ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika

ISSN(p): 2338-8323 | ISSN(e): 2459-9638 | Vol. 10 | No. 4 | Halaman 859 - 872
DOI : http://dx.doi.org/10.26760/elkomika.v10i4.859 Oktober 2022

Sistem Multi Agen untuk Pelayanan *Drone* pada *Groundbase Docking Station*

DIMAS NOVIAN ADITIA SYAHPUTRA[1] , RADEN SANGGAR DEWANTO[2] , DADET PRAMADIHANTO[3]

[1]Departemen Teknik Elektro, Politeknik Elektronika Negeri Surabaya
[2]Departemen Teknik Mekanika dan Energi, Politeknik Elektronika Negeri Surabaya
[3]Departemen Teknik Informasi dan Komputer, Politeknik Elektronika Negeri Surabaya
Email: dimasa52@gmail.com

Received 6 Juli 2022 **|** *Revised* 26 Juli 2022 **|** *Accepted* 3 Agustus 2022

ABSTRAK

Multi-Agent System (MAS) diajukan sebagai solusi untuk mengatasi permasalahan pada groundbase sebuah DDS, di mana pada groundbase terdapat AGV yang bertugas untuk membantu Drone beraktifitas di DDS hingga kemudian berangkat kembali menuju DDS lain. Metode auction serta contract antar agent digunakan dalam pemrosesan request dari Drone dan pembagian sumber daya. Pada MAS diterapkan algoritma prioritas sebagai solusi apabila terjadi konflik antar agen. Pengujian dengan simulasi pada CoppeliaSim dan ROS (Robot Operating System) menunjukkan bahwa penggunaan algoritma prioritas berdampak positif pada MAS yang dibuat. Pada DDS dengan skenario 11 AGV, terjadi peningkatan kemampuan DDS dalam menerima dan memproses request yang datang dari 57.9% menjadi 100%, serta pemecahan deadlock yang terjadi pada DDS dari 10 menjadi 0 sehingga seluruh request dapat terselesaikan.

***Kata kunci**: Multi-Agent System, Algoritma Prioritas, Drone Docking Station, AGV*

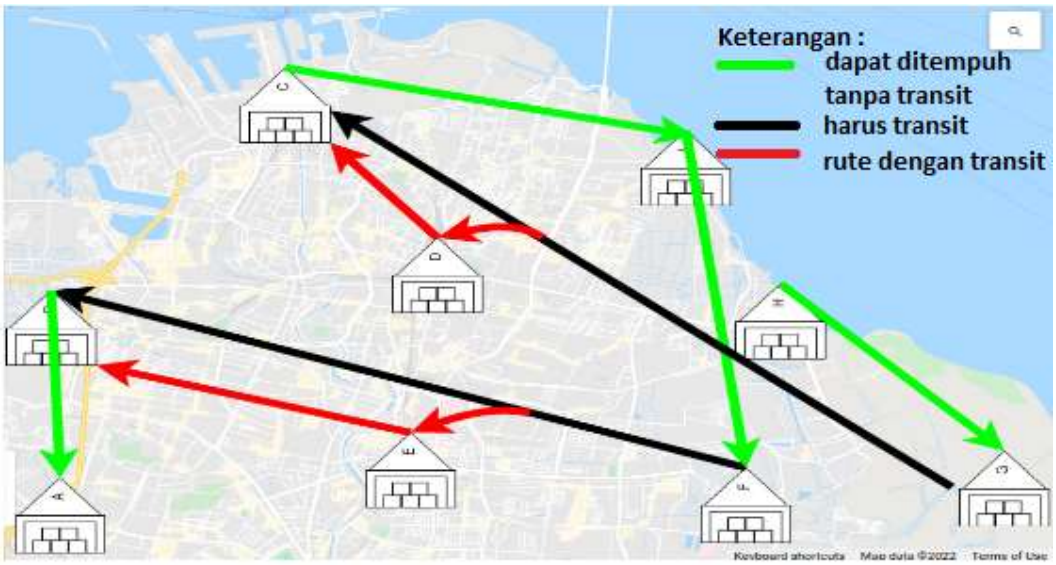
ABSTRACT

The Multi-Agent System (MAS) was proposed as a solution to overcome problems in the groundbase of a DDS, where on the groundbase there is an AGV whose job is to help drones carry out activities in DDS and then depart for another DDS. Auction methods and contracts between agents are used in processing requests from drones and sharing resources. In MAS, a priority algorithm is applied as a solution in the event of a conflict between agents. Tests with simulations on CoppeliaSim and ROS (Robot Operating System) show that the use of priority algorithms has a positive impact on the created MAS. In DDS with 11 AGV scenario, there is an increase in DDS ability to receive and process incoming requests from 57.9% to 100%, as well as solving deadlocks that occur in DDS from 10 to 0 so that all requests can be resolved.

