

Studi Implementasi *Multi-Agent Reinforcement Learning* untuk Optimasi Luas Cakupan Jaringan Sensor

Glenn Bonaventura WIJAYA
2017630019

Diajukan untuk memenuhi salah satu syarat mendapatkan gelar Sarjana Teknik di Program Studi Teknik Elektro Konsentrasi Mekatronika, Universitas Katolik Parahyangan.

Panitia Penguji :

Tua Agustinus Tamba, Ph.D., Pembimbing 1

Dr. Christian Fredy Naa, S.Si., M.Si., M.Sc.,
Penguji 1

Faisal Wahab, S.Pd., M.T., Penguji 2

***Study on Implementation of Multi-Agent
Reinforcement Learning Method for Optimizing
Sensor Network Coverage***

Glenn Bonaventura WIJAYA
2017630019

Diajukan untuk memenuhi salah satu syarat mendapatkan gelar Sarjana Teknik di Program Studi Teknik Elektro Konsentrasi Mekatronika, Universitas Katolik Parahyangan.

Panitia Penguji :

Tua Agustinus Tamba, Ph.D., Pembimbing 1

Dr. Christian Fredy Naa, S.Si., M.Si., M.Sc.,
Penguji 1

Faisal Wahab, S.Pd., M.T., Penguji 2

Lembar Persetujuan Selesai



Tugas Akhir berjudul:

Studi Implementasi *Multi-Agent Reinforcement Learning* untuk Optimasi Luas Cakupan Jaringan Sensor

oleh:

Glenn Bonaventura Wijaya
NPM : 2017630019

ini telah diujikan pada Sidang Tugas Akhir 2 (IME 184500) di Program Studi Sarjana Teknik Elektro Konsentrasi Mekatronika, Fakultas Teknologi Industri, Universitas Katolik Parahyangan serta dinyatakan SELESAI.

TANDA PERSETUJUAN SELESAI,

Bandung, 17 Agustus 2022

Ketua Program Studi Sarjana
Teknik Elektro Konsentrasi Mekatronika

Tua Agustinus Tamba, Ph.D.

Pembimbing Pertama,

Tua Agustinus Tamba, Ph.D.

**PERNYATAAN TIDAK MENCONTEK ATAU
MELAKUKAN TINDAKAN PLAGIAT**

Saya yang bertandatangan dibawah ini,

GLENN BONAVENTURA WIJAYA

Dengan ini menyatakan bahwa Buku Tugas Akhir dengan judul:

"STUDI IMPLEMENTASI *MULTI-AGENT REINFORCEMENT LEARNING*
UNTUK OPTIMASI LUAS CAKUPAN JARINGAN SENSOR"

adalah hasil pekerjaan Saya. Seluruh ide, pendapat atau materi dari sumber lain
telah dikutip dengan cara penulisan referensi yang sesuai.

Pernyataan ini Saya buat dengan sebenar-benarnya dan jika pernyataan ini tidak
sesuai dengan kenyataan maka Saya bersedia menanggung sanksi yang akan
dikenakan kepada Saya.

Bandung, 17 Agustus 2022



Glenn Bonaventura Wijaya

NPM: 2017630019

Abstrak

Penggunaan *quadrotor* sebagai alat pengamatan meningkat akhir-akhir ini. Kelebihan *quadrotor* dalam melakukan *vertical take-off and landing* (VTOL). Selain itu kemampuan *quadrotor* melakukan *aerial monitoring* adalah hal yang sangat cocok digunakan untuk mengamati daerah yang tidak dapat terjangkau manusia seperti daerah yang terkena bencana ataupun daerah sumber radioaktif. Terdapat banyak algoritma yang memanfaatkan *quadrotor* untuk melakukan *aerial monitoring* seperti *potential field* ataupun partisi Voronoi. Namun, algoritma tersebut membutuhkan data dan fungsi dari lingkungan yang hendak diuji. Sementara sulit untuk mendapatkan data dan fungsi dari lingkungan pada daerah yang tidak terjangkau manusia. Oleh sebab itu, pada Laporan Tugas Akhir, metode *Multi-Agent Reinforcement Learning* (MARL) digunakan untuk melakukan pengamatan. *Reinforcement learning* merupakan cabang *machine learning* yang membuat agen untuk melakukan pembelajaran sendiri tanpa label, input, dan *ouput* yang sudah terlebih dahulu diatur. Namun, pada implementasinya MARL membutuhkan banyak *computational cost*. Hal tersebut disebabkan karena agen tidak hanya memikirkan dirinya, namun kedua agen harus berkoordinasi dengan agen lainnya dan kedua agen saling bergerak. Kondisi agen yang selalu bergerak membuat lingkungan pembelajaran menjadi dinamis. Oleh karena itu, agen harus mengetahui kapan agen harus bergerak sendiri dan kapan agen harus berkoordinasi. Koordinasi yang dilakukan agen meliputi bertukar informasi dan negosiasi. Negosiasi digunakan agar kedua agen dapat menentukan aksi terbaik bagi kedua agen. Algoritma negosiasi yang digunakan berdasar pada *Nash Equilibrium*. Algoritma negosiasi yang digunakan adalah *non-strict* EDSP dan *meta equilibrium*. Agen yang dirancang harus mampu menekan jumlah gerakan dan perpindahan arah gerak dengan tujuan menghemat penggunaan daya, mempertahankan jarak antar kedua agen agar angin yang dihasilkan *propeller* tidak mengganggu agen lainnya jika terlalu dekat, dan kedua agen tidak dapat berkomunikasi bila terlalu jauh.

