

DISERTASI

**SISTEM KONTROL FORMASI BERDASARKAN
KOMBINASI METODA LOGIKA *FUZZY* TIPE-2 DAN
DINAMIK *PARTICLE SWARM OPTIMIZATION***



Oleh :
BAMBANG TUTUKO
03043681621003

**PROGRAM STUDI ILMU TEKNIK
PROGRAM DOKTOR FAKULTAS TEKNIK
UNIVERSITAS SRIWIJAYA
2019**

HALAMAN PENGESAHAN

SISTEM KONTROL FORMASI BERDASARKAN KOMBINASI METODA LOGIKA *FUZZY* TIPE-2 DAN DINAMIK *PARTICLE* *SWARM OPTIMIZATION*

LAPORAN AKHIR DISERTASI

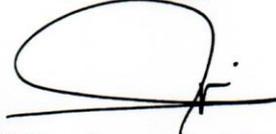
Diajukan Untuk Melengkapi Salah Satu Syarat
Memperoleh Gelar Doktor Bidang Ilmu Teknik Informatika
Fakultas Teknik Universitas Sriwijaya

Oleh :

BAMBANG TUTUKO
Nim : 03043681621003

Palembang, Agustus 2019

Promotor



Prof. Drs. Saparudin, MT. Ph.D

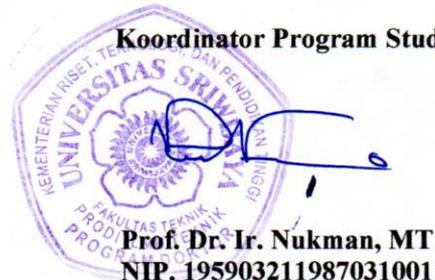
Mengetahui,

Dekan Fakultas Teknik



Prof. Ir. Subriyer Nasir, MS, Ph.D
NIP. 196009091987031004

Koordinator Program Studi



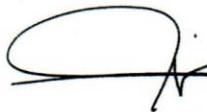
Prof. Dr. Ir. Nukman, MT
NIP. 195903211987031001

HALAMAN PERSETUJUAN

Karya Tulis Ilmiah berupa laporan akhir Disertasi dengan judul “ **SISTEM KONTROL FORMASI BERDASARKAN KOMBINASI METODA LOGIKA FUZZY TIPE-2 DAN DINAMIK PARTICLE SWARM OPTIMIZATION** “ telah dipertahankan di hadapan Tim Penguji Disertasi Program Doktor Bidang Ilmu Teknik Informatika Fakultas Teknik Universitas Sriwijaya. Palembang, 19 Juli 2019.

Promotor :

Prof. Drs. Saparudin, M.T, Ph.D

()

Ketua Tim Penguji :

1. Prof. Ir. Subriyer Nasir, MS, Ph.D
NIP. 196009091987031004

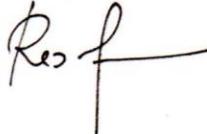
()

Anggota Tim Penguji :

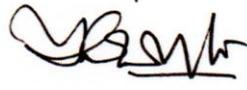
2. Prof. Dr. Ghazali bin Sulong (MSU)

()

3. Reza Firsandaya Malik, ST, MT, Ph.D
NIP. 197604252010121001

()

4. Dr. Yusuf Hartono, M.Sc
NIP. 196411161990031002

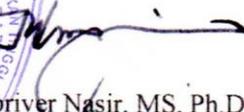
()

5. Prof. Dr. Ir. Nukman, MT
NIP. 195903211987031001

()

Mengetahui,

Dekan Fakultas Teknik
Universitas Sriwijaya

()
Prof. Ir. Subriyer Nasir, MS, Ph.D
NIP. 196009091987031004

Koordinator Program Studi

()
Prof. Dr. Ir. Nukman, MT
NIP. 195903211987031001

HALAMAN PERSETUJUAN

Karya Tulis Ilmiah berupa Laporan akhir Disertasi judul “ **SISTEM KONTROL FORMASI BERDASARKAN KOMBINASI METODA LOGIKA FUZZY TIPE-2 DAN DINAMIK PARTICLE SWARM OPTIMIZATION** “ telah dipertahankan di hadapan Tim Penguji Sidang Terbuka Promosi Doktor Bidang Ilmu Teknik Informatika Fakultas Teknik Universitas Sriwijaya. Palembang, 5 Agustus 2019.

Promotor :

Prof. Drs. Saparudin, M.T, Ph.D



Ketua Tim Penguji :

1. Prof. Dr. Ir. H. Anis Saggaff, MSCE
NIP. 196210281989031002

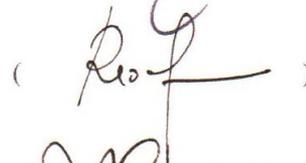
Anggota Tim Penguji :

2. Prof. Dr. Ghazali bin Sulong (MSU)



3. Prof. Ir. Subriyer Nasir, MS, Ph.D
NIP. 196009091987031004

4. Reza Firsandaya Malik, ST, MT, Ph.D
NIP. 197604252010121001



5. Dr. Yusuf Hartono, M.Sc
NIP. 196411161990031002



6. Prof. Dr. Ir. Nukman, MT
NIP. 195903211987031001

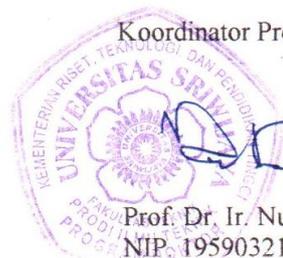
Mengetahui,

Dekan Fakultas Teknik
Universitas Sriwijaya



Prof. Dr. Ir. Subriyer Nasir, MS, Ph.D
NIP. 196009091987031004

Koordinator Program Studi



Prof. Dr. Ir. Nukman, MT
NIP. 195903211987031001

HALAMAN PERNYATAAN INTEGRITAS

Yang bertanda tangan dibawah ini :

Nama : Bambang Tutuko

NIM : 03043681621003

Tempat/Tanggal Lahir : Pekalongan, 12 Januari 1960

Jurusan/Program studi : Ilmu Teknik – Bidang Teknik Informatika

Fakultas : Teknik Universitas Sriwijaya

Alamat Rumah : Jl. Lunjuk Jaya Gg. Raflesia 31, Lorok Pakjo Ilir Barat I
Palembang

Email : tutukocn235@gmail.com

Dengan ini menyatakan dengan sesungguhnya bahwa disertasi yang berjudul
“ SISTEM KONTROL FORMASI BERDASARKAN KOMBINASI METODA
LOGIKA *FUZZY* TIPE-2 DAN DINAMIK *PARTICLE SWARM
OPTIMIZATION*”

Bebas dari plagiarisme dan bukan hasil karya orang lain

Apabila dikemudian hari ditemukan seluruh atau sebagian dari disertasi tersebut terdapat indikasi plagiarisme, saya bersedia menerima sanksi dengan peraturan perundang-undangan yang berlaku.

Demikian pernyataan ini saya buat dengan sesungguhnya tanpa ada paksaan dari siapapun dan untuk dipergunakan sebagaimana mustinya.



Palembang, Agustus 2019
Yang membuat pernyataan,

(Bambang Tutuko)
03043681621003

KATA PENGANTAR

Bismillahirrahmanirahim

Puji syukur dipanjatkan ke hadirat Allah SWT, atas limpahan rahmat dan hidayahnya dapat menyelesaikan penelitian Disertasi dengan judul “Sistem Kontrol Formasi Berdasarkan Kombinasi Metode Logika *Fuzzy* Tipe-2 dan Dinamik *Particle Swarm Optimization*”. Dalam kesempatan ini diucapkan terimakasih yang sedalam-dalamnya kepada Yth:

1. Ibu, Bapak yang tercinta
2. Prof. Drs. Saparudin, MT, PhD selaku Promotor
3. Prof. Dr. Ir. Anis Saggaff, MSCE, selaku Rektor Universitas Sriwijaya dan juga sebagai ketua tim sidang terbuka promosi Doktor.
4. Prof. Ir. Subriyer Nasir, MS, Ph.D, selaku Dekan Fakultas Teknik Universitas Sriwijaya dan juga sebagai ketua tim penguji komprehensif.
5. Prof. Dr. Ir. Nukman, MT, selaku Koordinator Program Studi Ilmu Teknik Program Doktor, Fakultas Teknik Universitas Sriwijaya
6. Prof. Dr. Ir. Kaprawi, DEA, Dipl. Ing. Ir. Amrifan Saladin Mohruni, Ph.D, terima kasih atas dukungannya
7. Teman-teman Jurusan Sistem Komputer Fasilkom Unsri, terima kasih atas dukungannya.
8. Pimpinan, staf, Adm Fasilkom Unsri terima kasih atas dukungannya
9. Istri dan anak-anaku, yang telah memberikan dorongan, pengertian dan pengorbanan.
10. Semua pihak yang telah memberikan bantuan hingga selesainya disertasi ini

Mudah-mudahan Allah SWT membalas semua amal kebajikan tersebut dan menjadikannya sebagai amal shaleh.

Salam Hormat,

Bambang Tutuko

RINGKASAN

SISTEM KONTROL FORMASI BERDASARKAN KOMBINASI METODE LOGIKA *FUZZY* TYPE-2 DAN DINAMIK *PARTICLE SWARM* *OPTIMIZATION*

Karya tulis ilmiah berupa Disertasi, 19 Juli 2019

Bambang Tutuko; dibimbing oleh Prof. Drs. Saparudin, MT, PhD.

Program Studi Doktor Ilmu Teknik, Fakultas Teknik, Universitas Sriwijaya

Sistem kontrol formasi merupakan isu penting pada aplikasi sistem robotik berkelompok (*swarm*), karena saat melakukan tugasnya sekelompok robot harus mengikuti lintasan yang telah ditentukan sambil mempertahankan pola spasial yang diinginkan. Untuk menghasilkan kinerja yang baik, model dan metode sistem kontrol yang sesuai harus ditentukan. Dalam hal robot berkelompok bersifat *leader-follower*, terdapat satu pemimpin dalam grup sementara robot lain mengikuti pemimpin dengan cara tertentu. Setiap robot harus memiliki kemampuan komunikasi terbatas dan tidak semua informasi global tersedia. Sehingga perancangan sistem kontrol untuk setiap robot harus didasarkan pada informasi lokal. Jika tidak ada pemimpin yang ditunjuk, maka semua robot harus berkoordinasi untuk pencapaian tujuan bersama.

Banyak metode sistem kontrol telah diajukan untuk memecahkan masalah dalam pengendalian formasi, namun, parameter stabilitas formasi, pengendalian pola formasi yang berbeda, dan ketidakpastian dalam formasi menjadi hal yang penting ketika sistem kontrol dirancang. Selain itu waktu pemrosesan yang cepat, beban komputasi yang terbatas dan mampu beradaptasi dengan perubahan lingkungan menjadi dasar yang penting lainnya. Untuk mengatasi hal tersebut, dalam penelitian ini akan dikembangkan algoritma baru dengan menggunakan Metode Sistem Logika *Fuzzy* Tipe-2 (SLFT2) dan Dinamik *Particle Swarm Optimization* (DPSO). Untuk proses validasi digunakan struktur robot pemimpin/pengikut (*leader-follower*). Dengan pertimbangan bahwa pendekatan tersebut memiliki kekurangan utama yaitu lemahnya stabilitas sistem, dikarenakan kegagalan pemimpin dalam mengeksplorasi lingkungan dapat menyebabkan kegagalan secara keseluruhan. Sehingga diperlukan perbaikan kinerja pengendalian agar kekurangan model robot *leader-follower* dapat diatasi.

Dari hasil pengujian dan validasi sistem kontrol, dengan metode yang diusulkan menghasilkan kinerja yang memuaskan dalam hal panjang rute tempuh, waktu pemrosesan, menghindari tabrakan dan pergerakan dalam formasi. Telah dilakukan perbandingan SLFT2-DPSO dengan metode yang telah ada, yaitu SLFT2- *Gregarious* PSO dan SLFT2-PSO, dengan hasil yang sangat memuaskan. Sistem kontrol yang diusulkan dalam melaksanakan tugasnya menghasilkan rute tempuh yang pendek, waktu pemrosesan yang cepat, pergerakan yang halus dan mampu menghindari tabrakan. Selain itu jika robot *leader* gagal dalam melaksanakan tugas, proses tetap dilanjutkan oleh robot *follower*. Sehingga dapat dinyatakan bahwa metode SLFT2-DPSO mampu memperbaiki kelemahan pada sistem robotika *leader-follower*.

Kata Kunci: Robot *Leader-Follower*, Kontrol Formasi, *Fuzzy* Tipe-2, PSO

SUMMARY

FORMATION CONTROL SYSTEM BASED ON COMBINATION OF TYPE-2 FUZZY LOGIC SYSTEM AND DYNAMIC PARTICLE SWARM OPTIMIZATION METHOD

Scientific Research in the Form of Dissertation, July, 19, 2019
Bambang Tutuko, Supervised by Prof. Drs. Saparudin, MT, PhD
Program Studi Doktor Ilmu Teknik, Fakultas Teknik, Universitas Sriwijaya

Formation control system an important issue in swarm robotic system applications, because when performing its tasks a group of robots must follow a predetermined path while maintaining the desired spatial pattern. To produce good performance, the appropriate model and control system method must be determined. In the case of leader-follower robots, there is one leader in the group while another robot follows the leader in a certain way. Every robot must have limited communication skills and not all global information is available. Hence, a control system design for each robot must be based on local information. If the leader failed or no leader in some situation, then all robots must coordinate to achieve the goals.

Many control system methods have been proposed to solve problems in formation control. However, formation stability parameters, different formation control patterns, and uncertainty in formations must be considered when the control systems designed. In addition, fast processing time, limited computational cost and capable to adapt to environmental changes are other important case. To overcome such problem, in this study a new algorithm will be developed using the Type-2 *Fuzzy* Logic System (SLFT2) and Dynamic Particle Swarm Optimization (DPSO) method. For the validation process, a robot leader/follower is utilized. Taking into account that the leader-follower approach has a major disadvantage, including the weakness of system stability. Because the failure of leaders to exploration the environment can cause overall failure condition. Therefore, the improvements in control performance are needed, to overcome the lack of leader-follower robot model.

As found in the results in the testing and validation process of the control system, the proposed method produces satisfactory performance in terms of optimization of travel routes, processing time, data generation, computational loads and being able to avoid collisions and maintain the robot movement in the formation. The comparison of SLFT2-DPSO method has been done with the existing methods, namely SLFT1-PSO, SLFT1-Gregarious PSO, SLFT2-PSO and SLFT2-Gregarious PSO. The result showed that the proposed SLFT2-DPSO method produce very satisfactory results. The proposed control system in carrying out its tasks produces short travel routes, fast processing times, smooth movements and able to avoid collisions. In addition, if the leader robot fails in carrying out the task, the process is continued by the follower robot. Therefore, it can be stated that the SLFT2-DPSO method is able to overcome the weakness in the leader-follower robotic system.

Keywords: Leader-Follower robot, Formation Control, Type-2 *Fuzzy* , PSO

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PENGESAHAN.....	ii
HALAMAN PERSETUJUAN.....	iii
HALAMAN PERNYATAAN INTEGRITAS	v
KATA PENGANTAR.....	vi
RINGKASAN	vii
SUMMARY	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL.....	xv
BAB I PENDAHULUAN.....	1
1.1. Umum.....	1
1.2. Latar Belakang Masalah.....	1
1.3. Rumusan Masalah	6
1.4. Tujuan Penelitian.....	7
1.5. Lingkup Penelitian	8
1.6. Urgensi Penelitian	9
1.7. Sistematika Penulisan Penelitian.....	10
BAB II STUDI PUSTAKA.....	11
2.1. Pendahuluan.....	11
2.2. Sistem Robotika Swarm.....	11
2.3. Sistem Kontrol Formasi	12
2.3.1 Sistem Kontrol Dengan Pendekatan Leader-Follower.....	13
2.3.2. Sistem Kontrol Pendekatan Berbasis Perilaku.....	14
2.3.3. Sistem Kontrol Dengan Pendekatan Berbasis Struktur Virtual.....	15
2.4. Teknik Komputasi Cerdas Pada Robot Swarm	17
2.5. Metode Sistem Kontrol Formasi	20
2.5.1. Sistem Logika Fuzzy Type-1.....	20
2.5.2. Sistem Logika Fuzzy Tipe-2	27

2.5.3. Particle Swarm Optimization.....	43
BAB III METODOLOGI PENELITIAN	50
3.1. Kerangka Penelitian.....	50
3.2. Pemodelan Sistem Robotik.....	53
3.2.1. Robotik Individu.....	53
3.2.2. Pemodelan Robot Leader-Follower.....	61
3.3. Pemodelan Lingkungan Pengujian	65
3.4. Pemodelan Robot Leader-follower.....	65
3.4.1. Pemodelan Sistem Robotik.....	65
3.4.2. Pemodelan Sistem Sensor Ultrasonik.....	66
3.5. Pemodelan Sistem Kontrol	68
3.5.1. Sistem Kontrol Pergerakan dengan Sistem Logika Fuzzy	68
3.5.2. Optimisasi Pencarian Target	76
3.5.3. Pemrosesan PSO dan Kondisi Berhenti.....	81
3.6. Sistem Kontrol Formasi Kombinasi Metode SLFT2-DPSO	84
3.7. Pengujian dan Validasi.....	86
3.7.1. Pengujian Sistem Pada Lingkungan Terstruktur.....	86
3.7.2. Pengujian Sistem Pada Lingkungan Tidak Terstruktur	87
3.8. Analisis Kinerja Sistem Kontrol Formasi	88
3.8.1. Trajectory.....	88
3.8.2. Beban Komputasi	88
3.8.3. Ketahanan	88
3.8.4. Waktu Pemrosesan Dalam Mencapai Target	89
BAB IV HASIL DAN PEMBAHASAN	90
4.1. Pendahuluan.....	90
4.2. Sistem Kontrol Pergerakan Robot	90
4.2.1. Metode SLFT1 dengan 2 Variabel Linguistik	90
4.2.2. Algoritma SLFT1 dengan 3 Variabel Linguistik	93
4.2.3. Algoritma SLFT2.....	97
4.3. Kinerja Sistem Kontrol Formasi Dalam Mengatur Pergerakan.....	99
4.3.1. Analisis Surfaces SLF	99
4.3.2. Pengontrolan Pergerakan Robot Leader-Follower	100

4.4. Sistem Kontrol Formasi Robot Leader-Follower Dalam Mencapai Target	113
.....	113
4.4.1. Analisis Surface DPSO	113
4.4.2. Kinerja DPSO Pada Sistem Pendeteksi Target.....	117
4.4.3. Kinerja Metode Hybrid SLFT2 dan DPSO	122
IV. 5. Ringkasan	128
BAB V KESIMPULAN DAN SARAN.....	130
5.1. Kesimpulan	130
5.2. Kontribusi Penelitian.....	131
5.3. Penelitian Selanjutnya	132
DAFTAR PUSTAKA	134
LAMPIRAN – 1	154
SURAT KETERANGAN PERBAIKAN UJIAN KOMPREHENSIF	
DISERTASI.....	154
LAMPIRAN – 2.....	160
SURAT KETERANGAN PERBAIKAN SIDANG TERBUKA PROMOSI	
DOKTOR	160

DAFTAR GAMBAR

Halaman

Gambar 2. 1. Sistem Formasi dengan Struktur Leader (L) dan Follower (F_n)	14
Gambar 2. 2. Perilaku kontrol dari robot individu	15
Gambar 2. 3. Pendekatan struktur virtual pada robot bergerak otonom	16
Gambar 2. 4. Sistem logika fuzzy tipe-1 (SLFT1)	20
Gambar 2. 5. Gabungan dari himpunan fuzzy	22
Gambar 2. 6. Irisan dari himpunan fuzzy	22
Gambar 2. 7. Komplemen dari himpunan fuzzy	23
Gambar 2. 8. Metode COG	26
Gambar 2. 9. Representasi himpunan fuzzy tipe-2 (Mendel, 2007)	28
Gambar 2. 10. (a) fungsi keanggotaan tipe-1 (b) Fungsi keanggotaan tipe-1 dengan ketidakpastian (c) fungsi keanggotaan sekunder	30
Gambar 2. 11. Footprint of uncertainty (FOU) (Mendel 2007)	31
Gambar 2. 12. Fungsi keanggotaan Tipe-2 Interval (Mendel 2007)	32
Gambar 2. 13. Teori irisan himpunan fuzzy tipe-2	33
Gambar 2. 14. Teori gabungan himpunan fuzzy tipe-2	34
Gambar 2. 15. Sistem Logika Fuzzy Tipe 2 (SLFT2)	35
Gambar 2. 16. Fungsi keanggotaan Gaussian	36
Gambar 2. 17. Sistem inferensi SLFT2 (Mendel 2007)	38
Gambar 3. 1. Kerangka Kerja Penelitian	52
Gambar 3. 2. Non-holonmik mobile robot	53
Gambar 3. 3. Perpindahan mobile robot	55
Gambar 3. 4. Kecepatan roda robot berdasarkan porosnya	56
Gambar 3. 5. Struktur kinematik dari perputaran roda pada bidang horizontal	58
Gambar 3. 6. Skema formasi robot leader- follower	62
Gambar 3. 7. Pseudo code lingkungan robot swarm	65
Gambar 3. 8. Pemodelan robot leader-follower	66
Gambar 3. 9. Posisi sensor ultrasonik dalam simulasi robot leader-follower	67
Gambar 3. 10. Model sensor dalam membaca jarak	67
Gambar 3. 11. Pseudo code sensor ultrasonik	68
Gambar 3. 12. Pseudo code fuzzifikasi	69

Gambar 3. 13 Pseudo Code fuzzifikasi f	69
Gambar 3. 14. Pseudo code kaidah dan inferensi f_{min}	70
Gambar 3. 15. Pseudo code kaidah dan inferensi f_{min}	71
Gambar 3. 16. Pseudo code Karnik-Mendel c_{kanan}	72
Gambar 3. 17. Pseudo code Karnik-Mendel c_{kiri}	72
Gambar 3. 18. Pseudo code keluaran kecepatan pada y_{kanan}	73
Gambar 3. 19. Pseudo Code keluaran kecepatan pada y_{kiri}	74
Gambar 3. 20. Pseudo Code keluaran perubahan sudut pada y_{kanan}	75
Gambar 3. 21. Pseudo code keluaran perubahan sudut pada y_{kiri}	76
Gambar 3. 22. Pseudo code dinamik PSO (DPSO)	84
Gambar 3. 23. Sistem kontrol formasi dengan metode SLFT2-DPSO	85
Gambar 4. 1. Fungsi keanggotaan masukan dengan 2 variabel linguistik	91
Gambar 4. 2. Fungsi keanggotaan keluaran dengan 2 variabel linguistik	91
Gambar 4. 3. Proses fuzzifikasi dengan 2 variabel linguistik	93
Gambar 4. 4. Hasil defuzzifikasi dengan 2 variabel linguistik	93
Gambar 4. 5. Fungsi keanggotaan masukan	94
Gambar 4. 6. Fungsi keanggotaan keluaran	94
Gambar 4. 7. Basis kaidah SLFT1	96
Gambar 4. 8. Hasil mekanisme mesin inference SLFT 1 dengan 3 variabel linguistik	96
Gambar 4. 9. Fungsi Keanggotaan Masukan	98
Gambar 4. 10 Fungsi Keanggotaan Keluaran	98
Gambar 4. 11. Proses fuzzifikasi algoritma SLFT2	99
Gambar 4. 12. Hasil defuzzifikasi algoritma SLFT 2	99
Gambar 4. 13. Surfaces SLFT1 dan SLFT2	100
Gambar 4. 14. Kinerja sistem logika fuzzy untuk pengontrolan gerak dalam formasi pada lingkungan sederhana	101
Gambar 4. 15. Respon waktu kecepatan menggunakan SLFT1 dan SLFT2 pada lingkungan sederhana	103
Gambar 4. 16. Respon waktu sudut kemudi robot menggunakan SLFT1 dan SLFT2 pada lingkungan sederhana	104

Gambar 4. 17. Kinerja sistem logika fuzzy untuk pengontrolan gerak dalam formasi pada lingkungan kompleks	105
Gambar 4. 18. Respon waktu kecepatan menggunakan SLFT1 dan SLFT2 pada lingkungan kompleks.....	106
Gambar 4. 19. Respon waktu sudut kemudi menggunakan SLFT1 dan SLFT2 pada lingkungan kompleks	108
Gambar 4. 20. Kinerja sistem kontrol formasi pada lingkungan tidak terstruktur	108
Gambar 4. 21. Kinerja posisi robot leader-follower dalam lingkungan kompleks 2 dengan SLFT1 8 kaidah, SLFT1 27 kaidah, dan SLFT2 8 kaidah.....	110
Gambar 4. 22. Kinerja orientasi robot leader-follower	111
Gambar 4. 23. Simulasi surface dari algoritma PSO.....	115
Gambar 4. 24. Kinerja robot leader-follower dalam mencapai target pada lingkungan sederhana dengan 3 halangan	117
Gambar 4. 25. Kinerja robot leader-follower pada lingkungan segi empat	118
Gambar 4. 26. Optimisasi rute dengan halangan bergerak.....	120
Gambar 4. 27. Optimisasi rute dengan halangan bergerak dan halangan tetap....	121
Gambar 4. 28. Kinerja sistem kontrol formasi untuk lingkungan U-Shape	123
Gambar 4. 29. Kinerja sistem kontrol formasi untuk lingkungan kompleks 1.....	124
Gambar 4. 30. Kinerja sistem kontrol formasi untuk lingkungan kompleks 2....	124
Gambar 4. 31. Sistem kontrol formasi untuk lingkungan berbentuk lingkaran ..	125
Gambar 4. 32. Sistem kontrol formasi untuk lingkungan berbentuk oval.....	125

DAFTAR TABEL

	Halaman
Tabel 2. 1. Perbandingan Sistem Robotika Swarm Dan Sistem Muti Agen Lain	12
Tabel 2. 2. Perbandingan Strategi Kontrol Formasi.....	16
Tabel 3. 1. Kombinasi masukan untuk tiap-tiap kaidah.....	70
Tabel 3. 2. Resume antara PSO standar dan Dinamik PSO	80
Tabel 4. 1. Linguistik fungsi masukan dengan 2 variabel.....	90
Tabel 4. 2. Linguistik fungsi keluaran SLFT 1 dengan 2 variabel.....	91
Tabel 4. 3. Basis kaidah SLFT1 dengan 2 variabel.....	92
Tabel 4. 4. Linguistik fungsi keanggotaan masukan dengan 3 variabel	93
Tabel 4. 5. Linguistik fungsi keanggotaan keluaran dengan 3 variabel.....	94
Tabel 4. 6. Basis kaidah dengan 3 variabel linguistik.....	95
Tabel 4. 7. Linguistik fungsi keanggotaan masukan.....	97
Tabel 4. 8. Linguistik fungsi keanggotaan keluaran	97
Tabel 4. 9. Perbandingan kinerja sistem kontrol formasi dengan	102
Tabel 4. 10. Perbandingan kinerja sistem kontrol formasi dalam melakukan pergerakan di berbagai lingkungan pengujian	112
Tabel 4. 11. Perbandingan hasil optimisasi berdasarkan jumlah iterasi	116
Tabel 4. 12. Kinerja PSO Dalam Mencapai Target Pada Sistem Kontrol Formasi	118
Tabel 4. 13. Titik koordinat awal dan akhir serta waktu yang ditempuh robot dalam mencapai target menggunakan PSO.....	119
Tabel 4. 14. Kinerja sistem kontrol pergerakan dan optimisasi rute pada halangan yang bergerak secara dinamis	122
Tabel 4. 15. Kinerja sistem kontrol pergerakan dan optimisasi rute pada halangan statis dan dinamis	122
Tabel 4. 16. Kinerja sistem kontrol hybrid untuk waktu tempuh.....	126
Tabel 4. 17. Kinerja sistem kontrol hybrid untuk beban komputasi	127

BAB I

PENDAHULUAN

1.1. Umum

Robot bergerak otonom merupakan sebuah robot individu yang mampu bergerak secara otomatis, dengan suatu sistem penginderaan, dan sistem penggerak agar dapat bereaksi dalam suatu lingkungan tertentu serta mampu melakukan tugas-tugas tanpa campur tangan manusia. Dalam kehidupan nyata robot ini telah diaplikasikan ke berbagai bidang, seperti di bidang pertanian (Pedersen et al., 2006), pelayanan medis (Chivarov et al., 2012), pertambangan/penggalan (Carpin, and Pagello, 2009), keamanan (Di Paola et al., 2010), militer (Mies, 2010), dan industri (Danget et al., 2013). Agar dapat berfungsi secara maksimal, sebuah robot bergerak otonom harus dapat menavigasi dengan aman melalui lingkungan yang tidak diketahui sekaligus menjalankan tugas yang telah dibebankan kepadanya. Agar tugas tersebut dapat dilaksanakan dengan baik maka sebuah robot harus memiliki beberapa informasi yang relevan tentang lingkungan tempatnya bergerak, sayangnya informasi yang ada sebagian besar tidak diketahui secara apriori di sebagian besar aplikasi. Berdasarkan hal tersebut, robot harus dapat membangun model atau peta lingkungan yang tidak diketahui sebelumnya. Model lingkungan yang dirancang, digunakan untuk memperkirakan posisi dan orientasi saat ini, dan untuk menavigasi ke titik target berdasarkan kinerja yang telah ditetapkan.

1.2. Latar Belakang Masalah

Dalam melaksanakan berbagai tugas pada lingkungannya, sebuah robot harus dilengkapi dengan sistem persepsi, sistem kontrol, dan sistem penggerak. Khususnya sistem persepsi beragam sistem sensorik yang digunakan untuk mengukur beberapa informasi eksternal dari lingkungan, antara lain sensor GPS, Ultrasonik, Infra-merah, Sonar, dan Laser, (Zhang, and Ghosh, 2000; Wijk, and Christensen, 2000; Tardos et al., 2002; Yu, and Zhang, 2006). Namun, semua sistem sensor tersebut cenderung menghadirkan ketidakpastian sudut yang tinggi,

derau pengukuran, mahal, dan konsumsi daya peralatan yang tinggi (Paya et al. 2017). Sebagai alternatif dari sistem persepsi lainnya yaitu dengan menggunakan sensor kamera (Gasparet al., 2000; Zhu et al., 2006; Bradley et al., 2005). Penggunaan sensor kamera pada sistem robotika memiliki banyak keunggulan antara lain, memberikan informasi lingkungan dalam jumlah besar, dalam bentuk data 3D yang bisa diekstrak darinya, berharga relatif murah dan konsumsi daya yang relatif rendah (Bradley et al., 2005; Paya et al. 2017). Sayangnya proses pengolahan citra dari ketersediaan gambar merupakan suatu penambahan tugas yang sulit selain dari pemetaan dan lokalisasi (Gasparet al., 2000; Paya et al. 2017).