***Keywords**: Multi-Agent System, Priority Algorithm, Drone Docking Station, AGV*

1. PENDAHULUAN

Delivery Drone sebagai bagian dari penerapan *Artificial Intelligent* di lingkungan *smart warehousing* dan otomatisasi sistem logistic **(Pandian, 2019**),menjadi alternatif pengiriman barang yang murah, cepat, dan efisien terutama di daerah perkotaan dengan lalu lintas yang padat. Di sisi lain penggunaan *delivery Drone* memiliki tantangan tersendiri yaitu keterbatasan jarak yang mampu ditempuh oleh *Drone* karena kapasitas baterai yang terbatas. Untuk membuat sistem pengiriman barang dengan *Drone* yang cepat dan efisien diperlukan banyak *Drone* (D) serta banyak *Drone Docking Station* (DDS) seperti yang ditunjukkan pada Gambar 1. DDS digunakan sebagai tempat untuk *Drone* transit untuk mengisi daya serta melakukan bongkar muat muatannya. DDS terdiri dari dua bagian yaitu bagian udara dan bagian darat, pada bagian udara mengatur lalu lintas *Drone* dari satu DDS ke DDS lain, sedangkan bagian darat yang disebut *Groundbase Drone Docking Station* bertugas untuk melayani *Drone* ketika mendarat. Pada GDDS terdapat beberapa *Automated Guided Vehicle* (AGV) yang berfungsi sebagai landasan bagi *Drone* serta melayani *Drone* dari tempat pendaratan *Runway* (R) menuju tempat *Charging Station* (CS) tempat pengisian daya *Drone* dan atau menuju *Loading Station* (LS) tempat untuk *Drone* bongkar muat. Jika CS,LS dan R penuh saat akan dituju oleh *Drone* maka AGV akan menuju *Parking Area* (PA) yang berfungsi sebagai tempat sementara AGV selama menunggu antrian. Menurut Rashidah **(Rashidah, dkk, 2018)** penggunaan AGV pada lingkungan gudang logistik seperti pada DDS ini dapat mengurangi *cycle time* untuk keseluruhan sistem dengan batasan jumlah AGV tertentu dan sistem yang terprogram. Menurut Osmond **(Osmond, dkk, 2019)** Dalam transportasi dan logistik yang kompleks dan dinamis, MAS banyak digunakan untuk mensimulasikan solusi untuk menyelesaikan permasalahan penentuan rute kendaraan (*vehicle routing problem*) dengan mengadopsi perilaku kerja sama, negosiasi, maupun komunikasi. Sehingga pendekatan *Multi-Agent System* dapat digunakan untuk memodelkan sistem GDDS.



Gambar 1. Banyak DDS dalam Suatu Kota

Penerapan MAS untuk kendali transportasi banyak robot telah dilakukan dengan penerapan pada berbagai bidang. Ribas **(Ribas, dkk, 2013)** telah berhasil membuat sebuah model kontroler untuk sistem transportasi AGV menggunakan *Agent-Based Model,* tersdapat dua kelas agen dalam sistem yang dibuat, yaitu agen yang mengatur pergerakan AGV dan agen

yang merepresentasikan elemen lain secara penuh. Sistemnya dibangun pada *framework* Netlogo sehingga memungkinkan melakukan konfigurasi ulang secara cepat. Kato (**Kato, dkk, 2020)** membangun model lingkungan simulasi MAS untuk gudang logistik yang memungkinkan pemodelan arsitektur agen dan arsitektur pesan terpadu dapat lebih murah dan lebih mudah dimodifikasi dibandingkan dengan lingkungan simulasi MAS konvensional. Kalisa **(Kalisa, dkk, 2016)** telah berhasil membuat arsitektur agen yang mampu melakukan penentuan tindakan dengan tepat dan efisien berdasarkan *database* yang tersedia dalam studi kasus *data warehousing*. Karagoz **(Karagoz, dkk, 2014)** telah berhasil membuat model koordinasi banyak robot berbentuk piringan yang *independent*, robot-robot yang ada juga bertindak sebagai *obstacle* dinamis. Setiap robot pada penelitian ini mengetahui dengan pasti posisi dan ukuran dari robot lain yang ada disekitarnya sehingga dapat ditentukan kecepatan aktual masing-masing robot. Gerrits **(Gerrits, dkk, 2019)** telah berhasil mendesain dan mengimplementasikan sebuah model simulasi berbasis agen untuk melakukan otomasi pada terminal peti kemas. Agen *Traffict Manager* yang dibuat mampu mengatur seluruh opersi yang ada di bagian pelabuhan termasuk memecahkan konflik dan mencegah *deadlock*.

Walaupun telah banyak dilakukan, kebanyakan sistem yang dibuat berada pada lingkungan berupa *grid* sehingga AGV hanya dapat berjalan melalui rute-rute tertentu, sistem komunikasi dan penentuan keputusan dari penelitian yang ada bersifat sentral yang berarti komunikasi antar agen harus melalui agen pusat sebagai perantara. Pada penelitian ini AGV dapat bergerak bebas pada lingkungan (lingkungan *non-grid*) sehingga meningkatkan kemungkinan terjadinya tabrakan karena AGV lain yang bergerak bebas pada area DDS juga bertindak sebagai *obstacle* dinamis bagi AGV lain, di lain sisi memungkinkan AGV mengambil rute yang lebih pendek daripada saat bergerak pada lingkungan berbentuk *grid* . MAS akan berusaha mengatur AGV-AGV dan apabila terjadi konflik saat bernavigasi dalam DDS, sistem akan menerapkan algoritma prioritas dan mempersilahkan AGV dengan prioritas tertinggi bergerak terlebih dahulu dan AGV lain menunggu. Pada bagian selanjutnya akan dijelaskan metode atau sistem yang digunakan pada penelitian ini.

2. METODE PENELITIAN

Tujuan penelitian ini adalah untuk membuat sebuah sistem multi agen sebagai pendekatan untuk mengatur sebuah grup AGV dalam GDDS. Sistem multi agen yang dibuat untuk melayani Drone dalam pengisian daya dan pemuatan memiliki arsitektur seperti yang ditampilkan pada Gambar 2. Pada sistem DDS terdapat 12 jenis agen, masing-masing agen memiliki tugas dan fungsinya dengan jumlah yang beragam untuk setiap jenis agen. Infrastruktur DDS yang dibangun memiliki keterbatasan kapasitas. Batas kapasitas ini dinyatakan pada Persamaan (1).

$$P \leq K \geq N \leq M \tag{1}$$

Dengan:

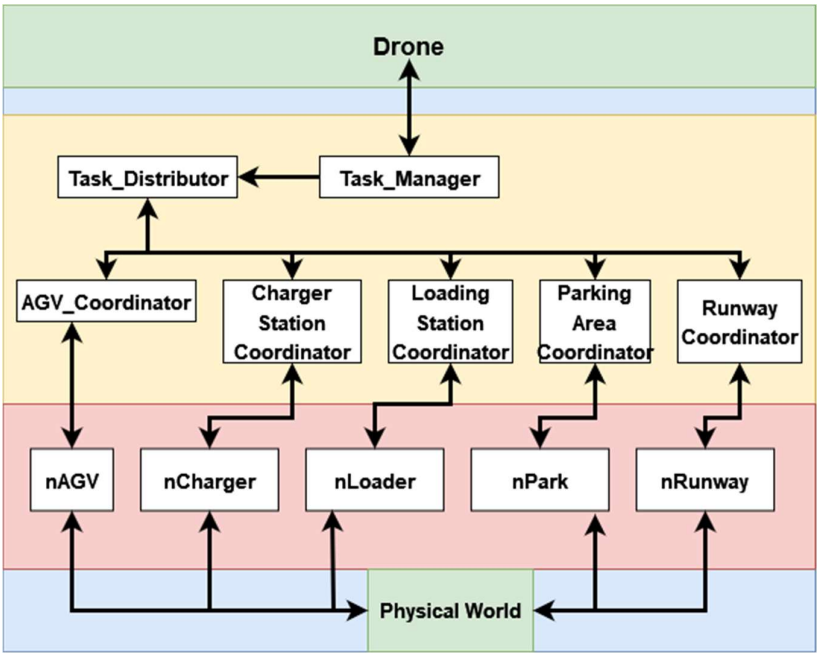
K = Jumlah AGV

N = Jumlah *Runway*

P = Jumlah *Drone*

M = Jumlah *Docking Station*

Kapasitas maksimal dari DDS sama dengan jumlah AGV pada DDS tersebut



Gambar 2. Arsitektur MAS

Tabel 1. Agen dalam MAS

No.	Nama Agen	Aktifitas Agen
1	*Task Manager*	-Menerima tugas dari drone -memutuskan untuk menerima atau menolak tugas -meneruskan tugas yang diterima pada Task distributor
2	*Task Distributor*	-Mendistribusikan Tugas -menentukan tujuan pertama dari Drone dalam GDDS -Mengirimkan perintah lelang AGV
3	*AGV Coordinator*	-Menentukan Tujuan ke-2 dari Drone -melakukan lelang AGV
4	*Charger Station Coordinator*	-Mengatur dan memilih charging area
5	*Loading Station Coordinator*	-Mengatur dan memilih loading area
6	*Parking Area Coordinator*	-Mengatur dan memilih parking area
7	*Runway Coordinator*	-Mengatur dan memilih runway area
8	nAGV	-Mengikuti lelang -Bernavigasi pada DDS untuk mengantar AGV -Memecahkan konflik antar AGV
9	*nCharger*	-Mengisi daya Baterai Drone
10	*nLoader*	-Bongkar muat muatan Drone
11	*nPark*	-Menampung sementara AGV dalam kondisi stanby atau saat CS,LS, atau R penuh
12	*nRunway*	-tempat AGV mengangkut dan menerbangkan drone
13	Drone	-memberi tugas pada GDDS

2.1 Metode *Auction and Contract*

Untuk melakukan pendistribusian tugas dengan efisien, setiap agen berinteraksi satu sama lain dengan negosiasi **(Axak, dkk, 2021)**. Protokol komunikasi antar agen berpengaruh pada metode interaksi agen serta penentuan keputusan secara kolektif. Pengalokasian dan penjawadwalan tugas dalam MAS merupakan salah satu penentuan keputusan yang berpengaruh pada proses seluruh sistem MAS. Salah satu metode yang digunakan dalam pengalokasian tugas adalah metode *Auction* **(Braquet, dkk, 2021),** dimana konsensus dicapai dengan menentukan agen terbaik dari sekumpulan agen yang siap melakukan tugas tersebut. Beberapa parameter seperti jumlah baterai, kemampuan angkut dan posisi dari sebuah AGV dapat digunakan sebagai properties untuk menghitung kemampuan agen tersebut dalam mengikuti lelang tugas sehingga agen yang terpilih dapat dipastikan mampu menyelesaikan tugas yang telah diambil **(Liu, dkk, 2018)**. Pada jurnal ini strategi pengalokasian tugas dilakukan dengan metode Auction bersyarat, dimana AGV dengan pada DDS memiliki propertis yang ditunjukkan pada Persamaan (2).

$$AGV_{id} = \{AGV_{battery}, AGV_{position}, AGV_{status}\} \quad (2)$$

Dengan :

AGV_{id} = nomor identitas AGV

$AGV_{battery}$ = tingkat keterisian baterai, AGV 0 hingga 100

$AGV_{position}$ = posisi AGV, 0 jika sedang bergerak, 1 jika berada di R, 2 jika berada pada CS, 3 jika berada pada LS, dan 4 jika berada pada PA

AGV_{status} = status AGV, 0 jika tersedia, 1 jika sedang bertugas

Dari *properties* tersebut, AGV yang diizinkan mengikuti *auction* adalah AGV dengan status 0 dan berada pada R atau PA, dengan AGV yang berada pada R memiliki prioritas penerimaan tugas lebih tinggi sehingga Drone dapat langsung mendarat pada AGV yang berada pada R. Proses auction tidak hanya dilakukan ketika tugas datang, apabila pada R terdapat titik yang tidak terisi AGV maka AGV yang berada pada PA dan dalam kondisi tersedia akan melakukan lelang untuk menentukan AGV mana yang menuju ke R untuk mengisi titik yang tersedia tersebut pada tahap ini ketersediaan baterai pada AGV menjadi pertimbangan utama, AGV dengan baterai tertinggi akan terpilih. Selain pengalokasian tugas yang tepat, strategi penjadwalan tugas yang tepat dapat mempercepat proses pengerjaan tugas oleh AGV **(Liu, dkk, 2019)**. Pada Gambar 2 ditampilkan hirarki antar agen dalam MAS. Hirarki ini menunjukkan rantsai komunikasi antar agen. Drone yang datang membawa informasi *Request* pendaratan serta keperluannya yang dinotasikan pada Persamaan (3).

$$task = \{task_{id}, task_{battery}, task_{needcharge}, task_{needload}, task_{destinationcost}\} \quad (3)$$

Dimana:

$task_{id}$ = nomor identitas *Drone*.

