



**PROPOSAL TUGAS AKHIR - EE184701**

**IMPLEMENTASI ALGORITMA HYBRID ROLLING-HORIZON COLLISION RESOLUTION DAN REGRET-BASED MARGINAL COST ALGORITHM PADA MULTI AGENT PICKUP AND DELIVERY**

**VINCENTIUS CHARLES MAYNAD**

**NRP 07111940000038**

Dosen Pembimbing

**Mochamad Sahal S.T., M.Sc.**

**NIP 197011191998021002**

**Yusuf Bilfaqih S.T., M.T.**

**NIP 197203251999031001**

**Program Studi Sarjana Teknik Elektro**

Departemen Teknik Elektro

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

2022

# **LEMBAR PENGESAHAN**

## **IMPLEMENTASI ALGORITMA HYBRID ROLLING-HORIZON COLLISION RESOLUTION DAN REGRET-BASED MARGINAL COST ALGORITHM PADA MULTI AGENT PICKUP AND DELIVERY**

### **PROPOSAL TUGAS AKHIR**

Diajukan untuk memenuhi salah satu syarat  
memperoleh gelar Sarjana Teknik pada  
Program Studi Sarjana Teknik Elektro  
Departemen Teknik Elektro  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember

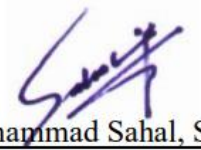
Oleh : **VINCENTIUS CHARLES MAYNAD**

NRP. 07111940000038

Disetujui oleh Tim Penguji Proposal Tugas Akhir :

- |   |                            |
|---|----------------------------|
| 1. Nama dan gelar pembimbing            | Mochamad Sahal S.T., M.Sc. |
| 2. Nama dan gelar ko-pembimbing/penguji | Yusuf Bilfaqih S.T., M.T.  |
| 3. Nama dan gelar penguji               |                            |
| 4. Nama dan gelar penguji               |                            |
| 5. Nama dan gelar penguji               |                            |

Surabaya, 12 Juni 2022  
Dosen Pembimbing I,



Mochammad Sahal, S.T., M.Sc.  
NIP : 197011191998021002

Dosen Pembimbing II,

Yusuf Bilfaqih, S.T., M.T.  
NIP : 197203251999031001

Mengetahui,  
Kaprodi Sarjana / Kepala Departemen

Menyetujui,  
Kepala Laboratorium Sistem dan  
Sibernetika

Dedet C. Riawan, S.T., M.Eng., Ph.D.  
NIP : 197311192000031001

Yusuf Bilfaqih, S.T., M.T.  
NIP : 197203251999031001

**SURABAYA**

**Juni, 2022**

## ABSTRAK

### IMPLEMENTASI ALGORITMA HYBRID ROLLING-HORIZON COLLISION RESOLUTION DAN REGRET-BASED MARGINAL COST ALGORITHM PADA MULTI AGENT PICKUP AND DELIVERY

Nama Mahasiswa / NRP : Vincentius Charles Maynad / 07111940000038  
Departemen : Teknik Elektro FTEIC - ITS  
Dosen Pembimbing : 1. Mochamad Sahal S.T., M.Sc.  
2. Yusuf Bilfaqih S.T., M

#### Abstrak

Dalam sistem gudang otomatis, beberapa agen bekerja sama memenuhi pesanan pelanggan. Setiap pesanan terdiri dari barang yang harus dikirim ke stasiun pengambilan untuk konsolidasi dan distribusi selanjutnya. Hal serupa terjadi pada pusat penyortiran otomatis. Disini, agen-agen ditugaskan untuk mengirim surat ke beberapa tempat tujuan yang sudah ditetapkan. Problema riil ini dapat dimodelkan sebagai permasalahan *Multi-agent Pickup and Delivery* (MAPD). MAPD terdiri dari dua prosedur penting yakni pembagian tugas atau *Task Assignment* (TA) dan penentuan jalur multi agen atau *Multi-Agent Path Finding* (MAPF). Kedua prosedur harus dilaksanakan dengan baik agar tujuan MAPD terpenuhi. Metode MAPD pada umumnya adalah membagi jalur setelah memberi tugas tanpa memperhatikan biaya yang diperlukan. Oleh karena itu, diperlukan algoritma yang membagi tugas setelah memperhitungkan biaya masing-masing tugas dan lebih fleksibel dalam penerapannya terutama dalam soal skalabilitas. Selain itu, terdapat asumsi tambahan seperti perhitungan kapasitas, yaitu robot dimungkinkan untuk melaksanakan beberapa tugas bersamaan. Pada tugas akhir ini, algoritma tersebut akan direalisasikan dengan mengombinasikan *Regret-based Marginal Cost Algorithm* (RMCA) dan *Rolling-Horizon Collision Resolution* (RHCR). Penggunaan algoritma tersebut diyakini bisa mempercepat waktu eksekusi dan meminimalkan biaya. Hasil yang didapatkan dari penelitian ini diharapkan dapat menjadi referensi teoritis dan praktis untuk penerapan MAPD kedepannya sekaligus otomasi industri tahap lanjut.