Selain pemilihan teknik yang sesuai pada sistem persepsi, sebuah robot juga harus memiliki sistem kontrol untuk mencapai kinerja yang baik pada semua kondisi lingkungan (Benavidez, Jamshidi, and Chair, 2011). Berbagai sistem kontrol konvensional telah digunakan untuk pengontrolan gerak robot antara lain, *path-planning* dan *potential field* yang menghasilkan kinerja yang baik (Berg, et al., 2006; Hernández-Martínez and Aranda-Bricaire, 2011; Benbouabdallah and Qi, 2013). Namun, dalam aplikasinya sistem kontrol konvensional bekerja hanya mengandalkan sistem persepsi sehingga membutuhkan biaya yang besar dalam menyelesaikan tugas yang kompleks karena memerlukan kombinasi sistem sensor yang banyak (Nurmaini and Primanita, 2012; Hoffmann, 2001). Selain itu model pemetaan lingkungan harus diturunkan, untuk mencapai hasil yang maksimal. Sehingga sebuah sistem persepsi dan sistem kontrol yang dipilih akan sangat menentukan kinerja pergerakan sebuah robot bergerak otonom tersebut.

Saat ini, telah banyak dikembangkan metode sistem kontrol menggunakan metode komputasi lunak (*soft computing*) (Gu and Hu, 2002; Wagner and Hagra, 2007; Park, et al, 2009; Nurmaini and Tutuko, 2011; Juang et al., 2016; Rao et al., 2017). Dibandingkan dengan teknik konvensional, teknik ini mampu bekerja tanpa perlu menurunkan persamaan analitis atau model matematika yang kompleks (Hagra et al., 2004; Nurmaini, and Primanita, 2012; Penget al., 2013). Pada teknik komputasi lunak pendekatan yang paling populer dan menjanjikan dalam menyelesaikan masalah sistem kontrol pergerakan robot adalah jaringan saraf tiruan (Gu and Hu, 2002; Park, et al, 2009), algoritma evolusi (Hoffmann,

2001; Hagraş et al., 2004; Wagner and Hagraş, 2007), sistem logika *fuzzy* (Nurmaini, and Primanita, 2012; Castillo et al., 2016; Benigni et al., 2017), algoritma *bio-inspired* (Penget et al., 2013; Ni et al., 2016), serta dengan kombinasi beberapa metode kecerdasan buatan (Hagraş et al., 2004; Nurmaini and Tutuko, 2011; Juang et al., 2016; Rao et al., 2017). Sayangnya metode tersebut memiliki beberapa kelemahan yaitu: (i) perubahan lingkungan yang tidak menentu dapat menyebabkan kegagalan dalam proses pengendalian, diakibatkan oleh ketidakpastian, ketidakakuratan dan ketidaktelitian (Hagraş, 2004); (ii) kinerja sistem kontrol hanya optimal sebagian (Abdessemed, et al., 2004); (iii) sistem kontrol yang dipilih hanya cocok pada satu kondisi lingkungan, jika kondisi tersebut berubah maka sistem akan gagal (Abdessemed, et al., 2004).

Berdasarkan hal tersebut di atas, perlunya suatu pemilihan strategi yang tepat dan metode yang sesuai dalam mempertimbangkan struktur sebuah sistem kontrol yang akan dipilih. Apalagi jika tugas yang diberikan harus diimplementasikan pada suatu lingkungan yang luas dan membutuhkan banyak perangkat pendeteksi. Selain itu permasalahan lainnya pengontrolan pergerakan robot harus memperhatikan berbagai keterbatasan platform tertanam (Faigl et al., 2013), penginderaan secara *onboard* (Dai and Lee, 2014), beban komputasi yang kecil (Yogeswaran and Ponnambalam, 2009; Pinciroli et al., 2012) dengan hasil yang maksimal dan dapat mengatasi ketidakpastian dan ketidaktelitian.

Saat ini telah diusulkan penggunaan banyak robot dalam menyelesaikan masalah kompleks di area lingkungan yang luas untuk mempermudah pekerjaan, dan menyelesaikan tugas yang diberikan kepadanya (Mohan, and Ponnambalam, 2009; Sharkey, 2007; Brambilla et al., 2013). Sayangnya pencapaian kinerja yang baik dapat dengan mudah diselesaikan oleh robot individu, tetapi menjadi masalah pada sistem multi-robot (Benavidez et al., 2011). Hal tersebut dikarenakan sistem multi-robot memiliki jumlah robot yang banyak, harus saling berkomunikasi dan berkoordinasi dalam menyelesaikan tugasnya (Brambilla et al., 2013; Mohan and Ponnambalam, 2009). Padahal dalam sistem multi-robot, masing-masing robot juga harus berperilaku sebagai diri sendiri dan perilaku dalam kelompok (Sharkey, 2007).

Salah satu jenis sistem multi-robot adalah robot *swarm* (berkelompok). Robot *swarm* merupakan robot dalam jumlah banyak dengan kesamaan fisik (Mohan & Ponnambalam, 2009), dengan dua roda terkendali dan satu roda bebas yang bergerak dengan kecepatan sama (Abdalla and Hamzah, 2013). Robot *swarm* menggunakan pendekatan koordinasi sejumlah robot saat melakukan berinteraksi. Mereka dapat bekerja sama untuk menyelesaikan tugas, khususnya dengan tugas diluar kemampuan robot individu (Kumra et al., 2009; Brambilla et al., 2013; Mohan and Ponnambalam, 2009). Hal tersebut dapat meningkatkan kinerja dengan fleksibilitas yang tinggi dan mempunyai ketahanan sistem secara keseluruhan (Sharkey, 2006; Beni, 2004, Sahin et al., 2007). Namun, kinerja dapat menurun pada aplikasi terdistribusi dalam lingkungan yang besar dan beragam seperti pencarian dan penyelamatan, operasi pengawasan, pencarian dan eksplorasi, pengawasan, survei, dan pemetaan (Bayindir, 2015).

Tantangan utama bagi perancangan pengontrolan robot *swarm* yaitu bagaimana membuat robot cerdas yang mampu beradaptasi berdasarkan interaksi dengan lingkungan dan robot lain (Nurmaini and Tutuko, 2014; Sahin et al., 2002). Namun, mempertahankan formasi dengan menghindari rintangan dan tabrakan antar robot dalam lingkungan yang dinamik dan tidak diketahui merupakan hal yang sulit dicapai oleh robot *swarm* (Tran and Lee, 2011; Olfati Sabber and Murray, 2002), karena diakibatkan oleh ketidakpastian dan ketidakakuratan dalam setiap proses pengontrolan.

Terdapat beberapa isu penting dalam pengontrolan formasi antara lain *formation acquisition* (kemampuan untuk mencapai bentuk geometris yang telah ditetapkan dari setiap posisi awal dan orientasi anggota *swarm*), *formation maintenance* (menjaga formasi selama bergerak dalam kelompok), dan *formation reconfiguration* (kemampuan untuk berpisah atau bergabung atau mengubah bentuk geometri selama beroperasi di lingkungan berbagai kondisi lingkungan) (Desai et al., 1998; Balch and Arkin, 1998; Brunete et al., 2012; Gazi et al., 2014). Sehingga dibutuhkan pendekatan yang dapat menangani situasi yang tidak menentu, ketidakpastian dan ketidakakuratan, tetapi sifat ketahanan harus tetap diutamakan dalam prosedur kontrol.

Logika *fuzzy* telah banyak digunakan untuk menyelesaikan persoalan ketidakpastian, ketidakakuratan dan ketidakteelitian dalam sistem robotika berkelompok (Mendel, 2001; El-Desouky and Hagrass, 2009). Namun sistem logika *fuzzy* type-1 (SLFT1) kurang menghasilkan respon yang baik pada kondisi lingkungan yang tidak terstruktur dan dinamik (Mendel, 2001; Nurmaini and Primanita, 2012). Pengembangan SLFT1 menjadi sistem logika *fuzzy* tipe-2 (SLFT2) meningkatkan kemampuan sistem kontrol pada robot *swarm* dalam hal pengontrolan formasi (Lei and Li, 2007; Hagrass, 2012; Karimi, 2012). Hal itu dikarenakan adanya perbaikan kinerja pengendali dengan memasukkan interval ketidakpastian di dalam fungsi keanggotaan *fuzzy*, baik masukan maupun keluaran sistem logika *fuzzy*. Sayangnya komputasi menjadi meningkat dan robot *swarm* belum memiliki sistem pengontrol pencapaian target atau optimisasi rute. Sehingga diperlukan metode *hybrid* untuk mengatasinya.

Sistem kontrol pada robot *swarm*, harus memperhitungkan waktu yang singkat, proses yang cepat, algoritma yang sederhana, tetapi tetap mampu menjaga formasi dan dapat menghindari tabrakan pada saat yang bersamaan (Chen and Wang, 2005; Gazi et al., 2014; Mohan and Ponnambalam, 2009). Banyak metode yang telah digunakan untuk memenuhi parameter tersebut, antara lain metode *path planning*, *artificial potential function* and *potensial field* (Barnes, 2007; Benbouabdallah and Qi, 2013; Kowdiki et al., 2012). Sayangnya metode ini sangat mengandalkan peta lingkungan yang dibangun, jika kondisi lingkungan berubah, maka robot akan gagal. Metode lainnya adalah penggunaan kecerdasan *swarm*, yang menunjukkan hasil yang memuaskan, karena menggunakan karakteristik hewan di alam dalam mencari makanan, antara lain *Particle Swarm Optimization* (PSO) (Pimenta et al., 2013; Rigatos, 2008; Pugh and Martinoli, 2007). Sayangnya banyak penerapan metode PSO untuk optimisasi pencarian rute target, sering kali menghasilkan kinerja optimal sebagian, dan dapat menurunkan kinerja pengontrolan formasi. Sehingga perlunya suatu metode PSO yang dapat bereaksi secara dinamik dalam kondisi yang tidak menentu.

Dari berbagai latar belakang masalah yang telah dibahas, maka dalam penelitian ini yang di usulkan adalah bagaimana meningkatkan kinerja sistem kontrol formasi robot *leader-follower* agar menghasilkan kinerja yang baik dalam

hal pergerakan menghindari halangan dan optimisasi rute. Keefektifan kombinasi metode sistem Logika *Fuzzy* Tipe-2 dan Dinamik *Particle Swarm Optimization* akan digunakan untuk mengatasi masalah dari sistem kontrol individual yang sering menjadi tawar menawar (*trade-off*) antara berbagai parameter. Diharapkan dengan strategi dan mekanisme terbaru yang diusulkan tersebut dapat memberikan kontribusi bagi penelitian sistem kontrol robotika pada kasus pengontrolan formasi pada robot *swarm*.

1.3. Rumusan Masalah

Untuk memecahkan tugas-tugas dalam aplikasi terdistribusi, kinerja robot *swarm* tergantung pada kemampuan individu robot dan struktur sistem kontrol (Sahin et al., 2002; Gazi, 2014), sistem komunikasi dan mekanisme koordinasi dengan robot lain untuk berinteraksi dengan lingkungan secara kolektif (Mohan and Ponnambalam, 2009). Kemampuan kolektif adalah kompetensi dalam memperoleh informasi dari lingkungan dan memproses informasi agar dapat bergerak di dalam lingkungan dengan cara berinteraksi dengan robot lain. Sehingga semua robot bergerak dalam arah dan tujuan yang telah ditetapkan, memiliki kemampuan menghindari tabrakan antar robot dan menghindari rintangan yang ada dalam lingkungan (Balch and Arkin, 1998; De La Cruz, and Carelli, 2008). Semua hal tersebut merupakan perilaku robot berkelompok (*swarm*) dalam melakukan aksinya.

Robot *swarm* yang beroperasi di lingkungan yang tidak diketahui, harus mampu menangani perubahan dinamis dari lingkungan. Sistem kontrol yang dimiliki harus dapat berfungsi secara individual untuk mengatur pergerakan, dan harus berfungsi dalam kelompok untuk mengatur formasi pergerakan (Benigni et al., 2017; Garro et al., 2007; Ghommam et al., 2011). Perancangan sistem kontrol formasi dari robot *swarm* dalam menangani ketidakpastian pada lingkungan adalah salah satu masalah yang paling menantang. Permasalahan yang sering terjadi adalah bagaimana mengontrol pergerakan formasi dari titik awal ke titik sasaran dalam suatu lingkungan tidak diketahui, dengan kemampuan menghindari rintangan dalam waktu yang singkat dan komputasi yang kecil.

Sehubungan dengan berbagai permasalahan yang ada dalam pengembangan sistem kontrol formasi, penelitian ini mengembangkan suatu metode kombinasi antara sistem kontrol formasi dan optimisasi rute pencapaian target. Sistem kontrol yang dirancang berhubungan dengan lingkungan yang dilalui untuk pengontrolan posisi dan orientasi robot menuju target yang telah ditentukan. Selain itu perlunya penyelesaian masalah ketidakpastian, ketidaktepatan dan ketidaktepatan karena perubahan lingkungan yang dinamik. Hasil yang diharapkan sistem akan memiliki kinerja yang lebih baik pada berbagai lingkungan pengujian dengan berbagai batasan dalam perancangan robot *swarm*.

Sehingga pertanyaan utama dari penelitian ini adalah:

Bagaimana merancang sistem kontrol formasi dengan strategi kontrol yang sederhana dan memiliki kinerja yang baik pada robot leader-follower dalam berbagai kondisi lingkungan?

Dengan demikian, isu-isu yang harus dipertimbangkan untuk memecahkan masalah tersebut adalah:

- a) Bagaimana memilih sistem kontrol yang tepat dan efisien untuk aplikasi sistem terdistribusi?
- b) Bagaimana mengembangkan strategi kontrol untuk mengontrol formasi robot *swarm* dengan kemampuan menghindari rintangan, mencapai target dan membentuk formasi dengan kinerja yang memuaskan?
- c) Bagaimana membuat perangkat lunak sistem kontrol *hybrid* berbasis teknik logika *fuzzy* tipe-2 dan *particle swarm optimization*?
- d) Bagaimana menguji, memvalidasi dan menganalisis hasil simulasi dari sistem kontrol yang dipilih pada berbagai kondisi lingkungan yang tidak diketahui?

1.4. Tujuan Penelitian

Tujuan utama dari penelitian ini adalah:

- a) Mengembangkan strategi sistem kontrol formasi pada robot *swarm* untuk meningkatkan kinerja dalam mempertahankan formasi serta dapat

menghindari tabrakan pada saat yang bersamaan;

- b) Merancang algoritma pengontrolan pergerakan robot *swarm* menggunakan metode sistem logika *fuzzy* tipe-2;
- c) Merancang algoritma optimisasi rute pencapaian target menggunakan metode dinamik *particle swarm optimization*;
- d) Merancang algoritma sistem kontrol formasi pada robot *swarm* [1] dengan metode *hybrid* sistem logika *fuzzy* tipe-2 (SLFT2), dan dinamik *particle swarm optimization* (DPSO);
- e) Melakukan validasi pendekatan, dengan beberapa teknik dengan metode kecerdasan buatan terhadap kinerja pergerakan, ketahanan terhadap gangguan perubahan lingkungan, waktu pemrosesan dan beban komputasi.

1.5. Lingkup Penelitian

Berikut adalah lingkup penelitian yang telah diidentifikasi:

1. Jenis robot berkelompok yang akan digunakan dalam penelitian ini adalah jenis *differential drive*;
2. Model robot *swarm* yang dipilih adalah jenis *leader-follower*;
3. Strategi kontrol formasi adalah berbasis perilaku 1 robot *leader* dan 1 robot *follower* yang meliputi, menghindari halangan, menjaga formasi kelompok dan mencapai target;
4. Validasi sistem kontrol formasi yang diusulkan hanya melalui simulasi;
5. Metode yang akan dibandingkan pada pengontrolan gerak adalah Sistem Logika *Fuzzy* Tipe-1;
6. Metode yang dibandingkan pada optimisasi rute adalah *Gregarious* PSO (GPSO) dan PSO;
7. Perbandingan analisis berdasarkan kinerja pergerakan, waktu tempuh, panjang rute tempuh, ketahanan terhadap perubahan lingkungan dan beban komputasi;

8. Pengujian dengan membuat kondisi lingkungan yang selalu berubah yaitu, sederhana, kompleks dan tidak terstruktur.

1.6. Urgensi Penelitian

Penggunaan pendekatan kecerdasan buatan dengan teknik komputasi lunak dalam masalah perancangan sistem kontrol robotika untuk sistem multi-robot sangat diperlukan untuk diteliti. Hal tersebut disebabkan dalam melakukan tugasnya sebuah robot akan menemukan ketidakpastian yang disebabkan oleh perubahan lingkungan, ketidakakuratan yang disebabkan oleh gangguan pada sistem pendeteksi dan ketidaktelitian pergerakan *aktuator* akibat kesalahan sistem persepsi. Jika pendekatan konvensional diterapkan maka akan membutuhkan peta lingkungan saat melakukan lokalisasi daerah operasi. Jika terjadi perubahan kondisi operasi lingkungan, pemetaan yang telah dilakukan tidak dapat digunakan dan robot dapat mengalami '*dead lock*' atau diam tanpa aksi.

Khususnya sistem kontrol formasi untuk robot *swarm*, pada lingkungan yang tidak diketahui, dengan situasi yang tidak menentu pendekatan kecerdasan buatan menunjukkan kinerja yang baik (Benigni et al., 2017; Garro et al., 2007; Ghommam et al., 2011). Selain itu pendekatan ini lebih sering digunakan dibandingkan metode konvensional, karena pendekatan analitis atau pemodelan matematika akan dihindari karena membuat sistem meningkat kompleksitasnya dan menghasilkan beban komputasi yang besar (De La Cruz, and Carelli, 2008). Mendeteksi objek di lingkungan yang tidak diketahui dan membentuk pola formasi mengelilingi target dengan bentuk yang kompleks merupakan tantangan lainnya (Ghommam et al., 201; Benigni et al., 2017). Sehingga interpretasi bentuk lingkungan yang tidak terstruktur sangat perlu untuk diperhatikan dalam mengenali objek lingkungan.

Selain itu struktur kontrol formasi dari robot *swarm* dalam menangani ketidakpastian lingkungan, ketidakakuratan sistem persepsi, dan ketidaktelitian sistem penggerak pada lingkungan adalah salah satu masalah yang paling menantang dalam navigasi robot *swarm*. Permasalahan yang sering terjadi adalah bagaimana mengontrol pergerakan formasi dari titik awal ke titik sasaran dalam suatu lingkungan tidak diketahui, dengan kemampuan menghindari rintangan

dalam waktu yang singkat, komputasi yang kecil dan kinerja pergerakan yang baik. Oleh karena itu penelitian akan sangat penting untuk dilakukan dalam memberikan kontribusi keilmuan dan kebaharuan dari metode yang digunakan pada sistem kontrol robotika.

1.7. Sistematika Penulisan Penelitian

Penelitian dengan topik merancang kontrol formasi dengan metode *hybrid* logika *fuzzy* tipe-2 dan dinamik *particle swarm optimization*, terdiri dari 5 bab. Pada Bab 1 berisi tentang latar belakang masalah mengapa perlunya penelitian ini dilakukan berdasarkan penelitian-penelitian terdahulu. Pada Bab 2 berisi tinjauan pustaka dari penelitian terkait bidang sistem robotika *swarm*, strategi kontrol, metode kontrol formasi dan berbagai aplikasi yang telah mengimplementasikan metode-metode telah diusulkan sebelumnya. Bab 3 berisi metodologi riset yang digunakan untuk menyelesaikan berbagai masalah yang ada, melalui tahapan-tahapan penelitian yang dirancang secara terstruktur. Bab 4 menjelaskan tentang hasil penelitian, pembahasan hasil penelitian, dan hasil yang diharapkan dari penelitian. Bab 5 menjelaskan tentang kesimpulan, kontribusi penelitian dan saran untuk penelitian lanjutan.

BAB II

STUDI PUSTAKA

2.1. Pendahuluan

Sistem kontrol formasi pada robot *swarm* memastikan proses berkoordinasi untuk bergerak mencapai target yang telah ditentukan dan berinteraksi dengan robot lain di lingkungan di sekitarnya tanpa mengalami tabrakan. Pada bab ini berisi gambaran singkat tentang permasalahan yang ada tentang: Bagaimana merancang pengendali pergerakan robot berkelompok dalam struktur *leader-follower* di lingkungan yang berubah secara dinamis dan tidak diketahui.

2.2. Sistem Robotika *Swarm*

Istilah “Robotika *Swarm*” mengacu pada sekelompok besar robot yang relatif sederhana serta berinteraksi dan bekerja sama untuk menyelesaikan tugas-tugas yang berada di luar kemampuan masing-masing robot individu (Sahin, 2004; Kumra et al., 2009; Brambilla et al., 2013; Mohan and Ponnambalam, 2009). Dalam penyelesaian tugas-tugas yang diberikan, kinerja tergantung dari koordinasi antar robot yang sederhana bukan pada perilaku individu yang kompleks dengan kemampuan sensorik yang kuat (Barca dan Sekercioglu, 2012).

Perilaku kecerdasan yang dimiliki oleh sistem multi-robot yang berinteraksi satu sama lain dalam kawanan terinspirasi dari kehidupan koloni hewan, seperti kawanan biologi serangga, semut dan ikan atau mamalia (Brambilla et al., 2013; Beni, 2014). Walaupun kemampuan yang dimiliki individu sangat sederhana, namun perilaku kelompok yang kompleks muncul dalam menyelesaikan tugas secara kolektif. Banyak peneliti terdahulu telah mengamati perilaku kelompok cerdas yang muncul dari sekelompok hewan/serangga sosial dengan kemampuan yang terbatas melalui komunikasi lokal dan transmisi informasi (Tan, and Zheng, 2013). Penggunaan berbagai sifat alami dari koloni hewan saat melakukan kerja sama dianalogikan sebagai penyelesaian tugas-tugas secara kolektif pada aplikasi sistem robotika *swarm*, telah menghasilkan kinerja yang baik dan lebih efektif (Brambilla. et al, 2013). Beberapa keuntungan lainnya adalah sulitnya untuk

memprediksi apa yang akan terjadi, menyebabkan robot tunggal rentan terhadap bagian kecil yang rusak, dan dapat mempengaruhi seluruh sistem (Barca and Sekercioglu, 2012). Selain itu, sistem robotika *swarm* memiliki sifat paralelisme tinggi dan sangat cocok untuk tugas-tugas skala besar (Tan and Zheng, 2013).

Kemampuan dan efisiensi sekelompok robot tergantung pada fungsi robot individu dan struktur kontrolnya (Mondada et al., 2004). Beberapa perilaku kooperatif muncul dari inspirasi alam dengan menyelesaikan tugas secara berkelompok berlaku untuk sistem multi-agen. Sistem ini melakukan proses koordinasi dan komunikasi dalam menyelesaikan masalah yang diberikan. Pada Tabel 2.1. Menampilkan beberapa karakteristik penting sistem multi-agen antara lain; robot *swarm*, multi-robot, *sensor network* dan multi-agen lainnya.

Tabel 2. 1. Perbandingan Sistem Robotika *Swarm* Dan Sistem Muti Agen Lain

	Robot <i>Swarm</i>	Multi Robot	<i>Sensor Network</i>	Multi Agen
kontrol	desentralisasi, otonom	sentralisasi	sentralisasi	sentralisasi, hierarki
populasi	besar	kecil	tetap	kecil
pergerakan	bergerak	bergerak	tetap	tetap, bergerak
fleksibel	tinggi	rendah	rendah	sedang
skalabiliti	tinggi	rendah	sedang	sedang
lingkungan	tidak diketahui	diketahui, tidak diketahui	diketahui	diketahui
aplikasi	pemantauan daerah berbahaya, militer	transportasi, robot sepak bola	perawatan medis, pengawasan	pengolahan sumber daya jaringan, kontrol distribusi

2.3. Sistem Kontrol Formasi

Sistem kontrol formasi merupakan bagian yang penting dalam aplikasi multi-robot atau robot *swarm*. Fungsi ini memungkinkan untuk mengatur seluruh kelompok robot tanpa menentukan secara langsung perintah untuk masing-masing robot. Sedangkan fungsi lainnya adalah mengkoordinasikan sekelompok robot

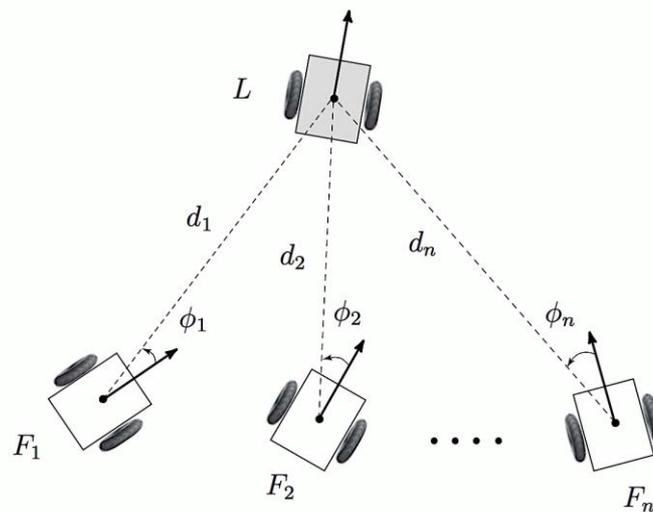
agar bergerak dengan formasi yang ditetapkan serta tetap menjaga pola formasi dalam mencapai tujuan. Masalah sistem kontrol formasi dapat dibagi menjadi berbagai tahap yaitu (i) akuisisi formasi (berkaitan dengan pencapaian bentuk geometris yang telah ditetapkan dari setiap posisi awal dan orientasi anggota *swarm*); (ii) pemeliharaan formasi (berkaitan dengan menjaga pembentukan formasi selama gerakan), dan (iii) konfigurasi ulang formasi (berkaitan dengan berbagai perilaku dalam formasi seperti berpisah, bergabung atau mengubah geometri membentuk formasi) (Gazi et al, 2014).

Untuk memastikan kinerja sistem kontrol dalam hal fleksibilitas tinggi, toleransi kesalahan, dan kemampuan beradaptasi pada sistem robot *swarm*, tiga bentuk pendekatan strategis sistem kontrol formasi telah diusulkan yaitu pendekatan berbasis *leader-follower*, berbasis perilaku dan berbasis struktur virtual (Sakurama et al., 2008; Soorki et al., 2011; Lewis and Tan, 1997; Antonelli et al., 2010; Beard et al., 2001; Lewis and Tan, 1997; Ren and Randal, 2004; Mataric, 1983; Balchand Arkin, 1998; Marino et.al, 2012; Das et al., 2002; Consolini et al., 2008; Cui et al., 2010; Egerstedt and Hu, 2001; Ren and Beard, 2004). Semua strategis tersebut akan dijelaskan pada sub bab selanjutnya.

2.3.1 Sistem Kontrol Dengan Pendekatan *Leader-Follower*

Pada pendekatan robot pemimpin-pengikut (*leader-follower*) seperti terlihat pada Gambar 2.1, terdapat hierarki antara robot pemimpin dan robot pengikut. Pendekatan tersebut membentuk struktur kontrol melalui perintah dari robot pemimpin, sedangkan robot pengikut melaksanakan perintah yang dihasilkan oleh robot pemimpin (Sakurama et al., 2008; Soorki et al., 2011). Pada lingkungan yang kompleks, seperti kondisi lingkungan yang tidak pasti dan tidak diketahui merupakan suatu gangguan. Sehingga sistem kontrol individu dalam pendekatan pemimpin-pengikut dapat menjamin stabilitas formasi. Namun kelemahan utama adalah apabila terjadi kegagalan pada robot pemimpin menyebabkan robot pengikut ikut gagal dan menunjukkan respon yang buruk karena kesalahan tersebut (Soorki et al., 2011; Peng et al., 2013).

Penelitian terdahulu dengan pendekatan *leader-follower* konvensional dengan skema *two leaders – single follower* menghasilkan kinerja yang baik (Tran and Lee, 2011) yaitu sekelompok robot bergerak sepanjang lintasan yang diinginkan dengan tetap mempertahankan formasi yang diinginkan, dalam hal apa pun dan kapan saja. Dari hasil simulasi menunjukkan bahwa stabilitas algoritma kontrol dapat dicapai dengan pengaturan parameter benar, dan algoritma dapat bekerja dengan baik dalam setiap skala formasi. Sayangnya, hanya dikondisi pada lingkungan bebas hambatan (Tran and Lee, 2011).

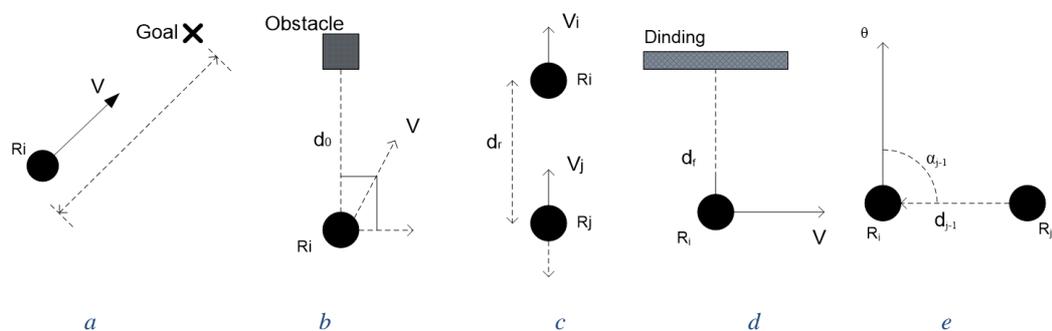


Gambar 2. 1. Sistem Formasi dengan Struktur Leader (L) dan Follower (F_n)

2.3.2. Sistem Kontrol Pendekatan Berbasis Perilaku

Strategi pendekatan berbasis perilaku pada koordinasi antar robot dikembangkan oleh Balch dan Arkin, 1998 untuk keperluan akuisisi dan stabilisasi formasi. Berbagai perilaku akan diintegrasikan agar semua robot dapat saling berkoordinasi dan berkomunikasi. Integrasi ini memungkinkan sistem untuk mencapai tujuan navigasi dan menghindari tabrakan sekaligus mempertahankan formasi geometris secara terstruktur (Balch and Arkin, 1998). Keuntungan utama strategi ini adalah dapat menghindari dan menangani tabrakan dengan mudah dengan reaksi yang dilakukan oleh setiap robot secara otonom (Lewis and Tan, 1997; Antonelli et al., 2010; Beard et al., 2001).