$task_{battery}$ = kondisi keterisian baterai *Drone* saat ini.

$task_{needcharg}$ = kebutuhan untuk melakukan daya, bernilai 1 jika membutuhkan pengisian daya dan bernilai 0 jika tidak membutuhkan pengisian daya.

$task_{needload}$ = kebutuhan untuk melakukan bongkar/muat, bernilai 1 jika

membutuhkan bongkar/muat dan bernilai 0 jika tidak membutuhkan bongkar/muat.

$task_{destinationcost}$ = battery yang dibutuhkan *Drone* untuk menuju tujuan pengiriman.

Request dari *Drone* akan diterima oleh agent task manager, informasi tersebut kemudian dicatat dalam database tugas dan diteruskan pada agen task distributor (TD). Agen TD akan meminta informasi ketersediaan AGV pada *Coordinator Runway* (CR) dan ketersediaan CS,PA serta LS pada agen *Coordinator Charging Station* (CCS), agen *Coordinator Parking Area* (CPA), dan *Coordinator Loading Station* (CLS) sehingga diperoleh empat himpunan data {C, P, L, A}, dimana C adalah himpunan kondisi CS, P himpunan kondisi PA , L himpunan kondisi LS, serta A kondisi ketersediaan AGV pada R dan PA. Jika A tidak tersedia maka *Drone* akan diinstruksikan untuk mencari DDS yang lain. Jika tersedia maka agen TD akan menentukan tujuan pertama dari *Drone* tersebut kemudian TD akan melelang tugas dengan mengirimkan informasi tentang tugas baru ke agen AGV yang tersedia, informasi tersebut ditampilkan pada Persamaan (4).

$$task = \{task_{id}, task_{battery}, task_{needcharge}, task_{needload}, task_{destinationcost}, task_{destination}\} \quad (4)$$

Dengan :

Keterangan $task_{id}$ hingga $task_{destinationcost}$ sama dengan keterangan Persamaan (3)

$task_{destination}$ = Tujuan AGV

Informasi tersebut kemudian dikirimkan pada setiap AGV yang dalam kondisi tersedia, AGV-AGV tersebut akan memperkirakan biaya tugas tersebut sehingga diperoleh sebuah set nilai pada Persamaan (5).

$$E_{task} = \{e_1, e_2, \dots, e_n\} \quad (5)$$

Kemudian dilakukan pemilihan AGV berdasarkan nilai terkecil dari set biaya tugas dari AGV-AGV dengan Persamaan (6).

$$AGV_{winner} = arg\ \min(E_{task}) \quad (6)$$

Setelah AGV dipilih maka kontrak antar Drone dan MAS telah terjalin, kontrak tersebut meliputi tujuan-tujuan yang akan dituju oleh Drone hingga siap untuk terbang kembali. Pengerjaan kontrak dimulai dengan *Drone* mendarat di atas AGV kemudian AGV akan bernavigasi ke tujuan yang telah ditentukan.

2.2 Algoritma Prioritas (AP)

Lingkungan *non grid* memungkinkan AGV bergerak bebas pada DDS memiliki keunggulan dibandingkan pada lingkungan berbentuk *Grid* dimana pada lingkungan *non grid* AGV dapat mengambil jarak terpendek dari satu titik ke titik yang lain. Di lain sisi pada lingkungan *non grid* kemungkinan terjadi tabrakan antar AGV saat bernavigasi pada DDS meningkat. Tabrakan atau konflik antar AGV ini dapat dibagi menjadi tabrakan yang dapat diselesaikan secara otomatis dan tabrakan yang tidak dapat diselesaikan secara otomatis **(Kapitan, dkk, 2017)**. Pengaturan kecepatan dan rute dari AGV digunakan untuk mencegah terjadinya tabrakan antar AGV pada lingkungan dengan bentuk *Grid* **(Pamosoaji, 2019)**. Selain pengaturan kecepatan dan rute, faktor *obstacle avoidance* dapat ditambahkan pada algoritma yang

digunakan pada strategi *path planning* untuk menghindari konflik *node* pada lingkungan AGV berbentuk grid sehingga *delay rate* akibat konflik berkurang **(Liyun, dkk, 2021)**. Pada jurnal ini diterapkan metode algoritma pemecah konflik dengan memanfaatkan status prioritas dari tugas yang sedang dikerjakan oleh AGV-AGV yang mengalami konflik. Strategi prioritas yang diterapkan pada algoritma tersebut adalah sebagai berikut:

1. Jika AGV di R tidak tersedia dan ada AGV di PA yang tersedia , maka AGV tersebut akan dipanggil menuju R dengan metode auction yang dijelaskan pada bagian sebelumnya.

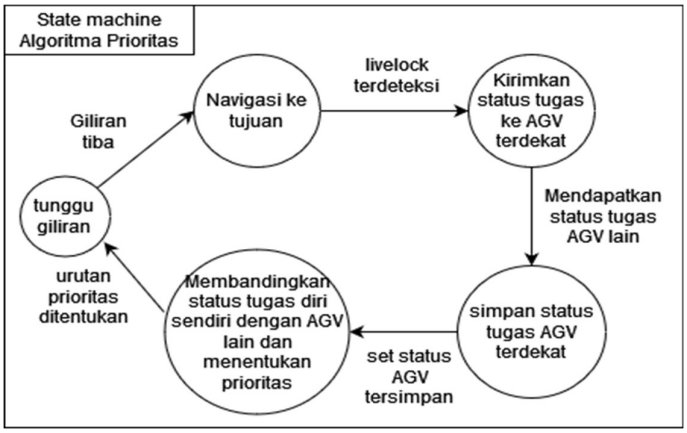
2. Jika terjadi konflik antar AGV saat melakukan navigasi, maka akan diterapkan aturan pada Persamaan (7).

$$AR_{load} > AR_{empty} > ACS > ALS > APA \quad (7)$$

dengan:

AR_{load} = AGV menuju ke R untuk menerbangkan D.
AR_{empty} = AGV menuju ke R untuk memuat D.
ACS = AGV menuju ke CS.
ALS = AGV menuju ke LS.
APA = AGV menuju ke PA.

