UAV biru. Seluruh spesifikasi UAV merah dan biru adalah sama kecuali UAV merah dilatih oleh algoritma MATAC sedang UAV biru menggunakan strategi yang lebih kaku. Kemenangan didapat oleh tim yang bisa menghancurkan seluruh UAV lawan. Sebagai validasi, MATAC dibandingkan dengan beberapa algoritma lain dan terlihat dari gambar 2.10 kalau MATAC adalah algoritma terbaik dengan kemungkinan menang yang lebih tinggi daripada algoritma-algoritma lain.



Gambar 2.10 Perbandingan rasio kemenangan MATAc dibanding algoritma-algoritma lain dari 300 percobaan

Hal yang bisa diambil dari *paper* ini adalah konsep CTDE dan perancangan solusi untuk mencapai hasil yang lebih baik dari berbagai parameter. Namun, disini belum ada pembagian tugas secara eksplisit karena skenario yang diungkap memang jauh berbeda. Pembagian tugas secara tidak langsung terjadi karena kelompok ingin mengusahakan lebih banyak UAV di kelompoknya yang berada di posisi unggul. Hal yang lebih ditonjolkan adalah kemampuan algoritma untuk mempelajari beberapa manuver kompleks seperti *baiting*.

2.1.3 UAV Formation Shape Control via Decentralized Markov Decision Processes [11]

Paper yang ditulis oleh Md Ali Azam, Hans D. Mittelmann, dan Shankarachary Ragi ini ingin melakukan kontrol formasi dari sekumpulan UAV menggunakan metode *Decentralized Markov Decision Processes* (Dec-MDP). Skenario yang digunakan adalah menggerakkan sistem multi-UAV dari posisi awal ke posisi lain sambil membentuk formasi tiga dimensi. Untuk mempermudah proses komputasi, digunakan metode *programming* dinamik, *Nominal Belief-state Optimization* (NBO). Salah satu perbedaan yang paling terlihat dari dua *paper* sebelumnya adalah penggunaan metode yang sepenuhnya desentralisasi. Konsep desentralisasi yang dimaksud adalah tiap UAV harus memaksimalkan vektor kontrol

$[a_k^i, a_{nn}^i]$ saat waktu k . a_k^i adalah vektor kontrol UAV i dan a_{nn}^i adalah vektor kontrol untuk tetangga-tetangga terdekatnya.

Asumsi untuk masing-masing UAV masih sama seperti sebelumnya, yakni memiliki jangkauan sensor dan komunikasi yang terbatas. Permasalahan kontrol formasi multi-UAV dimodelkan menggunakan model *Decentralized Markov Decision Process* (Dec-MDP). Ada beberapa komponen yang mengisi model Dec-MDP, antara lain himpunan agen I , state s , tindakan a , fungsi transisi state $s_{k+1}^i = \psi(s_k^i, a_k^i) + \mathcal{W}_k^i$, dan cost function $C(s_k, a_k)$. ψ adalah model gerak UAV yang dipengaruhi oleh state sekarang dan tindakan yang diambil atau bisa dianggap sebagai dinamika kontrol. Sedangkan untuk $C(s_k, a_k)$, didefinisikan $b_k^i(s_k^i, a_k^{nn})$ sebagai sistem state lokal UAV i . Ada pula d^i , posisi yang harus dituju, dan $d_{coll,thresh}$ adalah jarak aman antar UAV untuk menghindari tabrakan. Maka dari itu, cost function lokal dari UAV adalah

$$\begin{aligned} c(b_k^i, a_k^i, a_k^{nn}) &= w_1 [dist(s_k^{i,pos}, d^i) + dist(s_k^{nn,pos}, d^{nn})] \\ &+ w_2 [dist(s_k^i, s_k^{nn})^{-1} \mathbb{I}(dist(s_k^i, s_k^{nn}) < d_{coll,thresh})] \end{aligned} \quad (2.10)$$

dimana $s_k^{i,pos}$ adalah lokasi UAV i , w_1 dan w_2 adalah parameter pembobot, $dist(a, b)$ adalah jarak lokasi a dan b , serta $\mathbb{I}(a)$ adalah fungsi indikator. $\mathbb{I}(a) = 1$ jika argumen a benar dan 0 jika sebaliknya.

Dengan meminimalkan fungsi (2.10), tiap UAV akan mengoptimalkan perintah kontrolnya sendiri dan tetangga-tetangganya. Namun, hanya mengimplementasikan kontrol lokalnya sendiri tanpa memakai milik tetangga. Bagian pertama dari cost function akan membuat UAV mencapai destinasinya sedang bagian kedua akan meminimalkan tabrakan antar UAV. Tujuan akhir dari permasalahan Dec-MDP adalah meminimalkan ekspektasi kumulatif dari local cost masing-masing UAV pada horizon (H) tertentu.

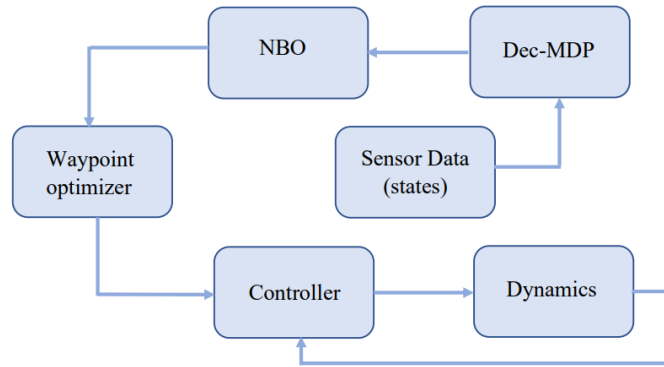
$$\min_{\{a_k^i, a_k^{nn}\}, k=0, \dots, H-1} E[\sum_{k=0}^{H-1} c(b_k^i, a_k^i, a_k^{nn}) | b_0^i] \quad (2.11)$$

$E[\cdot]$ menunjukkan evolusi stokastik dari *local state* seiring waktu yang muncul karena adanya variabel acak pada persamaan dinamika UAV (berupa *noise*).