Sayangnya jika kompleksitas meningkat, analisis matematis menjadi sulit dilakukan (Balch and Arkin, 1998), sehingga tidak ada definisi yang jelas untuk perilaku kelompok robot *swarm* untuk menjamin stabilitas formasi yang diinginkan. Pada strategi kontrol formasi dengan pendekatan berbasis perilaku, kombinasi dari beberapa sub perilaku dapat dilakukan untuk menghasilkan kinerja pergerakan yang baik, yaitu, bergerak menuju target, menghindari rintangan, mengikuti dinding, menghindari robot dan pembentukan menjaga formasi (Gambar 2.2).



Gambar 2. 2. Perilaku kontrol dari robot individu

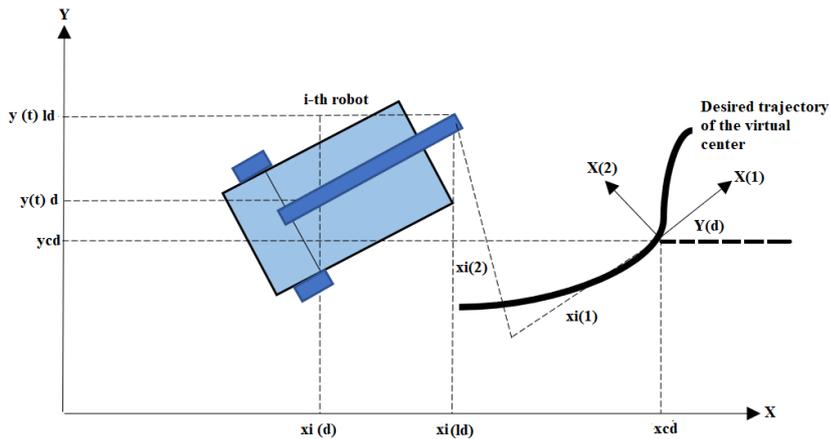
- a. bergerak ke target b. menghindari rintangan c. menghindari robot
d. mengikuti dinding e. menjaga formasi

2.3.3. Sistem Kontrol Dengan Pendekatan Berbasis Struktur Virtual

Dalam metode struktur virtual seluruh formasi dianggap sebagai satu kesatuan struktur untuk mencapai gerak yang diinginkan (Lewis and Tan, 1997; Ren and Randal, 2004; Antonelli et al., 2010). Pembentukan gerak dalam pendekatan ini sangat kaku karena hubungan geometris antara robot di sistem harus dipertahankan selama gerakan (Gambar 2.3). Oleh karena itu, umumnya sulit untuk melakukan pengontrolan formasi dan masalah dalam menghindari rintangan dengan parameter berubah terhadap waktu. Pendekatan struktur virtual tidak cocok untuk mengendalikan kelompok besar robot karena hubungan kendala antara robot menjadi lebih rumit sehingga kompleksitas meningkat.

Dalam beberapa kepustakaan, pendekatan ini digunakan dalam kontrol pembentukan formasi pesawat ruang angkasa dan kendaraan laut (Antonelli et al.,

2010; Ren and Randal, 2004). Kekuatan dari pendekatan ini adalah bahwa semua elemen dalam formasi memegang fungsi yang sama, dan formasi yang rapat selama manuver dapat diterapkan dan dipelihara. Namun, menghindari tabrakan sulit untuk dihindari karena tidak adanya interaksi antara kendaraan, serta kurang eksplisit umpan balik untuk formasi.



Gambar 2. 3. Pendekatan struktur virtual pada robot bergerak otonom

Secara umum, perbandingan antara strategi kontrol dalam sistem kontrol formasi dapat dilihat pada Tabel 2.2. berikut.

Tabel 2. 2. Perbandingan Strategi Kontrol Formasi

	<i>Behavior Based</i>	<i>Leader-Follower</i>	Struktur Virtual
Kontrol	desentralisasi (Sun et al., 2014)	sentralisasi (Tran dan Lee, 2011; Di Paola et al., 2010)	sentralisasi- desentralisasi (Lewis and Tan, 1997; Antonelli et al., 2010; Beard et al., 2001)
Kinerja	mempertahankan formasi, menghindari rintangan, bergerak mencapai tujuan (Balch, Arkin, 1998; Sun et al., 2014)	robot pemimpin mempertahankan lintasan dan pengikut melacak jarak relatif dari robot tetangga (Tran and Lee, 2011;	menjaga hubungan geometris antara robot di lingkungan dan harus mempertahankan selama melakukan gerakan (Antonelli et al., 2010)
Lingkungan	tidak diketahui (Xu et al., 2014)	tidak diketahui (Tran dan Lee, 2011)	tetap (Antonelli et al., 2010)
Aplikasi	penyelamatan dan pengawasan daerah berbahaya	patroli keamanan, militer, pemantauan (Di Paola et al., 2010;	kontrol pesawat ruang angkasa, satelit (Antonelli et al., 2010;

	(Tanner, and Piovesan, 2010)	Basilico et al., 2009)	Ren and Randal, 2004)
Kerugian	sulit untuk menganalisis sistem secara akurat, karena banyaknya perilaku (Balch, Arkin, 1998; Sun et al., 2014)	kegagalan robot pemimpin dalam mencapai target menyebabkan kegagalan seluruh sistem (Tran and Lee, 2011; Di Paola et al., 2010)	pengontrolan sangat kaku, susah untuk mengubah formasi terhadap waktu, dan kurang cocok untuk mengendalikan robot dalam kelompok besar ((Lewis and Tan, 1997; Antonelli et al., 2010; Beard et al., 2001)

2.4. Teknik Komputasi Cerdas Pada Robot *Swarm*

Prinsip-prinsip kolektif yang di implementasikan pada sistem robotika menggunakan kecerdasan *swarm* disebut robot *swarm*. Penelitian awal robot *swarm* dilakukan oleh (Dudek et al., 1993), diklasifikasikan berdasarkan ukuran robot, jangkauan komunikasi, topologi komunikasi, *bandwidth* komunikasi, kemampuan melakukan konfigurasi ulang, dan kemampuan pengontrolan formasi. Pembentukan formasi pada robot *swarm* secara umum untuk mempertahankan formasi dengan bentuk tertentu, seperti lingkaran (Defago and Konagaya, 2002), garis (Sugihara and Suzuki, 1990; Balch and Arkin, 1998) atau bahkan bentuk yang tidak menentu (Sahin et al., 2002).

Kecerdasan pada robot *swarm* merupakan kecerdasan yang menunjukkan perilaku dari kombinasi banyak individu (Sharkey, 2006). Karakteristik penting adalah dari penekanan biologis yang terinspirasi pada sistem kontrol lokal dan sistem komunikasi lokal sehingga munculnya perilaku global sebagai hasil dari pengorganisasian individu (Sharkey, 2007). Berdasarkan karakteristik penting tersebut menghasilkan berbagai ruang lingkup penelitian yang dapat dilakukan pada robot *swarm* yaitu sistem komunikasi, sistem kontrol, pemetaan dan lokalisasi, objek transportasi dan manipulasi, koordinasi gerak, pembelajaran dan tugas-tugas pengalokasian (Mohan and Ponnambalam, 2009).

Berbagai metode komputasi telah diimplementasikan untuk menyelesaikan masalah sistem kontrol pergerakan robot *swarm* salah satunya adalah metode *path-planning* dengan hasil yang cukup baik (Mohan and Ponnambalam, 2009). Metode *path-planning* menghasilkan kinerja yang efisien dan waktu pemrosesan

yang cepat baik pada kasus *local path-planning* (Langer et al., 2007; Lei and Li, 2007; Mohan and Ponnambalam, 2009) maupun *global path-planning* (Garro et al., 2007; Li et al., 2007; Kang et al., 2007). Selain itu metode ini juga menghasilkan kinerja pengontrolan yang cukup memuaskan pada lingkungan statis (Garro et al., 2007; Li et al., 2007) dan pada lingkungan dinamis (Berg et al., 2006; Tian et al., 2007). Namun metode ini harus mempersiapkan informasi di awal mengenai kondisi lingkungan, dan jika kondisi lingkungan tersebut berubah karena ketidakstrukturan dan ketidakpastian perubahan lingkungan maka metode ini akan mengalami kegagalan.

Beberapa metode dengan komputasi cerdas, seperti algoritma genetika, jaringan syaraf tiruan, algoritma semut dan *particle swarm optimization* sering digunakan karena kokoh dan kemampuan perhitungan simultan untuk memecahkan masalah kontrol pergerakan (Ghorbani, 2009; Sugawara, 2004; Qu, 2009; Eberhart, 1995 and Kennedy, 1995; Doctor, 2004; Garro et al., 2007). Salah satu metode untuk menghindari tabrakan pada robot bergerak otonom di lingkungan yang dinamik berdasarkan multi-sensor fusi data menggunakan metode *Support Vector Machine* (SVM) dengan hasil kemampuan beradaptasi yang baik (Tian et al., 2007). Selain itu banyak penelitian menggunakan metode PSO untuk optimisasi pergerakan bagi kendaraan tak berawak dalam mencapai target (Chen, 2006; Qin, 2004), navigasi robot sepak bola (Wang, 2006; Hao, 2007). Masalah lain yang telah diselesaikan adalah perencanaan gerakan secara *real-time*, menghasilkan waktu konvergen yang cepat, dan diimplementasikan dengan algoritma sederhana dan *trajectory* pergerakan yang halus dan efisien (Karimi, 2012),.

Sayangnya semua metode tidak mempertimbangkan ketidakpastian dalam proses pengendalian formasi. Ketidakpastian biasanya disebabkan oleh ketidaktelitian sensor, ketidakpresisian aktuator, dan perubahan lingkungan yang dinamik (Nurmaini and Primanita, 2012). Masalah tersebut dapat menyebabkan kesalahan perubahan posisi dan orientasi robot. Jika nilainya terakumulasi, maka dapat menghasilkan kesalahan pergerakan dalam pembentukan formasi. Berdasarkan penelitian-penelitian terdahulu metode logika *fuzzy* mampu

mengatasi semua bentuk ketidakpastian pada sistem kontrol robotika dengan hasil yang memuaskan (Saffiotti, 1997; Bazoula et al., 2008; Jamshidi et al., 1993) .

Logika *Fuzzy* memiliki kemampuan menangani masalah-masalah ketidakpastian tersebut (Zadeh, 1965; Klir and Yuan, 1995; Mendel, 2001). Teknik dari Logika *Fuzzy* yang sering digunakan di dalam suatu sistem kendali adalah sistem Logika *Fuzzy* Tipe-1 (SLFT1) (Mendel, 2001; El-Desouky and Hagra, 2009). Namun, SLFT1 memiliki sebuah keterbatasan, yaitu pada tingkatan tertentu sebuah himpunan *fuzzy* tipe-1 tidak lagi membawa sifat ketidakpastian saat masukan kepada sistem adalah besaran tegas (*crisp*) (El-Desouky and Hani, 2009). Hal ini dikarenakan pada tingkatan tertentu SLFT1 hanya memetakan besaran tegas menjadi besaran tegas yang baru antara nol dan satu, menghilangkan sifat ketidakpastian yang awalnya merupakan kelebihan yang ditawarkan oleh konsep Logika *Fuzzy*. Hasilnya memang tidak signifikan, tetapi akumulasinya sangat berpengaruh pada kinerja sistem kontrol (Klir and Yuan, 1995).

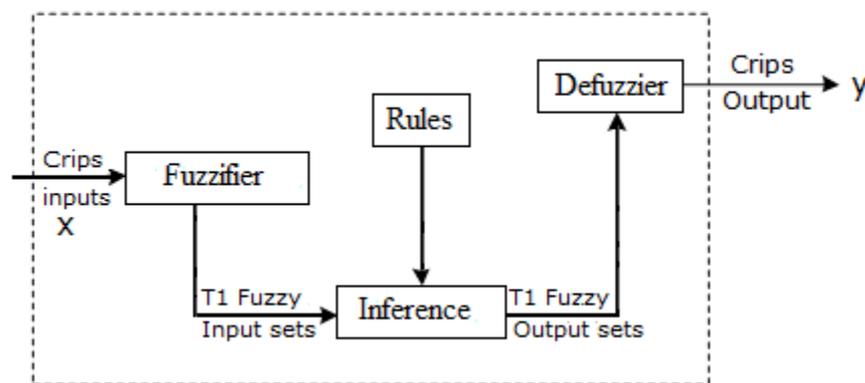
Hilangnya konsep ketidakpastian mengurangi kemampuan penanganan ketidakpastian dalam lingkungan tidak terstruktur. Maka dari itu untuk menyelesaikan masalah tersebut, dikembangkan algoritma sistem Logika *Fuzzy* Tipe-2 (SLFT2). Algoritma SLFT2 merupakan perluasan dari algoritma SLFT1. Sayangnya SLFT2 menghasilkan komputasi yang besar dan kompleks sehingga kurang efisien untuk diimplementasikan pada aplikasi *real-time* seperti pengontrolan gerak robot (El-Desouky and Hani, 2009; Hagra, 2012). Sehingga dikembangkan algoritma sistem logika *fuzzy* type-2 dengan fungsi keanggotaan yang lebih sederhana (Wu, 2013; Nurmaini and Tutuko, 2011). Fungsi keanggotaan dalam algoritma SLFT2 berupa interval dan memiliki dua derajat keanggotaan, yaitu derajat keanggotaan primer dan sekunder. Hal tersebut diharapkan dapat menyelesaikan masalah ketidakpastian yang terdapat pada SLFT1 (Nurmaini and Tutuko, 2011).

2.5. Metode Sistem Kontrol Formasi

Berbagai metode sistem kontrol formasi telah diteliti, dari sistem kontrol konvensional berbasis model matematis *Lyapunov* (Benzerrouk et al., 2010), *Lyapunov Redesign* (Soorki et al., 2011), Algoritma *Backstepping* (Li, et al., 2005), *Dynamic Feedback Linierization* (Gamage et al., 2009), pendekatan Kinematik (Kowdiki et al., 2012), Algoritma D^* (Berg et al., 2006) dengan hasil yang baik. Selain itu ada juga yang menggunakan sistem kontrol berbasis kecerdasan buatan *fuzzy logic* (Lei and Li, 2007), *Particle Swarm Optimization* (Rigatos, 2008), interval *fuzzy kohonen network* (Nurmaini et al., 2014), *Ant Colony Optimization* (Garro et al., 2007) pendekatan *bioinspired neurodynamic* (Peng et al., 2013) dengan hasil yang memuaskan. Dalam penelitian ini yang digunakan sebagai metode pengembangan sistem kontrol formasi, akan dijelaskan pada sub-bab selanjutnya.

2.5.1. Sistem Logika Fuzzy Type-1

Sistem logika *fuzzy* type-1 (SLFT1) dapat dilihat blok diagramnya pada Gambar 2.4. SLFT1 terdiri atas empat proses yaitu fuzzifikasi, basis kaidah, penarikan kesimpulan (*inferencing*), dan defuzzifikasi (Zadeh, 1965). Setiap proses-proses ini akan dijelaskan kemudian, bersama dengan dasar dari himpunan *fuzzy*.



Gambar 2. 4. Sistem logika *fuzzy* tipe-1 (SLFT1)

2.5.1.1. Himpunan *Fuzzy*

Pada suatu himpunan *fuzzy*, elemen-elemennya memiliki nilai derajat keanggotaan tertentu, sehingga nilainya di dalam himpunan memiliki nilai rentang dari suatu nilai kebenaran. Dapat juga sebagian bernilai benar atau salah dalam derajat tertentu. Derajat keanggotaan ini bernilai antara 0 (nol) sampai dengan 1 (satu). Pemetaan elemen-elemen dari suatu himpunan *fuzzy* ke derajat keanggotaannya dinyatakan dalam fungsi keanggotaan. Oleh karena itu, sebuah himpunan *fuzzy* dapat dinyatakan dalam bentuk fungsi keanggotaannya.

Himpunan *fuzzy* Tipe-1 adalah generalisasi dari himpunan crisp, yang nilai keanggotaannya antara 0 dan 1. Himpunan *fuzzy* dari A didefinisikan pada semesta pembicaraan X dan ditandai dengan tingkat keanggotaan (x) yang mengambil nilai dalam interval [0,1]. Himpunan *fuzzy* A atas semesta pembicaraan X didefinisikan sebagai (Mendel, 1995),

$$\mu_A(x): X \rightarrow [0,1] \quad (2.1)$$

Jika bentuk himpunan X adalah dalam domain kontinu dapat dinyatakan sebagai persamaan berikut,

$$A = \int_x \mu_A(x) / x \quad (2.2)$$

Tanda integrasi menunjukkan pengumpulan semua titik $x \in X$ yang diasosiasikan sebagai derajat keanggotaan $\mu_A(x)$.

Sedangkan apabila bentuk himpunan X adalah dalam domain diskrit, maka

$$A = \sum_{x_i} \in \frac{x_i \mu_A(x)}{x_i} \quad (2.3)$$

Tanda sigma menunjukkan pengumpulan semua pemotongan titik $x \in X$ yang diasosiasikan sebagai derajat keanggotaan $\mu_A(x)$.

Operator Himpunan *Fuzzy*

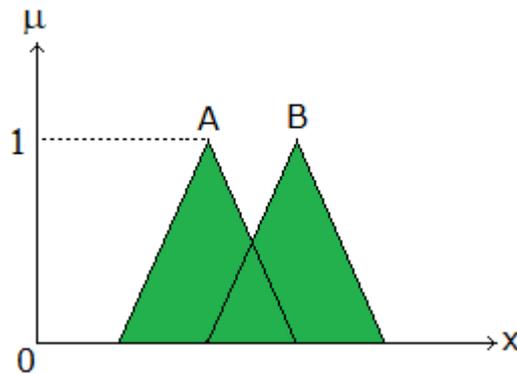
Berikut ini adalah teori fungsi operasi dari himpunan operasi gabungan, irisan dan komplemen yang didefinisikan dengan A , B dan C pada semesta pembicaraan X .

a) Operasi Gabungan

Operasi gabungan dari himpunan *fuzzy* pada semesta pembicaraan X dapat didefinisikan sebagai berikut :

$$\mu_{A \cup B}(x) = \mu_A(x) \vee \mu_B(x) \quad (2.4)$$

Tanda \vee menunjukkan operator maksimum.



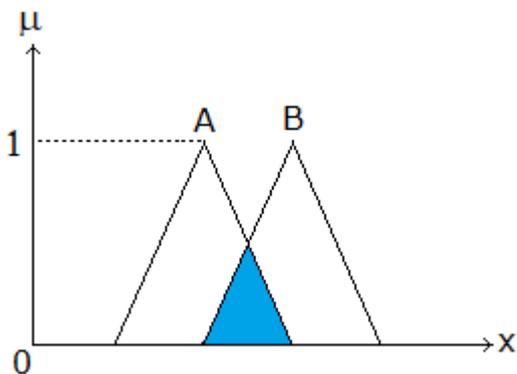
Gambar 2. 5. Gabungan dari himpunan *fuzzy*

b) Operasi Irisan

Operasi irisan dari himpunan *fuzzy* pada semesta pembicaraan X dapat didefinisikan sebagai berikut:

$$\mu_{A \cap B}(x) = \mu_A(x) \wedge \mu_B(x) \quad (2.5)$$

Tanda \wedge menunjukkan operator minimum.

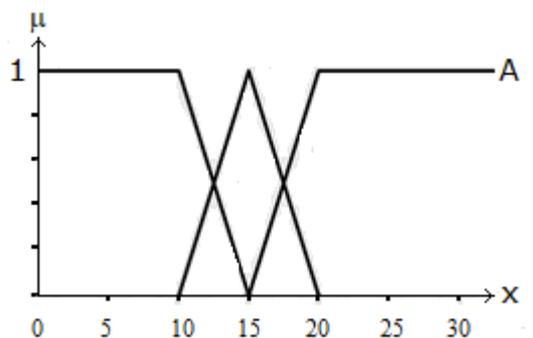


Gambar 2. 6. Irisan dari himpunan *fuzzy*

c) Operasi Komplemen

Komplemen dari sebuah himpunan dari semesta X , dan himpunan *fuzzy* A dapat didefinisikan sebagai berikut:

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad (2.6)$$



Gambar 2. 7. Komplemen dari himpunan fuzzy

2.5.1.2 Fuzzifikasi

Fuzzifikasi melakukan pemetaan dari input pemetaan $x = (x, \dots, x)$ ke dalam himpunan *fuzzy* di U . Himpunan *Fuzzy* dikaitkan dengan istilah yang muncul di anteseden atau konsekuensi dari aturan, dan dengan input ke dan output dari SLFT1. Basis kaidah *fuzzy* If-Then terdiri dari himpunan anteseden (input) dan konsekuen (output) memungkinkan objek yang dikontrol untuk mentolerir ketidakpastian dan ketidaktepatan saat melakukan aksinya. Dalam hal pengendalian robot bergerak otonom, anteseden memungkinkan pemrosesan input sensorik atau fuzzifikasi dan konsekuensinya memungkinkan defuzzifikasi untuk menghasilkan output untuk kontrol aktuator. Sistem logika *fuzzy* ini dikategorikan sebagai sistem logika *fuzzy* tipe-1 (SLFT1).

Proses ini melibatkan penggunaan himpunan *fuzzy* tipe-1, dan proses pemetaan nilai tegas (*crisp*) dari masukan ke dalam himpunan *fuzzy* menggunakan fungsi keanggotaan. Fungsi keanggotaan tipe-1 adalah fungsi yang merepresentasikan himpunan *fuzzy* tipe-1. Nilai masukan dipetakan kedalam derajat keanggotaan menggunakan fungsi ini. Sebagai salah satu contoh pernyataan matematis fungsi keanggotaan segitiga dan pernyataan matematis

fungsi keanggotaan trapesium ditunjukkan pada Persamaan (2.7) dan Persamaan (2.8).

$$\text{Segitiga}(x: a, b, c) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a < x \leq b \\ \frac{c-x}{c-b}, & b < x \leq c \\ 0, & c < x \end{cases} \quad (2.7)$$

$$\text{Trapezium}(x: a, b, c, d) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a < x \leq b \\ 1, & b < x \leq c \\ \frac{d-x}{d-c}, & c < x \leq d \\ 0, & d < x \end{cases} \quad (2.8)$$

Kedua persamaan (2.7) dan persamaan (2.8) dapat dinyatakan sebagai fungsi masukan maupun fungsi keluaran dari SLFT1.

II.5.1.3. Fuzzy Inference System

Sebuah *Fuzzy Inference System* (FIS), adalah sistem pengambilan keputusan pada konsep logika *fuzzy*, dimana derajat keanggotaan yang dihasilkan dari proses fuzzifikasi digabungkan berdasarkan aturan tertentu dan kemudian, kaidah-kaidah yang aktif dipotongkan ke himpunan kesimpulan (*rule firing*). Terdapat lebih dari satu metode yang dapat digunakan dalam sebuah FIS, tetapi yang paling banyak digunakan adalah metode inferensi Mamdani (Negnevitsky, 2005). Dibawah ini akan dijelaskan mengenai metode *inference* Mamdani.

Dalam proses FIS terdapat tiga jenis metode yang sering digunakan yaitu metode Mamdani, metode Sugeno dan metode Takagi-Sugeno. Ketiga metode tersebut memiliki perbedaan hanya pada fungsi keanggotaan keluaran. Sedangkan proses fuzzifikasi, basis kaidah, inferensi *fuzzy* dan defuzzifikasi memiliki cara yang sama. Tipe Mamdani memberikan cakupan pengetahuan yang lebih luas dalam merepresentasikan intuisi dan perilaku manusia. Sayangnya metode Mamdani mempunyai kelemahan dalam hal komputasi dikarenakan membutuhkan

memori yang cukup besar sehingga metode ini kurang cocok digunakan dalam mikrokontroler. Hal tersebut dikarenakan saat melakukan perhitungan yang berhubungan dengan fungsi keanggotaan output, memerlukan perhitungan rata-rata. Sedangkan metode Sugeno langsung dapat menentukan nilai keluaran karena fungsi keluarannya berbentuk singleton.

Dalam FIS terdapat implikasi yaitu proses untuk menghitung nilai yang diberikan dari *antecedent* ke *consequent* dan mengumpulkan semua *consequent* dari setiap kaidah ke dalam *consequent* yang baru untuk mendapatkan keputusan. Terdapat dua tipe implikasi yaitu Komposisi Max-Min dan Komposisi Max Product.

(a) Komposisi Max - Min

Dimisalkan R_1 dan R_2 adalah dua relasi *fuzzy* pada $X * Y$ dan $Y * Z$, Komposisi Max - Min dapat di definisikan sebagai:

$$R_1 \circ R_2 = \left\{ \left[(x, z), \max_y \min \left(\mu_{R_1}(x, y), \mu_{R_2}(y, z) \right) \right] \mid x \in X, y \in Y, z \in Z \right\} \quad (2.9)$$

(b) Komposisi Max Produk

Dimisalkan R_1 and R_2 adalah relasi dari *fuzzy* dalam $X * Y$ dan $Y * Z$, Komposisi Max Produk dapat di definisikan sebagai:

$$\mu_{R_1 \circ R_2} = \max_y \left[\mu_{R_1}(x, y), \mu_{R_2}(y, z) \right] \quad (2.10)$$

II.5.1.4. Basis Kaidah

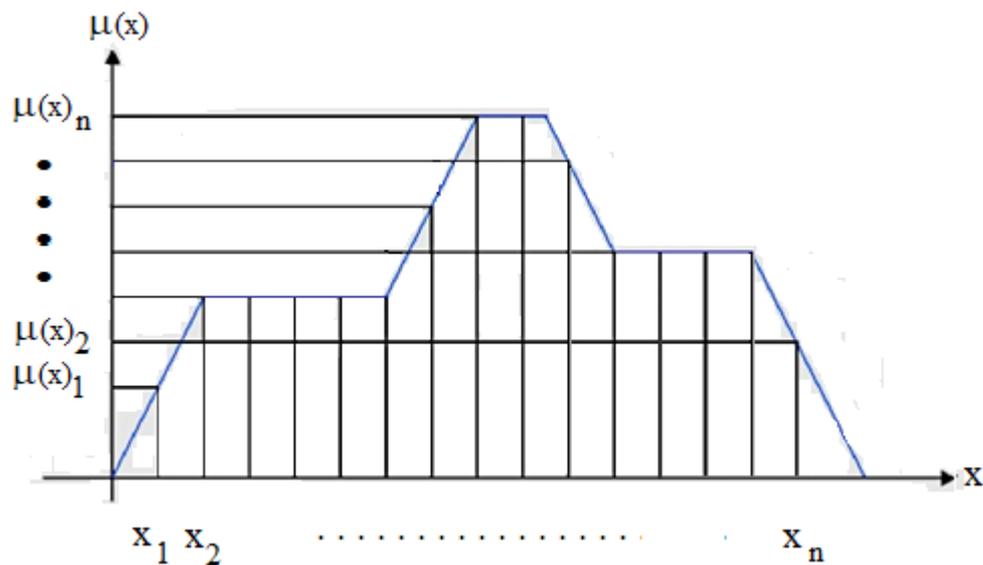
Dalam SLFT1, basis kaidah adalah kumpulan pernyataan If - Then. Dimana bagian dari aturan **IF** adalah anteseden, dan bagian **THEN** adalah konsekuensinya, dalam bentuk berikut (Mendel, 1995):

$$R^l : \text{IF } x_1 \text{ is } X_1^l \text{ and } \dots \text{ and } x_p \text{ is } X_p^l, \text{ THEN } y \text{ is } Y^l \quad l = 1, 2, \dots, N \quad (2.11)$$

dimana X_i^l ($i = 1, \dots, p$) dan Y^l adalah himpunan *fuzzy* tipe-1 dalam U_i dan V , berturut-turut, dan $x = (x_1, \dots, x_p) \in U_1 \times U_2 \times \dots \times U_p \equiv U$ dan $y \in V$ adalah variabel linguistik; x adalah input SLFT1, dan y adalah output SLFT1. Dalam mekanisme mesin inferensi, prinsip logika *fuzzy* digunakan untuk mengombinasikan kaidah If-Then. Dalam kombinasi beberapa kaidah memetakan himpunan *fuzzy* U ke himpunan *fuzzy* dalam V . Kemudian, defuzzifikasi akan membentuk pemetaan dari himpunan *fuzzy* V menjadi suatu nilai crisp output $y \in V$.

2.5.1.5. Defuzzifikasi

Defuzzifikasi adalah implementasi untuk merubah nilai *fuzzy* menjadi nilai nyata. Terdapat beberapa metode defuzzifikasi, namun metode yang paling terkenal adalah teknik titik tengah (Centre of Gravity (COG)). Secara matematis disebut titik gravitasi (2.19) dan digambarkan sebagai berikut (Gambar 2.8);



Gambar 2. 8. Metode COG

Dengan menggunakan persamaan (2.12) maka titik tengah dari metode COG dapat dihitung. Ide dasarnya adalah dengan mensampling semua nilai himpunan *fuzzy* menjadi nilai luaran sebagai berikut;

$$COG = \frac{\sum_{x=a}^b \mu_A(x)x}{\sum_{x=a}^b \mu_A(x)} \quad (2.12)$$

2.5.2. Sistem Logika *Fuzzy* Tipe-2

SLFT2 adalah penyempurnaan dari SLFT1, karena memiliki kemampuan untuk menangani ketidakpastian yang tidak dimiliki oleh SLFT1 (Karnik et al., 1999; Karnik and Mendel, 2001; Mendel, 2007; Mendel 2009). Pemrosesan dari SLFT2 mirip dengan SLFT1, pembeda utama dari kedua sistem ini adalah pada pemrosesan keluaran (*output processing*), dimana SLFT2 memiliki proses reduksi tipe (*type reducer*) sebelum proses defuzzifikasi (Mendel 2007). Selain itu sistem logika *fuzzy* tipe-2 (SLFT2) adalah generalisasi SLFT1 dalam hal merepresentasikan ketidakpastian. Tidak dibatasi hanya pada variable linguistic tetapi juga pada definisi fungsi keanggotaan (MFs) (Karnik and Mendel, 1998).