**Kata kunci:** *Multi-agent Pickup and Delivery* ( MAPD), *Regret-based Marginal Cost Algorithm* (RMCA), *Rolling-Horizon Collision Resolution* (RHCR)

# ABSTRACT

## IMPLEMENTATION OF ROLLING-HORIZON COLLISION RESOLUTION AND REGRET-BASED MARGINAL COST HYBRID ALGORITHM ON MULTI AGENT PICKUP AND DELIVERY

**Student Name / NRP** : Vincentius Charles Maynad / 07111940000038  
**Department** : Teknik Elektro FTEIC - ITS  
**Advisor** : 1. Mochamad Sahal S.T., M.Sc.  
2. Yusuf Bilfaqih S.T., M.T.

### Abstract

In automated warehouse system, several agents work together to fulfill costumers's orders. Each order consists of goods that must be send to a picking station for consolidation and subsequent distribution. Something similar happens in the automated sortation center. Here, agents are assigned to send letters to some predetermined destinations. This real problem can be modeled as a Multi-agent Pickup and Delivery (MAPD) problem. MAPD consists of two important procedures, namely the division of tasks or Task Assignment (TA) and the determination of multi-agent paths or Multi-Agent Path Finding (MAPF). Both procedures must be carried out properly so that the objectives of MAPD are met. The MAPD method in general is to divide the path after giving the task without considering the costs involved. Therefore, we need an algorithm that divides the tasks after calculating the cost of each task and is more flexible in its application, especially in terms of scalability. In addition, there are additional assumptions such as capacity calculations, i.e. the robot is possible to carry out several tasks simultaneously. In this final project, the algorithm will be realized by Regret-based Marginal Cost Algorithm (RMCA) and Rolling-Horizon Collision Resolution (RHCR). The use of that algorithm is believed to be able to speed up execution time and minimize costs. The results obtained from this study are expected to be a theoretical and practical reference for future MAPD applications as well as advanced industrial automation.

**Keywords:** *Multi-agent Pickup and Delivery ( MAPD), Regret-based Marginal Cost Algorithm (RMCA), Rolling-Horizon Collision Resolution (RHCR)*

## DAFTAR ISI

LEMBAR PENGESAHAN	i
ABSTRAK	iii
ABSTRACT	iv
DAFTAR ISI	v
DAFTAR GAMBAR	vi
DAFTAR TABEL	vii
DAFTAR SIMBOL	viii
BAB 1    PENDAHULUAN	9
1.1    Latar Belakang	9
1.2    Rumusan Masalah	9
1.3    Batasan Masalah	9
1.4    Tujuan	9
1.5    Manfaat	9
BAB 2    TINJAUAN PUSTAKA	11
2.1    Hasil Penelitian Terdahulu	11
2.2    Dasar Teori	12
BAB 3    METODOLOGI	14
3.1    Metode yang digunakan	14
3.2    Bahan dan peralatan yang digunakan	14
3.3    Urutan pelaksanaan penelitian	14
DAFTAR PUSTAKA	15
LAMPIRAN	16

## **DAFTAR GAMBAR**

## **DAFTAR TABEL**



## DAFTAR SIMBOL

$P^U$	kumpulan tugas ( <i>task set</i> )
$i$	tugas spesifik
$s_i$	lokasi asal dari tugas $i$
$g_i$	lokasi destinasi dari tugas $i$
$R$	set robot
$k_1^*$	agen dengan biaya terkecil pertama
$k_2^*$	agen dengan biaya terkecil kedua
$o_k$	urutan tindakan yang berurutan untuk agen $k$
$q_i^*$	tindakan ke- $i$
$p_i^*$	bentuk lain dari $q_i^*$
$t$	Langkah waktu ( <i>time instant</i> )
$\oplus$	penjumlahan Kronecker
$j$	satuan. Contohnya $g_i[j]$ artinya destinasi ke- $j$
$w$	horizon waktu
$h$	periode perencanaan ulang