Abstract

The use of quadrotors as a monitoring device has increased recently. One thing that differentiates quadrotors from other UAVs is the ability to do Vertical Take off and Landing (VTOL). The ability to do VTOL and aerial monitoring is very suitable for observing areas that can't be reached by humans, such as disastrous areas and radioactive contamination areas. There were many algorithms to do aerial monitoring such as Voronoi partitions and potential field algorithms. However, environmental data and function were needed to run these algorithms and it was hard to extract data and function from the environment. Therefore reinforcement learning algorithm was proposed to overcome the problems mentioned. Reinforcement learning is a branch of machine learning that allows agents to do learning without any predefined labels, inputs, and outputs. There will be two agents in the system, so the term reinforcement learning will be changed to multi-agent reinforcement learning. Multi-agent reinforcement learning is more complex than single-agent reinforcement learning and more computational cost were needed to run a multi-agent reinforcement learning algorithm. It was because agents don't learn for themselves but the agents need to coordinate with other agents and all agents are not stationary. This problem caused the environment to become more dynamic. Therefore agents need to know when to do their task alone or coordinate with another agent. The coordination itself consisted of information exchange and negotiations. Negotiations were done by both agents to determine the best course of action for both agents. The negotiations algorithm used by both agents were non-strict EDSP and meta equilibrium. The agent was designed to be able to reduce the amount of movement and the number of changing the direction of motion with the aim of saving power consumption; Maintaining the distance between the two agents so that the wind generated by the propeller does not interfere with the other agent if it is too close, and the two agents cannot communicate if it is too far away; Last, the agents have to be able to create most optimal coverage area with minimum overlap area.

Kata Pengantar

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa, karena dengan berkat dan rahmat-Nya, Laporan Tugas Akhir ini dapat diselesaikan. Laporan Tugas Akhir yang berjudul "Studi Implementasi Metode Multi-Agent Reinforcement Learning untuk Optimasi Luas Cakupan Jangkauan Sensor" disusun, sebagai syarat untuk mengikuti Sidang Laporan Tugas Akhir pada mata kuliah Tugas Akhir II (IME 184500) pada Program Studi Sarjana Teknik Elektro (Konsentrasi Mekatronika) Universitas Katolik Parahyangan. Disadari bahwa penulisan Laporan Tugas Akhir ini dapat terselesaikan karena ada banyak dorongan yang diberikan dari berbagai pihak. Oleh karena itu penulis ingin mengucapkan terima kasih kepada:

- Tua Agustinus Tamba, Ph.D. selaku dosen pembimbing penulis pada mata kuliah Tugas Akhir II di Program Studi Sarjana Teknik Elektro (Konsentrasi Mekatronika) Universitas Katolik Parahyangan.
- Bapak Surya Wijaya Loekman dan Ibu So Nini selaku orang tua penulis yang telah memberi dukungan materiil dan moril agar penulis selalu bersemangat dalam mengerjakan Laporan Tugas Akhir hingga selesai.
- Bapak Hartono Loekman dan Ibu Liem Kiem Hoa selaku kakek nenek dari penulis yang selalu memberikan dorongan moril, mendoakan dari jauh, serta memastikan penulis dalam keadaan yang baik-baik saja.
- Bapak Hiapman selaku paman dari penulis yang sudah memberikan dukungan moril dan materiil, selalu mendukung dari jauh, serta doa-doa yang sudah disampaikan.
- Frans Cevin Hutabarat dan Stephen Ivanda Gulo selaku teman bertukar pikiran dan berbagi pengalaman dalam menyusun Laporan Tugas Akhir.
- Radifsya Leyndiva, Faza Pahla, Johannes Ekachandra, Irvan Elmahendra, Akbar Zahran, Karl Abraham, dan Wendi Kristianto yang selalu siap memberi semangat dan dukungan moral kepada penulis ketika penulis sedang menghadapi masalah.
- Ikan koi yang ada pada kolam dibawah pohon *maple* yang selalu menjadi tempat penulis untuk menenangkan diri dan berteduh.

- Seluruh dosen pengajar dan tata usaha Program Studi Teknik Elektro Konsentrasi Mekatronika Universitas Katolik Parahyangan yang telah membantu memberikan ilmu dan dukungan serta ramah tamah kepada penulis.
- Seluruh pihak yang membantu penulis namun belum sempat penulis cantumkan.

Semoga Tuhan Yang Maha Esa membalas kebaikan semua pihak yang telah menolong. Penulis memohon maaf yang sebesar-besarnya jika terdapat kekurangan atau hal-hal yang kurang berkenan dalam penyusunan Laporan Tugas Akhir II ini. Akhir kata, terima kasih atas pengertian dan kerja sama Anda. Besar harapan penulis, agar Laporan Tugas Akhir II ini dapat memberikan kontribusi yang berguna bagi perkembangan Program Studi Sarjana Teknik Elektro (Konsentrasi Mekatronika) Universitas Katolik Parahyangan pada khususnya serta khazanah keilmuan Teknik Mekatronika pada umumnya.

Daftar Isi

Kata Pengantar	i
Daftar Isi	iii
Daftar Tabel	vii
Daftar Gambar	ix
Daftar Simbol dan Variabel	xi
Daftar Singkatan	xiii
1 Pendahuluan	1
1.1 Latar Belakang Masalah	1
1.1.1 Konsep Pengamatan Berbasis Quadrotor	1
1.1.2 <i>Reinforcement Learning</i> untuk <i>coverage monitoring</i>	3
1.2 Identifikasi dan Perumusan Masalah.	5
1.3 Batasan Masalah dan Asumsi	5
1.4 Tujuan Tugas Akhir	6
1.5 Manfaat Tugas Akhir	6
1.6 Metodologi Tugas Akhir	6
1.7 Sistematika Penulisan	7
2 Tinjauan Pustaka	9
2.1 <i>Machine Learning</i>	9
2.2 <i>Single Agent Reinforcement Learning</i>	10
2.3 <i>Markov Decision Process</i>	12
2.3.1 Keadaan	12
2.3.2 Aksi	12
2.3.3 Ganjaran	13
2.3.4 Probabilitas Transisi	13