Dengan adanya aturan pertama maka pada R akan selalu tersedia AGV untuk pendaratan selama kapasitas DDS masih tersedia. Semua proses antrian menuju CS, LS , dan R ditampung terlebih dahulu pada PA.



Gambar 3. FSM Algoritma Prioritas

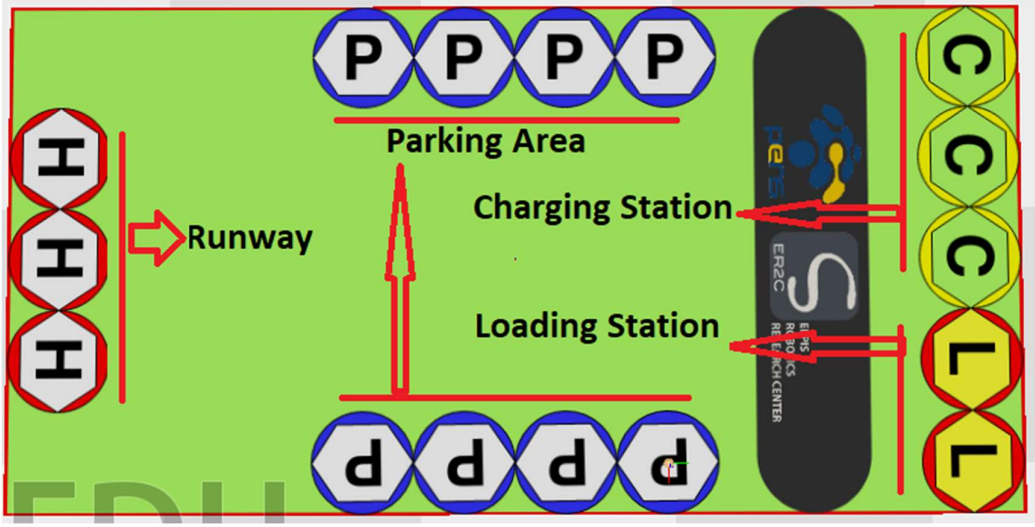
Pada saat AGV bernavigasi pada DDS dan terjadi *Livelock* atau *deadblock* antar AGV yang mengakibatkan AGV-AGV tersebut saling menunggu untuk mendapatkan lintasan terlebih dahulu maka agen AGV tersebut akan bernegosiasi untuk menentukan AGV mana yang lewat terlebih dahulu sesuai dengan aturan kedua, proses ini digambarkan pada Gambar 3.

3. HASIL DAN PEMBAHASAN

Model yang telah direncanakan pada bagian sebelumnya disimulasikan menggunakan *software* CoppeliaSim versi *Education* (*free*), pada simulasi AGV dimodelkan dengan robot Pioneer3dx

yang telah tersedia pada *model browser* CoppeliaSim. Model Pioneer3dx kemudian dimodifikasi *script*-nya sehingga dapat berkomunikasi dengan ROS. *Behavior* agen-agen dalam MAS berjalan pada *environment* ROS sedangkan AGV dan sensor-sensor pada DDS berjalan di *software* CoppeliaSim. Pengujian dilakukan sebanyak 50 kali dengan jumlah AGV (K) yang bervariasi antara 4 hingga 11 AGV pada lingkungan *Drone Docking Station* yang bentuknya tetap ditunjukkan pada Gambar 4, lingkungan ini terdiri dari 3 node R, 8 node PA, 3 node CS dan 2 node LS. Pada masing-masing *node* ditandai dengan simbol unik,yaitu :

1. *Runway* (R) = R1,R2,R3
2. *Charging Station* (CS) = C1,C2,C3
3. *Loading Station* (LS) = L1,L2
4. *Parking Area* (PA) = P1,P2,….,P8
5. AGV (A) = A1,A2,…,An



Gambar 4. *Environment* DDS di CoppeliaSim

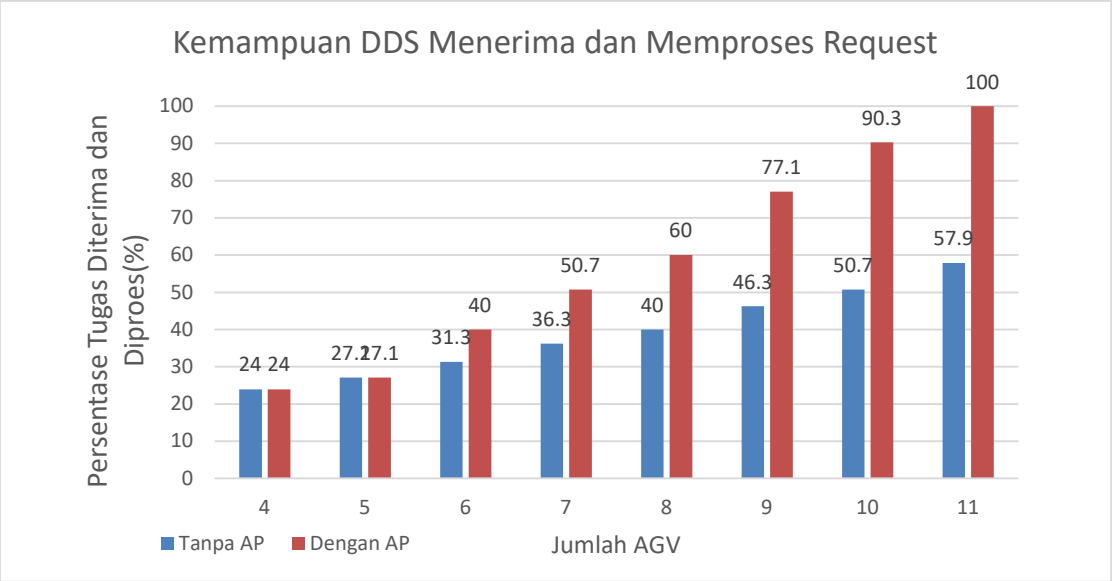
Pada setiap percobaan yang dilakukan terdiri dari 20 set *task request* yang bersifat acak dan diterima secara *online* dalam interval acak antara 1 hingga 100 unit waktu. Dari pengujian yang dilakukan didapatkan data perilaku agen-agen dalam MAS yang kemudian di analisa untuk menentukan performa dari MAS yang telah dirancang. Tolak ukur yang digunakan untuk mengukur performa sistem yang dirancang antara lain:

1. Kemampuan DDS menerima dan memproses *request* dari *Drone*.
2. Tingkat keberhasilan agen dalam melakukan distribusi tugas dan melakukan kontrak antar agen.
3. Tingkat keberhasilan agen pada MAS untuk menyelesaikan kontrak yang telah disepakati (S).