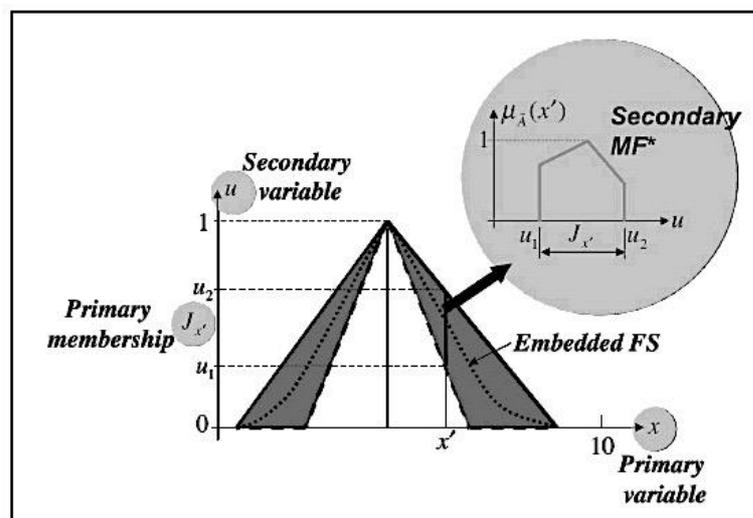
SLFT2 sangat berguna pada kondisi lingkungan yang tidak diketahui, dan terdapat ketidakpastian pengukuran (Mendel, 2000). Dengan metode SLFT2 memungkinkan untuk meminimisasi efek ketidakpastian yang tidak dapat dimodelkan secara langsung oleh SLFT1 (Mendel and Wu, 2002), karena SLFT2 merupakan peningkatan dalam bentuk interpretasi fungsi keanggotaan SLFT1 dengan menambahkan tingkat dimensi dari derajat (Mendel and Wu, 2002).

2.5.2.1. Himpunan *Fuzzy* Tipe-2

Secara umum mesin penalaran *fuzzy* dapat mengatasi ketidakpastian dengan memasukkan rentang nilai ketidakpastian antara [0,1] dalam SLFT1. Namun, untuk menentukan nilai eksak dari ketidakpastian seperti kesalahan pengukuran sangat sulit. Selain itu sangat tidak beralasan untuk menggunakan fungsi keanggotaan yang pasti untuk suatu nilai yang tidak pasti. Sehingga jenis himpunan *fuzzy* yang dapat mengatasi itu diperlukan, agar dapat mengatasi masalah ketidakpastian yang di sebut dengan himpunan *fuzzy* tipe-2 (Karnik and Mendel, 1998).

Tidak seperti himpunan *fuzzy* tipe-1, derajat keanggotaan adalah bernilai *crisp* antara [0,1], himpunan *fuzzy* tipe-2 di karakteristik oleh fungsi

keanggotaan *fuzzy*. Nilai keanggotaan atau derajat keanggotaan untuk setiap elemen dari himpunan *fuzzy* $[0,1]$ (John, 1998; Mendel and John, 2002). Himpunan tersebut dapat digunakan dalam situasi dimana ketidakpastian tentang derajat keanggotaan mereka seperti bentuk dari fungsi keanggotaan, atau berada dalam parameternya. Bentuk lainnya adalah dengan melihat SLFT1 sebagai pendekatan orde ke satu untuk merepresentasikan ketidakpastian. Sedangkan SLFT2 dapat dianggap menggunakan pendekatan orde ke dua (Mendel, 2000). Fungsi keanggotaan orde ke dua menambah dimensi pada himpunan *fuzzy* tipe-2. Fungsi keanggotaan 3 dimensi yang di wakikan pada fungsi keanggotaan dapat dilihat pada Gambar 2.9.



Gambar 2. 9.Representasi himpunan *fuzzy* tipe-2 (Mendel, 2007)

Untuk setiap nilai x adalah sub-set dari semesta pembicaraan X didefinisikan sebagai \tilde{A} (Karnik and Mendel, 2002),

$$\tilde{A} = \int_{x \in X} \mu_A(x) / x \quad (2.13)$$

atau,

$$\mu_{\tilde{A}}(x) = \int_u f_x(u)/u \quad , \quad u \in J_x \subseteq [0,1] \quad (2.14)$$

dimana x adalah variabel primer dengan domain X , $\mu_{\tilde{A}}(x)$ adalah nilai fungsi keanggotaan himpunan *fuzzy* tipe-2 dari x ; $f_x(u)$ adalah derajat sekunder dari u , bobot yang mungkin dapat ditandai untuk setiap u , u adalah variabel sekunder, yang memiliki domain J_x pada setiap nilai $x \in X$; J_x adalah variabel primer of x .

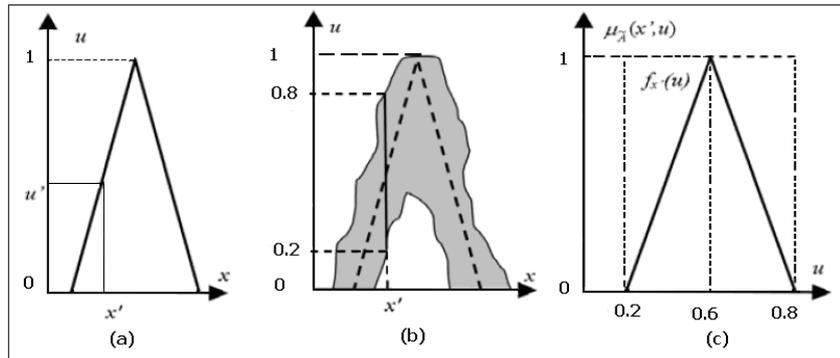
Suatu himpunan *fuzzy* tipe-2 ditandai dengan \tilde{A} di karakteristikkan oleh fungsi keanggotaan *fuzzy* tipe-2 $\mu_{\tilde{A}}(x,u)$, dimana $x \in X$ dan $u \in J_x$ dalam interval $0 \leq \mu_{\tilde{A}}(x,u) \leq 1$. \tilde{A} dapat diekspresikan sebagai berikut (Karnik and Mendel, 2002):

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} \mu_{\tilde{A}}(x,u)/(x,u) \quad J_x \subseteq [0,1] \quad (2.15)$$

Dimana $\int \int$ menunjukkan union untuk semua kemungkinan x dan u . Pada setiap nilai x , ($x = x'$), luas penampang berbentuk 2-D yang memiliki koordinat u dan $\mu_{\tilde{A}}(x',u)$ disebut *vertical slice* dari $\mu_{\tilde{A}}(x',u)$. Fungsi keanggotaan sekunder adalah sebuah *vertical slice* dari $\mu_{\tilde{A}}(x',u)$. Dalam hal $\mu_{\tilde{A}}(x = x',u)$ untuk $x' \in X$ dan $\forall u \in J_{x'} \subseteq [0,1]$ dapat dinyatakan sebagai berikut (Karnik and Mendel, 2002):

$$\mu_{\tilde{A}}(x = x',u) = \mu_{\tilde{A}}(x') = \int_{u \in J_{x'}} f_{x'}(u)/(u) \quad J_{x'} \subseteq [0,1] \quad (2.16)$$

Dimana $0 \leq f_{x'}(u) \leq 1$, karena $\forall u \in J_{x'} \subseteq [0,1]$ pada notasi $\mu_{\tilde{A}}(x')$ diturunkan dan $\mu_{\tilde{A}}(x)$ di refer sebagai fungsi keanggotaan sekunder. Mendel (2000) menyatakan bahwa, nama yang mewakili fungsi keanggotaan tipe-2 diasosiasikan sebagai fungsi keanggotaan sekunder, jika $f_x(u) = 1$, untuk $\forall u \in J_x \subseteq [0,1]$. Fungsi keanggotaan dalam bentuk himpunan interval dapat dilihat pada Gambar (2.10) berikut.



Gambar 2. 10.(a) fungsi keanggotaan tipe-1 (b) Fungsi keanggotaan tipe-1 dengan ketidakpastian (c) fungsi keanggotaan sekunder

2.5.2.2. Footprint of Uncertainty (FOU)

Elemen instrumentasi biasanya sering menunjukkan suatu nilai ketidaktepatan dan ketidakpastian yang disebut ketidakpastian yaitu *amplifier*, *sensor*, *digital to analog (DAC)*, *analog to digital converters (ADC)*, dan aktuator (Melin and Castillo, 2004). Sehingga dalam aksi kontrol yang dihasilkan akan merubah nilai aktuator menjadi tidak teliti. Hal tersebut dapat menyebabkan ketidak konsistenan pada perubahan lingkungan (Melin and Castillo, 2004).

Ketidakpastian ini menyebabkan kesulitan dalam menentukan *antecedent* dan *consequent* dari fungsi keanggotaan *fuzzy*. Dalam fungsi keanggotaan *fuzzy* tipe-2, \tilde{A} , ketidakpastian disebut “*footprint of uncertainty*” (FOU). Secara matematis dapat dinyatakan sebagai (Karnik and Mendel, 2002),

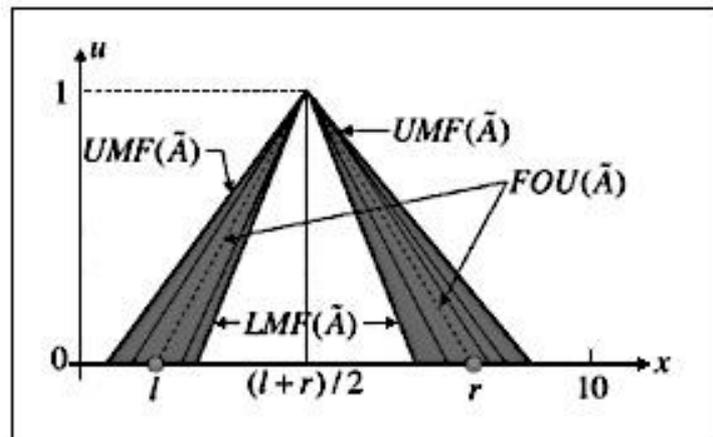
$$\text{FOU}(\tilde{A}) = \bigcup_{x \in X} J_x \quad (2.17)$$

FOU dari himpunan *fuzzy* tipe-2 dibatasi oleh dua fungsi keanggotaan *fuzzy* tipe-1, yaitu fungsi keanggotaan atas (*upper membership functions (UMFs)*) and fungsi keanggotaan bawah (*lower membership functions (LMFs)*) (Mendel, 2000) ditunjukkan pada Gambar 2.11. UMFs diasosiasikan sebagai batas atas dari FOU (\tilde{A}) dan dinyatakan sebagai $\bar{\mu}_{\tilde{A}}(x), \forall x \in X$ dan LMs diasosiasikan sebagai batas bawah dari FOU (\tilde{A}) dan dinyatakan sebagai $\underline{\mu}_{\tilde{A}}(x), \forall x \in X$, semua batas tersebut dapat dinyatakan dengan persamaan berikut (Karnik and Mendel, 2002):

$$\overline{\mu_{\tilde{A}}}(x) \equiv \overline{FOU(\tilde{A})} \quad \forall x \in X \quad (2.18)$$

$$\underline{\mu_{\tilde{A}}}(x) \equiv \underline{FOU(\tilde{A})} \quad \forall x \in X \quad (2.19)$$

Dengan teori FOU dan penentuan batas atas dan batas bawah dari fungsi keanggotaan maka dikenal dengan sebutan fungsi keanggotaan interval (Mendel, 2000).

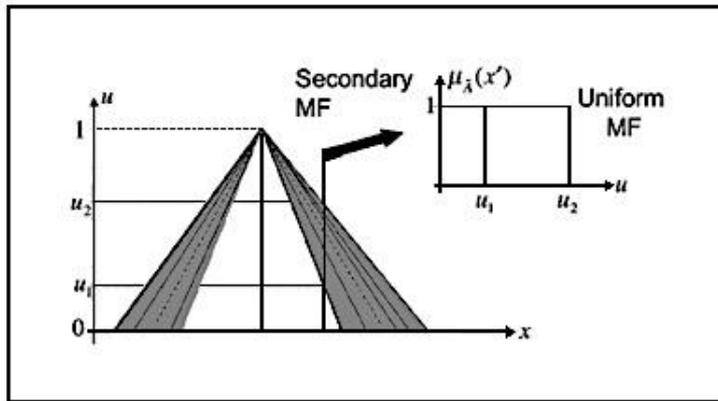


Gambar 2. 11. Footprint of uncertainty (FOU) (Mendel 2007)

2.5.2.3. Sistem Logika *Fuzzy* Tipe-2 Interval

SLFT2 dapat digunakan pada lingkungan ketika kondisi lingkungan sangat tidak pasti dalam menentukan derajat keanggotaan, seperti peralatan instrumentasi yang diinterferensi oleh suatu *noise*. Namun, pada umumnya SLFT2 menghasilkan komputasi yang besar (Liang and Mendel, 2000a; Wu and Mendel, 2001). Untuk menyederhanakannya fungsi keanggotaan sekunder di buat menjadi 0 (nol) yang disebut dengan fungsi keanggotaan interval (Mendel and John, 2002). Representasi fungsi keanggotaan interval dapat dilihat pada Gambar 2.12.

Dari Gambar 2.12, area yang diarsir di boboti secara uniform dalam FOU, garis tebal bagian atas disebut UMFs, dan bagian bawah disebut LMF, dan diantaranya adalah fungsi keanggotaan *fuzzy-1*.



Gambar 2. 12. Fungsi keanggotaan Tipe-2 Interval (Mendel 2007)

Jika setiap bobot memiliki derajat keanggotaan sama dengan 1 (satu), maka persamaan derajat keanggotaan *fuzzy-2* interval menjadi (Mendel, and john, 2002):

$$\mu_{\tilde{A}}(x) = \int_u 1/u \quad , \quad u \in J_x \subseteq [0,1] \quad (2.20)$$

Kemudian, himpunan *fuzzy-2* interval \tilde{A} dalam persamaan (2.5) dapat ditulis sebagai berikut (Wu and Mendel, 2001):

$$\tilde{A} = \int_{x \in X} \left[\int_{u \in [\underline{\mu}_{\tilde{A}}(x), \bar{\mu}_{\tilde{A}}(x)]} (1/u) / x \right] \quad (2.21)$$

dua titik akhir pada batas bawah integral diasosiasikan sebagai dua fungsi keanggotaan *fuzzy-1* dalam UMFs $\bar{\mu}_{\tilde{A}(x)}$ dan LMFs $\underline{\mu}_{\tilde{A}(x)}$ (Liang and Mendel 2000a).

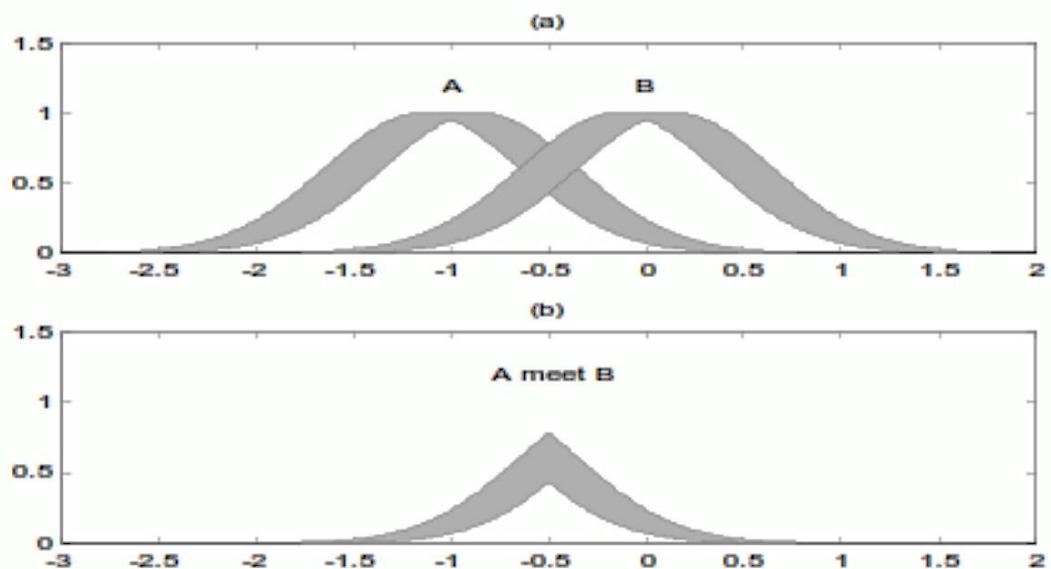
2.5.2.4. Operasi Himpunan *Fuzzy* Tipe-2 Interval

Untuk himpunan *fuzzy* tipe-2 interval, operator baru dinamakan sebagai irisan (*meet*) dan gabungan (*join*). Karnik and Mendel (2001a) menurunkan persamaan irisan dan gabungan pada himpunan *fuzzy* tipe-2 interval ke dalam himpunan *fuzzy* tipe-1 interval. Jika $F = \int_{v \in F} 1/v$ adalah himpunan *fuzzy* tipe-1

interval dalam domain $[l_f, r_f] \subseteq [0,1]$ dan $G = \int_{w \in G} 1/w$ adalah himpunan fuzzy tipe-1 interval lainnya dalam domain $[l_g, r_g] \subseteq [0,1]$. Maka irisan Q antara F dan G di bawah *product t-norm* dapat ditulis sebagai berikut (Karnik and Mendel, 2001a):

$$Q = F \prod G = \int_{q \in [l_f l_g, r_f r_g]} 1/q \quad (2.22)$$

Dari persamaan (2.22) setiap bentuk $F \prod G$ adalah sama dengan perkalian $v.w$ untuk nilai $v \in F$ dan $w \in G$ dalam bentuk terkecilnya adalah $l_f l_g$ dan terbesarnya adalah $r_f r_g$. Jika kedua himpunan F dan G dalam domain kontinu, maka $F \prod G$ juga dalam domain kontinu. Sehingga irisan himpunan fuzzy $F \prod G$ merupakan himpunan fuzzy tipe-1 interval dalam domain $[l_f l_g, r_f r_g]$. Dengan cara yang sama irisan dalam *product t-norm* dari himpunan fuzzy interval tipe 1 F_1, \dots, F_n memiliki domain $[l_1 r_1], \dots, [l_n r_n]$ berturut-turut, atau $[\prod_{0=1}^n l_o, \prod_{0=1}^n r_o]$ (Karnik and Mendel, 2001a). Representasi irisan antara himpunan F dan G dapat dilihat pada Gambar 2.13.

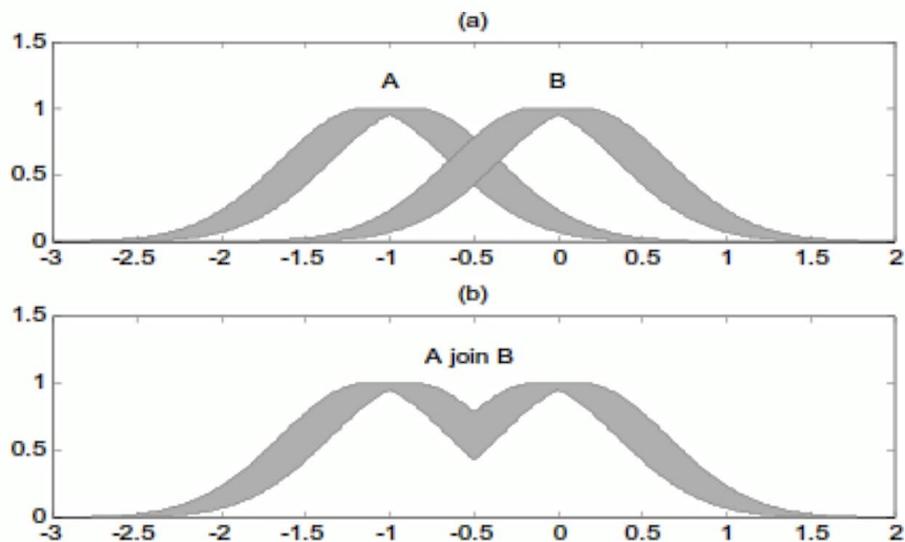


Gambar 2. 13. Teori irisan himpunan fuzzy tipe-2

Gabungan antara F dan G dapat diselesaikan sebagai berikut (Karnik and Mendel, 2001),

$$Q = F \amalg G = \int_{q \in [l_f \vee l_g, r_f \vee r_g]} 1/q \quad (2.23)$$

Dimana $q = v \vee w$ dan \vee menunjukkan operasi maksimum yang digunakan pada SLFT2 interval. Gabungan dari n himpunan *fuzzy* tipe-1 interval F_1, \dots, F_n dalam domain $[l_1 r_1], \dots, [l_n r_n]$, adalah himpunan interval dalam domain $(l_1 \vee l_2 \vee \dots \vee l_n), (r_1 \vee r_2, \vee \dots \vee r_n)$, atau $[\max(l_1, l_2, \dots, l_n), \max(r_1, r_2, \dots, r_n)]$. Representasi gabungan antara himpunan F dan G dapat dilihat pada Gambar 2.14.



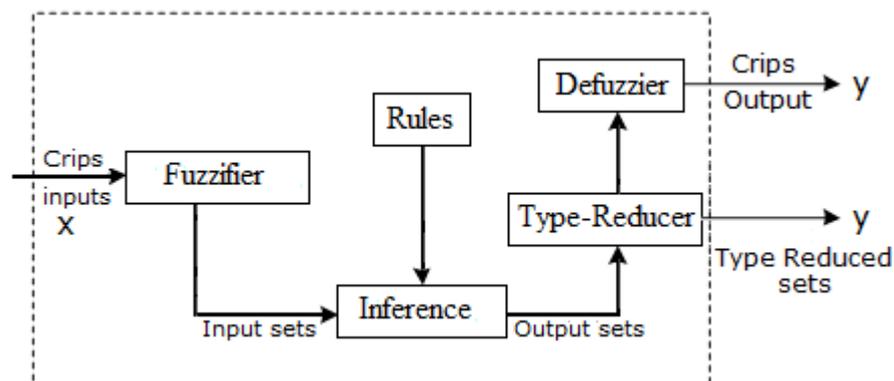
Gambar 2. 14. Teori gabungan himpunan *fuzzy* tipe-2

2.5.2.5. Pengendali Sistem Logika *Fuzzy* Tipe-2

Pada sistem pengendali SLFT2 memiliki karakteristik yang sama dengan SLFT1, yaitu menggunakan basis kaidah If-Then, tetapi *antecedent* dan *consequent* adalah himpunan *fuzzy* tipe-2. Selain itu untuk mengubah himpunan *fuzzy* tipe-2 agar menghasilkan nilai output maka pada SLFT2 memiliki tahapan reduksi tipe. Secara umum sistem pengendali dengan metode SLFT2 memiliki tahapan sebagai berikut fuzzifikasi, basis kaidah, mesin inferensi *fuzzy*, reduksi tipe dan defuzzifikasi (Liang and Mendel, 2000b). Secara umum SLFT2 bekerja

sebagai berikut: nilai *crisp* di fuzzifikasi kedalam bentuk himpunan *fuzzy* tipe-2 interval. Input dari *fuzzy* tipe-2 interval akan mengaktifkan mesin inferensi dan dengan basis kaidah akan menghasilkan keluaran menggunakan perhitungan *centroid* menjadi himpunan *fuzzy* tipe-1 yang disebut himpunan *fuzzy* yang direduksi (Hagras, 2004).

Ada dua metode yang dapat dilakukan untuk proses reduksi tipe yaitu (1) dengan metode iterative Karnik-Mendel (KM), dan (2) dengan menggunakan metode Wu-Mendel (Wu and Mendel, 2002). Setelah reduksi tipe dilakukan maka himpunan *fuzzy* di reduksi dilakukan defuzzifikasi dengan mencari rata-rata dari nilai keluaran. Gambar blok diagram proses *fuzzy* tipe-2 dengan fungsi keanggotaan interval dapat dilihat pada Gambar 2.15.



Gambar 2. 15. Sistem Logika Fuzzy Tipe 2 (SLFT2)

Fuzzifikasi

Fuzzifikasi memetakan nilai crisp $x = (x_1, \dots, x_p)^T \in X_1 \times X_2 \times \dots \times X_p \equiv X$, ke himpunan *fuzzy* tipe-2 interval \tilde{A}_x dalam X (Mendel, 2001). Dalam kasus ini, tipe fuzzifikasi singleton yang digunakan agar hanya satu titik bernilai tidak nol yang dihasilkan (Liang et al., 2000; Mendel, 2001). \tilde{A}_x adalah himpunan *fuzzy* tipe-2 interval *singleton* jika $\mu_{\tilde{A}_x}(x) = 1/1$ for $x=x'$ dan $\mu_{\tilde{A}_x}(x) = 1/0$ untuk semua $x \neq x'$.

Sebagai salah satu contoh fungsi keanggotaan *Gaussian*. Nilai masukan dari pengguna, yaitu σ dan c kan membentuk himpunan fungsi keanggotaan *Gaussian*

μ_i . Untuk menentukan batas-batas FOU dari himpunan interval *fuzzy* tipe 2, f_i^l nilai dari parameter masukan dihitung dengan rumus standar deviasi pada Persamaan (2.24)

$$f_i^l = \exp \left[-\frac{1}{2} \left(\frac{x_i - c_{i1}^l}{\sigma_i^l} \right)^2 \right], \sigma_i^l \in [\sigma_{i1}^l, \sigma_{i2}^l] \quad (2.24)$$

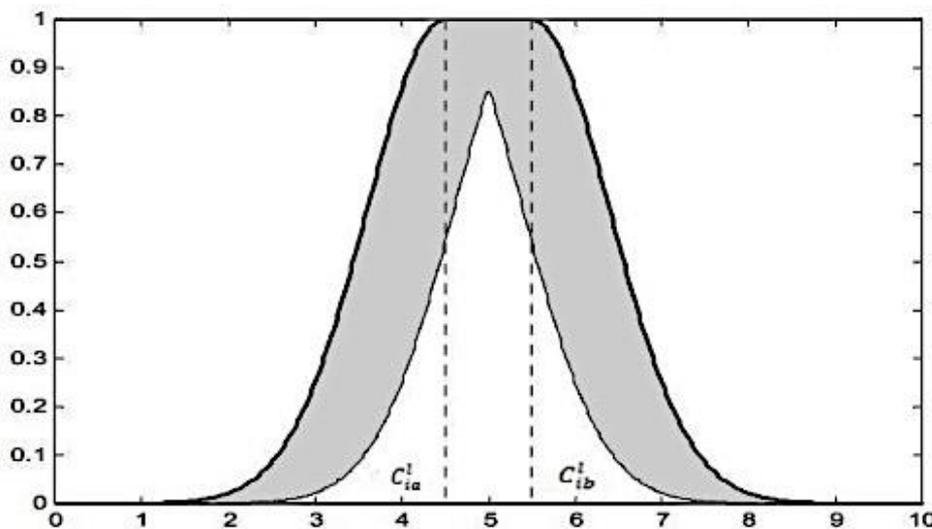
Maka persamaan fungsi keanggotaan pada \bar{f} ditunjukkan pada persamaan (2.24)

$$\bar{f} = \begin{cases} c_{i1}^l, \sigma_i^l; x_i & x_i < c_{i1}^l \\ 1, & c_{i1}^l \leq x_i \leq c_{i2}^l \\ c_{i2}^l, \sigma_i^l; x_i & x_i > c_{i2}^l \end{cases} \quad (2.25)$$

Sedangkan fungsi keanggotaan untuk interval *fuzzy* type-2 pada \underline{f} ditunjukkan pada persamaan (2.26)

$$\underline{f} = \begin{cases} (c_{i1}^l, \sigma_i^l; x_i), & x_i \leq \frac{c_{i1}^l + c_{i2}^l}{2} \\ (c_{i2}^l, \sigma_i^l; x_i), & x_i > \frac{c_{i1}^l + c_{i2}^l}{2} \end{cases} \quad (2.26)$$

dari kedua persamaan \bar{f} dan \underline{f} maka dapat digambarkan fungsi keanggotaan *Gaussian* seperti Gambar 2.16.



Gambar 2. 16.Fungsi keanggotaan Gaussian

- **Basis Kaidah**

Pada bagian ini berhubungan dengan sejumlah basis kaidah dalam himpunan *fuzzy* tipe-1 (Mendel, 2001). Basis kaidah dalam SLFT2 sama dengan

SLFT1, tetapi *antecedent* dan *consequent* direpresentasikan dengan himpunan *fuzzy* tipe-2 interval (Mendel 2001). Misalkan ada input sebanyak p berada pada $x_1 \in X_1, \dots, x_p \in X_p$ dan output c berada pada $y_1 \in Y_1, \dots, y_c \in Y_c$. Basis kaidah untuk SLFT2 dengan M kaidah dapat ditulis sebagai berikut (Mendel, 2001):

$$R^l = \{R_{M1}^1, R_{M2}^2, \dots, R_{MN}^M \quad (2.27)$$

Dimana i_{th} kaidah akan berbentuk sebagai berikut (Mendel, 2001):

$$R_{MIMO}^1 : \quad IF \ x_1 \text{ is } \tilde{F}_1^i, \dots, \text{ and } x_p \text{ is } \tilde{F}_p^i \text{ THEN } y_1 \text{ is } \tilde{G}_1^i, \dots, \text{ and } y_c \text{ is } \tilde{G}_c^i \\ i = 1, 2, \dots, M \quad (2.28)$$

Basis kaidah dari himpunan *fuzzy* tipe-2 direlasikan dari ruang input $x_1 \times \dots \times x_p$ ke ruang output y dari sistem inferensi *fuzzy* (Liang and Mendel, 2000a).