# BAB 1 PENDAHULUAN

## 1.1 Latar Belakang

Dalam sistem gudang otomatis, tim robot bekerja bersama-sama untuk memenuhi satu set pesanan pelanggan. Setiap pesanan terdiri dari satu atau lebih item yang ditemukan di lantai gudang, yang harus dikirim ke stasiun pengambilan untuk konsolidasi dan pengiriman. Di pusat penyortiran otomatis, sementara itu, masalah serupa muncul. Di sini, tim robot ditugaskan membawa tugas-tugas surat dari salah satu dari beberapa stasiun emitor, di mana paket baru tiba, ke tumpukan tugas yang diurutkan, semuanya terikat untuk fasilitas pemrosesan yang sama di mana mereka akan dikirimkan [1]. Sukses praktis di kedua konteks ini bergantung pada komputasi solusi tepat waktu untuk masalah optimasi menantang yang dikenal dalam literatur sebagai *Multi Agent Pickup and Delivery* (MAPD) atau penjemputan dan pengiriman multi agen [2].

Tim agen harus menjalankan sekumpulan tugas dalam lingkungan yang dikenal. Untuk menjalankan tugas, agen harus bergerak terlebih dahulu dari lokasi saat ini ke lokasi pengambilan tugas dan kemudian ke lokasi pengiriman tugas. Masalah pada algoritma MAPD adalah menetapkan tugas ke agen dan merencanakan jalur bebas tabrakan bagi mereka untuk menjalankan tugas mereka [3]. Penetapan tugas (TA) [4] dan *Multi Agent Path Finding* (MAPF) [5] itu sendiri susah untuk dikendalikan, membuat MAPD sangat menantang untuk dipecahkan dalam praktik. Penerapan RMCA memungkinkan agen membawa barang berkapasitas dan melaksanakan tugas dengan waktu yang lebih singkat daripada algoritma TA lainnya [1]. Sedangkan NHCR dapat digunakan untuk meningkatkan kuantitas tugas ke skala yang lebih besar tetapi tetap efektif dan fleksibel [6].

## 1.2 Rumusan Masalah

Tugas pengambilan dan pengiriman barang pada industri menggunakan sistem multi agen (MAPD) berkaitan erat dengan penetapan atau pendelegasian tugas (TA) dan perancangan jalur yang akan dilalui oleh masing-masing agen (MAPF). Hal yang ingin dicapai adalah waktu yang cepat, biaya minimum, skalabilitas, serta jalur yang bebas tabrakan (dengan agen lain). Permasalahan yang dapat dirumuskan pada penelitian ini adalah bagaimana cara mencapai hal tersebut dengan lebih efektif dan efisien.

## 1.3 Batasan Masalah

Ruang lingkup/pembatasan masalah dalam tugas akhir ini adalah metode yang digunakan hanya berupa integrasi dan pengujian modifikasi dari metode sebelumnya demi mengetahui efek atau hasilnya. Selain itu, penerapan yang dilakukan tidak dilakukan pada properti asli namun berupa simulasi menggunakan bahasa pemrograman C++ untuk menunjukkan prosesnya karena keterbatasan dana dan waktu. Batas maksimal jumlah agen yang digunakan pun akan disesuaikan dengan kemampuan laptop yang digunakan sebab semakin banyak agen berarti semakin banyak komputasi yang diperlukan. Harapannya dapat mencapai setidaknya 1000 agen. Jumlah kapasitas maksimal yang diasumsikan adalah 5. Sedangkan ukuran dari peta gudang diasumsikan 21x35 kotak, namun akan lebih bila jumlah agen semakin banyak.

## 1.4 Tujuan

Pada penilitan ini, akan dikombinasikan metode RMCA yang dikembangkan dengan NHCR untuk menyelesaikan tugas pengambilan dan pengiriman multi agen (MAPD). Hasil penelitian ini diharapkan dapat menjadi solusi dan referensi untuk penerapan MAPD yang lebih

efektif, efisien, dan fleksibel dalam skalabilitas pada berbagai lokasi penyimpanan yang memerlukan pemindahan barang intensif.