3.1 Uji Coba Kemampuan DDS Menerima dan Memproses *Request* dari *Drone*.

Pada simulasi, jumlah AGV sebagai *landing pad* pada DDS menjadi batasan jumlah *request* yang dapat dikerjakan oleh DDS pada satu waktu. Pada 50 kali simulasi dengan 20 set *request* pada setiap simulasinya yang diumpankan secara acak pada interval 1 hingga 100 unit waktu menunjukkan peningkatan jumlah *request* yang dapat dilayani oleh MAS berbanding lurus dengan jumlah AGV. Gambar 5 menunjukkan pada skenario DDS dengan 4 AGV hingga 11

AGV berturut-turut 24%, 27.1%, 31.3%, 36.3%, 40%, 46.3%, 50.7%, 57.9%. Percobaan ini dilakukan tanpa menerapkan algoritma tugas prioritas sehingga terjadi hambatan dalam navigasi yang memperlambat proses penanganan tugas ketika lalu lintas dalam DDS semakin padat. Uji coba berikutnya dilakukan dengan menggunakan algoritma prioritas. Hasil percobaan dengan menerapkan algoritma prioritas pada MAS menunjukkan peningkatan signifikan *request* yang berhasil dilayani. Perbedaan jumlah *request* yang diterima terlihat pada jumlah AGV 6 sampai dengan 11, hal ini karena pada jumlah AGV 4 hingga 5 sistem belum mengalami *deadblock* dan *Livelock* sehingga ada atau tidaknya algoritma prioritas tidak berpengaruh, sedangkan pada DDS dengan 6 AGV mulai terjadi *Livelock* yang membutuhkan algoritma prioritas untuk penyelesaiannya. Gambar 5 menunjukkan penggunaan algoritma prioritas memberikan peningkatan kemampuan DDS dalam menerima dan memproses request yang datang semakin meningkat seiring bertambahnya AGV yang bertugas, Dengan Penerapan AP penerimaan request meningkat secara signifikan, dimana pada skenario DDS dengan 4 AGV hingga 11 AGV berturut-turut 24%, 27.1%, 40%, 50.7%, 60%, 77.1%, 90.3%, 100%.



Gambar 5. Kemampuan DDS Menerima dan Memproses *Request*

3.2 Uji Coba Tingkat keberhasilan Agen dalam Melakukan Distribusi Tugas dan Melakukan Kontrak Antar Agen

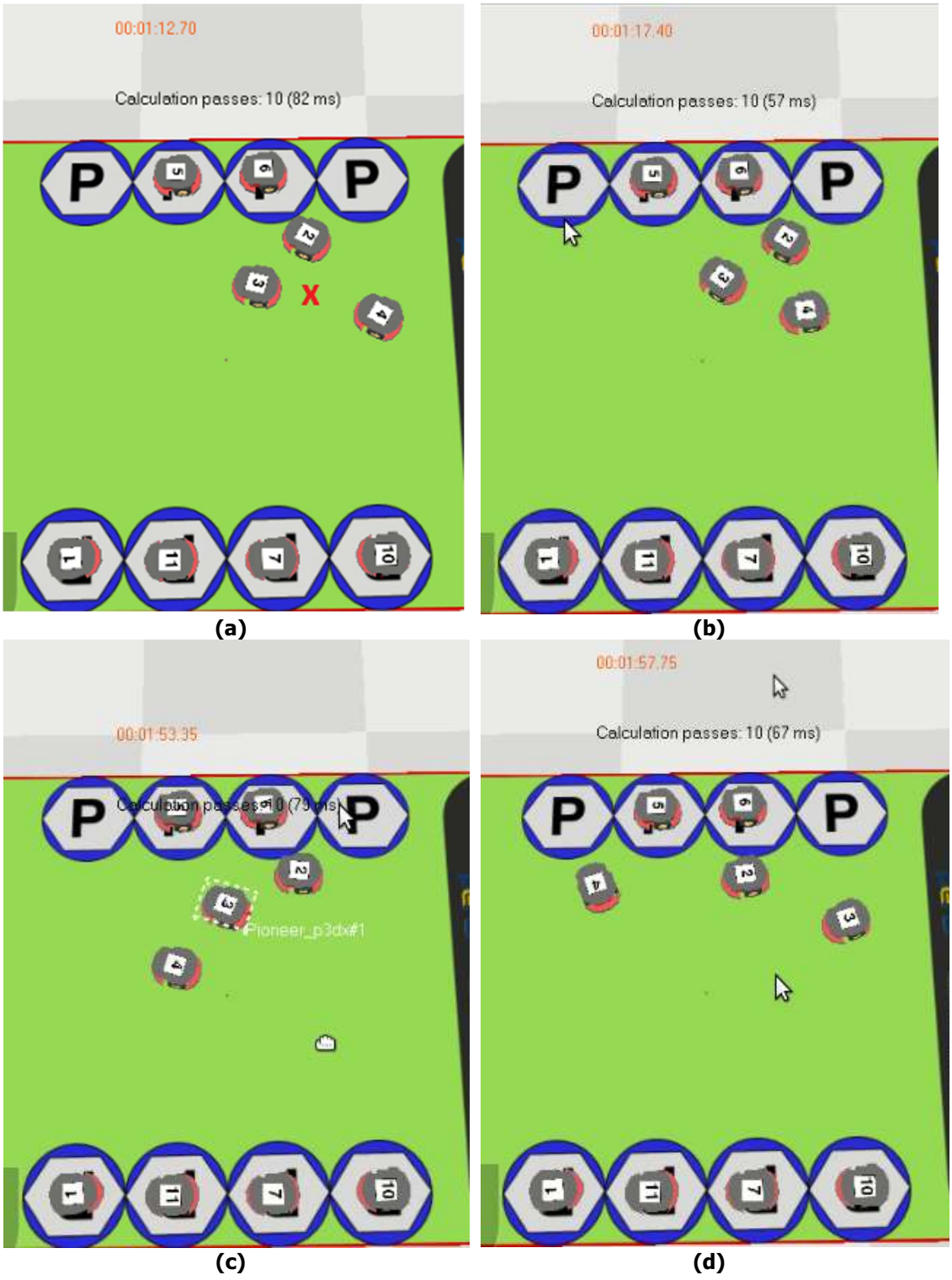
Pengujian terhadap MAS yang diajukan menunjukkan bahwa perilaku agen-agen pada sistem dalam melakukan pembagian tugas dapat berkerja dengan baik. Sistem kontrak antar AGV dan *node* pada DDS mampu mencegah lebih dari satu agen memiliki tujuan yang sama. Pada pengujian ini tingkat keberhasilan agen-agen pada sistem yang diajukan adalah 100%. Tabel 2 menunjukkan informasi *request* dari *Drone*, AGV yang terpilih untuk melayani *Drone* tersebut serta rute destinasi dari AGV tersebut.