Karena memperoleh ekspektasi dari persamaan (2.11) secara tepat tidak dapat dilakukan, digunakan pendekatan NBO. NBO mendekati ekspektasi dengan mengasumsikan bahwa semua variabel acak masa depan memiliki nilai nominal berupa nilai rata-rata. Karena variabel acak yang disebutkan di atas dimodelkan sebagai Gaussian rata-rata nol, nilai nominalnya hanyalah nol. Singkatnya, pendekatan NBO mendekati fungsi biaya kumulatif dalam Persamaan (2) dengan mengganti ekspektasi yang awalnya berupa lintasan acak *state* dari waktu ke waktu dengan urutan *state* yang diperoleh dengan mengganti variabel acak masa depan dengan nol. Fungsi objektifnya adalah sebagai berikut

$$J(b_0^i) \approx \sum_{k=0}^{H-1} c(\hat{b}_k^i, a_k^i, a_k^{nn}) \quad (2.12)$$

dimana $\hat{b}_1^i, \hat{b}_2^i, \dots, \hat{b}_{H-1}^i$ adalah urutan nominal *state* lokal.

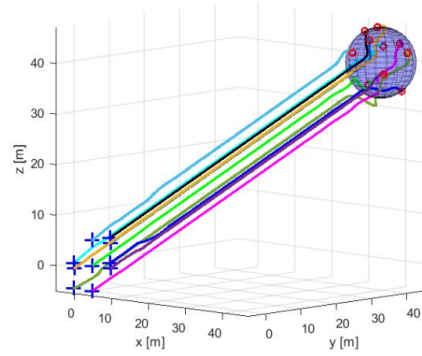


Gambar 2.11 Arsitektur kontrol UAV pada [11]

Kontrol UAV yang dilakukan bisa dilihat pada gambar 2.11. Kontrol dilakukan secara linier dengan mengasumsikan sudut-sudut gerak UAV cukup kecil sehingga melinierkan fungsi trigonometri yang ada.

Pengujian untuk algoritma yang sudah dirancang dilakukan menggunakan aplikasi Matlab dengan dua skenario, tanpa rintangan dan dengan rintangan. Seluruh UAV sadar dengan bentuk dan posisi rinci dari formasi yang diinginkan. Masing-masing UAV secara acak memilih posisi

yang akan dituju dan menggunakan NBO untuk mencapainya. Untuk mengukur performa algoritma kontrol formasi, didefinisikan parameter *benchmark* T_c dan T_f . T_c adalah waktu perhitungan perintah kontrol yang optimal. Sedangkan T_f adalah waktu yang dibutuhkan multi-UAV untuk mencapai bentuk formasi. Selain itu, metode Dec-MDP akan dibandingkan dengan pendekatan sentralisasi sebagai pembanding.

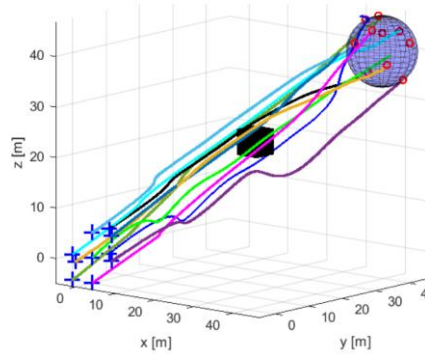


Gambar 2.12 Simulasi yang dilakukan pada [11]

Bila rintangan diperhitungkan, persamaan (2.10) perlu dimodifikasi menjadi

$$\begin{aligned}
 c(b_k^i, a_k^i, a_k^{nn}) &= w_1 [dist(s_k^{i,pos}, d^i) + dist(s_k^{nn,pos}, d^{nn})] \\
 &+ w_2 [dist(s_k^i, s_k^{nn})^{-1} \mathbb{I}(dist(s_k^i, s_k^{nn}) < d_{coll,thresh})] \\
 &+ w_3 [dist(s_k^i, s_k^{obs})^{-1} \mathbb{I}(dist(s_k^i, s_k^{obs}) < d_{coll,obs})]
 \end{aligned} \tag{2.13}$$

dimana s_k^{obs} adalah lokasi dari rintangan, $d_{coll,obs}$ adalah batas aman terhadap rintangan.

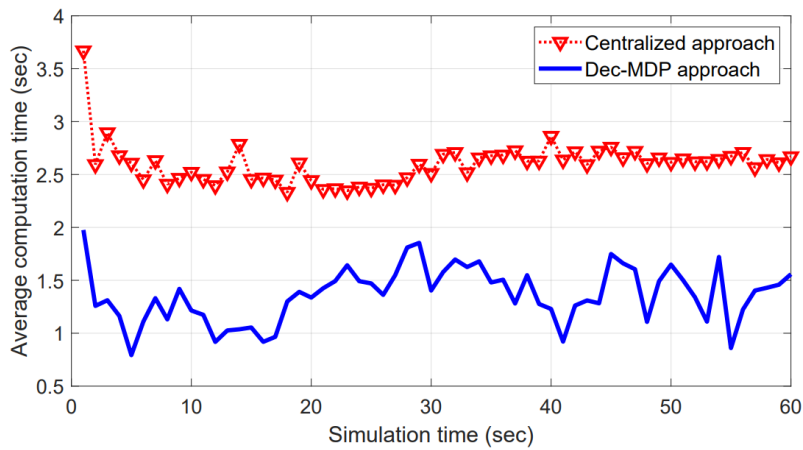


Gambar 2.13 Simulasi dengan rintangan

Berdasarkan parameter *benchmark* pada tabel 2.1, terlihat kalau pendekatan pendekatan secara desentralisasi lebih baik daripada sentralisasi.