### **1.5 Manfaat**

Manfaat penelitian ini adalah meningkatkan otomasi industri terutama pada tempat penyimpanan, meningkatkan pemanfaatan teknologi, dan peningkatan efektifitas serta efisiensi proses logistik.

## BAB 2 TINJAUAN PUSTAKA

### 2.1 Hasil Penelitian Terdahulu

Agen bergerak dengan mengikuti berbagai metode baik dengan dikontrol langsung oleh pengguna menggunakan *remote control* atau dengan cara yang lebih modern, robot otonom, yang bergerak secara mandiri. Gerakan tersebut diproses oleh robot yang disebut *path – planning*. Di sebagian besar aplikasi *path planning*, objek (agen) diharapkan mencapai target yang telah ditentukan sebelumnya dengan mengikuti jalur terpendek dan menghindari rintangan. Masalah navigasi ini juga disebut optimasi penghindaran rintangan (*obstacle avoidance*) [7]. Umumnya algoritma Dijkstra adalah yang sering digunakan untuk menemukan rute terpendek, namun seiring waktu, muncul berbagai modifikasi agar performa yang dihasilkan semakin meningkat [8]. Algoritma dua tahap adalah salah satunya. Pada tahap pertama, metode yang optimal untuk satu langkah, diterapkan secara iteratif. Dalam setiap langkah, rintangan pada garis lurus antara posisi saat ini dan target diasumsikan sebagai rintangan tunggal. Itu diwujudkan dengan menggunakan representasi geometris. Pada tahap kedua dari algoritma, rintangan direpresentasikan dengan bujur sangkar, yang panjang sisinya diatur sesuai dengan jarak minimum antar rintangan. Dengan diskritisasi, masalahnya menjadi jalur terpendek dalam grafik, dan diselesaikan dengan menerapkan algoritma Dijkstra [7]. Bahkan kita bisa mengembangkannya pada rintangan yang bersifat dinamis, meskipun masih cukup menantang secara komputasional karena memerlukan penambahan variabel waktu pada dimensi pencarian [9].

Metode diatas sudah cukup baik dan efektif, namun masih berlaku hanya kepada satu agen. Kelebihan lain dari robot kecil adalah pada kuantitas (*multi agent*). Jumlah yang lebih banyak tentu bisa memberikan keuntungan walaupun juga diperlukan mekanisme baru untuk menghubungkan masing – masing robot kepada yang lain. Pendekatan baru berbasis AI (*Artificial Intelligent*) untuk MAPF (*Multi Agent Path Finding*) [5], salah satunya disebut PRIMAL, memasukkan MAPF ke dalam kerangka pembelajaran di mana agen diharapkan untuk mempelajari kebijakan terdesentralisasi penuh di bawah demonstrasi sistem utama [10]. Metode pembelajaran seperti itu disebut juga *reinforcement learning*.

Pengimplementasian MAPF yang lain adalah MAPD (*Multi-Agent Pickup and Delivery*) yaitu versi kontinu atau *lifelong* dari MAPF [2]. Untuk menjalankan tugas, agen harus bergerak terlebih dahulu dari lokasi saat ini ke lokasi pengambilan tugas dan kemudian ke lokasi pengiriman tugas. Masalah MAPD adalah untuk menetapkan tugas ke agen (TA) [4] dan merencanakan jalur bebas tabrakan bagi mereka untuk menjalankan tugas mereka [3]. Ada dua jenis model untuk masalah ini, yaitu *online* dan *offline* [2], [3]. *Online* berarti jumlah tugas yang dimiliki bisa dimanipulasi secara *real time* sedangkan *offline* berarti jumlah tugas sudah ditetapkan dari awal dan tidak dapat dirubah.

Untuk menyelesaikan MAPD, terdapat berbagai algoritma baik dari sisi pembagian tugas maupun penentuan jalur. Contohnya seperti ICBS (*Task Conflict-Based Search*) [11], *Task Assignment and Prioritized path planning* (TA-Prioritized) [3], *Task Assignment and Hybrid path planning* (TA-Hybrid) [3]. Ada pula algoritma khusus untuk menyelesaikan masalah pembagian tugas (TA) yang optimal semacam CENTRAL [2], *Token Passing* (TP) [2], *Token Passing with Task Swap* (TPTS) [2], maupun *Genetic Algorithm* (GA) [12]. Sedangkan yang khusus MAPF ada *Travelling Salesman Problem* (TSP) spesial yang dimodifikasi dari TSP biasa untuk menghindari tabrakan dan memperkirakan waktu perjalanan antar lokasi [3]. Selain itu, dikembangkan beberapa skenario tambahan layaknya kapasitas yang menghasilkan metode

seperti *Regret-based Marginal Cost Algorithm* (RMCA) [1] dan tenggat waktu tugas (*deadline*) [13]. Selain itu, terdapat pula metode peningkatan MAPF seperti *Rolling-Horizon Collision Resolution* (RHCR) yang bisa meningkatkan kualitas solusi berbagai metode MAPF serta diimplementasikan ke lebih banyak agen maupun jenis peta [6].