2.3.5	<i>Q-Value</i>	14
2.4	<i>Multi-Agent Reinforcement Learning</i> dengan Interaksi Ruang Tersebar	15
2.4.1	Kerangka Koordinasi pada Ruang Tersebar	15
2.4.2	Menyiarkan Keadaan-Aksi Gabungan	16
2.5	<i>Markov Game</i>	19
2.6	<i>Normal Form Game</i>	20
2.6.1	<i>Prisoner's Dilemma</i>	20
2.6.2	<i>Minimum Variance Method</i>	25
3	Perancangan Sistem	27
3.1	Penentuan Daerah <i>Coverage</i> Agen	27
3.1.1	Perhitungan Luas <i>Coverage</i> dengan Memperhitungkan Batas Lingkungan	30
3.1.2	Perhitungan Luas Daerah yang Menumpuk (<i>Overlap</i>)	31
3.2	Garis Besar Pemelajaran	37
3.2.1	Pemberian Karakteristik pada Agen	39
3.2.2	Penentuan Relasi Keadaan-Aksi Untuk Berkoordinasi	41
3.3	<i>Single Agent Reinforcement Learning</i>	42
3.3.1	Keadaan	42
3.3.2	Aksi	43
3.3.3	Ganjaran	44
3.3.4	Probabilitas Transisi	45
3.3.5	Penentuan Keadaan Berikutnya	47
3.3.6	Penentuan <i>Q-value</i> pada relasi aksi dan keadaan	48
3.3.7	Algoritma Pemelajaran	49
3.4	<i>Multi-Agent Reinforcement Learning</i>	50
3.4.1	Mengetahui Kedua Agen Ada pada Jarak Aman atau Tidak?	50
3.4.2	Ganjaran MARL	51
3.4.3	Kondisi Relasi Keadaan-Aksi Koordinasi	53
3.4.4	Algoritma Penyiaran Keadaan-Aksi Gabungan	53
3.4.5	Negosiasi Agen	56
4	Analisis Sistem	63
4.1	Analisa Hasil Pemelajaran 1 Agen Menggunakan SARL	63
4.1.1	Analisa Rata-Rata Agen	63
4.1.2	Analisa <i>Randomness</i> dari Hasil SARL	67
4.1.3	Analisa Pemilihan Metode ϵ -greedy dan Besaran <i>epsilon</i> yang digunakan	68
4.1.4	Analisa Set Aksi yang Diambil Agen	69
4.2	Analisa Hasil Pemelajaran Menggunakan MARL	76
4.2.1	Analisa Rata-Rata <i>Return</i> pada MARL	76
4.2.2	Analisa Set Aksi yang Digunakan Kedua Agen	77
4.3	Analisa Penekanan <i>Computational Cost</i>	80

5	Simpulan dan Saran	83
5.1	Simpulan	83
5.2	Saran	84
	Daftar Pustaka	85
	Lampiran A Lampiran 1	89
A.1	Grafik Pemelajaran SARL	89
A.2	Grafik <i>Randomness</i> pada Pemelajaran	90
A.3	Grafik pengaruh <i>epsilon</i> pada Pemelajaran	91
A.4	Grafik Pemelajaran MARL	92

Daftar Tabel

- 4.1 Pengaruh algoritma interaksi ruang tersebar pada simulasi pembelajaran. 81

Daftar Gambar

1.1	Ilustrasi <i>field of view</i> pada <i>quadrotor</i>	2
1.2	Diagram <i>feedback</i> algoritma <i>reinforcement learning</i> yang menggambarkan interaksi antara agen dan lingkungan.	3
2.1	Diagram alir <i>single agent reinforcement learning</i>	11
2.2	Contoh kasus ketika kedua agen harus melakukan <i>knowledge transfer</i>	18
2.3	Nilai utilitas (a) kedua tahanan, (b) tahanan A, (c) tahanan B, berdasarkan semua kemungkinan aksi yang dapat diambil.	21
2.4	Contoh <i>strategy profile</i> yang tidak memiliki Nash Equilibrium	23
2.5	Contoh <i>strategy profile</i> yang memiliki <i>nash equilibrium</i> dan <i>non-strict EDSP</i>	24
3.1	Ilustrasi luas daerah <i>coverage</i> pada 1 agen.	28
3.2	Ilustrasi luas <i>coverage</i> agen pada posisi (a) <i>cell</i> (1,2,2) (b) <i>cell</i> (3,4,2)	30
3.3	Ilustrasi luas <i>coverage</i> agen ketika kedua agen (a) menghasilkan <i>overlap</i> dan (b) tidak menghasilkan <i>overlap</i>	32
3.4	Ilustrasi koordinat daerah <i>overlap</i> yang dihasilkan oleh dua agen. . .	34
3.5	Kemungkinan posisi agen untuk menghasilkan <i>coverage</i> optimal pada lingkungan sebesar [6,6,2] <i>cell</i>	37
3.6	Kemungkinan posisi agen-2 untuk menghasilkan <i>coverage</i> optimal jika agen-1 berada pada posisi <i>cell</i> (2,2,2) pada lingkungan sebesar [6,6,2] <i>cell</i>	38
3.7	Peta <i>Q-value</i> pada ketinggian $z = 3$ ketika agen melakukan SARL dengan (a) $\epsilon = 0.98$ dan (b) $\epsilon = 0.20$	40
3.8	Keadaan (a) awal dan (b) terminal dari sebuah agen pada satu episode	43
4.1	Plot rata-rata <i>return</i> pada setiap 50 episode.	64
4.2	Plot jumlah rata-rata agen bergerak ke luar lingkungan dan jumlah rata-rata agen berpindah arah gerak per 20 episode	66
4.3	Plot rata-rata <i>return</i> per 50 episode dari 5 percobaan yang berbeda.	67
4.4	Plot nilai <i>return</i> pada 5 percobaan dengan <i>epsilon</i> yang berbeda. . .	68
4.5	Ilustrasi agen dengan keadaan awal pada <i>cell</i> (1, 1, 1)	70
4.6	Ilustrasi agen dengan keadaan awal pada <i>cell</i> (16, 1, 1)	71