Tabel 2.1 Waktu rata-rata T_f

	Dec-MDP	Sentralisasi
T_f (sekon)	16.7	25.98



Gambar 2.14 Perbandingan T_c dari kedua pendekatan

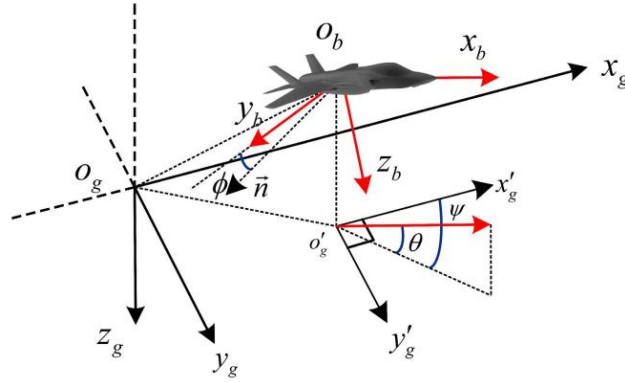
Hal yang bisa diambil dari *paper* ini adalah konsep Dec-MDP untuk memodelkan dan menyelesaikan permasalahan kontrol formasi. Namun, masalah yang diangkat belum mempertimbangkan skenario yang ingin diangkat pada tesis ini secara lengkap, antara lain *target tracking*. Selain itu, pemodelan yang digunakan masih menggunakan metode Dec-MDP, bukan Dec-POMDP. Target dari masing-masing UAV juga sudah ditentukan, meskipun UAV mana menuju ke target mana dipilih secara acak.

2.2 Teori Dasar

Pada sub bab ini dituliskan beberapa teori dasar yang dibutuhkan untuk menunjang perumusan dan penyelesaian masalah yang dihadapi dalam mengerjakan tesis. Pada bagian awal terdapat teori mengenai UAV secara umum dan konsep gerak dari UAV. Lalu, dilanjutkan dengan bahasan algoritma MDP dan penyelesaiannya.

2.2.1 Model dinamika UAV [10], [12], [13]

Model UAV yang akan digunakan pada penelitian ini adalah *drone fixed-wing* yang memiliki tiga derajat kebebasan (3-DOF). Pergerakannya dibagi berdasarkan *earth frame* F_g dan *body frame* F_b .



Gambar 2.15 Ilustrasi model *drone fixed-wing* dengan tiga derajat kebebasan. Jika dilihat dari titik origin dan sumbu-sumbu koordinat, F_g dapat dinotasikan sebagai (o_g, x_g, y_g, z_g) dimana $o_g x_g$ adalah arah utara, $o_g y_g$ menunjuk ke timur, dan $o_g z_g$ mengarah ke tanah. F_g ini berguna untuk menunjukkan posisi dan orientasi dari UAV. Sedangkan F_b adalah kerangka acuan yang titik asalnya adalah titik massa UAV. Persamaan gerak UAV pada F_g bisa dituliskan dengan

$$\begin{cases} \dot{x} = v \cos \theta \cos \psi \\ \dot{y} = v \cos \theta \sin \psi \\ \dot{z} = -v \sin \theta \end{cases} \quad (2.14)$$

dimana \dot{x} , \dot{y} , dan \dot{z} adalah perubahan posisi UAV pada sumbu x , y , dan z , secara berurutan. v adalah kecepatan UAV, θ adalah sudut *pitch*, dan ψ adalah sudut *yaw*.

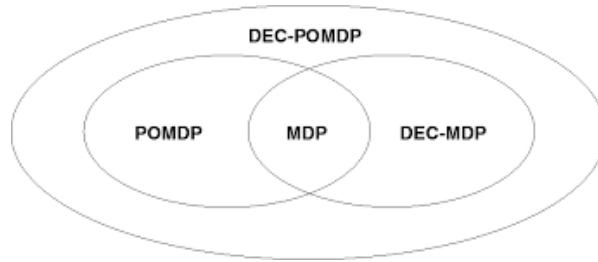
Diasumsikan bahwa tidak ada selip selama gerakan UAV, daya dorong mesin ke depan sepanjang arah sumbu x UAV, dan UAV terbang di lingkungan tanpa angin. Kelebihan gaya dorong dan aerodinamis yang bekerja pada UAV dalam penerbangan dapat diuraikan menjadi beban tangensial dan normal. Persamaan dinamis pada pusat massa UAV yang mencakup v , θ , dan ψ dapat ditulis sebagai

$$\begin{cases} \dot{v} = g(n_x - \sin \theta) \\ \dot{\theta} = \frac{g}{v}(n_z \cos \phi - \cos \theta) \\ \dot{\psi} = \frac{gn_z \sin \phi}{v \cos \theta} \end{cases} \quad (2.15)$$

dengan \dot{v} , $\dot{\theta}$, dan $\dot{\psi}$ adalah percepatan, *pitch rate*, dan *yaw rate*, secara berurutan. n_x adalah *tangential overload*, n_z adalah *normal overload*, g adalah percepatan gravitasi, sedangkan ϕ adalah sudut *roll*.

2.2.2 Decentralized Partially Observable Markov Decision Process (Dec-POMDP) [3], [10]

Model POMDP dapat dianggap sebagai generalisasi dari MDP yang mempertimbangkan ketidakpastian dalam *state* prosesnya. MDP sendiri adalah *framework* matematis yang digunakan untuk memodelkan permasalahan pengambilan keputusan sambil mempertimbangkan keacakan. Jika *state* pada MDP (standar) diasumsikan sepenuhnya dapat diamati setiap saat, informasi tentang status POMDP mungkin tidak lengkap.



Gambar 2.16 Ilustrasi hubungan Dec-POMDP terhadap algoritma MDP lainnya

Pada setiap langkah t , setiap UAV membuat keputusan aksinya berdasarkan informasi lokal dan semua UAV melaksanakan aksi bersama untuk merubah lingkungan. Bila diasumsikan ada n UAV, model Dec-POMDP didefinisikan sebagai

$$(N, S, A, T, R, O, Z, \gamma) \quad (2.16)$$

dimana N adalah himpunan n UAV, S adalah *state space* dari state s , $s \in S$, $A: \{A^{(1)}, A^{(2)}, \dots, A^{(n)}\}$ menotasikan *joint action space* dari tiap UAV, $a^{(i)} \in A^{(i)}$, probabilitas $P(s'|s, a) \rightarrow [0, 1]$ menotasikan model probabilitas transisi dari *state* s ke *state* selanjutnya s' , $s' \in S$, setelah mengeksekusi *joint action* $a: \{a^{(1)}, a^{(2)}, \dots, a^{(n)}\}$, $R(s, a): \{r^{(1)}, r^{(2)}, \dots, r^{(n)}\}$ adalah fungsi *joint reward* yang

didapatkan setelah mengeksekusi a pada *state* s . $O: \{O^{(1)}, O^{(2)}, \dots, O^{(n)}\}$ menotasikan *joint observation space* dari semua UAV, $o: \{o^{(1)}, o^{(2)}, \dots, o^{(n)}\}$ dan $o^{(i)} \in O^{(i)}$, $Z: o^{(i)} = Z(s, i)$ adalah model observasi individu dari tiap UAV berdasarkan *state* s . Sedangkan $\gamma \in [0, 1]$ adalah konstanta *discount factor*. Pada Dec-POMDP, fungsi *reward* adalah parameter yang menunjukkan hubungan antar agen. Jika $r^{(1)} = r^{(2)}$ maka mereka kooperatif. Sebaliknya, jika $r^{(1)} = -r^{(2)}$ maka mereka kompetitif.