## 2.2 Dasar Teori

### 2.2.1 Multi Agent Pickup and Delivery (MAPD)

Di MAPD, diberikan satu set tugas dan tim agen kooperatif (setara dengan robot). Pekerjaan MAPD ada dua: pertama, kita harus menetapkan setiap tugas ke beberapa robot; kedua, kita perlu menemukan untuk setiap robot satu set jalur bebas tabrakan yang menjamin setiap tugas yang diberikan berhasil dilakukan. Masing-masing aspek ini, *task assignment* (TA) dan *Multi-agent Path Finding* (MAPF) itu sendiri sulit dikendalikan, yang membuat MAPD sangat menantang untuk dipecahkan dalam praktik.

### 2.2.2 Regret-based Marginal Cost Algorithm (RMCA)

RMCA merupakan pengembangan dari algoritma lainnya, yaitu *Marginal-cost Based Task Selection Algorithm* (MCA). MCA hanya mengatur perihal *task assignment* saja atau membagi tugas ke para agen. Sedangkan RMCA menerapkan bentuk informasi yang memiliki pandangan kedepan untuk memilih tugas yang tepat untuk ditugaskan pada tiap iterasi. Algoritma ini memilih tugas berikutnya berdasarkan selisih harga marginal dari memasukan tugas ke rute robot terbaik dan rute robot kedua terbaik lalu memberikannya ke robot dengan harga marginal yang lebih rendah.

Untuk tiap tugas  $i$  pada set tugas  $P^U$ , kita gunakan  $k_1^*$  untuk menunjukan agen yang memasukan  $i$  ke rutanya dengan biaya perjalanan terkecil sambil menghindari tabrakan, dimana...

$$(k_1^*, q_1^*, q_2^*) = \arg \min_{\substack{k_1 \in R, \\ 1 < q_1 \leq |o_k|, \\ q_1 < q_2 \leq |o_k| + 1}} \{t((o_k \oplus_{q_1} s_i) \oplus_{q_2} g_i) - t(o_k)\}$$

Agen terbaik kedua,  $k_2^* \in R \setminus \{k_1^*\}$  untuk menanggapi  $i$  adalah...

$$(k_2^*, p_1^*, p_2^*) = \arg \min_{\substack{k_2 \in R, \\ 1 < p_1 \leq |o_k|, \\ p_1 < p_2 \leq |o_k| + 1}} \{t((o_k \oplus_{p_1} s_i) \oplus_{p_2} g_i) - t(o_k)\}$$

Terdapat dua metode untuk menentukan tugas mana  $i^* \in P^U$  yang akan ditugaskan, RMCA(a) atau *absolute regret* dan RMCA(r) atau *relative regret*. Sedangkan untuk penyelesaian MAPF, RMCA masih mengandalkan algoritma lain, antara lain A\* atau PBS.

Keunggulan dari RMCA adalah waktu eksekusi yang lebih cepat daripada algoritma MAPD lain. RMCA juga memungkinkan proses *task assignment* dilakukan dengan memperhitungkan kapasitas. Hal ini membuat RMCA bisa menyelesaikan tugas lebih cepat lagi. Selain itu, RMCA bisa diterapkan pada permasalahan *lifelong* MAPD yang berarti tugas baru tiba terus-menerus.

### 2.2.3 Rolling-Horizon Collision Resolution (RHCR)

RHCR bukanlah algoritma yang berdiri sendiri, melainkan mengembangkan algoritma MAPF yang sudah ada. Perkembangan yang dimaksud adalah RCHR membuat penerapan algoritma menjadi *windowed* (berjendela). RHCR memiliki dua parameter yang perlu ditentukan oleh pengguna, *time horizon*  $w$  dan *replanning period*  $h$ . Horizon waktu menunjukkan bahwa penyelesaian MAPF berjendela mampu menyelesaikan tumbukan selama horizon dengan  $w$  langkah waktu. Sedangkan periode perencanaan ulang berarti penyelesaian MAPF berjendela perlu perencanaan ulang sekali tiap  $h$  langkah waktu.  $w$  sebaiknya lebih besar sama dengan  $h$  untuk menghindari tumbukan.