Tabel 2. Data Tugas dan Tujuan AGV

task Id	kondisi baterai %	Keperluan Charging	Keperluan Loading/ Unloading	Minimal Kebutuh Baterai %	AGV Terpilih	Destinasi					
						1	2	3	4	5	6
1	43	1	0	100	A1	C1	P2	R3	-	-	-
2	29	0	1	28	A2	L1	P1	R2	-	-	-
3	83	1	0	92	A3	C2	P3	R1	-	-	-
4	65	0	1	42	A4	L2	P1	R2	-	-	-
5	23	1	0	73	A6	C1	P2	R3	-	-	-
6	72	1	1	90	A5	C2	L2	P3	R2	-	-
7	100	0	1	37	A7	L1	P1	R1	-	-	-
8	81	1	0	89	A1	C1	P4	R3	-	-	-
9	82	0	1	57	A2	L1	P5	R1	-	-	-
10	58	0	1	58	A8	L2	P3	R3	-	-	-
11	44	0	1	41	A3	P2	L1	R1	-	-	-
12	10	1	1	14	A9	C1	L2	R2	-	-	-
13	95	0	1	69	A4	L1	P6	R2	-	-	-
14	41	0	1	38	A6	L2	R2	-	-	-	-
15	65	1	0	93	A10	P1	C3	R3	-	-	-
16	20	1	1	74	A5	C2	L1	R1	-	-	-
17	44	1	0	71	A11	C1	R1	-	-	-	-
18	37	1	0	64	A7	P1	C3	R3	-	-	-
19	54	0	1	36	A1	P3	L2	R2	-	-	-
20	95	1	1	96	A6	P2	C2	L1	R1	-	-

3.1.3. Uji Coba Tingkat Keberhasilan MAS dalam Menyelesaikan Tugas

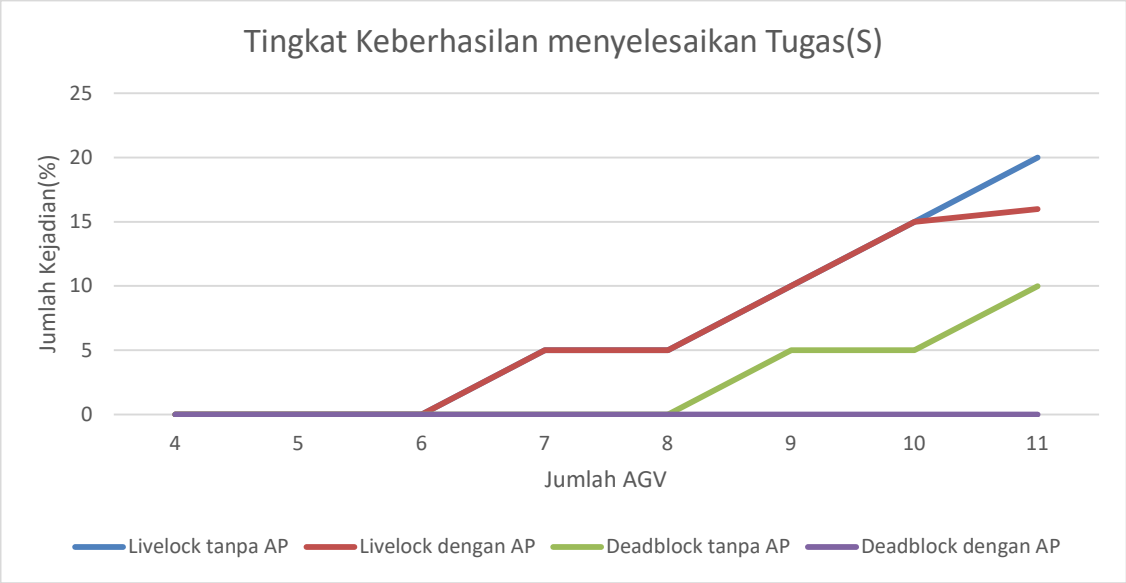
Pada sistem yang diajukan DDS memiliki ukuran sekecil mungkin, hal ini memungkinkan terjadinya *deadblock* pada saat beberapa agen AGV sedang bergerak dari satu *node* ke *node* lain karena AGV dapat bergerak bebas pada lingkungan DDS seperti yang ditunjukkan pada Gambar 6 (a) dengan kondisi A2,A3, dan A4 akan bertemu di titik yang ditandai dengan tanda X warna merah kemudian saling mengunci satu sama lain. Dengan adanya AP, AGV tersebut akan mengubah *state*-nya menuju *state* AP dengan mekanisme *timeout* sehingga terbebas dari *deadblock*, kemudian akan menjadi *Livelock* yang dapat dipecahkan oleh AP, Proses pemecahan ini ditampilkan pada Gambar 6 (b) hingga Gambar 6 (d). Pada Gambar 7 menunjukkan perbedaan jumlah *Deadlock* dan *Livelock* pada beberapa AGV saat bernavigasi dengan AP dan tanpa AP. Pada system dengan AP mampu mencegah *deadblock* dan memecahkan *Livelock* yang terjadi sehingga seluruh tugas dapat diselesaikan.