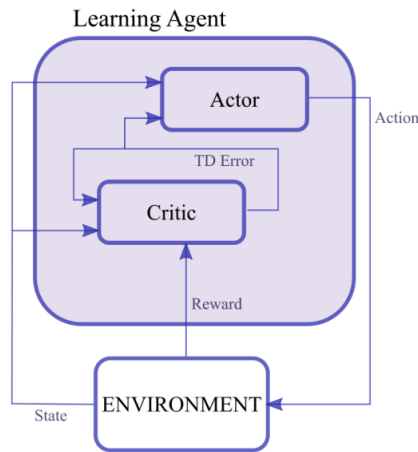
Tiap UAV akan menentukan tindakannya berdasarkan informasi observasi lokal masing-masing. Kebijakan dari agen i bisa ditulis sebagai $\pi^i = O^{(i)} \rightarrow A^{(i)}$. Sedangkan kebijakan gabungan dari masing-masing UAV adalah $\pi: \{\pi^{(1)}, \pi^{(2)}, \dots, \pi^{(n)}\}$. Mengingat *joint observation* o dan kebijakan gabungan π , $V_\pi^{(i)}(o) = E_\pi \left[\sum_{t=0}^{\infty} \gamma^t r_t^{(i)} | o_{t=0} = o \right]$ menotasikan fungsi *state-value* dari UAV- i dan fungsi *action-value* dari tindakan gabungan a adalah $Q_\pi^{(i)}(o, a) = E_\pi \left[\sum_{t=0}^{\infty} \gamma^t r_t^{(i)} | (o, a)_{t=0} = (o, a) \right]$. Tujuan utama dari model Dec-POMDP adalah menemukan π terbaik, π^* yang memaksimalkan *reward* pada persamaan $Q_\pi^{(i)}(o, a)$.

2.2.3 Reinforcement Learning : Actor-Critic [9]

Tujuan utama dari Dec-POMDP adalah merumuskan kebijakan π terbaik untuk digunakan oleh para UAV. Salah satu cara untuk menyelesaikannya adalah menggunakan metode MARL, yaitu *Actor-Critic*. Metode ini menggunakan fungsi kebijakan (*policy function*) dan fungsi nilai (*value function*, bisa *action-value* ataupun *state-value*) untuk mendapatkan solusi kebijakan. Ada beberapa variasi dari algoritma ini, antara lain *Q actor-critic* dan *advantage actor-critic*.

Pada prakteknya, menemukan π^* bukanlah persoalan yang mudah. Maka dari itu, pada MARL, sering kali kebijakan Dec-POMDP direpresentasikan sebagai *neural network*. Jaringan kebijakan dinotasikan sebagai π_w dimana w adalah parameter jaringan. Dengan kata lain, masing-masing kebijakan dari UAV ke-1 hingga ke- n akan diberikan w tertentu.

$V_{\pi_w}(o)$ adalah pengembalian yang diharapkan saat *state* waktu tertentu ketika suatu tindakan dipilih berdasarkan kebijakan π_w . Sama dengan fungsi kebijakan, fungsi nilai juga didekati sebagai *neural network* dengan parameter v sehingga $V_{\pi_w}(o) \approx V_v^{\pi_w}(o)$. Algoritma *actor-critic* menggunakan fungsi nilai saat menghitung *policy gradient* untuk memperbarui fungsi kebijakan. Ketika seorang "aktor" memilih tindakan di bawah kebijakan saat ini π_w , seorang "kritikus" memperbarui parameter v dari jaringan nilai dengan memasukkan *reward* yang didapat oleh tindakan yang dipilih. Kemudian, *critic* menggunakan fungsi nilai yang diperbarui untuk memandu *actor* dalam memperbarui parameter w untuk fungsi kebijakannya.



Gambar 2.17 Ilustrasi algoritma *actor-critic*

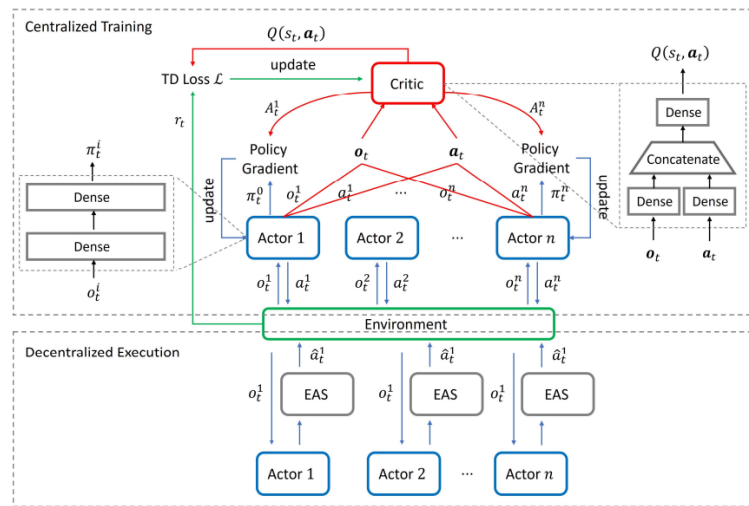
2.2.4 Centralized Training Decentralized Execution (CTDE) [5], [10]

Konsep atau arsitektur dari MARL secara umum bisa dibagi menjadi tiga, yaitu sentralisasi (*centralized*), terdistribusi (*distributed*), dan CTDE. Ketiganya memiliki kelebihan dan kekurangan masing-masing. Pemilihan arsitektur mana yang sebaiknya digunakan ditentukan oleh fungsi dan tujuan yang diinginkan perancang.