Di setiap episode MAPF Berjendela, katakanlah, mulai dari langkah waktu  $t$ , RHCR pertama-tama memperbarui lokasi awal  $s_i$  dan urutan lokasi tujuan  $g_i$  untuk setiap agen  $a_i$ . RHCR menetapkan mulai lokasi  $s_i$  agen  $a_i$  ke lokasinya pada *timestep*  $t$ . Kemudian, RHCR menghitung batas bawah pada jumlah langkah waktu  $d$  bahwa agen  $a_i$  perlu mengunjungi semua lokasi yang tersisa di  $g_i$ , yaitu...

$$d = \text{dist}(s_i, g_i[0]) + \sum_{j=1}^{|g_i|-1} \text{dist}(g_i[j-1], g_i[j])$$

di mana  $\text{dist}(x, y)$  adalah jarak dari lokasi  $x$  ke lokasi  $y$  dan  $|x|$  adalah kardinalitas pada barisan  $x$ .  $d$  lebih kecil dari  $h$  menunjukkan bahwa agen  $a_i$  mungkin selesai mengunjungi semua nya lokasi tujuan dan kemudian menganggur sebelum episode MAPF berjendela berikutnya dimulai pada *timestep*  $t + h$ . Untuk menghindari situasi ini, RHCR terus menetapkan lokasi tujuan baru untuk agen  $a_i$  sampai  $d$  jam. Setelah lokasi awal dan lokasi tujuan urutan untuk semua agen tidak memerlukan pembaruan lagi, RHCR memanggil pemecah MAPF berjendela untuk menemukan jalur untuk semua agen yang memindahkan mereka dari lokasi awal mereka ke semua lokasi tujuan mereka dalam urutan yang diberikan oleh urutan lokasi tujuan mereka dan bebas tumbukan untuk  $w$  langkah waktu pertama. Akhirnya, itu memindahkan agen untuk  $h$  langkah waktu di sepanjang jalur yang dihasilkan dan menghapus lokasi tujuan yang sudah dikunjungi dari urutan lokasi tujuan mereka.

Keunggulan RHCR bukan pada penambahan performa seperti kecepatan, melainkan fokus ke fleksibilitas (pada ruang waktu dan bisa diterapkan ke berbagai peta *warehouse*) dan skalabilitas (dimana lebih banyak agen diperhitungkan). Selain itu, RHCR bisa diterapkan pada permasalahan *lifelong* MAPD yang berarti tugas baru tiba terus-menerus.

## BAB 3 METODOLOGI

### 3.1 Metode yang digunakan

#### 1. Studi Literatur

Pada tahap studi literatur dilaksanakan untuk mempelajari dan mengkaji teori-teori serta mengumpulkan data penelitian mengenai perancangan sistem multi agen untuk tugas pengambilan dan pengiriman. Sumber-sumber yang digunakan pada studi literatur ini dipastikan memiliki kredibilitas yang terpercaya, seperti jurnal atau paper dari IEEE, arXiv, dan lain-lain.

#### 2. Perancangan Sistem

Pada tahap perancangan sistem, pertama dilakukan pembuatan program, algoritma dan lingkungan simulasi. Selanjutnya akan dilakukan proses penetapan parameter-parameter, seperti jumlah agen, jumlah tugas, posisi pengambilan serta tujuan kirim, dan lain-lain. Hasil dari kegiatan-kegiatan tersebut berupa simulasi penyelesaian tugas oleh agen-agen. Simulasi dan perancangan akan dilakukan dengan menggunakan bahasa pemrograman C++.

#### 3. Pengujian Sistem dan Evaluasi

Pada tahap pengujian sistem akan dilakukan dengan mendeteksi keakuratan dari algoritma yang dipakai. Pengujian didasarkan pada beberapa variabel seperti jumlah agen, waktu tempuh, dan jumlah tugas. Selanjutnya hasil dari pengujian akan dianalisa dan dievaluasi. Algoritma akan dievaluasi dan diperbaiki hingga mencapai tingkat akurasi tertentu.