4.7	Ilustrasi agen dengan keadaan awal pada <i>cell</i> (1, 16, 1)	72
4.8	Ilustrasi agen dengan keadaan awal pada <i>cell</i> (16, 16, 1)	73
4.9	Ilustrasi agen dengan keadaan awal pada <i>cell</i> (8, 8, 1)	74
4.10	Ilustrasi agen dengan keadaan awal pada <i>cell</i> (10, 7, 1)	75
4.11	Plot dari <i>return</i> pada percobaan menggunakan metode pembelajaran MARL	76
4.12	Ilustrasi gerakan kedua agen saat melakukan SARL ketika (a) agen-1 berada pada <i>grid</i> [1 1 1] dan (b) agen-2 berada pada <i>grid</i> [3 1 1]. . .	78
4.13	Ilustrasi gerakan kedua agen ketika melakukan MARL.	78
4.14	Ilustrasi gerakan kedua agen saat melakukan SARL ketika (a) agen-1 berada pada <i>grid</i> [8 8 1] dan (b) agen-2 berada pada <i>grid</i> [4 1 1]. . .	79
4.15	Ilustrasi gerakan kedua agen ketika melakukan MARL.	80
A.1	Grafik <i>return</i> yang didapatkan dan <i>step</i> yang dibutuhkan pada setiap <i>episode</i>	90
A.2	Grafik aksi keluar lingkungan dan aksi yang mengubah arah gerak setiap <i>episode</i>	90
A.3	Grafik <i>return</i> dari 5 buah percobaan dengan parameter yang sama. .	91
A.4	Grafik <i>return</i> dari 5 buah percobaan dengan parameter <i>epsilon</i> yang berbeda.	91
A.5	grafik <i>return</i> hasil simulasi <i>multi-agent reinforcement learning</i>	92

Daftar Simbol dan Variabel

(x, y, z)	Urutan posisi <i>grid</i> yang diisi <i>quadrotor</i> .
$[X, Y, Z]$	Besar lingkungan
θ_x	Sudut FOV yang dibentuk sensor yang sejajar dengan sumbu x kartesian.
θ_y	Sudut FOV yang dibentuk sensor yang sejajar dengan sumbu y kartesian.
$x+$	Bergerak 1 <i>grid</i> ke kanan.
$x-$	Bergerak 1 <i>grid</i> ke kiri.
$y+$	Bergerak 1 <i>grid</i> ke depan.
$y-$	Bergerak 1 <i>grid</i> ke belakang.
$z+$	Bergerak 1 <i>grid</i> ke atas.
$z-$	Bergerak 1 <i>grid</i> ke bawah.
i	Agen yang ditunjuk.
N	Jumlah agen yang melakukan koordinasi
k	<i>Step</i> yang ditunjuk.
K	<i>Step</i> maksimum pada satu episode.
e	Episode yang ditunjuk.
$s_{i,k}$	Keadaan dari agen- i pada <i>step</i> k .
S_k	Keadaan gabungan dari agen yang berkoordinasi pada <i>step</i> k .
$a_{i,k}$	Aksi yang diambil agen- i pada <i>step</i> k .
A_k	Aksi gabungan kedua agen pada <i>step</i> k .
a_{\max}	Aksi dengan <i>Q-value</i> tertinggi pada keadaan s .
$r(s, a)$	Ganjaran yang didapatkan agen ketika berada pada keadaan s dan mengambil aksi a .
$R(S, A)$	Ganjaran yang didapatkan kedua agen ketika berada keadaan gabungan S dan mengambil aksi gabungan A .
$\{S\}$	Semua kemungkinan keadaan agen.
$\{A\}$	Set aksi yang dimiliki agen.
G_e	<i>Return</i> yang didapatkan agen pada <i>episode</i> - e .
G_e^*	<i>Return</i> optimal yang didapatkan agen pada <i>episode</i> - e .
$Q_i(s, a)$	<i>Q-value</i> agen- i ketika berada pada keadaan s dan
$Q_i^{CT}(S, A)$	<i>Transferred Q-value</i> agen- i ketika berada pada keadaan gabungan S dan mengambil aksi gabungan A .
$Q_i^J(S, A)$	<i>Join Q-value</i> agen- i ketika pada keadaan
$Q_\pi(s, a)$	<i>Q-value</i> agen yang sudah optimal. gabungan S dan mengambil aksi gabungan A . mengambil aksi a .
$P(s, a, s')$	Probabilitas transisi agen pada keadaan s , mengambil aksi a dan bergerak menuju keadaan s' .
$\pi(s, a)$	<i>Policy</i> .