(a) **(b)**

(c) **(d)**

Gambar 6. *Deadblock* dan Pemecahannya



Gambar 7. Tingkat Keberhasilan Menyelesaikan Tugas

4. KESIMPULAN

Setelah melakukan pengujian dan analisis dari simulasi MAS untuk pelayanan *Drone* pada GDDS, maka diperoleh kesimpulan bahwa Komunikasi dan koordinasi antar agen dalam mendistribusikan tugas dapat dilakukan dengan baik oleh MAS yang dibuat dengan metode *auction* dan *contract*. Kontrak yang terjalin antar agen mampu mencegah terjadinya konflik sehingga sistem dapat menyelesaikan 100% *request* yang diterima. Algoritma prioritas yang diterapkan memberikan dampak positif bagi tingkat keberhasilan AGV dalam menyelesaikan tugas yang diterima DDS dengan mengurangi *Deadlock* yang terjadi secara signifikan.

Penelitian ini masih dapat dikembangkan lebih lanjut lagi, yaitu dengan menambahkan variasi jenis AGV, drone, dan bentuk DDS yang lain. Pengukuran energi juga dapat ditambahkan sebagai acuan performa MAS pada penerapan di dunia *real*.

UCAPAN TERIMA KASIH

Ucapan terimakasih kepada Kementerian Riset, Teknologi, dan Pendidikan Tinggi Republik Indonesia beserta Laboratorium Robotic and Intelligent Systems Centre (RoISC) dan Politeknik Elektronika Negeri Surabaya (PENS) atas dukungan berupa finansial dan non-finansial yang telah diberikan sehingga penelitian ini dapat terlaksana dengan baik.

DAFTAR RUJUKAN

Axak, N., Korablyov, M., & Ushakov, M. (2021). The Development of a Multi-Agent System for Controlling an Autonomous Robot. *CEUR Workshop Proceedings*, *3013*, (pp. 96–105).

Braquet, M., & Bakolas, E. (2021). Greedy Decentralized Auction-based Task Allocation for Multi-Agent Systems. *IFAC-PapersOnLine*, *54*(20), (pp. 675–680).

Gerrits, B., Mes, M., & Schuur, P. (2019). A simulation model for the planning and control of AGVs at automated container terminals. *Proceedings - Winter Simulation Conference*, *2018-Decem*(i), (pp. 2941–2952).

Kalisa Wilson, Ndatinya Eustache and Gakiza Canisius, 2016. Agent-based software architecture for decision making in the data warehouse*, International Journal of Current Research*, 8, (01), 25174-25178.

Kapitan, R., Veretilnyk, T., & Demyanenko, V. (2017). Development of a Multi-Agent Collision Resolution System At the Supply of Spare Parts and Components To the Production Equipment of Industrial Enterprises. *EUREKA: Physics and Engineering*, *6*(6), 27–34.

Karagoz, C. S., Bozma, H. I., & Koditschek, D. E. (2014). Coordinated navigation of multiple independent disk-shaped robots. *IEEE Trans. Robot.*, *30*(6), 1289–1304.

Kato, T., & Kamoshida, R. (2020). Multi-agent simulation environment for logistics warehouse design based on self-contained agents. *Applied Sciences (Switzerland)*, *10*(21), 1–20.

Liu, Y., Ji, S., Su, Z., & Guo, D. (2019). Multi-objective AGV scheduling in an automatic sorting system of an unmanned (intelligent) warehouse by using two adaptive genetic algorithms and a multi-adaptive genetic algorithm. *PLoS ONE*, *14*(12), 1–21.

Liu, W., Chen, D., & Guo, J. (2018). Goal-Capability-Commitment based Mediation for Multi-Agent Collaboration. *Proceedings of the 2018 IEEE 22nd International Conference on Computer Supported Cooperative Work in Design, CSCWD 2018*, (pp. 353–358).

Liyun, X., Ning, W., & Xufeng, L. (2021). Study on Conflict-free AGVs Path Planning Strategy for Workshop Material Distribution Systems. *Procedia CIRP*, (pp. 1071–1076).

Osmond, A. B., & Suhono Harso Supangkat. (2019). Platform dan Pemodelan Kerjasama Multi Agen untuk Layanan Pengiriman Barang. *Jurnal Sistem Cerdas*, *2*(1), 22–34.

Pamosoaji, A. K. (2019). Perencanaan Rute dan Kecepatan AGV pada Sistem Pergudangan Menggunakan Algoritma Ant Colony Optimization. *SAINTEK: Jurnal Ilmiah Sains Dan Teknologi Industri*, *3*(2), 52.

Pandian, D. A. P. (2019). Artificial Intelligence Application in Smart Warehousing Environment for Automated Logistics. *Journal of Artificial Intelligence and Capsule Networks*, *2019*(2), 63–72.

Rashidah Mohamad, N., Hafidz Fazli Md Fauadi, M., Azni Jafa, F., Zaki Mohamed Noor, A., & Hisham Nordin, M. (2018). Simulation-Based Multi-Objective Optimization for Distributed Material Transportation System. *International Journal of Engineering & Technology*, *7*(3.20), 92.

Ribas-Xirgo, L., & Chaile, I. F. (2013). Multi-agent-based controller architecture for AGV

systems. *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, (pp. 0–3).