#### 4. Penyusunan Laporan Tugas Akhir

Hasil akhir dari penelitian ini akan disusun dalam bentuk buku. Isi buku tersebut meliputi proses penelitian dari awal hingga akhir.

### 3.2 Bahan dan peralatan yang digunakan

- Komputer atau laptop
- Program C++ untuk perancangan algoritma dan simulasi
- Algoritma MAPD modifikasi

### 3.3 Urutan pelaksanaan penelitian

Kegiatan	Minggu Ke -															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Studi Literatur																
Perancangan sistem																
Pengujian sistem dan Evaluasi																
Penyusunan Laporan Tugas Akhir																

## DAFTAR PUSTAKA

- [1] Z. Chen, J. Alonso-Mora, X. Bai, D. D. Harabor, and P. J. Stuckey, "Integrated Task Assignment and Path Planning for Capacitated Multi-Agent Pickup and Delivery," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 5816–5823, 2021, doi: 10.1109/LRA.2021.3074883.
- [2] H. Ma, T. K. S. Kumar, J. Li, and S. Koenig, "Lifelong multi-Agent path finding for online pickup and delivery tasks," *Proc. Int. Jt. Conf. Auton. Agents Multiagent Syst. AAMAS*, vol. 2, no. Aamas, pp. 837–845, 2017.
- [3] M. Liu, H. Ma, J. Li, and S. Koenig, "Task and path planning for multi-agent pickup and delivery," *Proc. Int. Jt. Conf. Auton. Agents Multiagent Syst. AAMAS*, vol. 2, pp. 1152–1160, 2019.
- [4] G. A. Korsah, M. B. Dias, and A. Stentz, "A Comprehensive Taxonomy for Multi-Robot Task Allocation Background : Gerkey and Matarì ' , " *Ijrr*, pp. 1–29, 2013.
- [5] R. Stern *et al.*, "Multi-agent pathfinding: Definitions, variants, and benchmarks," *Proc. 12th Int. Symp. Comb. Search, SoCS 2019*, pp. 151–158, 2019.
- [6] J. Li, A. Tinka, S. Kiesel, J. W. Durham, T. K. Satish Kumar, and S. Koenig, "Lifelong multi-agent path finding in large-scale warehouses," *Proc. Int. Jt. Conf. Auton. Agents Multiagent Syst. AAMAS*, vol. 2020-May, pp. 1898–1900, 2020.
- [7] N. Gasilov, M. Doğan, and V. Arici, "Two-stage shortest path algorithm for solving optimal obstacle avoidance problem," *IETE J. Res.*, vol. 57, no. 3, pp. 278–285, 2011, doi: 10.4103/0377-2063.83650.
- [8] A. Turkey Rashid, A. Abdulkareem Ali, L. Fortuna, A. Rashid, A. A. Ali, and M. Frasca, "26 Path planning and obstacle avoidance based on shortest distance algorithm," no. December, 2017, [Online]. Available: <https://www.researchgate.net/publication/322775664>.
- [9] M. Phillips and M. Likhachev, "SIPP: Safe interval path planning for dynamic environments," *Proc. - IEEE Int. Conf. Robot. Autom.*, vol. 1, pp. 5628–5635, 2011, doi: 10.1109/ICRA.2011.5980306.
- [10] L. Zhiyao and G. Sartoretti, "Deep reinforcement learning based multi-agent pathfinding," *Artif. Intell. a Mod. approach*, pp. 1–8, 2020.
- [11] C. Henkell, J. Abbenseth, and M. Toussaintl, "An Optimal Algorithm to Solve the Combined Task Allocation and Path Finding Problem," *IEEE Int. Conf. Intell. Robot. Syst.*, pp. 4140–4146, 2019, doi: 10.1109/IROS40897.2019.8968096.
- [12] A. C. L. C. Queiroz, H. S. Bernardino, A. B. Vieira, and H. J. C. Barbosa, "Solving Multi-Agent Pickup and Delivery Problems Using a Genetic Algorithm," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 12320 LNAI, no. February 2021, pp. 140–153, 2020, doi: 10.1007/978-3-030-61380-8\_10.
- [13] X. Wu *et al.*, "Multi-Agent Pickup and Delivery with Task Deadlines," *14th Int. Symp. Comb. Search, SoCS 2021*, no. ii, pp. 206–208, 2021.



## **LAMPIRAN**