$\pi^*(s, a)$	<i>Optimal policy.</i>
ϵ	<i>Explore rate.</i>
γ	<i>Discount rate.</i>
α	<i>Learning rate.</i>
U^{NE}	Utilitas agen yang memenuhi <i>nash equilibrium</i> .
U^{NSEDSP}	Utilitas agen yang memenuhi <i>non-strict EDSP</i> .
U^{META}	Utilitas agen yang memenuhi <i>meta equilibrium</i> .
σ	Variansi.
μ	Rata-rata.
$\delta_{\min,x}$	Representasi sumbu x dari titik koordinat pojok kiri bawah dari <i>coverage</i> agen.
$\delta_{\min,y}$	Representasi sumbu y dari titik koordinat pojok kiri bawah dari <i>coverage</i> agen.
$\delta_{\max,x}$	Representasi sumbu x dari titik koordinat pojok kanan atas dari <i>coverage</i> agen.
$\delta_{\max,y}$	Representasi sumbu y dari titik koordinat pojok kanan atas dari <i>coverage</i> agen.
$\delta_{\min,x}^*$	Representasi sumbu x dari titik koordinat pojok kiri bawah dari <i>coverage</i> agen dengan mempertimbangkan lingkungan.
$\delta_{\min,y}^*$	Representasi sumbu y dari titik koordinat pojok kiri bawah dari <i>coverage</i> agen dengan mempertimbangkan lingkungan.
$\delta_{\max,x}^*$	Representasi sumbu x dari titik koordinat pojok kanan atas dari <i>coverage</i> agen dengan mempertimbangkan lingkungan.
$\delta_{\max,y}^*$	Representasi sumbu y dari titik koordinat pojok kanan atas dari <i>coverage</i> agen dengan mempertimbangkan lingkungan.
λ	Luas <i>coverage</i> agen tunggal.
λ^*	Luas <i>coverage</i> agen tunggal dengan memperhitungkan lingkungan.
$\beta_{\min,x}$	Representasi sumbu x dari titik koordinat pojok kiri bawah dari <i>overlap</i> yang dibentuk agen.
$\beta_{\min,y}$	Representasi sumbu y dari titik koordinat pojok kiri bawah dari <i>overlap</i> yang dibentuk agen.
$\beta_{\max,x}$	Representasi sumbu x dari titik koordinat pojok kanan atas dari <i>overlap</i> yang dibentuk agen.
$\beta_{\max,y}$	Representasi sumbu y dari titik koordinat pojok kanan atas dari <i>overlap</i> yang dibentuk agen.
ρ	Luas <i>overlap</i> yang dibentuk dua agen
Λ	Luas <i>coverage</i> multi-agen.
λ^*	Luas <i>coverage</i> multi-agen dengan mempertimbangkan lingkungan dan <i>overlap</i> .

Daftar Singkatan

VTOL	<i>Vertical Take-Off and Landing</i>
SARL	<i>Single Agent Reinforcement Learning</i>
MARL	<i>Multi Agent Reinforcement Learning</i>
FTI	Fakultas Teknologi Industri
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
UNPAR	Universitas Katolik Parahyangan

Bab 1

Pendahuluan

Bab 1 membahas latar belakang masalah studi implementasi metode *multi-agent reinforcement learning* untuk optimasi luas cakupan jaringan sensor, serta menyertakan latar belakang masalah, batasan, asumsi, tujuan, dan manfaat dari perancangan *multi-agent reinforcement learning* untuk optimasi luas cakupan jaringan sensor.

1.1 Latar Belakang Masalah

Dalam mengamati daerah di luar jangkauan manusia, seperti daerah yang terkena bencana [1], daerah yang terkena erosi [2], lokasi sumber radioaktif [3], atau melakukan monitor pertanian dari hama hewan pengerat [4], penggunaan jaringan sensor dapat dijadikan alternatif yang lebih aman, akurat, dan optimal. Namun, penggunaan jaringan sensor dalam melakukan pengamatan harus memperhitungkan posisi jaringan sensor agar dapat mengamati suatu daerah dengan optimal dan tidak saling menumpuk; Jarak antar sensor, agar sensor dapat saling berkomunikasi sehingga tidak saling mengganggu; Serta optimasi penggunaan daya pada penyimpanan daya sensor. Pada laporan ini, sensor yang ditinjau merupakan sensor yang diletakkan di bagian bawah *quadrotor* dan menghadap ke bawah untuk mengamati daerah di bawah *quadrotor* tersebut.

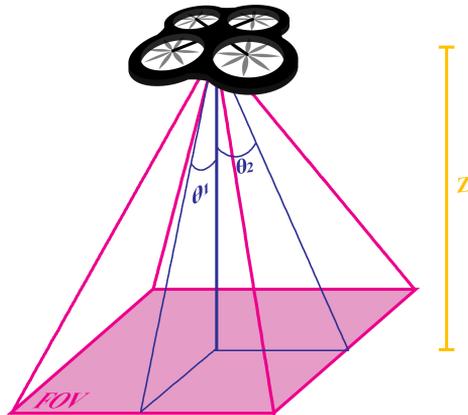
1.1.1 Konsep Pengamatan Berbasis Quadrotor

Penggunaan *quadrotor* sebagai alat pengamatan meningkat akhir-akhir ini. *Quadrotor* memiliki kelebihan untuk monitor *aerial view* yang sangat baik sehingga digunakan untuk melakukan observasi dan memonitor objek statis maupun bergerak. Selain itu *quadrotor* sering kali dapat melaksanakan tugas pengamatan pada lingkungan yang ekstrem dan berbahaya bagi manusia [5]. *Quadrotor* memiliki kelebihan untuk memberikan informasi dengan biaya yang murah dan fleksibel. Menurut Ozdemir [6],

yang membedakan *quadrotor* dengan sub tipe *Unmanned Aerial Vehicle* (UAV) lainnya adalah kemampuan untuk melakukan *vertical take-off and landing* (VTOL). VTOL memungkinkan *quadrotor* untuk melakukan lepas landas dan pendaratan secara vertikal dan tidak membutuhkan landasan pacu. Selain itu menurut Mian [7], *quadrotor* memiliki 6 derajat kebebasan yaitu bergerak secara translasi pada sumbu x , y , z serta rotasi sebesar ϕ, θ, ψ . *Quadrotor* mampu bergerak translasi ke kanan ($x+$), kiri ($x-$), maju ($y+$), mundur ($y-$), naik ($z+$), turun ($z-$), dan diam. Dalam penerapannya, fungsi *quadrotor* dalam bidang pengamatan dapat digunakan sebagai:

1. Monitor daerah yang terkena bencana [1].
2. Monitor daerah yang terkena erosi [2].
3. Menentukan lokasi sumber radioaktif menggunakan sistem monitor radioaktif [3].
4. Monitor pertanian dari hama hewan pengerat [4]

Agar dapat melakukan pengamatan, sebuah sensor dipasang di bagian bawah *quadrotor*. Sensor dihadapkan ke bawah sehingga sensor dapat memindai objek yang hendak diamati. Daerah yang terekam oleh sensor disebut *field of view* (FOV). Dapat diamati pada **Gambar 1.1**, FOV yang ditinjau pada Tugas Akhir ini merupakan area berbentuk segi empat yang terbentuk dari dua buah sudut θ_x dan θ_y serta tinggi z .



Gambar 1.1 Ilustrasi *field of view* pada *quadrotor*

Berdasarkan **Gambar 1.1**, FOV dapat berubah ukuran bergantung pada ketinggian dari *quadrotor* tersebut. Semakin tinggi posisi *quadrotor* relatif terhadap bidang yang diamati, semakin besar pula FOV yang terbentuk.

Dalam melakukan pengamatan menggunakan *quadrotor*, banyak riset dan algoritma seperti pada [8–10] yang ditemukan seperti partisi Voronoi [8] dan metode *potential*

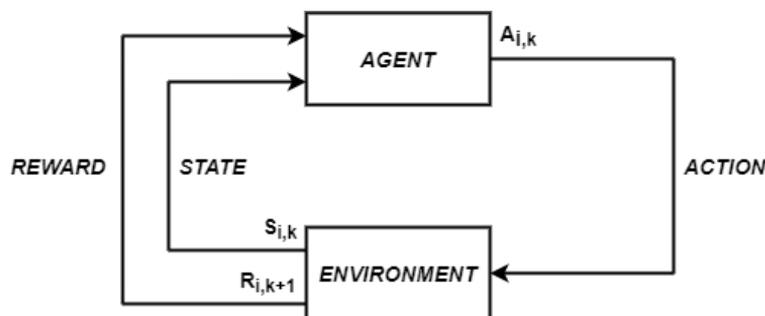
field [9]. Partisi Voronoi membagi daerah yang diamati menjadi beberapa bagian berbentuk *convex polytope* dan memberikan informasi pada daerah yang dibagi menurut probabilitas dan fungsi nilai yang dimiliki [11]. *Potential field* memberikan nilai virtual pada daerah yang diamati dan membuat sensor bergerak menuju nilai terbesar atau terkecil dan menghindari rintangan [9]. Namun, pada kebanyakan riset, penulis banyak melakukan asumsi mengenai model matematika dari lingkungan, seperti jalur *quadrotor* yang akan melakukan pengamatan [10]. Pada kenyataannya, untuk menemukan model lingkungan yang akurat sangat sulit dan cenderung tidak mungkin. Hal tersebut disebabkan karena data yang ada pada kondisi lingkungan yang sebenarnya terbatas atau bahkan tidak ada [12].

Menurut Pham [12], salah satu metode yang dapat digunakan untuk melakukan adalah *machine learning*. *Machine learning* merupakan disiplin ilmu yang memikirkan algoritma untuk membangun sebuah sistem komputer yang dapat meningkatkan kinerjanya melalui pengalaman [13].

1.1.2 Reinforcement Learning untuk coverage monitoring

Machine learning memiliki tiga buah sub-kategori: *supervised learning*, *unsupervised learning*, dan *reinforcement learning*. Berbeda dengan *supervised learning* dan *unsupervised learning*, *reinforcement learning* merupakan algoritma yang memiliki interaksi antara pengguna dan lingkungan yang diamati.

Algoritma tanpa model seperti *reinforcement learning* dapat digunakan sebagai salah satu pendekatan untuk menyelesaikan masalah mengenai model matematika lingkungan dan sifat-sifat kompleks dari sistem. Metode seperti *supervised learning* dan *unsupervised learning* membutuhkan input yang pada kasus ini merupakan dinamika dan model matematika dari lingkungan [14]. *Reinforcement learning*, di lain sisi, memungkinkan pengguna untuk melakukan tugasnya tanpa harus memperhitungkan model dari lingkungan karena hanya mengandalkan interaksi agen dan lingkungan [15].



Gambar 1.2 Diagram *feedback* algoritma *reinforcement learning* yang menggambarkan interaksi antara agen dan lingkungan.

Seperti yang dapat diamati pada **Gambar 1.2**, agen akan mendapatkan sebuah keadaan (*state*) dari lingkungan (*environment*). Berdasarkan keadaan yang ada, agen lalu melakukan sebuah aksi (*action*) terhadap lingkungan. Aksi yang diberikan agen menentukan ganjaran yang diterima. Aksi yang diberikan agen juga mengubah keadaan dari agen tersebut pada lingkungan. Setelah melakukan aksi, agen mendapatkan ganjaran (*reward*) dan keadaan yang baru dari lingkungan. Ganjaran terbesar yang diterima agen yang kemudian dijadikan sebagai acuan dalam metode menyelesaikan tugasnya. Proses tersebut menggambarkan satu *step* pada pembelajaran. *Step* tersebut akan terus diulang hingga agen menyelesaikan tugasnya.

Ketika agen berhasil menyelesaikan tugasnya, satu episode dinyatakan selesai. Akumulasi ganjaran yang didapat selama satu episode dinamakan *return*. *Reinforcement learning* dirancang untuk membuat agen dapat menyelesaikan episode dengan *return* menuju optimal pada tiap episode.