Pada konsep terdistribusi, tiap agen akan dilatih secara terpisah dari agen-agen lain dan jaringan kebijakannya akan menentukan tindakan yang akan diambil selanjutnya berdasarkan observasi lokal. Masing-masing agen akan melihat agen lain sebagai bagian dari lingkungan sehingga karakter lingkungan akan berubah jika

kebijakan agen lain berubah. Maka dari itu, sistem menjadi dinamis dan tidak stasioner atau *non-Markovian*.

Konsep sentralisasi dapat mengatasi sifat tidak stasioner lingkungan dengan mempelajari kebijakan bersama dari semua agen. Masukan yang diberikan adalah observasi gabungan dari tiap agen dan keluarannya tidak lain adalah tindakan gabungan untuk semua agen. Kelemahan metode ini ada pada skalabilitas dimana jumlah agen yang begitu banyak akan meningkatkan kompleksitas komputasi secara eksponensial.

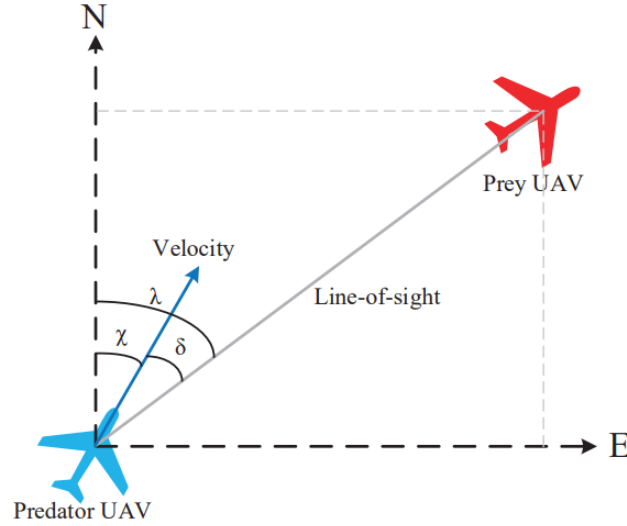


Gambar 2.18. Ilustrasi arsitektur CTDE

Dengan adanya kelemahan-kelemahan diatas, konsep CTDE menjadi opsi yang menarik untuk dipilih. Metode sentralisasi digunakan selama pelatihan dan setelahnya, agen akan mengambil keputusan hanya berdasarkan observasi lokal menggunakan jaringan kebijakan yang sudah dilatih. Pada masa pelatihan, digunakan *centralized critic* yang menerima informasi *state* serta tindakan gabungan sebagai masukan dan menghasilkan estimasi *reward*. Dengan demikian, tiap agen bisa mendapatkan informasi dari agen lain. Sementara itu, masing-masing agen akan mempelajari kebijakan terdesentralisasi yang hanya bergantung pada *state* lokal untuk eksekusi. Kebijakan tersebut tidak memerlukan informasi dari agen lain, yang membantu mengurangi masalah pada kondisi dengan agen skala besar.

2.2.5 Proportional Pursuit (PP) dan Proportional Navigation (PN) [12]

Pada kontrol pemanduan *Proportional Pursuit* (PP), perubahan sudut arah *pursuer* sebanding dengan selisih sudut antara arah kecepatan dan garis pandang ke *evader*. Sedangkan pada *Proportional Navigation* (PN), perubahan sudut arah *pursuer* sebanding dengan kecepatan angular dari garis pandang ke *evader*.



Gambar 2.19. Ilustrasi arah gerak, garis pandang, dan sudut deviasi.

Seperti pada gambar 2.19, arah gerak *pursuer* atau predator didefinisikan sebagai sudut antara proyeksi kecepatan *pursuer* terhadap arah utara.

PP menggerakkan sudut deviasi *pursuer* dari arah kecepatan dan garis pandang ke *evader*. Sudut deviasinya dapat dituliskan sebagai

$$\delta = \chi - \lambda \quad (2.17)$$

Perubahan χ didapatkan dari persamaan

$$\dot{\chi} = -K\delta \quad (2.18)$$

dimana K adalah konstanta positif. Sebaliknya, pada PN, perubahan χ didapatkan dengan menggunakan persamaan

$$\dot{\chi} = N\dot{\lambda} \quad (2.19)$$

dimana N juga adalah konstanta positif.

Kedua metode bisa digabungkan dengan menjumlahkan persamaan (2.18) dan (2.19),

$$\dot{\mathcal{X}} = N\dot{\lambda} - K\delta \quad (2.20)$$

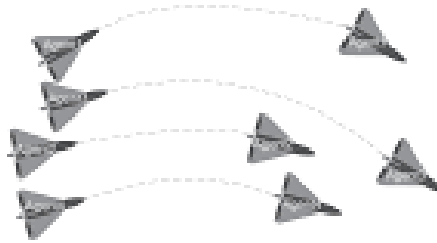
Menggunakan gabungan kedua strategi kontrol akan meningkatkan kecepatan respon dan mengurangi *overshoot* saat pengejaran.

Halaman ini sengaja dikosongkan

BAB 3

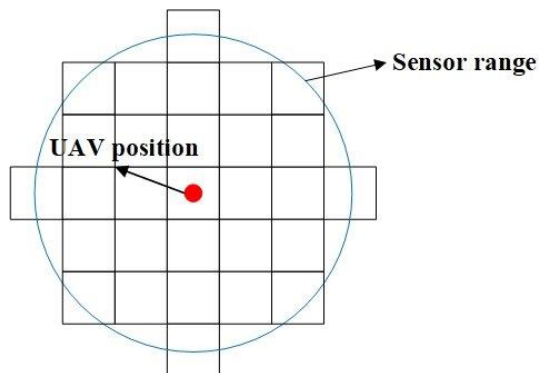
METODOLOGI PENELITIAN

Bab ini membahas permasalahan yang diangkat, konseptualisasi sistem yang diusulkan, langkah-langkah perancangan, serta pengujian yang akan dilakukan.