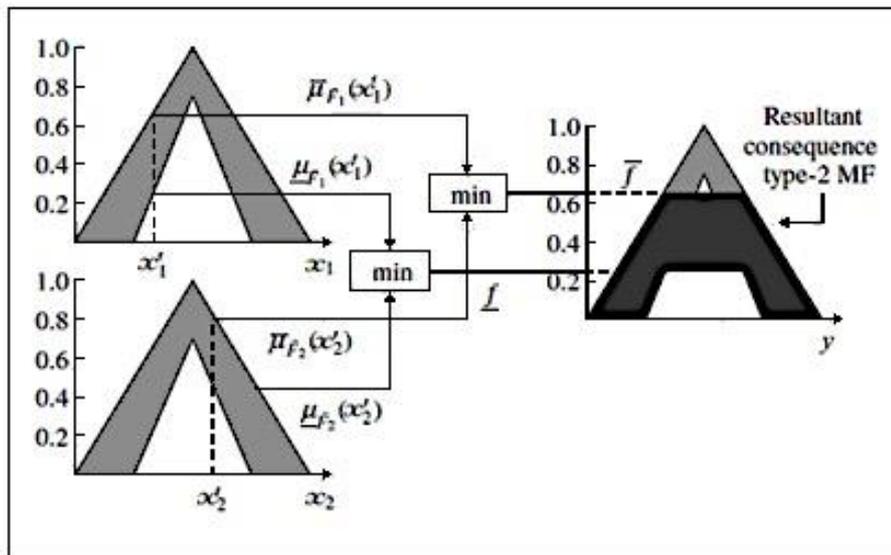
Basis kaidah SLF (tipe-1 atau tipe-2) memiliki batasan yaitu jika jumlah aturan meningkat secara eksponensial dengan jumlah variabel yang terlibat (Wang, 1999). Untuk sistem yang memiliki basis kaidah yang besar jumlahnya, perhitungan himpunan *fuzzy* yang direduksi tipe sebanding dengan jumlah aturan dan prosesnya akan memakan waktu lama untuk mengumpulkan setiap siklus untuk bertemu. Oleh karena itu membatasi aplikasi *real-time* dari SLFT2 (Wu dan Mendel, 2002). Untuk mengatasi dua masalah ini, beberapa peneliti menggunakan dekomposisi hierarki masalah kontrol dengan membagi ruang input untuk analisis dengan membagikannya di antara beberapa perilaku (Tunstel et al., 1997; Hagra, 2004).

Sistem Inferensi Fuzzy

Blok mesin inferensi *fuzzy* pada Gambar 2.17 memberikan pemetaan dari input SLFT2 ke output SLFT2 (Liang dan Mendel, 2000a). Dalam proses ini beberapa anteseden dalam basis kaidah dihubungkan menggunakan operasi irisan. Nilai keanggotaan dalam himpunan input digabungkan dengan nilai dalam himpunan output menggunakan komposisi *extended sup-star*.

Komposisi *extended sup-star* dapat diperoleh hanya dengan memperluas komposisi *extended sup-star* fungsi keanggotaan *fuzzy* tipe-1 dengan mengganti fungsi keanggotaan *fuzzy* tipe-1 dengan fungsi keanggotaan *fuzzy* tipe-2 interval. Operasi yang digunakan adalah operasi gabungan dan operasi irisan (Karnik dan Mendel, 2002). Komposisi *extended sup-star* adalah perhitungan untuk SLFT1, dan juga untuk SLFT2 interval. Basis aturan yang digunakan dapat dianggap sebagai sekelompok basis yang disebutkan sebelumnya yaitu memetakan dari multi input ke single output. Sehingga setiap kaidah memiliki p inputs $x^1 \in X_1, \dots, x_p \in X_p$ dan satu output $y_k \in Y_k$ (Mendel, 2001):

$$R_{MISO}^1 : \text{IF } x_1 \text{ is } \tilde{F}_1^i, \dots, \text{ and } x_p \text{ is } \tilde{F}_p^i \text{ THEN } y_k \text{ is } \tilde{G}_k^i \quad i=1,2, \dots, \quad (2.29)$$



Gambar 2. 17. Sistem inferensi SLFT2 (Mendel 2007)

R_{MISO}^1 digambarkan oleh fungsi keanggotaan $\mu_{R^i}(x, y_k) = \mu_{R^i}(x_1, \dots, x_p, y_k)$ dan sehubungan dengan penelitian Mendel (2001), maka $\mu_{R^i}(x, y_k)$ dapat ditulis kembali sebagai berikut;

$$\mu_{R^i}(x, y_k) = \mu_{\tilde{A}^i \rightarrow \tilde{G}_k^i}(x, y_k) = \mu_{\tilde{F}_1^i}(x_1) \dots \mu_{\tilde{F}_p^i}(x_p) \dots \mu_{\tilde{G}_k^i}(y_k) \quad (2.30)$$

Jika $\tilde{F}_1^i \times \dots \times \tilde{F}_p^i = \tilde{A}^i$ dalam persamaan (2.30) maka:

$$R^l : \tilde{F}_1^l \times \dots \times \tilde{F}_p^l \rightarrow \tilde{G}^l = \tilde{A}^l \rightarrow \tilde{G}^l \quad l = 1, \dots, M \quad (2.31)$$

R^l digambarkan sebagai fungsi keanggotaan MFs $\mu_{R^l}(x, y) = \mu_{R^l}(x_1, \dots, x_p, y)$ dimana,

$$\mu_{R^l}(x, y) = \mu_{\tilde{A}^l \rightarrow \tilde{G}^l}(x, y)$$

Secara umum, dimensi p dari input ke basis kaidah R^l diberikan dalam himpunan *fuzzy* tipe-1 \tilde{A}_x yang memiliki nilai,

$$\mu_{\tilde{A}_x}(x) = \mu_{\tilde{x}_1}(x_1) \prod \dots \prod \mu_{\tilde{x}_p}(x_p) = \prod_{i=1}^p \mu_{\tilde{x}_i}(x_i) \quad (2.32)$$

dimana $\tilde{X}_i (i = 1, \dots, p)$ adalah semua label dari himpunan *fuzzy* yang menggambarkan input. Setiap basis kaidah R^l menentukan sebuah aturan dalam himpunan *fuzzy* tipe-2 $\tilde{B}^l = \tilde{A}_x \circ R^l$ yaitu (Liang and Mendel, 2000a):

$$\mu_{\tilde{B}^l}(y) = \mu_{\tilde{A}_x \circ R^l} = \prod_{x \in X} [\mu_{\tilde{A}_x}(x) \prod \mu_{R^l}(x, y)] \quad y \in Y \quad l = 1, \dots, M \quad (2.33)$$

Kemudian hasil operasi pada persamaan (2.23) menghasilkan himpunan yang aktif, $\prod_{i=1}^p \mu_{\tilde{x}_i}(x'_i \equiv F^l(x'))$, dalam bentuk himpunan *fuzzy* tipe-1 interval yang dapat dinyatakan sebagai (Liang and Mendel, 2000a),

$$F^l(x') = [\underline{f}^l(x'), \overline{f}^l(x')] = [\underline{f}^l, \overline{f}^l] \quad (2.34)$$

Interval yang aktif pada persamaan (2.35) yaitu level bawah

$\underline{f}^l(x')$ dan level atas $\overline{f}^l(x')$, yang di hitung menggunakan minimum *t-norm*,

berikut:

Level bawah,

$$\underline{f}^l(x') = \min [\underline{\mu}_{\tilde{F}_1^l}(x'_1), \dots, \underline{\mu}_{\tilde{F}_p^l}(x'_p)] \quad (2.35)$$

Level atas,

$$\underline{f}^l(x') = \min[\underline{\mu}_{\tilde{F}_1^l}(x_1'), \dots, \underline{\mu}_{\tilde{F}_p^l}(x_p')] \quad (2.36)$$

Operasi dari $\underline{f}^l(x')$ dihitung menggunakan *t-norm* batas bawah dari FOU (\tilde{G})

dan $\overline{f}^l(x')$ dihitung menggunakan *t-norm* batas atas dari FOU (\tilde{G}).

Reduksi Himpunan Tipe-2 menjadi Himpunan Tipe-1

Reduksi tipe telah diusulkan oleh Karnik dan Mendel (1998) yaitu perpanjangan defuzzifikasi tipe-1, yang diperoleh dengan menerapkan Prinsip Perluasan (Zadeh, 1975) ke metode defuzzifikasi tertentu. Ini mewakili pemetaan SLFT2 interval menjadi SLFT1 interval (Karnik dan Mendel, 1998; Karnik et al., 1999). Himpunan yang direduksi tipenya ini kemudian di defuzzifikasi untuk mendapatkan output. Perhitungan himpunan yang direduksi tipenya dibagi menjadi dua tahap. Tahap pertama adalah perhitungan *centroid* dari set konsekuensi interval tipe-2 dari setiap aturan yang dilakukan sebelumnya. Tahap kedua terjadi setiap siklus kontrol untuk menghitung set tipe-reduksi yang kemudian defuzzifikasi untuk menghasilkan output yang sesuai (Hagras, 2004).

Ada banyak jenis metode reduksi tipe seperti *centroid*, *center-of-sets*, *center-of sums*, dan *height type-reduction*. Namun, dalam proses SLFT2 interval, terlepas dari metode reduksi tipe dan bagaimana inputnya dimodelkan (misalnya, sebagai singleton, himpunan *fuzzy* tipe-1, atau himpunan *fuzzy* tipe-2), himpunan tereduksi tipe selalu merupakan himpunan interval dan ditentukan oleh titik ujung kiri $y_l(x)$ dan titik ujung kanan $y_r(x)$. Umumnya metode yang sering digunakan mereduksi tipe untuk himpunan *fuzzy* tipe-2 adalah *centroid*, yang dapat dinyatakan sebagai (Karnik dan Mendel, 2001; Wu dan Mendel, 2001):

$$Y_{TR}(x) = [y_l(x), y_r(x)] \quad (2.37)$$

Penyelesaian secara iterative dari metode reduksi tipe adalah menggunakan algoritma *center-of-sets (cos)* (Karnik and Mendel, 1999). Tipe reduksi jenis ini membangkitkan himpunan *fuzzy* tipe-1 yang disebut $Y_{\cos}(x)$ (Karnik and Mendel, 1999):

$$y_{\cos}(x) = [y_l, y_r] = \int_{y^1 \in [y_l^1, y_r^1]} \dots \int_{y^M \in [y_l^M, y_r^M]} \int_{f^1 \in [\underline{f}^1, \bar{f}^1]} \dots \int_{f^M \in [\underline{f}^M, \bar{f}^M]} 1 / \frac{\sum_{i=1}^M f^i y^i}{\sum_{i=1}^M f^i} \quad (2.38)$$

Interval ditentukan dari nilai y_l dan y_r , yang berhubungan dengan *centroid* dari himpunan *fuzzy* tipe-2 konsekuen \tilde{G}^i . Persamaan dapat dinyatakan sebagai (Karnik *et al.*, 1999):

$$C_{\tilde{G}^i} = \int_{\theta_1 \in J_{y^1}} \dots \int_{\theta_N \in J_{y^N}} 1 / \frac{\sum_{i=1}^N y_i \theta_i}{\sum_{i=1}^N \theta_i} = [y_l^i, y_r^i] \quad (2.39)$$

Dari hasil operasi anteseden, maka basis kaidah yang aktif dalam himpunan *fuzzy* tipe-2 interval adalah (Mendel, 2000):

$$F^i(x) = [\underline{f}^i(x), \bar{f}^i(x)] \equiv [\underline{f}^i, \bar{f}^i], \quad i = 1, 2, \dots, M \quad (2.40)$$

dimana $\underline{f}^i(x)$ and $\bar{f}^i(x)$ menyatakan derajat kaidah yang aktif bagian bawah dan atas dari i th kaidah dan M adalah jumlah kaidah. Jika tipe reduksi *center of sets* digunakan, maka perhitungan *centroid* adalah sebagai berikut (Karnik and Mendel, 1999),

$$y_l(x) = y_l [\bar{f}^i(x), \dots, \bar{f}^{L^*(x)}(x), \dots, \underline{f}^{L^*(x)+1}(x), \dots, \underline{f}^M(x), y_l^1, \dots, y_l^M] \quad (2.41)$$

$$y_r(x) = y_r [\bar{f}^i(x), \dots, \bar{f}^{R^*(x)}(x), \dots, \underline{f}^{R^*(x)+1}(x), \dots, \underline{f}^M(x), y_r^1, \dots, y_r^M] \quad (2.42)$$

dimana y_L^i dan y_R^i ($i = 1, \dots, M$) adalah *centroid* kiri dan titik akhir dari sebelah kanan untuk semua konsekuen. $L^*(x)$ dan $R^*(x)$ adalah nilai yang sangat penting dan tergantung dengan nilai input. Setelah $L^*(x)$ dan $R^*(x)$ ditentukan, maka titik akhir reduksi tipe dapat ditentukan.

Algoritma Karnik-Mendel akan melakukan dua iterasi pada nilai $L^*(x)$ dan $R^*(x)$ (Karnik dan Mendel, 2000). Jika diasumsikan y_r^i dan y_l^i ($i = 1 \dots M$, dengan

M adalah jumlah kaidah) diurutkan secara menaik, algoritma karnik-mendel untuk menghitung y_r dapat dideskripsikan sebagai berikut (Wu, 2013):

1. Inisialisasi nilai f_n

Hitung semua nilai f_n menggunakan persamaan (2.24)

$$f^n = \frac{f^n + \bar{f}^n}{2}, n = 1, 2, 3, \dots, N \quad (2.42)$$

Hitung y' dengan menggunakan persamaan (2.25)

$$y' = \frac{\sum_{n=1}^N \bar{y}^n f^n}{\sum_{n=1}^N f^n} \quad (2.43)$$

Nilai dari y' berbeda-beda, tergantung pada reduksi tipe yang digunakan.

2. Menentukan $k(1 \leq k \leq N - 1)$ agar:

$$y_{k^r} \leq y \leq y_{k+1^r} \quad (2.44)$$

3. Tentukan:

$$f^n = \begin{cases} f^n, n \leq k \\ \bar{f}^n, n > k \end{cases} \quad (2.45)$$

Hitung y_r dengan menggunakan persamaan (2.28),

$$y_r = \frac{\sum_{n=1}^N \bar{y}^n f^n}{\sum_{n=1}^N f^n} \quad (2.46)$$

4. Periksa apakah nilai y' sama dengan y_r
 - a. Apabila keduanya bernilai sama, maka $y' = y_r$
 - b. Apabila tidak sama, ubah nilai $y' = y_r$ dan ulangi langkah 2 sampai y_r ditemukan.

Sedangkan langkah-langkah untuk menentukan y_l adalah sebagai berikut:

1. Inisialisasi nilai f_n

Hitung semua nilai f_n dengan (2.29)

$$f^n = \frac{f^n + \bar{f}^n}{2}, n = 1, 2, 3, \dots, N \quad (2.47)$$

Dan hitung y' dengan (2.30)

$$y' = \frac{\sum_{n=1}^N \underline{y}^n f^n}{\sum_{n=1}^N f^n} \quad (2.48)$$

2. Menentukan $k(1 \leq k \leq N - 1)$ agar:

$$y_{k^l} \leq y \leq y_{k+1^l} \quad (2.49)$$

3. Tentukan:

$$f^n = \begin{cases} \bar{f}^n, & n \leq k \\ \underline{f}^n, & n > k \end{cases} \quad (2.50)$$

Dan menghitung y_l dengan (2.33)

$$y_l = \frac{\sum_{n=1}^N \underline{y}^n f^n}{\sum_{n=1}^N f^n} \quad (2.51)$$

4. Periksa apakah nilai y' sama dengan y_l
- Apabila keduanya bernilai sama, maka $y' = y_l$
 - Apabila tidak sama, ubah nilai $y' = y_l$ dan ulangi langkah 2 sampai y_l ditemukan.

Defuzzifikasi

Setelah tahapan reduksi tipe, maka yang terakhir adalah defuzzifikasi (Liang and Mendel, 2000). Tahapan defuzzifikasi merupakan proses yang terakhir dari algoritma logika *fuzzy*, untuk menghasilkan nilai nyata sebagai hasil keputusan. Proses ini dilakukan dengan menghitung nilai tegas dari suatu himpunan tereduksi dengan merata-rata nilai paling kiri y_l dan nilai paling kanan y_r , atau dapat dinyatakan sebagai:

$$y = \frac{y_l + y_r}{2} \quad (2.52)$$

2.5.3. Particle Swarm Optimization

Selain tugas menghindari halangan dan berjalan dalam formasi yang telah ditentukan, sekelompok robot memiliki tugas penting lainnya yaitu untuk

mencapai target. Beberapa sistem kontrol formasi menggunakan metode optimisasi rute dengan metode kecerdasan *swarm* seperti, *particle swarm optimization*, algoritma semut, algoritma lebah (Yogeswaran and Ponnambalam, 2006; Sharkey, 2006; Pugh, and Martinoli, 2007). Sayangnya metode-metode tersebut hanya optimal sebagian, dan sistem robotika dapat terjebak dalam kondisi tertentu dan tidak dapat melanjutkan proses pencarian target. Khususnya untuk sistem robot *leader-follower*, kondisi tersebut dapat menyebabkan robot *leader-follower* diam pada daerah tertentu dan gagal mencapai target. Dari berbagai hal tersebut, diperlukan untuk mengontrol robot *leader-follower* dengan menggunakan metode yang dapat mengatasi ketidakpastian dan mampu mengoptimisasi rute pergerakan posisi dan orientasi saat menuju target.

Metode *Particle Swarm Optimization* (PSO) merupakan satu algoritma yang sangat sederhana dan mudah untuk diimplementasikan. Dengan pemilihan parameter-parameter optimisasi yang sesuai dapat membuat PSO menghasilkan kinerja yang baik. Antara lain dengan mengatur fungsi inersia dalam mencari bobot yang sesuai. Pada sub-bab berikut ini akan dijelaskan mengenai metode tersebut. Metode PSO didasarkan pada algoritma *behaviorally inspired* sebagai alternatif dari algoritma genetika, yang sering disebut *evolution-based procedures*. Algoritma PSO merupakan sebuah algoritma yang mampu meniru kemampuan hewan yang hidup secara berkelompok, setiap individu pada PSO akan dianggap sebuah partikel. Algoritma ini pertama kali diperkenalkan oleh (Eberhart and Kennedy 1995; Kennedy, 2001). Algoritma ini merupakan bagian dari algoritma komputasi evolusi antara lain algoritma genetika, dan pemrograman genetik.

Dalam konteks optimasi multi-variabel, kawanan diasumsikan mempunyai ukuran tertentu atau tetap dengan setiap partikel posisi awalnya terletak di suatu lokasi yang acak dalam ruang multidimensi (Eberhart and Kennedy 1995; Kennedy, 2001). Setiap partikel diasumsikan memiliki dua karakteristik: posisi dan kecepatan. Setiap partikel bergerak dalam ruang (*space*) tertentu dan mengingat posisi terbaik yang pernah dilalui atau ditemukan terhadap sumber makanan atau nilai fungsi objektif. Setiap partikel menyampaikan informasi atau posisi bagusnya kepada partikel yang

lain dan menyesuaikan posisi dan kecepatan masing-masing berdasarkan informasi yang diterima mengenai posisi yang bagus tersebut (Eberhart, and Shi, 2000; Eberhart, and Shi, 2001; Eberhart and Kennedy 1995; Kennnedy, 2001).

PSO terdiri dari konsep yang sangat sederhana, dan dapat diimplementasikan hanya dalam beberapa baris kode komputer (program), algoritma ini hanya membutuhkan operator matematika secara *primitive* dan secara komputasional tidak mahal baik dari segi persyaratan memori maupun kecepatan (Eberhart, and Shi, 2001; Nurmaini *et al.*, 2014). Setiap partikel diperlakukan seperti sebuah titik pada suatu dimensi ruang tertentu. Kemudian terdapat dua faktor yang memberikan karakter terhadap status partikel pada ruang pencarian yaitu posisi partikel dan kecepatan partikel (Kennedy and Eberhart, 1995). Berikut ini merupakan formulasi matematika yang menggambarkan posisi dan kecepatan partikel pada suatu dimensi ruang tertentu:

$$X_i(t) = x_{i1}(t), x_{i2}(t), \dots, x_{iN}(t)$$

$$V_i(t) = v_{i1}(t), v_{i2}(t), \dots, v_{iN}(t)$$

dimana X = posisi partikel; V = kecepatan partikel; i = indeks partikel;

t = iterasi ke- t ; N = ukuran dimensi ruang

Terdapat dua macam jenis algoritma PSO antara lain standar PSO dan PSO dengan perbaikan yang akan dijelaskan pada sub bab berikut ini.

2.5.3.1. Standar *Particle Swarm Optimization*

Setiap individu dalam kelompok partikel terdiri dari tiga vektor berdimensi D , di mana D adalah dimensi ruang pencarian, dimana posisi saat ini x_i^k , posisi terbaik sebelumnya p_i^k , dan kecepatan v_i^k (Eberhart, and Shi, 2000). Posisi saat ini x_i^k dapat dianggap sebagai seperangkat koordinat yang menggambarkan titik di dalam suatu ruang. Pada setiap iterasi algoritma, posisi saat ini dievaluasi sebagai solusi masalah. Jika posisi itu lebih baik dari yang telah ditemukan sejauh ini, maka koordinatnya tersimpan di vektor kedua, p_i^k . Nilai

hasil fungsi terbaik sejauh ini tersimpan dalam variabel yang bisa disebut $pbest_i^n$ (untuk "best sebelumnya"), untuk perbandingan pada iterasi selanjutnya. Tujuannya, adalah untuk terus mencari posisi dan pembaharuan p_i^k dan $pbesti$ yang lebih baik. Poin baru dipilih dengan menambahkan koordinat v_i^k ke x_i^k , dan algoritma beroperasi dengan menyesuaikan v_i^k , yang dapat dilihat secara efektif sebagai ukuran langkah (Eberhart and Shi, 2000; Eberhart and Shi, 2001; Eberhart and Kennedy 1995; Kennneddy, 2001).

Permasalahan jika tidak berlakunya nilai *particle best* karena perubahan lingkungan, dapat diselesaikan dengan membuat posisi terakhir masing-masing partikel menjadi nilai *particle best* dari partikel tersebut, seperti yang ditunjukkan pada persamaan (2.53).

$$\forall p_i^k, x_i^k : \text{lingkungan berubah} \rightarrow p_i^k = x_i^k \quad (2.53)$$

Ketika PSO menjadi konvergen, maka nilai faktor sosial akan mendekati nol. Sementara itu, berubahnya posisi nilai *particle best* menjadi posisi terakhir *agent* akan membuat nilai seperti yang ditunjukkan pada persamaan (2.54):

$$p_i^k - x_i^k = 0 \quad (2.54)$$

Persamaan PSO standar dapat dilihat pada persamaan (2.55) dan persamaan (2.56) berikut;

$$v_i^{k+1} = v_i^k + c_1 * rand1 * (p_i^k - x_i^k) + c_2 * rand2 * (p_g^k - x_i^k), \quad (2.55)$$

$$x_i^{k+1} = x_i^k + (v_i^{k+1}); i = 1, 2, \dots, n, \quad (2.56)$$

Berdasarkan persamaan (2.55), maka dapat disederhanakan menjadi:

$$v_i^{k+1} = \underbrace{v_i^k + c_1 * rand * (0)}_{\text{Individu}} + \underbrace{c_2 * rand * (p_g^k - x_i^k)}_{\text{Sosial}} \quad (2.57)$$

dimana:

i	= partikel ke i ,
k	= iterasi PSO ke k
c_1	= tingkat akselerasi untuk faktor kognitif, bernilai antara 0 - 1
c_2	= tingkat akselerasi untuk faktor sosial, bernilai antara 0- 1
V_i^k	= kecepatan partikel ke i pada iterasi ke k
x_i^k	= posisi partikel ke i pada iterasi ke k
p_i^k	= <i>local best</i> dari partikel ke i pada iterasi ke k
p_g^k	= <i>global best</i> pada iterasi ke k
rand	= fungsi acak, bernilai antara 0 – 1

2.5.3.2. Strategi Perubahan Bobot Inersia Untuk *Particle Swarm Optimization*

Bobot inersia memainkan peran penting dalam proses memberikan keseimbangan antara proses eksplorasi dan eksploitasi. Bobot Inersia menentukan tingkat kontribusi dari kecepatan partikel sebelumnya terhadap kecepatannya pada langkah waktu saat ini. Standar PSO, yang diusulkan oleh Eberhart dan Kennedy pada tahun 1995, tidak memiliki bobot inersia (Eberhart, and Kennedy, 1995). Pada tahun 1998, pertama kali Shi dan Eberhart mempresentasikan konsep bobot inersia dengan memperkenalkan bobot Inersia Konstan (Eberhart and Shi, 2000). Mereka menyatakan bahwa bobot inersia yang besar memudahkan pencarian global sementara bobot inersia yang kecil memfasilitasi pencarian lokal.

Penyesuaian secara dinamis dari bobot inersia telah diusulkan oleh banyak peneliti yang dapat meningkatkan kemampuan PSO. Eberhart dan Shi (2001) mengusulkan strategi bobot inersia random dan secara eksperimental menemukan bahwa strategi ini meningkatkan konvergensi PSO dalam iterasi awal algoritma. Strategi Penurunan Linier meningkatkan efisiensi dan kinerja PSO (Eberhart dan Shi, 2001). Ditemukan secara eksperimental bahwa *Inersia Weight* dari 0.9 hingga 0.4 memberikan hasil yang sangat baik. Terlepas dari kemampuannya untuk optimal secara global, karena partikel dapat masuk ke kondisi lokal optimal.

Al Hasan et al. (2007) memperkenalkan teknik *Particle Swarm* yang dioptimalkan yang menggunakan Simulasi *Annealing* untuk mengoptimalkan

bobot inersia. Teknik yang diusulkan menghasilkan jauh lebih baik dalam hal kecepatan konvergensi. Chen et al. (2006) menyajikan dua strategi bobot inersia eksponen alami yang didasarkan pada tujuan dasar penurunan bobot inersia. Secara eksperimental, dua strategi ini lebih cepat selama tahap awal proses pencarian dan memberikan hasil yang lebih baik untuk sebagian besar masalah optimasi berkelanjutan. *Chaotic Inersia Weight* telah diusulkan oleh Feng et al. (2007). Teknik ini menghasilkan kinerja yang baik, dengan tahap pencarian kasar dan tahap pencarian per menit secara bergantian dalam semua proses evolusi.

Malik et al. (2007) menggunakan kombinasi fungsi *Sigmoid* dan fungsi Linier untuk meningkatkan bobot inersia. Mereka menemukan bahwa fungsi Sigmoid menghasilkan fungsi *fitness* yang minimum dan fungsi linear dapat meningkatkan bobot inersia sehingga kemampuan konvergensi menjadi lebih cepat. Hal tersebut menghasilkan peningkatan kecepatan yang besar dalam kemampuan konvergensi dan gerakan agresif yang menyempit menuju daerah solusi.

Gao et al. (2008) mengusulkan algoritma PSO baru yang menggabungkan *Logaritma Decreasing Inertia Weight* dengan operator mutasi *Chaos*. *Logaritma Decreasing Inertia Weight* dapat meningkatkan kecepatan konvergensi, sementara mutasi *Chaos* dapat meningkatkan kemampuan untuk melompat keluar dari kondisi lokal optimal. Untuk mengatasi konvergensi prematur dan kejadian osilasi periode selanjutnya dari PSO standar, Bobot Inersia Pengurangan Eksponen dan mutasi stokastik untuk menghasilkan PSO yang lebih baik yang menggunakan Bobot Inersia Pengurangan Eksponen bersama dengan mutasi stokastik *piecewise* untuk partikel optimal global saat ini dan selama waktu berjalan, sehingga memperkuat kemampuan solusi optimal parsial (Gao, An and Liu, 2008).

Algoritma optimasi partikel Gregarious PSO (GPSO) diusulkan untuk meningkatkan keragaman dan kemampuan pencarian lokal populasi (Kessentini, and Barchiesi, 2015). Metode ini memiliki komputasi yang sederhana digunakan untuk membandingkan kinerja partikel yang berbeda. Operator mutasi berdasarkan evolusi diferensial (DE) dirancang dan digunakan untuk memperbarui posisi terbaik partikel dan posisi populasi global terbaik. Tujuannya adalah untuk meningkatkan kemampuan pencarian lokal dan probabilitas untuk

menemukan optimal global. Strategi regenerasi digunakan untuk memperbarui partikel yang mengalami stagnasi sehingga semakin meningkatkan keragaman GPSO.

2.5.3.3. Particle Swarm Optimization Dengan Perbaikan

Untuk melakukan perbaikan kinerja dari algoritma PSO standar maka pada bagian ini akan diperkenalkan sebuah nilai yang disebut bobot inersia (w). Dalam implementasinya, ditemukan bahwa kecepatan partikel dalam persamaan standar PSO yang di-*update* terlalu cepat dan nilai minimum fungsi tujuan sering terlewati. Sehingga perlu adanya revisi atau perbaikan terhadap algoritma PSO standar. Perbaikan itu berupa penambahan suatu bobot inersia w untuk mengurangi kecepatan. Biasanya nilai w dibuat sedemikian hingga semakin meningkat iterasi yang dilalui, semakin mengecil kecepatan partikel. Nilai ini bervariasi secara linier dalam rentang 0.9 hingga 0.4. Secara matematis perbaikan PSO dengan menambahkan nilai bobot w pada persamaan (2.55) yang dapat dituliskan kembali sebagai berikut,

$$v_i^{k+1} = w_i^k v_i^k + c_1 r_1 (p_i^k - x_i^k) + c_2 r_2 (p_g^k - x_i^k), \quad (2.58)$$

$$x_i^{k+1} = x_i^k + (v_i^{k+1}); i = 1, 2, \dots, n, \quad (2.59)$$

Bobot inersia ini diusulkan oleh (Eberhart and Kennedy 1995). Untuk meredam kecepatan selama iterasi, yang memungkinkan partikel konvergen ke titik target secara lebih akurat dan efisien dibandingkan dengan algoritma aslinya. Nilai bobot inersia yang tinggi menambah porsi pencarian global (*global exploration*), sedangkan nilai yang rendah lebih menekankan pencarian lokal (*local search*). Untuk tidak terlalu menitik-beratkan pada salah satu bagian dan tetap mencari area pencarian yang baru dalam ruang berdimensi tertentu, maka perlu dicari nilai bobot inersia w yang secaraimbang menjaga pencarian global dan lokal.