Dalam melakukan optimasi luas jangkauan *multi sensor* optimal, agen merupakan *quadrotor* yang dilengkapi sensor; lingkungan merupakan lingkungan pengamatan dalam bentuk tiga dimensi yang diberikan *grid* (x, y, z); aksi merupakan perpindahan satu *grid* yang dilakukan *quadrotor* ke kanan ($x+$), kiri ($x-$), maju ($y+$), mundur ($y-$), naik ($z+$), turun ($z-$); Serta ganjaran merupakan nilai yang didapat agen berdasarkan aksi yang dipilih.

Untuk melakukan tugasnya dalam pengamatan lapangan, satu buah *quadrotor* belum tentu dapat melakukan tugasnya dengan optimal. Ketika dibutuhkan lebih dari satu sensor, metode *single agent reinforcement learning* belum dapat digunakan untuk menyelesaikan permasalahan karena pada sistem multi-agen, agen tidak hanya berinteraksi dengan lingkungan saja tetapi juga berinteraksi dengan agen lain. Hal tersebut membuat sistem yang ada pada lingkungan menjadi lebih dinamis karena agen lainnya bergerak dan tidak diam di satu tempat. Pada sistem multi-agen, kemungkinan aksi terbaik yang dimiliki satu agen belum tentu merupakan aksi yang terbaik juga untuk setiap agen. Oleh sebab itu, aksi yang diambil kedua agen harus melewati koordinasi antar agen yang saling berhubungan. Metode *Multi-Agent Reinforcement Learning* (MARL) mampu menentukan keputusan bersama dari beberapa agen yang berlaku sistem multi-agen dan memberikan lebih banyak keuntungan pada sistem multi-agen [12].

Karena pada sistem multi-agen setiap agen harus memperhitungkan pilihan aksi agen lainnya, maka para agen tersebut harus menentukan kesepakatan dalam menentukan pilihan. Penentuan kesepakatan dapat dievaluasi melalui *nash equilibrium* [16]. Menurut Hu dan Wellman [16], *nash equilibrium* merupakan metode evaluasi yang mana agen bertindak seefektif mungkin berdasarkan agen lainnya dan agen tersebut harus bertindak rasional mungkin terhadap agen lainnya. Berdasarkan *nash equilibrium*, algoritma negosiasi seperti *non-strict* EDSP dan *meta equilibrium* dapat dilakukan.

MARL memiliki *computational cost* yang relatif jauh lebih tinggi dengan peningkatan secara eksponensial pada tiap agen yang bekerja pada sistem [17]. Pada SARL, agen

hanya memperhitungkan relasi antara agen dan lingkungan. Pada MARL dengan dua agen, agen-1 memperhitungkan relasi antara agen tersebut dengan lingkungan dan agen lainnya. *Computational cost* memengaruhi kinerja *hardware* dan waktu perhitungan dalam melakukan pembelajaran. Untuk mengoptimalkan *computational cost*, maka negosiasi pada MARL hanya digunakan pada keadaan yang mengharuskan agen untuk berkoordinasi seperti, ketika *coverage* yang dibentuk agen membentuk *overlap* yang lebih besar dari *threshold overlap* dan/atau jarak kedua agen terlalu dekat atau terlalu jauh.

Dalam menyusun tugas akhir dengan judul *Studi Implementasi Metode Multi-Agent Reinforcement Learning untuk Optimasi Luas Cakupan Jaringan Sensor* beberapa hal yang harus dipertimbangkan adalah optimasi luas cakupan jaringan sensor dengan mempertimbangkan daerah *coverage* optimal, penggunaan daya secara optimal, serta jarak antar sensor agar dapat melakukan komunikasi antar sensor. Algoritma yang digunakan untuk menentukan posisi dan jalur yang dirintis adalah *reinforcement learning*. *Reinforcement learning* merupakan cabang dari *machine learning* yang tidak membutuhkan set data input dan mengandalkan relasi antar agen dan lingkungan. Sensor yang digunakan merupakan sensor yang dipasang pada bagian bawah *quadrotor*. *Quadrotor* yang digunakan berjumlah dua buah sehingga algoritma *reinforcement learning* harus dikembangkan menjadi *multi-agent reinforcement learning*. MARL memiliki *computational cost* yang jumlahnya bertambah secara eksponensial sesuai dengan jumlah agen. Untuk mengoptimalkan *computational cost*, maka jumlah MARL akan diterapkan jika dan hanya jika kedua agen sedang berada dalam keadaan yang mengharuskan agen untuk berkoordinasi.

1.2 Identifikasi dan Perumusan Masalah.

Dari beberapa permasalahan yang dipaparkan pada subbab 1.1, maka masalah yang dapat diidentifikasi adalah sebagai berikut:

1. Bagaimana cara mengimplementasikan *multi-agent reinforcement learning* untuk optimasi luas cakupan jaringan sensor?
2. Bagaimana cara menemukan *field coverage* optimal dan menghindari *overlap* menggunakan metode MARL tanpa mengetahui model dari lingkungan?
3. Bagaimana cara menentukan posisi setiap *quadrotor* agar tidak keluar dari sinyal jangkauan komunikasi dengan *quadrotor* lainnya?
4. Bagaimana cara mengoptimalkan *computational cost* pada MARL?

1.3 Batasan Masalah dan Asumsi

Untuk memecahkan masalah pada identifikasi dan perumusan masalah, terdapat beberapa batasan masalah dan asumsi yang dapat diperhatikan:

1. *Quadrotor* yang digunakan pada sistem multi-agen berjumlah dua buah.
2. Lingkungan berbentuk tiga dimensi yang dipartisi *grid* berbentuk kubus dengan titik pusat kubus dijadikan titik acuan posisi dari *quadrotor*.
3. FOV area berbentuk proyeksi limas segi empat dengan luas area segi empat yang dihitung sebagai luas *coverage*.
4. Sudut proyeksi sensor sebesar 45 derajat ($\theta_x = \theta_y = 45^\circ$)
5. Percobaan dan analisa dilakukan secara virtual pada program Matlab.