Gambar 3.1 Ilustrasi skenario penelitian yang akan dibuat

Pada sebuah lingkungan diinisiasi beberapa agen berupa UAV yang terbagi menjadi dua kelompok, *pursuer* dan *evader*. UAV *pursuer* adalah UAV yang bertugas untuk melacak dan menghancurkan semua UAV *evader*. Sebaliknya, UAV *evader* adalah UAV yang berusaha untuk menghindari *pursuer*. Ilustrasinya dapat dilihat pada gambar 3.1. Diasumsikan kalau tiap UAV memiliki sensor tiga dimensi yang terbatas sehingga masing-masing kelompok tidak bisa mengetahui posisi UAV dari kelompok lain jika berada di luar jangkauan sensornya.



Gambar 3.2 Ilustrasi jangkauan sensor UAV yang terbatas

Adapun konsep penghancuran yang dimaksud adalah UAV *pursuer* akan menabrakan dirinya ke UAV *evader*. Dibutuhkan satu *pursuer* untuk

menghancurkan satu *evader*. Konsep ini dikenal dengan sebutan *kamikaze* atau bunuh diri. Keseluruhan proses simulasi bisa dikatakan selesai jika seluruh *evader* sudah dihancurkan atau sudah tidak ada lagi UAV *pursuer* yang tersisa. Namun, permasalahan yang dibawa menjadi semakin rumit karena pada awalnya, kelompok *evader* akan bergerak bersama dalam suatu formasi bukan terpisah-pisah.

Secara menyeluruh, skenario yang dibahas sebelumnya memiliki tiga permasalahan utama jika dilihat dari sudut pandang *pursuer*, yakni

1. *Tracking*, yaitu melakukan pelacakan posisi *evader*.
2. *Following* dan *task allocating*, yaitu menganalisa jumlah *evader*, memanggil bantuan jika diperlukan, lalu membagi *evader* diantara *pursuer*.
3. *Intercepting*, yaitu menghancurkan seluruh *evader* sesuai dengan pembagian.

Karena lebarnya skenario yang dilakukan, diperlukan beberapa asumsi dan strategi kontrol yang berbeda dari sebelumnya untuk memastikan seluruh tugas bisa diselesaikan. Sebagai contoh, memanggil bantuan tidak pernah dipertimbangkan pada *paper-paper* di studi literatur. Hal ini dikarenakan skenario yang dibahas hanya sebagian, misalnya hanya sampai *following*, dan *evader* bergerak sendiri-sendiri secara acak. Jika satu *pursuer* menemukan kelompok *evader* lalu langsung menghancurkan salah satu *evader* maka teman-teman *pursuer* yang lain masih harus melakukan *tracking* dan menjadikan proses pencarian kurang efektif.

Untuk menyelesaikan tugas-tugas diatas, digunakan algoritma pengambilan keputusan berdasarkan MDP. Berikut beberapa langkah yang akan dilakukan untuk melaksanakan penelitian.

3.1 Pemodelan UAV



Gambar 3.3 Drone kamikaze

Jenis UAV yang akan digunakan pada penelitian adalah UAV tipe *fixed-wing* atau *drone*. Pemodelan matematis yang digunakan sesuai dengan dasar teori sub bab 2.2.1 dimana *drone* akan dimodelkan dari pergerakannya dan diasumsikan memiliki kontrol dalam yang baik sehingga bisa mengikuti posisi ataupun kecepatan referensi yang diberikan. Model yang sederhana dipilih agar penelitian bisa difokuskan pada perancangan algoritma pengambilan keputusan. Persamaan gerak UAV adalah

$$\begin{cases} \dot{x} = v \cos \theta \cos \psi \\ \dot{y} = v \cos \theta \sin \psi \\ \dot{z} = -v \sin \theta \end{cases} \quad (3.1)$$

dimana \dot{x} , \dot{y} , dan \dot{z} adalah perubahan posisi UAV pada sumbu x , y , dan z , secara berurutan. v adalah kecepatan UAV, θ adalah sudut *pitch*, dan ψ adalah sudut *yaw*. Sedangkan persamaan dinamis pada pusat massa UAV yang mencakup v , θ , dan ψ dapat ditulis sebagai

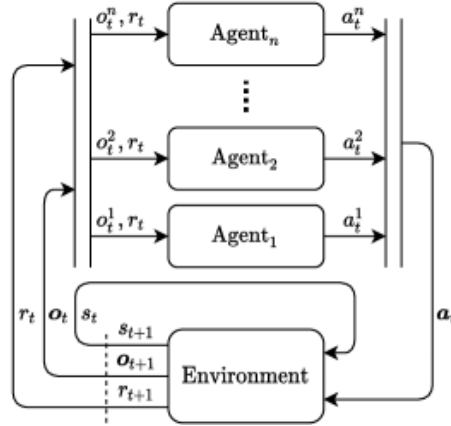
$$\begin{cases} \dot{v} = g(n_x - \sin \theta) \\ \dot{\theta} = \frac{g}{v}(n_z \cos \phi - \cos \theta) \\ \dot{\psi} = \frac{gn_z \sin \phi}{v \cos \theta} \end{cases} \quad (3.2)$$

dengan \dot{v} , $\dot{\theta}$, dan $\dot{\psi}$ adalah percepatan, *pitch rate*, dan *yaw rate*, secara berurutan. n_x adalah *tangential overload*, n_z adalah *normal overload*, g adalah percepatan gravitasi, sedangkan ϕ adalah sudut *roll*.

3.2 Pemodelan Dec-POMDP

Selain *plant*, permasalahan yang sudah dijelaskan sebelumnya perlu dimodelkan ke bentuk MDP. Diingat kembali kalau MDP adalah *framework* matematis yang digunakan untuk memodelkan permasalahan pengambilan keputusan sambil mempertimbangkan keacakan. Bagaimana MDP bekerja, terutama pada sistem multi-agen dapat dilihat pada gambar 3.4. Berdasarkan gambar 3.4, tindakan yang diambil oleh masing-masing agen dipengaruhi oleh kondisi dari lingkungan (*environment*). Kemudian, lingkungan akan bereaksi dengan memberikan data hasil observasi (o_t) dan *reward* (r_t). Data observasi dan *reward* akan diolah oleh masing-masing agen (atau berbeda bergantung pada konsep

learning yang dipilih). Dari hasil pengolahan data, tiap agen akan memilih tindakan apa yang sebaiknya diambil berikutnya.