BAB III METODOLOGI PENELITIAN

Metodologi yang digunakan pada bab ini adalah berfokus pada merancang sistem kontrol formasi pada robot berkelompok menggunakan metode *hybrid* Sistem Logika *Fuzzy* Tipe-2 dan dinamik *Particle Swarm Optimization* (SLFT2-DPSO). Setiap tahap perancangan akan dibahas secara detil dalam bab ini, yang meliputi: perancangan model sistem robotika, perancangan perangkat lunak sistem kontrol formasi, pengujian, validasi dan verifikasi serta analisis, penarikan kesimpulan dan saran untuk penelitian masa depan. Pelaksanaan pada masing-masing tahapan merupakan satu kesatuan dalam bentuk kerangka penelitian yang akan dijelaskan dalam sub bab berikut ini.

3.1. Kerangka Penelitian

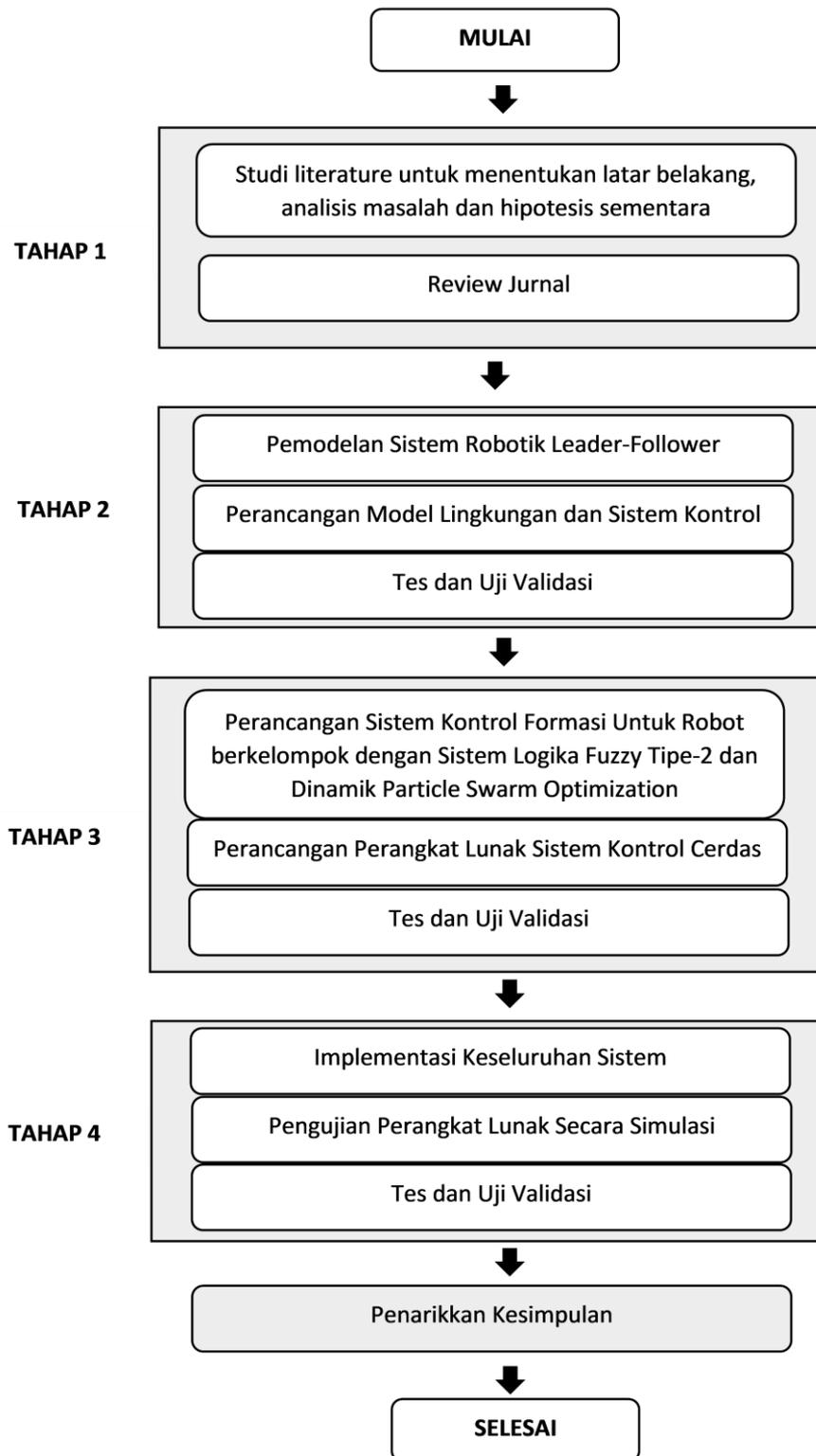
Pengembangan rancangan sistem kontrol formasi pada robot *swarm leader-follower* dilakukan dengan menggunakan strategi berbasis perilaku. Proses keseluruhan terdiri dari empat fase utama, yaitu tahapan awal penelitian yaitu studi pustaka, pengembangan model matematis sistem robotika, perancangan sistem kontrol formasi, dan pengujian dan validasi sistem yang diusulkan. Keseluruhan tahapan kegiatan penelitian diuraikan pada Gambar 3.1.

Tahap 1: Pada tahap awal penelitian dilakukan studi pustaka yang berhubungan tentang analisis isu utama yang terlibat dalam merancang sistem kontrol formasi pada robot *swarm* menggunakan metode *hybrid* sistem logika *fuzzy* tipe-2 (SLFT2) dan dinamik *Particle Swarm Optimization* (DPSO). Studi dilakukan untuk melihat dan mengidentifikasi suatu persyaratan untuk mengembangkan suatu metode baru dalam menyelesaikan masalah dalam pengendalian formasi pada robot *swarm leader-follower*.

Tahap 2: Pada tahap ini dilakukan permodelan sistem multi-robot dengan mempertimbangkan kondisi dan persyaratan yang dibutuhkan dalam robot *swarm*. Meliputi permodelan kontrol formasi yang sederhana, dengan kinerja ketahanan pada individu robot dan robot *swarm*, mampu beradaptasi dengan ketidakpastian dan perubahan lingkungan. Untuk mencapai keberhasilan dari langkah ini, semua proses (termasuk sistem sensor, penggerak, dan lingkungan pengujian) harus dimodelkan dan dikoordinasikan secara akurat untuk mencapai tujuan yang diharapkan. Semua parameter dirancang dalam bentuk model simulasi.

Tahap 3 : Pada tahap ini dilakukan pengembangan algoritma pengendali kontrol formasi dengan kombinasi SLFT2 dan DPSO. Pengembangan perangkat lunak bagian dalam penelitian ini berkonsentrasi pada tiga tahap. Tahap pertama adalah merancang algoritma SLFT2 untuk pengendali formasi robot *swarm*. Tahap kedua merancang algoritma optimisasi menggunakan metode DPSO. Tahap ketiga mengombinasikan metode SLFT2 dan DPSO untuk sistem kontrol formasi.

Tahap 4 : Validasi dan pengujian pada tugas kontrol formasi. Pengujian yang dilakukan, dibagi dalam beberapa tahap untuk melihat kinerja sistem pengendali SLFT2 dan DPSO yang diusulkan dimulai dari lingkungan sederhana, lingkungan kompleks tanpa gangguan dan lingkungan kompleks dengan gangguan. Metode yang digunakan diuji dan dievaluasi dengan menggunakan metode lainnya sebagai pembanding sistem yang diusulkan dalam hal ketahanan, keandalan dan penggunaan memori. Seluruh proses dalam metodologi penelitian yang diusulkan ditunjukkan pada Gambar 3.1 dalam bentuk kerangka kerja penelitian berikut ini.

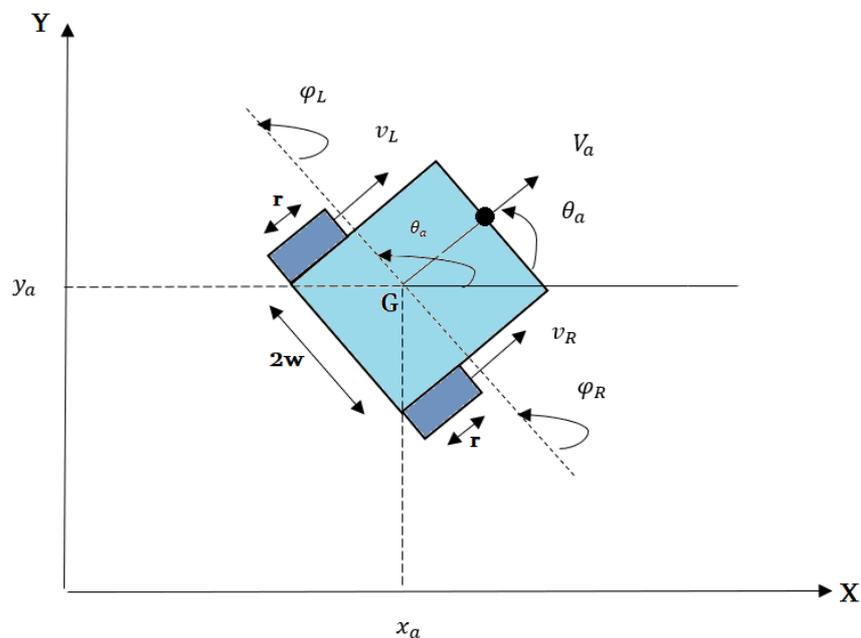


Gambar 3. 1. Kerangka Kerja Penelitian

3.2. Pemodelan Sistem Robotik

3.2.1. Robotik Individu

Pada penelitian ini menggunakan robot non-holonmik berpengerak *diferensial* (*Differential Drive Mobile Robot*) (DDMR). Robot jenis ini memiliki konstruksi dengan ciri yang khas yaitu, mempunyai 2 aktuator, 2 roda untuk menggerakkan keseluruhan badan robot tersebut dan 1 roda bebas. Roda penggerak kiri dan kanan dikemudikan secara terpisah dan diasumsikan hanya bergerak pada bidang horizontal seperti Gambar 3.2 berikut ini.



Gambar 3. 2. Non-holonmik *mobile robot*

Berdasarkan Gambar 3.2, robot diasumsikan berada dalam koordinat Cartesian xy . Parameter-parameter dalam Gambar 3.2 adalah sebagai berikut:

- $2w$: Lebar yang diukur dari garis tengah roda ke roda
- r : Diameter roda (roda kiri dan kanan adalah sama dan sebangun)
- θ_a : Sudut arah hadap robot
- $\dot{\theta}_a$: Turunan sudut arah hadap robot
- v_a : Kecepatan linier robot
- v_R : Kecepatan linier roda kanan
- v_L : Kecepatan linier roda kiri

- ϕ_R : Sudut pergerakan roda kanan
 ϕ_L : Sudut pergerakan roda kiri

Dengan mengabaikan analisis dari roda bebas, konfigurasi dari robot bergerak otonom dapat dideskripsikan ke dalam tiga variabel umum q yang menggambarkan posisi robot. Koordinat umum q dan masukan kendali u dapat didefinisikan sebagai berikut:

$$\begin{aligned} q &= [x_a, y_a, \theta_a]^T \\ u &= [\dot{\phi}_R, \dot{\phi}_L]^T \end{aligned} \quad (3.1)$$

Bentuk umum persamaan kinematika untuk DDMR dapat dinyatakan dalam persamaan sebagai berikut:

$$\begin{pmatrix} \dot{x}_a \\ \dot{y}_a \\ \dot{\theta}_a \end{pmatrix} = T_{NH}(q)u \quad \text{atau} \quad \begin{pmatrix} \dot{x}_a \\ \dot{y}_a \\ \dot{\theta}_a \end{pmatrix} = T_{NH} \begin{pmatrix} \dot{\phi}_R \\ \dot{\phi}_L \end{pmatrix} \quad (3.2)$$

Dimana $[\dot{x}_a, \dot{y}_a, \dot{\theta}_a]^T$ merupakan turunan pertama dari persamaan posisi robot. T_{NH} adalah transformasi nonholonomik robot dan $\dot{\phi}_R$ dan $\dot{\phi}_L$ merupakan kecepatan angular roda kiri dan roda kanan.

Pada Gambar 3.2 diasumsikan o adalah koordinat Cartesian xy . Kecepatan roda kanan (v_R) dan roda kiri (v_L) robot didapatkan dari persamaan kecepatan v_a ditambahkan dengan setengah dari lebar robot dan perpindahan putaran sudut roda $\dot{\theta}_a$. Sehingga kecepatan roda kiri dan kanan didefinisikan dalam persamaan (3.3) dan (3.4) berikut:

$$v_R = v_a + b\dot{\theta}_a \quad (3.3)$$

$$v_L = v_a - b\dot{\theta}_a \quad (3.4)$$

Dengan menyubstitusikan Persamaan (3.3) dan Persamaan (3.4) maka didapatkan persamaan kecepatan linier robot v_a dan persamaan angular robot $\dot{\theta}_a$ sebagai berikut:

$$v_R + v_L = 2v_a$$

$$v_a = \frac{1}{2}(v_R + v_L) \quad (3.5)$$

$$v_R - v_L = 2b\dot{\theta}_a$$

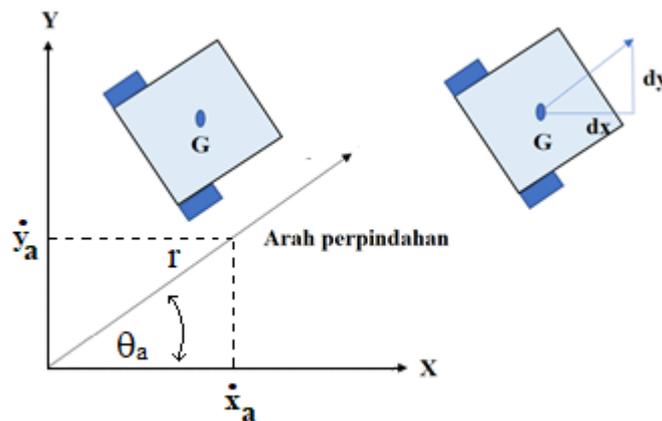
$$\dot{\theta}_a = \frac{v_R - v_L}{2b} \quad (3.6)$$

3.2.1.1. *Non-holonomic Constraints*

Robot bergerak otonom mempunyai sifat-sifat non-holonomik, yang berarti robot ini mempunyai pergerakan yang dibatasi. Karakteristik *non-holonomic constraint* yang harus dipenuhi yaitu:

$$A(q)\dot{q} = 0 \quad (3.7)$$

Persamaan (3.7) dapat direpresentasikan seperti Gambar 3.3 dan menjelaskan sifat *non-holonomic constraints*.



Gambar 3. 3. Perpindahan *mobile robot*

Dari gambar 3.2. dan gambar 3.3. kecepatan sudut roda kanan dan roda kiri adalah sebagai berikut :

$$\dot{\theta}_R = \frac{1}{2\pi R} V_R \quad (3.8)$$

$$\dot{\theta}_L = \frac{1}{2\pi R} V_L \quad (3.9)$$

Dalam penelitian ini model robot diasumsikan berjalan tanpa slip antara dua roda, sebagai acuan analisa komponen x dan y dapat dinotasikan dalam dua persamaan batasan non-holonomik yang dapat dinyatakan sebagai berikut :

$$\dot{x}_a = V_a \cos \theta_a$$

$$\dot{y}_a = V_a \sin \theta_a$$

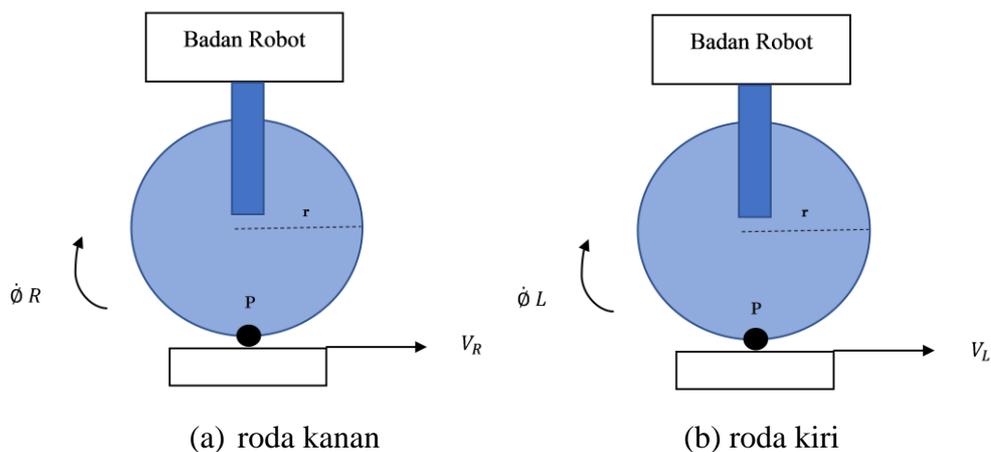
Dengan :

$$V_a = r \dot{\theta}_a = \frac{\dot{x}_a}{\cos \theta_a} = \frac{\dot{y}_a}{\sin \theta_a}$$

Maka :

$$\dot{x}_a \sin \theta_a - \dot{y}_a \cos \theta_a = 0 \quad (3.10)$$

Selanjutnya, dikarenakan tidak ada slip antara roda terhadap lantai, maka persamaan batasan *pure rolling* untuk masing-masing roda kanan dan kiri dapat ditunjukkan oleh Gambar 3.4 berikut ini,



Gambar 3. 4. Kecepatan roda robot berdasarkan porosnya

Roda dan lantai diasumsikan tidak slip, pertemuan antara roda dan lantai dianggap titik *P* terlihat pada Gambar 3.4. Sehingga kecepatan roda kiri dan roda kanan didapatkan dari perubahan putaran sudut roda kiri dan kanan berturut-turut

$\dot{\phi}_R$ dan $\dot{\phi}_L$ yang dikalikan dengan jari-jari r roda, yang dapat dilihat pada Persamaan (3.11) dan Persamaan (3.12) berikut ini:

$$v_R = r\dot{\phi}_R \quad (3.11)$$

$$v_L = r\dot{\phi}_L \quad (3.12)$$

Dengan menggunakan persamaan kecepatan roda (3.3) dan (3.4), dan mensubstitusikan Persamaan (3.11) dan Persamaan (3.12) pada persamaan kecepatan roda tersebut. Sehingga akan didapatkan persamaan konstrain pada roda kanan Persamaan (3.13) dan roda kiri Persamaan (3.14).

$$v_a = \dot{x}_a \cos \theta + \dot{y}_a \sin \theta$$

Dari persamaan (3.3) didapat :

$$v_R = \dot{x}_a \cos \theta + \dot{y}_a \sin \theta + b\dot{\theta}_a$$

Dari persamaan (3.9) didapat :

$$r\dot{\phi}_R = \dot{x}_a \cos \theta + \dot{y}_a \sin \theta + b\dot{\theta}_a$$

$$\dot{x}_a \cos \theta + \dot{y}_a \sin \theta + b\dot{\theta}_a - r\dot{\phi}_R = 0 \quad (3.13)$$

$$v_a = \dot{x}_a \cos \theta + \dot{y}_a \sin \theta$$

Dari persamaan (3.4) didapat :

$$v_L = \dot{x}_a \cos \theta + \dot{y}_a \sin \theta - b\dot{\theta}_a$$

Dari persamaan (3.10) didapat :

$$r\dot{\phi}_L = \dot{x}_a \cos \theta + \dot{y}_a \sin \theta - b\dot{\theta}_a$$

$$\dot{x}_a \cos \theta + \dot{y}_a \sin \theta - b\dot{\theta}_a - r\dot{\phi}_L = 0 \quad (3.14)$$

Dari Persamaan konstrain (3.10), (3.13) dan Persamaan (3.14) tersebut dapat ditulis kembali menjadi bentuk matriks berikut:

$$A_q = \begin{bmatrix} \sin \theta & -\cos \theta & 0 & 0 & 0 \\ \cos \theta & \sin \theta & b & -r & 0 \\ \cos \theta & \sin \theta & -b & 0 & -r \end{bmatrix} \quad (3.15)$$

3.2.1.2 Persamaan *Steering System*

Persamaan *steering system* (sistem kemudi) digunakan untuk mengendalikan pergerakan dari robot bergerak otonom. Persamaan sistem kemudi robot bergerak otonom mempunyai batasan kinematika, yaitu roda diasumsikan tidak slip, maka posisi awal pergerakan roda ditunjukkan pada Gambar 3.5. Dapat terlihat bahwa posisi awal roda terhadap bidang Cartesian xy , diperoleh persamaan posisi awal roda robot yaitu Persamaan (3.16), Persamaan (3.17), dan Persamaan (3.18).

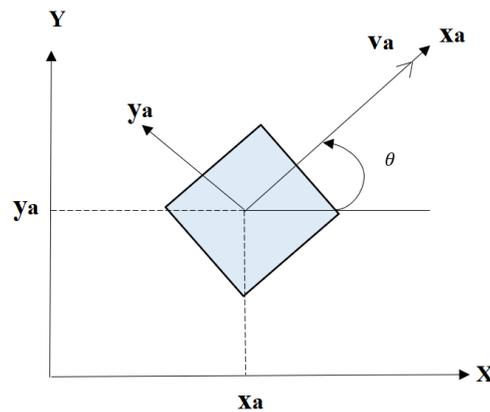
$$q = [x_a, y_a, \theta_a]^T \quad (3.16)$$

dimana,

$$\begin{aligned} x_a &= v_a \sin \theta \\ y_a &= v_a \cos \theta \\ \theta_a &= v_a \end{aligned} \quad (3.17)$$

Turunan pertama dari Persamaan (3.17) sebagai berikut:

$$\begin{aligned} \dot{x}_a &= v_a \cos \theta \\ \dot{y}_a &= v_a \sin \theta \\ \dot{\theta}_a &= \omega \end{aligned} \quad (3.18)$$



Gambar 3. 5. Struktur kinematik dari perputaran roda pada bidang horizontal

Transformasi non-holonmik didapatkan dari menyubstitusikan Persamaan (3.17) ke dalam Persamaan (3.18) lalu menyubstitusikan kembali Persamaan

(3.11) dan Persamaan (3.12). Maka transformasi non-holonmik ditunjukkan pada Persamaan (3.19), Persamaan (3.20) dan Persamaan (3.21) sebagai berikut:

$$\begin{aligned}\dot{x}_a &= v_a \cos \theta \\ \dot{x}_a &= \frac{v_R + v_L}{2} \cos \theta \\ \dot{x}_a &= \frac{r\phi_R \cos \theta + r\phi_L \cos \theta}{2}\end{aligned}\quad (3.19)$$

$$\begin{aligned}\dot{y}_a &= v_a \sin \theta \\ \dot{y}_a &= \frac{v_R + v_L}{2} \sin \theta \\ \dot{y}_a &= \frac{r\phi_R \sin \theta + r\phi_L \sin \theta}{2}\end{aligned}\quad (3.20)$$

$$\begin{aligned}\dot{\theta}_a &= \frac{v_R - v_L}{2b} \\ \dot{\theta}_a &= \frac{r\phi_R - r\phi_L}{2b}\end{aligned}\quad (3.21)$$

Persamaan transformasi non-holonmik (3.19), Persamaan (3.20), dan Persamaan (3.21) ini dapat dituliskan dalam bentuk matriks menjadi

$$T_{NH}(q) = \begin{bmatrix} \frac{r}{2} \cos \theta & \frac{r}{2} \cos \theta \\ \frac{r}{2} \sin \theta & \frac{r}{2} \sin \theta \\ \frac{r}{2b} & -\frac{r}{2b} \end{bmatrix}\quad (3.22)$$

Selanjutnya, setelah persamaan matriks transformasi non-holonmik diperoleh. Persamaan kinematika DDMR mengasumsikan kecepatan sudut u sebagai input robot. Dimana:

$$\begin{aligned}\dot{\phi}_R &= u_1 \\ \dot{\phi}_L &= u_2 \\ \dot{\theta}_a &= \omega\end{aligned}\quad (3.23)$$

Sehingga model kinematik dari gerakan DDMR sebagai berikut:

$$\dot{q} = T_{NH}(q) u$$

$$\begin{bmatrix} \dot{x}_a \\ \dot{y}_a \\ \dot{\theta}_a \end{bmatrix} = \begin{bmatrix} \frac{r}{2} \cos \theta & \frac{r}{2} \cos \theta \\ \frac{r}{2} \sin \theta & \frac{r}{2} \sin \theta \\ \frac{r}{2b} & -\frac{r}{2b} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (3.24)$$

Untuk memperoleh persamaan *input* robot, perlu diingat bahwa roda mempunyai batasan *pure rolling*. Dari persamaan kecepatan untuk masing-masing roda yaitu Persamaan (3.3) dan Persamaan (3.4) kemudian menyubstitusikan persamaan batasan *pure rolling* Persamaan (3.11) dan Persamaan (3.12) ke dalam persamaan kecepatan untuk masing-masing roda. Maka dapat diperoleh persamaan *input* robot berikut:

$$\begin{aligned} v_R &= v_a + b\dot{\theta}_a \\ r\dot{\phi}_R &= v_a + b\dot{\theta}_a \\ ru_1 &= v_a + b\omega \\ u_1 &= \frac{v_a}{r} + \frac{b\omega}{r} \end{aligned} \quad (2.25)$$

$$\begin{aligned} v_L &= v_a + b\dot{\theta}_a \\ r\dot{\phi}_L &= v_a + b\dot{\theta}_a \\ ru_2 &= v_a + b\omega \\ u_2 &= \frac{v_a}{r} + \frac{b\omega}{r} \end{aligned} \quad (2.26)$$

Kemudian *input* robot pada Persamaan (3.25) dan Persamaan (3.26) dapat dituliskan dalam bentuk matriks menjadi Persamaan (3.27) berikut:

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{r} & \frac{b}{r} \\ \frac{1}{r} & -\frac{b}{r} \end{bmatrix} \begin{bmatrix} v_a \\ \omega \end{bmatrix} \quad (3.27)$$

Persamaan kinematika DDMR dapat diperoleh dengan menyubstitusikan Persamaan (3.27) ke dalam Persamaan (3.24) sebagai berikut:

$$\begin{aligned}
\begin{bmatrix} \dot{x}_a \\ \dot{y}_a \\ \dot{\theta}_a \end{bmatrix} &= \begin{bmatrix} \frac{r}{2} \cos \theta & \frac{r}{2} \cos \theta \\ \frac{r}{2} \sin \theta & \frac{r}{2} \sin \theta \\ \frac{r}{2b} & -\frac{r}{2b} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \\
\begin{bmatrix} \dot{x}_a \\ \dot{y}_a \\ \dot{\theta}_a \end{bmatrix} &= \begin{bmatrix} \frac{r}{2} \cos \theta & \frac{r}{2} \cos \theta \\ \frac{r}{2} \sin \theta & \frac{r}{2} \sin \theta \\ \frac{r}{2b} & -\frac{r}{2b} \end{bmatrix} \begin{bmatrix} \frac{1}{r} & \frac{b}{r} \\ \frac{1}{r} & -\frac{b}{r} \end{bmatrix} \begin{bmatrix} v_a \\ \omega \end{bmatrix} \\
\begin{bmatrix} \dot{x}_a \\ \dot{y}_a \\ \dot{\theta}_a \end{bmatrix} &= \begin{bmatrix} \frac{r}{2} \cos \theta \frac{1}{r} + \frac{r}{2} \cos \theta \frac{1}{r} & \frac{r}{2} \cos \theta \frac{b}{r} + \frac{r}{2} \cos \theta \left(-\frac{b}{r}\right) \\ \frac{r}{2} \sin \theta \frac{1}{r} + \frac{r}{2} \sin \theta \frac{1}{r} & \frac{r}{2} \sin \theta \frac{b}{r} + \frac{r}{2} \sin \theta \left(-\frac{b}{r}\right) \\ \frac{r}{2b} \frac{1}{r} + \left(-\frac{r}{2b}\right) \frac{1}{r} & \frac{r}{2b} \frac{b}{r} + \left(-\frac{r}{2b}\right) \left(-\frac{b}{r}\right) \end{bmatrix} \begin{bmatrix} v_a \\ \omega \end{bmatrix} \\
\begin{bmatrix} \dot{x}_a \\ \dot{y}_a \\ \dot{\theta}_a \end{bmatrix} &= \begin{bmatrix} \frac{1}{2} \cos \theta + \frac{1}{2} \cos \theta & \frac{b}{2} \cos \theta - \frac{b}{2} \cos \theta \\ \frac{1}{2} \sin \theta + \frac{1}{2} \sin \theta & \frac{b}{2} \sin \theta - \frac{b}{2} \sin \theta \\ \frac{1}{2b} - \frac{1}{2b} & \frac{1}{2} + \frac{1}{2} \end{bmatrix} \begin{bmatrix} v_a \\ \omega \end{bmatrix} \\
\begin{bmatrix} \dot{x}_a \\ \dot{y}_a \\ \dot{\theta}_a \end{bmatrix} &= \begin{bmatrix} \cos \theta(t) & 0 \\ \sin \theta(t) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_a \\ \omega \end{bmatrix} \tag{3.28}
\end{aligned}$$

3.2.2. Pemodelan Robot *Leader-Follower*

Pada sub bab ini akan menguraikan pemodelan rumus *multi-robot* berbasiskan pendekatan *Leader-Follower*. Berdasarkan Gambar 3.6., diasumsikan dengan roda dan lantai tidak slip. Sehingga persamaan *kinematic constraint non-holonomic* ditunjukkan pada persamaan berikut:

$$\dot{y}_m \cos \theta_m - \dot{x}_m \sin \theta_m = b \dot{\theta}_m \tag{3.29}$$

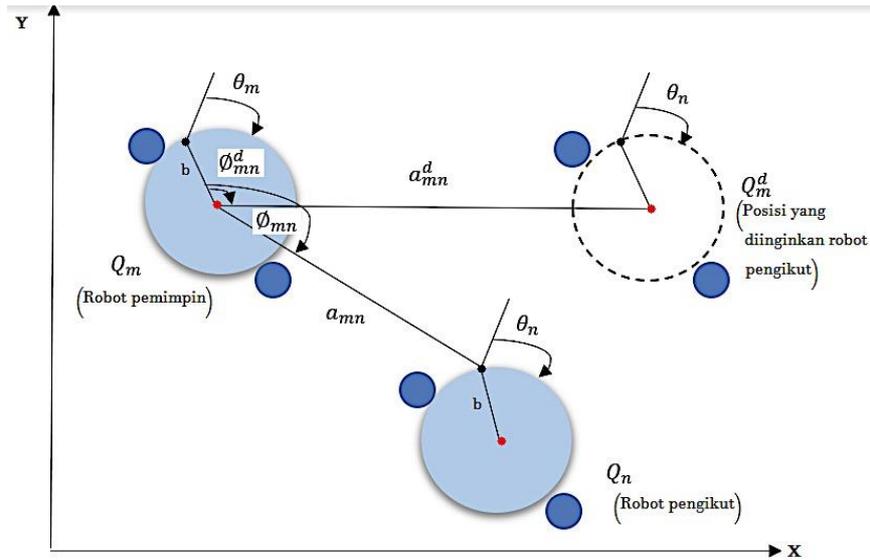
Dimana b merupakan jarak antara tepi robot ke titik setimbangan. Dari Persamaan (3.29) didapatkan model kinematika *non-holonomic* robot yang dituliskan kembali menjadi persamaan berikut:

$$\dot{Q}_m = \begin{bmatrix} \dot{x}_m \\ \dot{y}_m \\ \dot{\theta}_m \end{bmatrix} = \begin{bmatrix} \cos \theta_m & -b \sin \theta_m \\ \sin \theta_m & b \cos \theta_m \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_m \\ \omega_m \end{bmatrix} \quad (3.30)$$

Dimana

v_m : Kecepatan linier

ω_m : Kecepatan sudut pada robot



Gambar 3. 6. Skema formasi robot *leader-follower*

Dari Gambar 3.6 ditunjukkan robot *follower* R_b mengikuti robot *leader* R_a .