1.4 Tujuan Tugas Akhir

Tujuan dari perancangan *reinforcement learning* kooperatif pada *quadrotor* untuk *field coverage* adalah untuk merancang algoritma MARL yang dapat menentukan posisi dari sistem *multi-agent* pada *quadrotor* untuk melakukan *field monitoring* optimal dengan memaksimalkan *coverage* bagi masing-masing agen dan meminimalkan *overlap* bagi sistem multi-agen. Untuk mencapai koordinat tersebut, agen harus memperhatikan jarak antar agen serta optimasi penggunaan daya masing-masing agen.

1.5 Manfaat Tugas Akhir

Berdasarkan tujuan pada subbab 1.4, maka manfaat dari perancangan *reinforcement learning* kooperatif pada *quadrotor* untuk *field coverage* adalah:

1. Mampu memahami algoritma *reinforcement learning* pada sistem *multi-agent*
2. Mampu merancang sistem *multi-agent* untuk yang dapat mencari posisi *quadrotor* yang dapat memaksimalkan luas *coverage* yang dibentuk dengan mempertimbangkan jarak aman antar *quadrotor* dan optimasi penggunaan daya pada *quadrotor*.

1.6 Metodologi Tugas Akhir

Adapun dalam menyusun tugas akhir, metodologi yang digunakan dimulai dengan melakukan pengenalan dan pendalaman materi mengenai *reinforcement learning* untuk memahami konsep dasar dari topik yang dipilih dengan sebuah studi literatur dari beberapa jurnal dan buku. Setelah pemahaman konsep dan ilmu yang didapatkan sudah dianggap cukup, algoritma *field monitoring* dirancang untuk menghitung luas daerah optimal yang ditinjau oleh *quadrotor* luas daerah optimal yang digunakan nantinya dipakai untuk menjadi parameter apakah agen berhasil melakukan tugasnya. Selanjutnya, nilai optimal dari fungsi ganjaran untuk *Single Agent Reinforcement Learning* (SARL) dan *Multi-Agent Reinforcement Learning* (MARL) ditentukan, angka tersebut ditentukan untuk mengetahui parameter apa yang menjadi prioritas

bagi agen dalam menjalankan tugasnya. Setelah menentukan fungsi ganjaran, algoritma SARL kemudian dirancang. Ketika algoritma SARL yang dilakukan dirasa sudah cukup baik, maka algoritma untuk transfer informasi dan negosiasi ditentukan. Ketika algoritma negosiasi dirasa sudah cocok untuk digunakan, maka algoritma MARL kemudian dirancang. Pelatihan dilakukan menggunakan algoritma SARL dan MARL yang sudah dirancang lalu hasil dari pelatihan tersebut di analisa dengan melakukan contoh kasus. Langkah terakhir dari penelitian ini adalah dengan melakukan verifikasi pergerakan dari *quadrotor* untuk memastikan bahwa pergerakan *quadrotor* sesuai dengan simulasi pergerakan pada tahap pelatihan.

1.7 Sistematika Penulisan

Laporan Tugas Akhir ini dibagi menjadi 5 bab, yakni sebagai berikut:

1. **Bab 1 Pendahuluan.** Dalam bab ini dijelaskan mengenai latar belakang masalah, identifikasi dan perumusan masalah, batasan masalah dan asumsi, tujuan Tugas Akhir, manfaat Tugas Akhir, metodologi Tugas Akhir serta sistematika penulisan Laporan Tugas Akhir.
2. **Bab 2 Tinjauan Pustaka.** Bab ini berisi tentang dasar teori yang digunakan yaitu *reinforcement learning*, *multi-agent reinforcement learning*, *Markov Decision Process*, interaksi ruang tersebar, *Markov game*, dan *normal form game*. Teori-teori dasar ini diperoleh melalui proses telaah pustaka yang intensif pada sejumlah pustaka yang direkomendasikan oleh dosen pembimbing.
3. **Bab 3 Perancangan Sistem.** Dalam bab ini perancangan algoritma dipaparkan secara mendalam dengan dasar yang sudah dibahas pada Bab 2. Bab 3 memaparkan algoritma *coverage* untuk satu dan dua agen, penentuan aksi, ganjaran, keadaan, *update Q-Value* dan *joint Q-value*, serta negosiasi yang digunakan.
4. **Bab 4 Implementasi dan Analisa.** Berdasarkan algoritma yang dirancang pada Bab 3, maka hasil dari pembelajaran akan dibahas pada Bab 4. Hasil pembelajaran yang dianalisa antara lain adalah analisa hasil pembelajaran 1 agen yang meninjau *rata-rata step* dan ganjaran pada agen, seberapa acak pemilihan aksi pada agen, analisa dan alasan pemilihan metode pengambilan keputusan *ϵ -greedy*, dan contoh kasus serta ilustrasi pergerakan agen. Sedangkan pada MARL, analisa yang dilakukan meninjau analisa rata-rata *return* yang didapatkan masing-masing agen dan uji kasus serta ilustrasi pergerakan agen untuk memastikan negosiasi berjalan dengan baik. Serta pembuktian penekanan *computational cost* menggunakan metode interaksi ruang tersebar.
5. **Bab 5 Kesimpulan dan Saran.** Pada bab ini, Laporan Tugas Akhir disimpulkan. Kemudian penulis memberikan saran untuk pengembangan yang dapat dilakukan untuk riset selanjutnya.