Gambar 3.4 Diagram blok interaksi agen terhadap lingkungan

Adapun variasi Dec-POMDP akan dipilih karena paling sesuai dengan kondisi yang digambarkan pada penjelasan diawal bab. Cara memodelkannya sempat dijelaskan pada dasar teori sub bab 2.2.2. Pada setiap langkah t , setiap UAV membuat keputusan aksinya berdasarkan informasi lokal. Bila diasumsikan ada n UAV, model Dec-POMDP didefinisikan sebagai

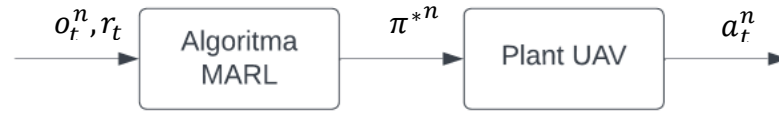
$$(N, S, A, T, R, O, Z, \gamma) \quad (3.3)$$

dimana N adalah himpunan n UAV, S adalah *state space* dari state s , $s \in S$, $A: \{A^{(1)}, A^{(2)}, \dots, A^{(n)}\}$ menotasikan *joint action space* dari tiap UAV, $a^{(i)} \in A^{(i)}$, probabilitas $P(s'|s, a) \rightarrow [0, 1]$ menotasikan model probabilitas transisi dari *state* s ke *state* selanjutnya s' , $s' \in S$, setelah mengeksekusi *joint action* $a: \{a^{(1)}, a^{(2)}, \dots, a^{(n)}\}$, $R(s, a): \{r^{(1)}, r^{(2)}, \dots, r^{(n)}\}$ adalah fungsi *joint reward* yang didapatkan setelah mengeksekusi a pada *state* s . $O: \{O^{(1)}, O^{(2)}, \dots, O^{(n)}\}$ menotasikan *joint observation space* dari semua UAV, $o: \{o^{(1)}, o^{(2)}, \dots, o^{(n)}\}$ dan $o^{(i)} \in O^{(i)}$, $Z: o^{(i)} = Z(s, i)$ adalah model observasi individu dari tiap UAV berdasarkan *state* s . Sedangkan $\gamma \in [0, 1]$ adalah konstanta *discount factor*.

Untuk permasalahan yang diangkat pada tesis, *state* untuk masing-masing UAV bisa berupa posisi serta kecepatan teman atau target, informasi yang didapat dari komunikasi dengan UAV lain, dst. Sedangkan *action* bisa berupa perubahan

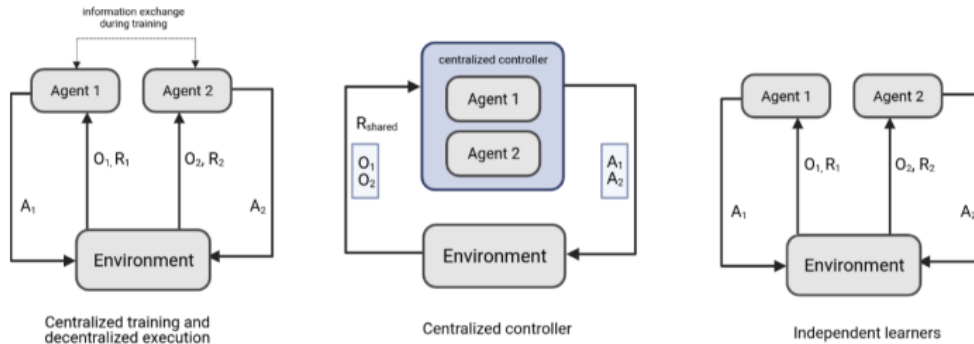
kecepatan atau arah dari UAV, perubahan *mode* operasional, menginisiasi komunikasi, dll.

3.3 Merancang algoritma MARL



Gambar 3.5 Rincian pada tiap blok *Agent* dari gambar 3.4

Untuk menghasilkan keputusan yang baik, merancang kebijakan (*policy*) dari masing-masing agen atau UAV, π^n , menjadi suatu hal yang penting. Berdasarkan [9], untuk menentukan kebijakan terbaik, π^*^n , digunakan algoritma *reinforcement learning*, atau tepatnya MARL untuk persoalan multi-agen. Posisi MARL bisa dilihat dari gambar 3.5. Terdapat cukup banyak macam algoritma MARL yang bisa digunakan dan dipilih algoritma *actor-critic* untuk tesis ini. Penjelasan algoritma bisa dilihat pada dasar teori sub bab 2.2.3.



Gambar 3.6 Gambaran skema pelatihan

Peningkatan terhadap *actor-critic* akan dilakukan. Salah satunya adalah dengan menerapkan konsep CTDE seperti pada dasar teori sub bab 2.2.4. Dengan CTDE, proses terberat dilakukan pada saat *training* sehingga proses eksekusi menjadi ringan. Proses *training* pada CTDE sendiri akan dicoba untuk ditingkatkan dengan mengembangkan pendekatan lain, yaitu menggabungkan proses *learning* secara sentralisasi dan desentralisasi. Penelitian-penelitian sebelumnya hanya menggunakan salah satu metode saja. Pendekatan ini seharusnya bisa diwujudkan

dengan menambahkan satu *critic* lagi untuk masing-masing agen diluar *critic* bersama.

3.4 Metode penghancuran target

Setelah pembagian tugas dilakukan, masing-masing pursuer akan mencegah dan menghancurkan target menggunakan metode *Proportional Pursuit* (PP) dan *Propotional Navigation* (PN) seperti yang dijelaskan pada [12]. Kedua metode ini dirasa cukup dengan asumsi kecepatan *pursuer* lebih cepat daripada *evader*.

3.5 Hipotesa Penelitian

Hipotesa dari penelitian yang akan dilakukan adalah:

1. Keseluruhan permasalahan dapat dimodelkan menjadi satu atau lebih model Dec-POMDP.
2. Dengan algoritma MARL yang dirancang, tiap *pursuer* dapat dilatih untuk memilih tindakan terbaik pada tiap kondisi hingga seluruh tugas dapat diselesaikan.