Berikut parameter-parameter yang didefinisikan dari Gambar 3.6 sebagai berikut:

R_m : Robot *leader*

R_n : Robot *follower*

Q_m : Posisi aktual robot *leader*

Q_n : Posisi aktual robot *follower*

Q_n^d : Posisi robot *follower* yang diharapkan

a_{mn} : Jarak aktual antar robot *leader* dan *follower*

ϕ_{mn} : Sudut aktual pergerakan roda antara *follower* dan *leader*

a_{mn}^d : Jarak robot *leader* dan robot *follower* yang diharapkan

ϕ_{mn}^d : Sudut pergerakan roda robot *leader* dan robot *follower* yang diharapkan

Hubungan geometris antar robot didapatkan dari posisi robot *follower* Q_n^d yang diharapkan kemudian diimplementasikan dalam bentuk persamaan berikut:

$$\begin{aligned} Q_n^d &= [x_n^d, y_n^d, \theta_n^d]^T \\ &= \begin{bmatrix} x_m - b \cos \theta_m + a_{mn}^d \cos(\phi_{mn}^d + \theta_m) \\ y_m - b \sin \theta_m + a_{mn}^d \sin(\phi_{mn}^d + \theta_m) \\ \theta_m \end{bmatrix} \end{aligned} \quad (3.31)$$

Posisi aktual Q_n pada robot *follower* R_n dapat dilihat pada persamaan berikut:

$$\begin{aligned} Q_n &= [x_n, y_n, \theta_n]^T \\ &= \begin{bmatrix} x_m - b \cos \theta_m + a_{mn} \cos(\phi_{mn} + \theta_m) \\ y_m - b \sin \theta_m + a_{mn} \sin(\phi_{mn} + \theta_m) \\ \theta_n \end{bmatrix} \end{aligned} \quad (3.32)$$

Jarak aktual antar robot a_{mn} di sumbu x dan y pada koordinat cartesian berikut:

$$a_{mn} = \sqrt{a_{mnx}^2 + a_{mny}^2} \quad (3.33)$$

Dimana a_{mnx} dan a_{mny} dinotasikan sebagai jarak relatif aktual antara *leader* dan *follower* terhadap koordinat Cartesian sumbu x dan y masing-masing sebagai berikut:

$$\begin{aligned} a_{mnx} &= x_m - x_n - b \cos \theta_m \\ &= -a_{mn} \cos(\phi_{mn} + \theta_m) \\ a_{mny} &= y_m - y_n - b \sin \theta_m \\ &= -a_{mn} \sin(\phi_{mn} + \theta_m) \end{aligned} \quad (3.34)$$

Hasil turunan pertama dari Persamaan (3.34) disubstitusikan pada Persamaan (3.30) menjadi

$$\begin{aligned}
\dot{a}_{mnx} &= \dot{x}_m - \dot{x}_n + b\dot{\theta}_m \sin \theta_m \\
&= v_m \cos \theta_m - b\omega_m \sin \theta_m - (v_n \cos \theta_n - b\omega_n \sin \theta_n) + b\omega_m \sin \theta_m \\
&= v_m \cos \theta_m - v_n \cos \theta_n + b\omega_n \sin \theta_n
\end{aligned} \tag{3.35}$$

$$\begin{aligned}
\dot{a}_{mny} &= \dot{y}_m - \dot{y}_n - b\dot{\theta}_m \cos \theta_m \\
&= v_m \sin \theta_m + b\omega_m \cos \theta_m - (v_n \sin \theta_n + b\omega_n \cos \theta_n) - b\omega_m \cos \theta_m \\
&= v_m \sin \theta_m - v_n \sin \theta_n - b\omega_n \cos \theta_n
\end{aligned} \tag{3.36}$$

Dimana v_m dan ω_m merupakan kecepatan linier dan kecepatan sudut robot *leader* R_m , v_n dan ω_n merupakan kecepatan linier dan kecepatan sudut robot *follower* R_n . Hasil turunan dari Persamaan (3.33) disubstitusikan ke Persamaan (3.30) sebagai berikut:

$$\begin{aligned}
\dot{\alpha}_{mn} &= \frac{1}{\sqrt{a_{mnx}^2 + a_{mny}^2}} (a_{mnx} \cdot \dot{a}_{mnx} + a_{mny} \cdot \dot{a}_{mny}) \\
&= \frac{1}{a_{mn}} \{v_m (a_{mnx} \cos \theta_m + a_{mny} \sin \theta_m)\} \\
&\quad - \frac{1}{a_{mn}} \{v_n (a_{mnx} \cos \theta_n + a_{mny} \sin \theta_n)\} \\
&\quad + \frac{1}{a_{mn}} \{-b\omega_n (-a_{mnx} \sin \theta_n + a_{mny} \cos \theta_n)\} \\
&= -v_m \cos \phi_{mn} + v_n \cos \beta_{mn} + b\omega_n \sin \beta_{mn}
\end{aligned} \tag{3.37}$$

Dimana $\beta_{mn} = \phi_{mn} + \theta_m - \theta_n$ sehingga menjadi persamaan berikut:

$$\begin{aligned}
a_{mnx} \cos \theta_m + a_{mny} \sin \theta_m &= -a_{mn} \cos \phi_{mn} \\
-a_{mnx} \sin \theta_m + a_{mny} \cos \theta_m &= -a_{mn} \sin \phi_{mn} \\
a_{mnx} \cos \theta_n + a_{mny} \sin \theta_n &= -a_{mn} \cos \beta_{mn} \\
-a_{mnx} \sin \theta_n + a_{mny} \cos \theta_n &= -a_{mn} \sin \beta_{mn}
\end{aligned}$$

Dari Gambar (3.6), $\phi_{mn} = \arctan \left(\frac{a_{mny}}{a_{mnx}} \right) - \theta_m + \pi$. Sehingga turunan pada persamaan sudut pergerakan roda sebagai berikut:

$$\begin{aligned}
\dot{\phi}_{mn} &= \left[\arctan \left(\frac{a_{mny}}{a_{mnx}} \right) - \theta_m + \pi \right]' \\
&= \frac{1}{a_{mn}} \{v_m \sin \phi_{mn} - v_n \sin \beta_{mn} + b\omega_n \cos \beta_{mn}\} - \omega_m
\end{aligned} \tag{3.38}$$

Dari persamaan (3.37) dan (3.38) model kinematika pada robot *leader-follower* dapat dituliskan kembali sebagai berikut:

$$\dot{a}_{mn} = -v_m \cos \phi_{mn} + v_n \cos \beta_{mn} + b\omega_n \sin \beta_{mn}, \quad (3.39)$$

$$\dot{\phi}_{mn} = \frac{1}{a_{mn}} \{v_m \sin \phi_{mn} - v_n \sin \beta_{mn} + b\omega_n \cos \beta_{mn}\} - \omega_m \quad (3.40)$$

3.3. Pemodelan Lingkungan Pengujian

Skala lingkungan adalah 1,4:1 cm, di mana 1 cm di nyata lingkungan adalah sama dengan 1,4 piksel pada aplikasi. Ukuran pemodelan lingkungan terbentuk ruang 961 x 561 piksel. Model robot kemudian akan diletakkan pada lingkungan berdasarkan koordinatnya pada sumbu x dan y . Berikut *pseudo code* lingkungan robot pada Gambar 3.7.

```

x_rectbegin[10], y_rectbegin[10], x_rectend[10], y_rectend[10] : integer
for i: integer 10 do
    x_rectbegin[i] → kiri
    y_rectbegin[i] → atas
    x_rectend[i] → lebar
    x_rectend[i] → tinggi
endfor
write('set rectangle(x_rectbegin,y_rectbegin,x_rectend,y_rectend)')

```

Gambar 3. 7. *Pseudo code* lingkungan robot swarm

3.4. Pemodelan Robot *Leader-follower*

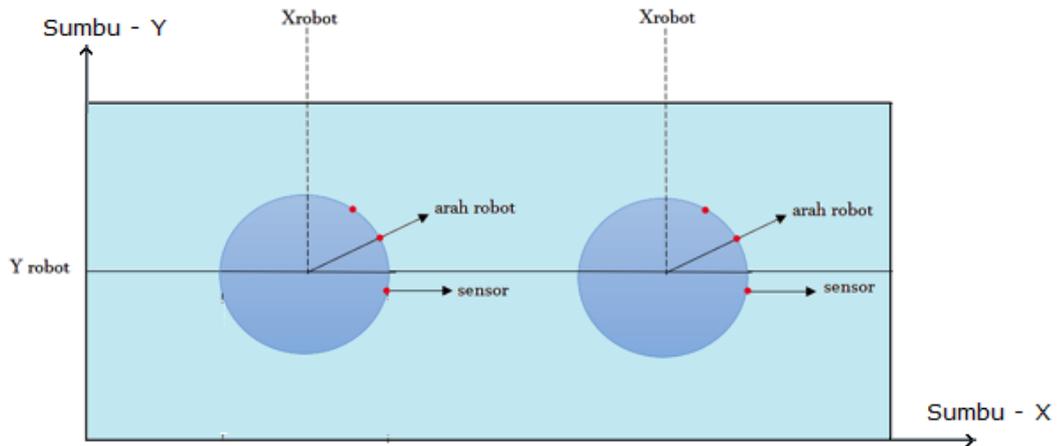
3.4.1. Pemodelan Sistem Robotik

Pada penelitian ini menggunakan robot yang bertipe *Differential Drive Mobile Robot* yaitu, robot bergerak otonom dengan dua roda penggerak kiri dan kanan yang dikemudikan secara terpisah dan diasumsikan hanya bergerak pada bidang horizontal. Robot *swarm* yang digunakan sebanyak dua robot yang disebut robot *leader* dan robot *follower*. Robot *leader* dan robot *follower* divisualisasikan sebagai bidang lingkaran yang berjalan berdasarkan nilai posisi dan orientasi. Letak model robot *swarm leader-follower* pada lingkungan ditentukan oleh koordinat titik pusatnya pada lingkungan, yaitu x_{robot} dan y_{robot} , sedangkan arah robot adalah besar sudutnya terhadap sumbu x positif berupa θ_{robot} . Besar bidang

lingkaran ditentukan oleh besar lebar robot, seperti yang diilustrasikan pada Gambar 3.8.

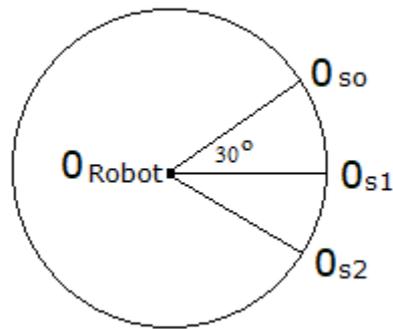
3.4.2. Pemodelan Sistem Sensor Ultrasonik

Pada simulasi, model robot *swarm* memiliki tiga buah model sensor ultrasonik pada bagian depan, samping kiri dan samping kanan, seperti pada Gambar 3.9.



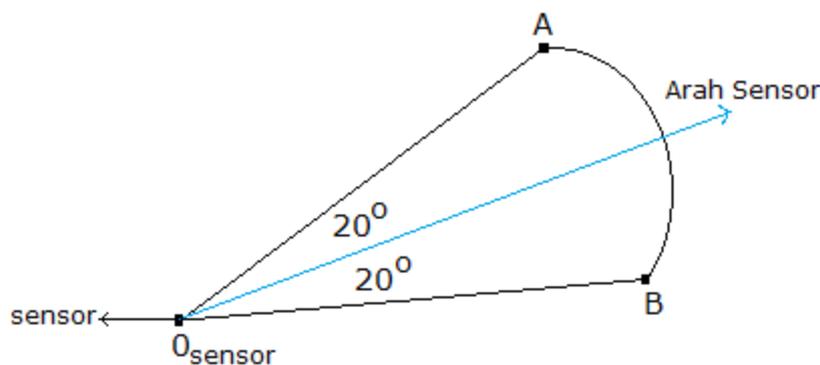
Gambar 3. 8. Pemodelan robot leader-follower

Sensor-sensor tersebut kemudian disebut sebagai sensor 0, sensor 1, dan sensor 2. Letak dari masing-masing sensor dipisahkan masing-masing sebesar 30° dari θ_{robot} . Dalam simulasi dianggap sensor 1 berada di posisi paling depan, sehingga arah sensor 1 adalah sama dengan arah robot, dan besar sudutnya terhadap sumbu x positif adalah sama dengan θ_{robot} . Koordinat dari sensor 1 O_{s1} ditentukan dengan mentranslasi titik dari O_{robot} searah dengan O_{robot} sejauh setengah dari lebar robot. Koordinat dari sensor 0 didapatkan dengan melakukan rotasi dari titik O_{s1} terhadap O_{robot} sebesar -30° sedangkan nilai arahnya adalah -30° dari θ_{robot} .



Gambar 3. 9. Posisi sensor ultrasonik dalam simulasi robot leader-follower

Koordinat dari sensor 2 didapatkan dengan melakukan rotasi dari titik O_{s1} sebesar 30° terhadap O_{robot} , dan nilai arahnya didapatkan dengan menambahkan 30° dari θ_{robot} . Pada aplikasi ini robot *swarm* berbentuk melingkar dengan 35 piksel diameter lingkaran atau 25 cm. Model sensor ultrasonik yang digunakan pada model robot *swarm* divisualisasikan sebagai titik pada lingkungan yang gerakannya mengikuti gerakan dari model robot *swarm*.



Gambar 3. 10. Model sensor dalam membaca jarak

Letak sensor ditentukan dengan merotasi titik pada bagian depan robot berdasarkan jarak sudutnya dari pusat robot. Pada aplikasi yang dirancang, besar bacaan dari model sensor didasarkan pada keadaan ideal, dimana model sensor akan membaca jarak terdekat dengan halangan yang berada pada juring AO_{sensor} seperti diilustrasikan pada Gambar 3.10. Keadaan ideal pada sebuah sensor ultrasonik adalah nilai besar sudut $AO_{sensor}B$ sebesar 40° , dan panjang bacaan maksimum, yaitu garis $O_{sensor}A$ dan $O_{sensor}B$ adalah sebesar 300 cm. Pada perangkat lunak ini, hasil bacaan model sensor untuk mendapatkan jarak

terdekat antara titik model sensor dengan bidang persegi model halangan. *Pseudo code* sensor ultrasonik dapat dilihat pada Gambar 3.11 berikut ini,

<pre> <u>Prosedur</u> set_xyrobot: $x, y \rightarrow \underline{real}$ xyrobot. $x \rightarrow x$ xyrobot. $y \rightarrow y$ <u>Prosedur</u> jarakbacasensor: <i>rectangle</i> rect $i \rightarrow \underline{integer}$ <u>for</u> $i \rightarrow 0$ <u>to</u> 5 <u>do</u> $i \rightarrow \text{jarakbacasensor}$ <u>endfor</u> <u>Prosedur</u> set_arah_robot: arah $\rightarrow \underline{real}$ <u>if</u> ($\text{arah} > 360$) <u>while</u> ($\text{arah} > 360$) <u>do</u> arah $\rightarrow \underline{integer} (\text{arah}) \bmod 360$ <u>endwhile</u> <u>endif</u> <u>if</u> ($\text{arah} < 0$) <u>while</u> ($\text{arah} < 360$) <u>do</u> arah $\rightarrow \text{arah} + 360$ <u>endwhile</u> <u>endif</u> <u>write</u> ('arahrobot \rightarrow arah') </pre>	<pre> <u>prosedur</u> updatekecepatan sensor0, sensor1, sensor2 sensor3, kecepatan: <u>real</u> kecepatan $\rightarrow 3$ <u>if</u> ($\text{sensor3} < \text{sensor0}$) <u>if</u> ($\text{sensor3} < 80$) kecepatan $\rightarrow \text{kec. kiri} (\text{sensor3}, \text{sensor2})$ <u>else</u> <u>if</u> ($\text{sensor0} > \text{sensor3}$) <u>if</u> ($\text{sensor1} < 80$) kecepatan $\rightarrow \text{kec. kanan} (\text{sensor0}, \text{sensor1})$ <u>endif</u> <u>endif</u> <u>endif</u> </pre>
--	---

Gambar 3. 11. *Pseudo code* sensor ultrasonik

3.5. Pemodelan Sistem Kontrol

3.5.1. Sistem Kontrol Pergerakan dengan Sistem Logika Fuzzy

3.5.1.1. Fuzzifikasi

Fuzzifikasi merupakan suatu proses untuk mengubah suatu *input* dari bentuk nilai pasti (*crisp input*) menjadi *fuzzy input* (variabel linguistik) yang biasa disajikan dalam bentuk himpunan-himpunan *fuzzy* dengan suatu fungsi keanggotaannya masing-masing. Fungsi keanggotaan yang digunakan pada penelitian yaitu fungsi keanggotaan Gaussian. Nilai masukan dari pengguna, yaitu σ dan c akan membentuk himpunan fungsi keanggotaan Gaussian μ_i , untuk menentukan batas-batas FOU dari himpunan interval *fuzzy* tipe 2, f_i^l nilai dari parameter masukan dihitung dengan rumus standar deviasi pada persamaan (2.24) pada Bab 2. Dapat dituliskan kembali sebagai berikut ini,

$$f_i^l = \exp \left[-\frac{1}{2} \left(\frac{x_i - c_{i1}^l}{\sigma_i^l} \right)^2 \right], \sigma_i^l \in [\sigma_{i1}^l, \sigma_{i2}^l]$$

Fungsi keanggotaan pada \bar{f} dituliskan kembali sebagai berikut,

$$\bar{f} = \begin{cases} (c_{i1}^l, \sigma_i^l; x_i), & x_i < c_{i1}^l \\ 1, & c_{i1}^l \leq x_i \leq c_{i2}^l \\ (c_{i2}^l, \sigma_i^l; x_i), & x_i > c_{i2}^l \end{cases}$$

Pseudo code nilai fuzzifikasi \bar{f} adalah seperti Gambar 3.12 sebagai berikut,

Hitung nilai x_i untuk mendapatkan nilai fuzzifikasi \bar{f} dengan persamaan
 If $(x_i < c_{i1}^l)$ then $\underline{\mu}_i^l = c_{i1}^l, \sigma_i^l$
 If $(x_i \geq c_{i1}^l)$ and $(x_i \leq c_{i2}^l)$ then $\underline{\mu}_i^l = 1$
 if $(x_i > c_{i2}^l)$ then $\underline{\mu}_i^l = c_{i2}^l, \sigma_i^l$

Gambar 3. 12. *Pseudo code* fuzzifikasi

Sedangkan persamaan untuk interval *fuzzy* tipe-2 pada \underline{f} dapat dituliskan kembali sebagai berikut,

$$\underline{f} = \begin{cases} (c_{i1}^l, \sigma_i^l; x_i), & x_i \leq \frac{c_{i1}^l + c_{i2}^l}{2} \\ (c_{i2}^l, \sigma_i^l; x_i), & x_i > \frac{c_{i1}^l + c_{i2}^l}{2} \end{cases}$$

Pseudo code nilai fuzzifikasi \underline{f} adalah adalah seperti Gambar 3.13 sebagai berikut:

Hitung nilai x untuk mendapatkan nilai fuzzifikasi \underline{f} dengan persamaan
 if $x_i \leq \frac{c_{i1}^l + c_{i2}^l}{2}$ then $\underline{\mu}_i^l = c_{i1}^l, \sigma_i^l$
 if $x_i > \frac{c_{i1}^l + c_{i2}^l}{2}$ then $\underline{\mu}_i^l = c_{i2}^l, \sigma_i^l$

Gambar 3. 13 *Pseudo Code* fuzzifikasi \underline{f}

3.5.1.2. Kaidah dan Inferensi

Banyak kombinasi kaidah yang ada pada aplikasi adalah sebanyak 8 kaidah, banyak dari kaidah didapatkan dari kombinasi 3 masukan yang masing-masing memiliki dua fungsi keanggotaan untuk robot *leader* dan robot *follower*. Kombinasi dari kaidah-kaidah dapat dilihat pada Tabel 3.1.

Tabel 3. 1.Kombinasi masukan untuk tiap-tiap kaidah

Sensor0	Sensor1	Sensor2	Kecepatan	Perubahan Sudut
Dekat	Dekat	Dekat	Lambat	Kiri
Dekat	Dekat	Jauh	Lambat	Kanan
Dekat	Jauh	Dekat	Lambat	Lurus
Dekat	Jauh	Jauh	Sedang	Kanan
Jauh	Dekat	Dekat	Lambat	Kiri
Jauh	Dekat	Jauh	Sedang	Kiri
Jauh	Jauh	Dekat	Sedang	Kiri
Jauh	Jauh	Jauh	Cepat	Lurus

Tiap-tiap baris dari kaidah kemudian akan digunakan untuk menentukan nilai hasil fuzzifikasi yang disimpan dalam inferensi. Untuk menentukan nilai hasil inferensi, dicari nilai minimum dari \bar{f} dan \underline{f} untuk masing-masing baris kaidah. Nilai-nilai ini kemudian disimpan sebagai \bar{f}_{min} dan \underline{f}_{min} . Untuk mendapatkan nilai $\bar{\mu}_{min}$ untuk tiap-tiap baris kaidah dilakukan langkah-langkah pada Gambar 3.14 seperti berikut ini:

<p>Simpan nilai \bar{f} untuk masing-masing x_1, x_2, x_3 masukan dari sensor sebagai $\bar{f}_{1dekat}, \bar{f}_{2dekat}, \bar{f}_{3dekat}$ dan $\bar{f}_{1jauh}, \bar{f}_{2jauh}, \bar{f}_{3jauh}$</p> <p>Tentukan nilai \bar{f}^{in1} sesuai dengan kombinasi masukan tiap-tiap kaidah</p> <p>Jika sensor 0 = dekat, $\bar{f}^{in1} = \bar{f}_{1dekat}$</p> <p>Jika sensor 0 = jauh, $\bar{f}^{in1} = \bar{f}_{1jauh}$</p> <p>Tentukan nilai \bar{f}^{in2} sesuai dengan kombinasi masukan tiap-tiap kaidah</p> <p>Jika sensor 1 = dekat, $\bar{f}^{in2} = \bar{f}_{2dekat}$</p> <p>Jika sensor 1 = jauh, $\bar{f}^{in2} = \bar{f}_{2jauh}$</p> <p>Tentukan nilai \bar{f}^{in3} sesuai dengan kombinasi masukan tiap-tiap kaidah</p> <p>Jika sensor 2 = dekat, $\bar{f}^{in3} = \bar{f}_{3dekat}$</p> <p>Jika sensor 2 = jauh, $\bar{f}^{in3} = \bar{f}_{3jauh}$</p> <p>Tentukan nilai $\bar{f}_{min} = \min(\bar{f}^{in1}, \bar{f}^{in2}, \bar{f}^{in3})$</p>

Gambar 3. 14. Pseudo code kaidah dan inferensi \bar{f}_{min}

Nilai \bar{f}_{min} dari tiap-tiap kombinasi masukan dihitung dan disimpan pada nilai \underline{f}_{min} . *Pseudo code* nilai \underline{f}_{min} dilakukan sebagai langkah-langkah seperti Gambar 3.15 sebagai berikut:

Simpan nilai \underline{f} untuk masing-masing x_1, x_2, x_3 masukan dari sensor sebagai $\underline{f}_{1dekat}, \underline{f}_{2dekat}, \underline{f}_{3dekat}$ dan $\underline{f}_{1jauh}, \underline{f}_{2jauh}, \underline{f}_{3jauh}$

Tentukan nilai \underline{f}^{in1} sesuai dengan kombinasi masukan tiap-tiap kaidah

Jika sensor 0 = dekat, $\underline{f}^{in1} = \underline{f}_{1dekat}$,

Jika sensor 0 = jauh, $\underline{f}^{in1} = \underline{f}_{1jauh}$,

Tentukan nilai \underline{f}^{in2} sesuai dengan kombinasi masukan tiap-tiap kaidah

Jika sensor 1 = dekat, $\underline{f}^{in2} = \underline{f}_{2dekat}$,

Jika sensor 1 = jauh, $\underline{f}^{in2} = \underline{f}_{2jauh}$

Tentukan nilai \underline{f}^{in3} sesuai dengan kombinasi masukan tiap-tiap kaidah

Jika sensor 2 = dekat, $\underline{f}^{in3} = \underline{f}_{3dekat}$,

Jika sensor 2 = jauh, $\underline{f}^{in3} = \underline{f}_{3jauh}$

Tentukan nilai $\bar{f}_{min} = \min(\underline{f}^{in1}, \underline{f}^{in2}, \underline{f}^{in3})$

Gambar 3. 15. *Pseudo code* kaidah dan inferensi \underline{f}_{min}

Nilai dari \bar{f}_{min} dan \underline{f}_{min} dari masing-masing kombinasi kaidah dihitung dan disimpan sebagai himpunan $\bar{f}_{min1}, \bar{f}_{min2}, \dots, \bar{f}_{minN}$ dan $\underline{f}_{min1}, \underline{f}_{min2}, \dots, \underline{f}_{minN}$ untuk $N =$ banyak kaidah.

3.5.1.3. Reduksi Tipe

Penelitian ini menggunakan reduksi tipe *center of sets*, dimana nilai himpunan *fuzzy* keluaran pada algoritma Logika *Fuzzy* terlebih dahulu dihitung dan disimpan. Nilai tengah (*centroid*) dari himpunan *fuzzy* interval tipe 2 μ_A^I , adalah dua nilai c_{kiri} dan c_{kanan} yang dihitung dengan mengimplementasikan algoritma iteratif Karnik-Mendel untuk menentukan pusat dari himpunan. *Pseudo code* Karnik-Mendel untuk nilai c_{kanan} adalah seperti Gambar 3.16 sebagai berikut:

```

Tentukan banyak potongan banyak_pot
Tentukan interval =  $\frac{\bar{c} - \bar{a}}{\text{banyak\_pot}}$ 
Tentukan nilai  $x_1 = \bar{a}$ 
Hitung nilai-nilai  $x_i = x_{i-1} + \text{interval}$ , untuk  $i=2,3,\dots,\text{banyak\_pot}$ 
Hitung nilai-nilai  $\bar{f}_i = \text{fuzzifikasi}(x_i)$  pada  $\bar{\mu}_A$ , untuk  $i=1,2,\dots,\text{banyak\_pot}$ 
Hitung nilai-nilai  $\underline{f}_i = \text{fuzzifikasi}(x_i)$  pada  $\underline{\mu}_A$ , untuk  $i=1,2,\dots,\text{banyak\_pot}$ 

Hitung nilai-nilai  $f_i = \frac{\bar{f}_i + \underline{f}_i}{2}$  untuk  $i=1,2,\dots,\text{banyak\_pot}$ 

Hitung nilai  $c_1 = \frac{\sum_{i=1}^N x_i f_i}{\sum_{i=1}^N f_i}$ , dengan  $N = \text{banyak\_pot}$ 
Tentukan nilai  $c_2 = -1$ 
Tentukan nilai break=false;
While (break=false) do
    Tentukan nilai k=0
    Cari nilai k, dimana  $c_i \leq f_k \leq c_{i+1}$ , dan  $i=1,2,\dots,\text{banyak\_pot}$ 
    Hitung nilai  $c_2 = \frac{\sum_{i=1}^k y_i \bar{f}_i + \sum_{i=k+1}^N y_i \underline{f}_i}{\sum_{i=1}^k \bar{f}_i + \sum_{i=k+1}^N \underline{f}_i}$ , dimana  $N = \text{banyak\_pot}$ 
    Jika  $c_1 = c_2$ , break = true
    Jika  $c_1 \neq c_2$ ,  $c_1 = c_2$ 
endwhile
Tentukan nilai  $c_{\text{kanan}} = c_2$ 

```

Gambar 3. 16. Pseudo code Karnik-Mendel c_{kanan}

Sedangkan untuk mencari nilai c_{kiri} dari μ_A^I implementasi pseudo code Karnik-Mendel untuk mencari centroid adalah seperti Gambar 3.17 sebagai berikut:

```

Tentukan banyak potongan banyak_pot
Tentukan interval =  $\frac{\bar{c} - \bar{a}}{\text{banyak\_pot}}$ 
Tentukan nilai  $x_1 = \bar{a}$ 
Hitung nilai-nilai  $x_i = x_{i-1} + \text{interval}$ , untuk  $i=2,3,\dots,\text{banyak\_pot}$ 
Hitung nilai-nilai  $\bar{f}_i = \text{fuzzifikasi}(x_i)$  pada  $\bar{\mu}_A$ , untuk  $i=1,2,\dots,\text{banyak\_pot}$ 
Hitung nilai-nilai  $\underline{f}_i = \text{fuzzifikasi}(x_i)$  pada  $\underline{\mu}_A$ , untuk  $i=1,2,\dots,\text{banyak\_pot}$ 

Hitung nilai-nilai  $f_i = \frac{\bar{f}_i + \underline{f}_i}{2}$  untuk  $i=1,2,\dots,\text{banyak\_pot}$ 

Hitung nilai  $c_1 = \frac{\sum_{i=1}^N x_i f_i}{\sum_{i=1}^N f_i}$ , dengan  $N = \text{banyak\_pot}$ 
Tentukan nilai  $c_2 = -1$ 
Tentukan nilai break=false;
While (break=false) do
    Tentukan nilai k=0
    Cari nilai k, dimana  $c_i \leq f_k \leq c_{i+1}$ , dan  $i=1,2,\dots,\text{banyak\_pot}$ 
    Hitung nilai  $c_2 = \frac{\sum_{i=1}^k y_i \bar{f}_i + \sum_{i=k+1}^N y_i \underline{f}_i}{\sum_{i=1}^k \bar{f}_i + \sum_{i=k+1}^N \underline{f}_i}$ , dimana  $N = \text{banyak\_pot}$ 
    Jika  $c_1 = c_2$ , break = true
    Jika  $c_1 \neq c_2$ ,  $c_1 = c_2$ 
endwhile
Tentukan nilai  $c_{\text{kiri}} = c_2$ 

```

Gambar 3. 17. Pseudo code Karnik-Mendel c_{kiri}

Langkah-langkah untuk menghitung y_{kanan} pada *pseudo code* Logika Fuzzy dengan himpunan keluaran kecepatan adalah seperti Gambar 3.18 sebagai berikut:

Tentukan nilai-nilai $y_{kanan}[i]$ untuk $i=1,2,\dots,$ banyak kaidah, sesuai dengan keluaran dari tiap kombinasi baris kaidah.

$$y_{kanan}[i] = c_{knlambat}, \text{ jika keluaran} = \text{lambat}$$

$$y_{kanan}[i] = c_{knsedang}, \text{ jika keluaran} = \text{sedang}$$

$$y_{kanan}[i] = c_{kncepat}, \text{ jika keluaran} = \text{cepat}$$

Urutkan himpunan y_i beserta urutan kombinasi kaidahnya secara menaik untuk $i=1,2,\dots,$ banyak kaidah.