3.6 Rencana Pengujian

Pengujian akan dilakukan untuk membuktikan hipotesa yang telah dibuat.

Rencana pengujian yang akan dibuat adalah sebagai berikut:

1. Pengujian dengan jumlah *pursuer* lebih banyak dari jumlah *evader*.
2. Pengujian dengan jumlah *pursuer* lebih sedikit dari jumlah *evader*.
3. Pengujian dengan jumlah *pursuer* sama dengan jumlah *evader*.

3.7 Kriteria Pengujian

Kriteria pengujian pada algoritma yang akan dilakukan berdasarkan masing-masing rencana pengujian adalah:

1. Seluruh *evader* hancur dan jumlah *pursuer* yang tersisa adalah jumlah awal dikurangi jumlah *evader*.
2. Beberapa *evader* hancur sejumlah *pursuer* dan jumlah *pursuer* yang tersisa adalah nol.
3. Seluruh *evader* hancur dan jumlah *pursuer* yang tersisa adalah nol.

BAB 4

RENCANA DAN JADWAL KEGIATAN

4.1 Jadwal Kegiatan

Rencana jadwal kegiatan penelitian selama kurang lebih satu semester ditunjukkan pada Tabel 4.1.

Tabel 4.1 Perencanaan Jadwal Kegiatan

No.	Kegiatan	Bulan					Target yang ingin dicapai
		Agu	Sep	Okt	Nov	Des	
1	Studi literatur						Mempelajari penelitian sebelumnya dan memahami landasan teori yang berkaitan dengan penelitian
2	Perancangan awal sistem						Validasi model UAV, pembuatan lingkungan, dan perancangan algoritma
3	Perancangan sistem akhir dan pengujian						Penerapan dan melakukan pengujian terhadap algoritma
4	Analisa dan tinjauan ulang						Melakukan analisa terhadap hasil.
5	Penyusunan laporan akhir						Mendeskripsikan hasil penelitian secara sistematis

							dan jelas untuk penguji
6	Konsultasi dengan dosen pembimbing						Melakukan diskusi kemajuan, memperoleh masukan, koreksi, dan pemecahan masalah

DAFTAR PUSTAKA

- [1] K. Guo, X. Li, and L. Xie, “Simultaneous cooperative relative localization and distributed formation control for multiple UAVs,” *Sci. China Inf. Sci.*, vol. 63, no. 1, pp. 2019–2021, 2020, doi: 10.1007/s11432-018-9603-y.
- [2] Y. Cao and Y. Sun, “Necessary and sufficient conditions for consensus of third-order discrete-time multi-agent systems in directed networks,” *J. Appl. Math. Comput.*, vol. 57, no. 1–2, pp. 199–210, 2018, doi: 10.1007/s12190-017-1101-8.
- [3] W. ZHOU, J. LI, Z. LIU, and L. SHEN, “Improving multi-target cooperative tracking guidance for UAV swarms using multi-agent reinforcement learning,” *Chinese J. Aeronaut.*, vol. 35, no. 7, pp. 100–112, 2022, doi: 10.1016/j.cja.2021.09.008.
- [4] S. Mukhopadhyay and B. Jain, “Multi-agent markov decision processes with limited agent communication,” *IEEE Int. Symp. Intell. Control - Proc.*, pp. 7–12, 2001, doi: 10.1109/isic.2001.971476.
- [5] S. Huang, H. Zhang, and Z. Huang, “Multi-UAV Collision Avoidance using Multi-Agent Reinforcement Learning with Counterfactual Credit Assignment,” 2022, [Online]. Available: <http://arxiv.org/abs/2204.08594>.
- [6] H. R. Lee and T. Lee, “Multi-agent reinforcement learning algorithm to solve a partially-observable multi-agent problem in disaster response,” *Eur. J. Oper. Res.*, vol. 291, no. 1, pp. 296–308, 2021, doi: 10.1016/j.ejor.2020.09.018.
- [7] L. Yue, R. Yang, J. Zuo, M. Yan, X. Zhao, and M. Lv, “Factored Multi-Agent Soft Actor-Critic for Cooperative Multi-Target Tracking of UAV Swarms,” *Drones*, vol. 7, no. 3, p. 150, 2023, doi: 10.3390/drones7030150.
- [8] J. Chen, X. Lan, Y. Zhou, and J. Liang, “Formation Control with Connectivity Assurance for Missile Swarms by a Natural Co-Evolutionary Strategy,” *Mathematics*, vol. 10, no. 22, 2022, doi: 10.3390/math10224244.

- [9] J. Kim, H. Oh, B. Yu, and S. Kim, "Optimal Task Assignment for UAV Swarm Operations in Hostile Environments," *Int. J. Aeronaut. Sp. Sci.*, vol. 22, no. 2, pp. 456–467, 2021, doi: 10.1007/s42405-020-00317-z.
- [10] S. Li, Y. Jia, F. Yang, Q. Qin, H. Gao, and Y. Zhou, "Collaborative Decision-Making Method for Multi-UAV Based on Multiagent Reinforcement Learning," *IEEE Access*, vol. 10, no. September, pp. 91385–91396, 2022, doi: 10.1109/ACCESS.2022.3199070.
- [11] M. A. Azam, H. D. Mittelman, and S. Ragi, "Uav formation shape control via decentralized markov decision processes," *Algorithms*, vol. 14, no. 3, 2021, doi: 10.3390/a14030091.
- [12] B. Tong, J. Liu, and H. Duan, "Multi-UAV Interception Inspired by Harris' Hawks Cooperative Hunting Behavior," *2021 IEEE Int. Conf. Robot. Biomimetics, ROBIO 2021*, pp. 1656–1661, 2021, doi: 10.1109/ROBIO54168.2021.9739214.
- [13] Y. Hou, X. Liang, L. He, and J. Zhang, "Time-Coordinated Control for Unmanned Aerial Vehicle Swarm Cooperative Attack on Ground-Moving Target," *IEEE Access*, vol. 7, pp. 106931–106940, 2019, doi: 10.1109/ACCESS.2019.2932625.