Hitung nilai-nilai $f_{min i} = \frac{\bar{f}_{min i} + \underline{f}_{min i}}{2}$ untuk $i=1,2,\dots,$ banyak kaidah

Hitung nilai $y_{kanan} = \frac{\sum_{i=1}^N y_{kanan}[i] f_{min i}}{\sum_{i=1}^N f_{min i}}$, dengan $N =$ banyak kaidah

Tentukan nilai $y_{kanan}'' = -1$

Tentukan nilai $k_lama = 0$

Tentukan nilai $k_baru = 0$

Tentukan nilai $break = false$;

While ($break = false$) do

Tentukan nilai $k = 0$

Cari nilai k_baru , dimana $y_{kanan}[i] \leq f_{min k_baru} \leq y_{kanan}(i+1)$ dengan $i=1,2,\dots,$ banyak kaidah

Jika $k_baru \leq k_lama$, $break = true$;

Hitung nilai $y_{kanan}'' = \frac{\sum_{i=1}^k y_i f_{min i} + \sum_{i=k+1}^N y_i \bar{f}_{min i}}{\sum_{i=1}^k \underline{f}_{min i} + \sum_{i=k+1}^N \bar{f}_{min i}}$, diman $N =$ banyak kaidah

Jika $y_{kanan}' = y_{kanan}''$ $break = true$

Jika $y_{kanan}' \neq y_{kanan}''$

$y_{kanan}' = y_{kanan}''$

$k_lama = k_baru$

endwhile

Tentukan nilai $y_{kanan} = y_{kanan}''$

Gambar 3. 18. Pseudo code keluaran kecepatan pada y_{kanan}

Nilai y_{kiri} pada *pseudo code* Logika Fuzzy dengan keluaran himpunan keanggotaan kecepatan kemudian juga dihitung dengan langkah-langkah seperti Gambar 3.19 sebagai berikut:

Tentukan nilai-nilai $y_{kiri}[i]$ untuk $i=1,2,\dots,$ banyak kaidah, sesuai dengan keluaran dari tiap kombinasi baris kaidah.

$y_{kiri}[i] = c_{krlambat}$, jika keluaran = lambat

$y_{kiri}[i] = c_{krse dang}$, jika keluaran = sedang

$y_{kiri}[i] = c_{krcepat}$, jika keluaran = cepat

Urutkan himpunan y_{li} beserta urutan kombinasi kaidahnya secara menaik untuk $i=1,2,\dots,$ banyak kaidah.

Hitung nilai-nilai $f_{\min i} = \frac{\bar{f}_{\min i} + \underline{f}_{\min i}}{2}$ untuk $i=1,2,\dots,$ banyak

kaidah

Hitung nilai $y_{kiri}' = \frac{\sum_{i=1}^N y_{kiri}[i] f_{\min i}}{\sum_{i=1}^N f_{\min i}}$, dengan $N =$ banyak kaidah

Tentukan nilai $y_{kiri}'' = -1$

Tentukan nilai $k_lama = 0$

Tentukan nilai $k_baru = 0$

Tentukan nilai $break = false$;

While ($break = false$) do

Tentukan nilai $k = 0$

Cari nilai k_baru , dimana $y_{kiri}[i] \leq f_{\min k_baru} \leq y_{kiri(i+1)}$ dengan $i=1,2,\dots,$ banyak kaidah

Jika $k_baru \leq k_lama$, $break = true$;

Hitung nilai $y_{kiri}'' = \frac{\sum_{i=1}^k y_{li} \underline{f}_{\min i} + \sum_{i=k+1}^N y_{li} \bar{f}_{\min i}}{\sum_{i=1}^k \underline{f}_{\min i} + \sum_{i=k+1}^N \bar{f}_{\min i}}$, dimana N

$=$ banyak kaidah

Jika $y_{kiri}' = y_{kiri}''$ $break = true$

Jika $y_{kiri}' \neq y_{kiri}''$

$y_{kiri}' = y_{kiri}''$

$k_lama = k_baru$

endwhile

Tentukan nilai $y_{kiri} = y_{kiri}''$

Gambar 3. 19. Pseudo Code keluaran kecepatan pada y_{kiri}

Langkah-langkah untuk menghitung y_{kanan} pada pseudo code Logika Fuzzy dengan himpunan keluaran perubahan sudut adalah seperti Gambar 3.20 sebagai berikut:

Tentukan nilai-nilai $y_{kanan}[i]$ untuk $i=1,2,\dots,$ banyak kaidah, sesuai dengan keluaran dari tiap kombinasi baris kaidah.

$y_{kanan}[i] = c_{knlkiri}$, jika keluaran = kiri

$y_{kanan}[i] = c_{knslurus}$, jika keluaran = lurus

$y_{kanan}[i] = c_{knkanan}$, jika keluaran = kanan

Urutkan himpunan y_{li} beserta urutan kombinasi kaidahnya secara menaik untuk $i=1,2,\dots,$ banyak kaidah.

Hitung nilai-nilai $f_{\min i} = \frac{\bar{f}_{\min i} + \underline{f}_{\min i}}{2}$ untuk $i=1,2,\dots,$ banyak kaidah

Hitung nilai $y_{kanan}' = \frac{\sum_{i=1}^N y_{kanan}[i] f_{min i}}{\sum_{i=1}^N f_{min i}}$, dengan $N = \text{banyak kaidah}$

Tentukan nilai $y_{kanan}'' = -1$
 Tentukan nilai $k_{lama} = 0$
 Tentukan nilai $k_{baru} = 0$
 Tentukan nilai $break = false$;
 While ($break = false$) do
 Tentukan nilai $k = 0$
 Cari nilai k_{baru} , dimana $y_{kanan}[i] \leq f_{min k_{baru}} \leq y_{kanan}(i+1)$ dengan
 $i = 1, 2, \dots, \text{banyak kaidah}$
 Jika $k_{baru} \leq k_{lama}$, $break = true$;
 Hitung nilai $y_{kanan}'' = \frac{\sum_{i=1}^k y_{li} f_{min i} + \sum_{i=k+1}^N y_{li} \bar{f}_{min i}}{\sum_{i=1}^k f_{min i} + \sum_{i=k+1}^N \bar{f}_{min i}}$, diman $N = \text{banyak kaidah}$

Jika $y_{kanan}' = y_{kanan}''$ $break = true$
 Jika $y_{kanan}' \neq y_{kanan}''$
 $y_{kanan}' = y_{kanan}''$
 $k_{lama} = k_{baru}$

endwhile
 Tentukan nilai $y_{kanan} = y_{kanan}''$

Gambar 3. 20. Pseudo Code keluaran perubahan sudut pada y_{kanan}

Sedangkan langkah-langkah untuk menghitung y_{kiri} pada pseudo code Logika Fuzzy dengan himpunan keluaran perubahan sudut adalah seperti Gambar 3.21 sebagai berikut:

Tentukan nilai-nilai $y_{kiri}[i]$ untuk $i = 1, 2, \dots, \text{banyak kaidah}$, sesuai dengan keluaran dari tiap kombinasi baris kaidah.

$y_{kiri}[i] = c_{krkiri}$, jika keluaran = kiri
 $y_{kiri}[i] = c_{krlurus}$, jika keluaran = lurus
 $y_{kiri}[i] = c_{krkanan}$, jika keluaran = kanan

Urutkan himpunan y_{li} beserta urutan kombinasi kaidahnya secara menaik untuk $i = 1, 2, \dots, \text{banyak kaidah}$.

Hitung nilai-nilai $f_{min i} = \frac{\bar{f}_{min i} + f_{min i}}{2}$ untuk $i = 1, 2, \dots, \text{banyak kaidah}$

Hitung nilai $y_{kiri}' = \frac{\sum_{i=1}^N y_{kiri}[i] f_{min i}}{\sum_{i=1}^N f_{min i}}$, dengan $N = \text{banyak kaidah}$

Tentukan nilai $y_{kiri}'' = -1$
 Tentukan nilai $k_{lama} = 0$
 Tentukan nilai $k_{baru} = 0$
 Tentukan nilai $break = false$;
 While ($break = false$) do
 Tentukan nilai $k = 0$
 Cari nilai k_{baru} , dimana $y_{kiri}[i] \leq f_{min k_{baru}} \leq y_{kiri}(i+1)$ dengan

```

i=1,2,...,banyak kaidah
Jika k_baru ≤ k_lama, break=true;
    Hitung nilai "y_kiri" =  $\frac{\sum_{i=1}^k y_i \underline{f}_{\min i} + \sum_{i=k+1}^N y_i \bar{f}_{\min i}}{\sum_{i=1}^k \underline{f}_{\min i} + \sum_{i=k+1}^N \bar{f}_{\min i}}$ , dimana N
    =banyak kaidah
    Jika "y_kiri' = "y_kiri" break = true
    Jika "y_kiri' ≠ "y_kiri"
        "y_kiri' = "y_kiri"
        k_lama=k_baru
    endwhile
    Tentukan nilai "y_kiri" = "y_kiri"

```

Gambar 3. 21. *Pseudo code* keluaran perubahan sudut pada y_{kiri}

3.5.1.4. Defuzzifikasi

Nilai tegas dari suatu himpunan tereduksi dapat dihitung dengan menjumlahkan y_{kanan} dan y_{kiri} yang telah didapatkan dari hasil reduksi tipe dan membaginya dua seperti Persamaan (2.34).

3.5.2. Optimisasi Pencarian Target

Optimisasi pencarian target adalah suatu proses yang dilakukan oleh robot *leader-follower* untuk mencapai target dengan jarak terpendek, waktu yang cepat dan menghasilkan nilai optimal global. Namun harus diperhatikan bahwa metode optimisasi harus sederhana dengan komputasi yang kecil karena keterbatasan robot *swarm* dengan model *leader-follower* yang identik, sederhana dengan kemampuan prosesor terbatas. Sehingga yang dibutuhkan adalah bagaimana menghasilkan nilai fungsi *fitness* menuju nol sehingga menghasilkan solusi yang mengerucut dengan bobot yang besar agar waktu konvergensi menjadi cepat dan optimal. Pada model yang diusulkan yaitu pendekatan Dinamik PSO dilakukan dengan mengimprovisasi bobot inersia menjadi nilai yang adaptif, berubah sesuai dengan kondisi lingkungan dan melakukan pengukuran fungsi fitness. Dalam penelitian ini untuk melihat kinerja dari model optimisasi DPSO akan dibandingkan dengan *Gregarious* PSO (GPSO) karena model GPSO menghasilkan proses komputasi yang sederhana dengan kinerja yang baik

(Kessentini, and Barchiesi, 2015). Model ini lebih sesuai untuk diimplementasikan pada robot *swarm* dengan memori dan *processor* terbatas.

3.5.2.1. Pendekatan Dinamik PSO

Dalam hal menentukan metode optimisasi yang cocok bagi sistem kontrol formasi *leader-follower* yang perlu diperhatikan adalah metode yang sederhana dengan beban komputasi yang tidak besar. Hal tersebut dikarenakan robot *leader-follower* merupakan salah satu jenis robot yang sederhana dan identik dengan sistem sensor, prosesor dan penggerak terbatas.

Metode optimisasi yang digunakan pada sistem kontrol pergerakan robot *leader-follower* pada penelitian ini yaitu dengan menggunakan metode dinamik PSO. Dimana metode PSO diimplementasikan dengan konsep '*gbest*' (*Globally Oriented*) yaitu orientasi secara global. Setiap robot memiliki pergerakan yang mempunyai koordinat pada jalur masing-masing. Koordinat tersebut akan dikirimkan melalui data serial sehingga dapat ditentukan oleh robot *leader*. Ketika robot *leader* telah ditentukan maka robot *follower* lainnya akan mengikuti *leader*.

Ketika robot tidak mendeteksi halangan (*obstacle*) dan robot mendeteksi adanya target, maka metode PSO akan mulai diproses. PSO menggunakan perhitungan yang menggunakan jarak setiap posisi robot, dalam penelitian ini posisi tiap robot adalah setiap perubahan jarak robot dalam mendekati target. *Global best* dan $f() \text{ global best}$ telah ditentukan sebelumnya karena *global best* adalah posisi optimal yang akan dicapai oleh robot. Dalam hal ini *global best* adalah jarak minimal robot dari target dan $f() \text{ global best}$ adalah besarnya nilai target maksimal yang dibaca oleh robot. Sedangkan *particle best* dan $f() \text{ particle best}$ adalah nilai inputan dari sensor robot, dimana *particle best* adalah jarak posisi robot dari target dan $f() \text{ particle best}$ adalah besarnya nilai dari target pada jarak tersebut.

Dengan menganalisis persamaan (2.37) dan (2.38), dapat diperoleh bahwa setiap partikel mengikuti dua nilai 'terbaik', nilai terbaik global saat ini dan solusi terbaik yang telah dicapai sejauh ini. Kecepatan partikel dengan cepat mendekati nol, yang menyebabkan partikel berhenti dengan solusi secara lokal. Jika persamaan (2.37) diulang kembali maka dapat dinyatakan sebagai berikut;

$$v_i^{k+1} = v_i^k + c_1 * rand1 * (p_i^k - x_i^k) + c_2 * rand2 * (p_g^k - x_i^k), \quad (3.41)$$

$$x_i^{k+1} = x_i^k + (v_i^{k+1}); i = 1, 2, \dots, n, \quad (3.42)$$

dimana nilai $c_1=c_2$ adalah dua bilangan positif dengan nilai konstan, $rand1$ dan $rand2$ adalah bernilai 1; dan bobot inersia w_i tidak ada.

Dengan nilai yang telah ditentukan diharapkan sistem akan cepat konvergen, sayangnya nilai tersebut sering membuat solusi optimal secara lokal. Namun untuk mencari nilai w_i yang paling cocok, merupakan hal yang sulit. Berbagai metode telah dikembangkan. Pada penelitian ini untuk memperbaiki kekurangan dari algoritma standar PSO digunakan teknik bobot inersia Terbaik Global-Lokal. Perhitungannya bobot inersia didasarkan pada fungsi terbaik lokal dan global dari partikel-partikel di setiap generasi. Hal itu tidak mengambil nilai konstan atau secara linear mengurangi nilai waktu yang bervariasi. Untuk mengatasi kelemahan konvergensi prematur ke lokal minimum, strategi bobot inersia adaptif diusulkan untuk meningkatkan kemampuan pencariannya. Teknik ini mengontrol keragaman populasi dengan penyesuaian secara adaptif dari bobot inersia dan untuk menjaga keseimbangan antara sifat sosial dan individu.

Modifikasi persamaan (3.41) menjadi persamaan (3.43) dengan menambah bobot inersia adalah sebagai berikut:

$$v_i^{k+1} = w_i * v_i^k + c_1 * rand1 * (p_i^k - x_i^k) + c_2 * rand2 * (p_g^k - x_i^k) \quad (3.43)$$

Faktor bobot inersia adalah $w \geq 0$. Dengan asumsi bahwa w yang relatif besar memiliki kemampuan pencarian yang lebih global sementara hasil w yang relatif kecil menghasilkan konvergensi yang lebih cepat.

Dalam persamaan (3.43), nilai $c_1=c_2 = 2$, $rand1$ dan $rand2$ adalah dua bilangan yang terdistribusi secara uniform antara [0.4, 0.5]. w_i adalah bobot inersia yang telah di pilih agar mendapatkan kinerja yang baik. Bobot inersia ini menggunakan hukum menurun secara linier (persamaan 3.44) yang berubah terhadap waktu dengan nilai w yaitu:

$$w = w_{ini} - \left(\frac{iter_{max} - iter}{iter_{max}} \right) * (w_{max} - w_{min}) \quad (3.44)$$

Posisi partikel sekarang adalah posisi partikel sebelumnya ditambahkan dengan kecepatan pergerakannya. Faktor kecepatan yang dihitung dengan menggunakan persamaan (3.44) akan menentukan pergerakan robot selanjutnya. Kecepatan ini diperoleh dengan menggunakan informasi dari partikel itu sendiri dan partikel-partikel lainnya.

Informasi dari partikel itu sendiri disebut dengan komponen kognitif, dimana nilainya merupakan jarak dari posisi partikel tersebut dengan posisi terbaik yang telah diperoleh partikel tersebut selama pencarian. Posisi terbaik partikel tersebut diberi nama *local best* (p_i). Setiap partikel memiliki informasi *local best* masing-masing. Dari *local best* yang telah diperoleh dari setiap partikel, tentu ada lokasi yang memiliki posisi terbaik di antara semua *local best* lainnya. Posisi terbaik ini disebut sebagai *global best* (p_g). Nilai *global best* inilah yang disebut dengan komponen sosial dari DPSO.

Untuk mengetahui seberapa baik posisi yang telah ditemukan, maka diperlukan suatu fungsi yang dapat mengukurnya. Fungsi ini disebut sebagai fungsi *fitness* $f(x)$ yaitu;

$$f' = \frac{f(p_g(t)) - f(p_g(t-1))}{f(p_g(t))} \quad (3.45)$$

Fungsi ini memiliki domain vektor posisi x . Makin baik nilainya, makin dekat posisi tersebut dengan solusi. Dengan fungsi ini, posisi-posisi yang telah ditemukan bisa dibandingkan kedekatannya dengan solusi yang dicari. *Fitness function* berbeda pada tiap-tiap permasalahan bergantung pada masalah yang dihadapi. Pada parameter $c1$ dan $c2$, jika $c1$ lebih besar maka pengaruh komponen kognitifakan lebih besar daripada komponen sosial dan begitu pula sebaliknya. Komponen kognitif yang lebih besar akan membuat partikel bergerak lebih ke arah *local best* sedangkan jika komponen sosial yang lebih besar maka partikel bergerak lebih ke arah *global best*.

Dari bobot inersia yang telah dipilih, maka proses optimisasi akan di bandingkan dengan salah satu varian PSO yaitu *Gregarious* PSO (GPSO) (Kessentini and Barchiesi, 2015). Pada pemrosesan GPSO juga menggunakan sifat menurun secara linier yang berubah terhadap waktu dengan k adalah konstanta *Gregarious* sehingga pemilihan bobot inersia w sebagai berikut (Kessentini and Barchiesi, 2015) :

$$w = w_{max} - \left(\frac{w_{max} - w_{min}}{iter_{max}} \right) * k \quad (3.46)$$

Bobot inersia w yang direkomendasikan adalah bervariasi dari nilai [0.4, 0.9] sedangkan $c_1 = c_2 = 1.5$ atau persamaan (3.47) dapat disederhanakan menjadi (Kessentini and Barchiesi, 2015) :

$$w = 0.9 - 0.5 * \frac{t}{T} \quad (3.47)$$

dimana t adalah jumlah langkah saat ini dan T adalah maksimum jumlah langkah yang ditentukan. GPSO di pilih karena komputasi yang sederhana, dengan kinerja yang telah terbukti baik.

Secara umum maka algoritma PSO standar dan PSO perbaikan dalam hal ini DPSO dapat dinyatakan dalam Tabel 3.2 sebagai berikut :

Tabel 3. 2. Resume antara PSO standar dan Dinamik PSO

PSO	PSO perbaikan (DPSO)
<p>Prosedur berikut dapat digunakan untuk mengimplementasikan algoritma PSO persamaan ;</p> <p>1) Inisialisasi, $w = 1$, $c_1 = c_2 = 1$, $r_1 = r_2 = random$ dan jumlah <i>swarm</i>;</p> <p>2) Inisialisasi kawanan (<i>swarm</i>) dengan menetapkan posisi acak di “<i>problem hyperspace</i>” untuk setiap partikel;</p>	<p>Prosedur berikut dapat digunakan untuk mengimplementasikan algoritma DPSO persamaan 3.43;</p> <p>1) Inisialisasi w, $(w_{max} - w_{min})w = w_{ini} - \left(\frac{iter_{max} - iter}{iter_{max}} \right) * (w_{max} - w_{min})$</p> <p>$c_1 = c_2 = 2$, $r_1 = r_2 = random$ dan jumlah <i>swarm</i>;</p>

<p>3) Mengevaluasi fungsi fitness untuk setiap partikel;</p> <p>4) Untuk setiap partikel individu, bandingkan nilai fitness partikel dengan $pbest$ nya. Jika nilai saat ini lebih baik daripada nilai $pbest$, maka atur nilai ini sebagai $pbest$ dan saat ini posisi partikel, x_i sebagai p_i;</p> <p>5) Identifikasi partikel yang memiliki nilai fitness terbaik. Itu nilai fungsi fitnessnya diidentifikasi sebagai $gbest$ dan posisi sebaga p_g;</p> <p>6) Perbarui semua lokasi dan posisi di seluruh partikel dengan menggunakan langkah (1) dan (2);</p> <p>7) Ulangi langkah 2–5 hingga kriteria berhenti terpenuhi misalnya dalam hal jumlah iterasi maksimum atau nilai fitness yang cukup baik.</p>	<p>2) Inisialisasi kawanan (<i>swarm</i>) dengan menetapkan posisi acak di “<i>problem hyperspace</i>” untuk setiap partikel;</p> <p>3) Mengevaluasi fungsi fitness untuk setiap partikel;</p> $f' = \frac{f(p_g(t)) - f(p_g(t-1))}{f(p_g(t))}$ <p>4) Untuk setiap partikel individu, bandingkan nilai fitness partikel dengan $pbest$ nya. Jika nilai saat ini lebih baik daripada nilai $pbest$, maka atur nilai ini sebagai $pbest$ dan saat ini posisi partikel, x_i sebagai p_i;</p> <p>5) Identifikasi partikel yang memiliki nilai fitness terbaik. Itu nilai fungsi fitnessnya diidentifikasi sebagai $gbest$ dan posisi sebaga p_g;</p> <p>6) Perbarui semua lokasi dan posisi di seluruh partikel dengan menggunakan langkah (1) dan (2);</p> <p>7) Ulangi langkah 2–5 hingga kriteria berhenti terpenuhi misalnya dalam hal jumlah iterasi maksimum, konvergensi atau nilai fitness yang cukup kecil.</p>
---	--

3.5.3. Pemrosesan PSO dan Kondisi Berhenti

Algoritma PSO adalah kondisi tertentu yang dapat menyebabkan pencarian berhenti. Terdapat beberapa kondisi yang dapat dijadikan parameter untuk membuat algoritma PSO berhenti, yaitu :

1. Iterasi telah melebihi batas waktu yang ditentukan

Batas waktu dalam PSO menjadi hal yang penting dalam kondisi ini. Jika batas waktu terlalu singkat, pencarian akan berhenti sebelum solusi ditemukan.

Jika batas waktu terlalu panjang, pada pencarian yang gagal akan terus berjalan walaupun PSO sudah mencapai titik konvergen dimana pergerakan menjadi tidak signifikan.

2. Solusi yang dapat diterima sudah ditemukan

Jika dalam pencarian telah ditemukan solusi yang dapat diterima, maka pencarian akan dihentikan. Batas yang diterima adalah ambang batas solusi yang didapat apakah sudah sesuai dengan apa yang diinginkan. Jika ambang batas terlalu jauh, solusi yang ditemukan mungkin tidak cukup sesuai dengan apa yang diinginkan. Sementara ambang batas yang terlalu kecil akan membuat PSO cukup kesulitan untuk menemukan solusi yang dapat memenuhinya.

3. Tidak ada peningkatan di beberapa iterasi

Terdapat beberapa cara untuk mengetahui apakah peningkatan sudah cukup baik, sehingga iterasi dapat dilanjutkan atau tidak. Jika perubahan posisi partikel cukup kecil, maka dapat dipastikan PSO sudah menuju ke titik konvergen. Cara lain, jika perhitungan kecepatan pada PSO memberikan hasil yang mendekati nol maka partikel hampir tidak bergerak, dan perhitungan PSO dapat dihentikan.

4. Radius normal dari kumpulan partikel mendekati nol

Radius normal dihitung dengan persamaan (3.48) sebagai berikut :

$$R_{norm} = \frac{R_{max}}{\text{diameter}(S)} \quad (3.48)$$

Diameter (S) merupakan diameter dari kumpulan partikel awal dan R_{max} merupakan jarak maksimum partikel dari *global best*.

$$R_{max} = \|x_m - p_g\| \quad m = 1, \dots, n \quad (3.49)$$

$$R_{max} \geq \|x_i - p_g\| \quad \forall i = 1, 2, 3 \dots, n \quad (3.50)$$

Jika R_{max} mendekati nol, maka kumpulan partikel (n) memiliki kemungkinan kecil untuk melakukan peningkatan. Perlu di berikan definisi batasan cukup dekat dengan nol. Jika batasan terlalu besar, maka PSO akan berhenti sebelum solusi

optimum ditemukan. Sebaliknya, jika batasan terlalu kecil, akan dibutuhkan iterasi yang lebih panjang untuk dapat mencapai batasan tersebut.

5. Perubahan solusi yang ditemukan mendekati nol

Kondisi ini terjadi dengan melihat perubahan solusi yang telah ditemukan. Perubahan solusi dihitung dengan persamaan (3.48). pada Algoritma DPSO dimodifikasi dengan menambahkan indeks peningkatan $f(x)$ untuk partikel dalam kawanan. $f(x)$ adalah fungsi *fitness* yang menghitung seberapa besar solusi telah ditemukan pada posisi x . Jika f terlalu kecil, maka kumpulan partikel dapat diasumsikan sudah konvergen menuju satu titik. Di sini juga dibutuhkan batasan seberapa dekat perubahan yang terjadi dengan nol. Seperti pada kondisi-kondisi sebelumnya, jika ambang batas terlalu besar, maka DPSO akan berhenti sebelum solusi yang optimal ditemukan, begitu pula sebaliknya.

Algoritma PSO bersifat konvergen jika pada iterasi tertentu seluruh partikel akan menuju ke satu titik *global best*. Saat hal ini terjadi, maka kemungkinan pergerakan partikel-partikel tersebut tidak akan menjadi terlalu signifikan yang membuat perhitungan PSO menjadi tidak efisien. Pada saat inilah akan ditentukan apakah pencarian akan tetap dilanjutkan atau tidak. Konvergen dalam PSO belum tentu berarti partikel sudah menemukan titik optimum (lokal atau global). Konvergen di sini berarti PSO telah mencapai titik keseimbangan. Partikel-partikel bergerak hanya menuju ke satu titik, bisa itu berupa titik optimum atau bukan.

Berdasarkan algoritma yang dijelaskan sebelumnya dapat dibuat *pseudo code* seperti Gambar 3.22, berikut ini:

```

Deklarasi :
  pbest, gbest : integer
  w: integer
Deskripsi :
  for → i
  if f(xi) > f(pbest) then
    pbest ← xi
  end if
  if f(xi) > f(gbest) then
    gbest ← xi
  end if
  end for
  w ← w = wini - (itermax - iter/itermax) * (wmax - wmin)
  for → i
  vik+1 ← wi * vik + c1 * rand * (pik - xik) + c2 * rand * (pgk - xik)
  end for

```

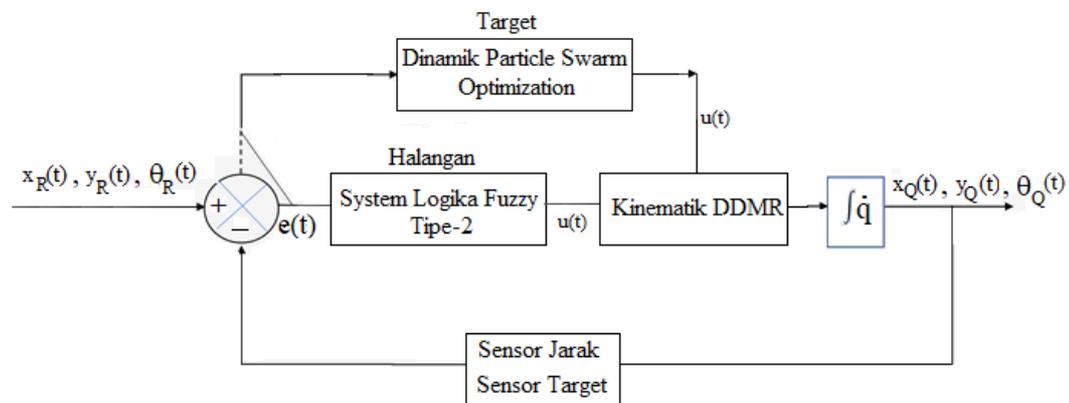
Gambar 3. 22. Pseudo code dinamik PSO (DPSO)

3.6. Sistem Kontrol Formasi Kombinasi Metode SLFT2-DPSO

Objek penelitian yang dipilih pada penelitian ini yaitu perancangan pengendali logika *fuzzy* tipe-2 dan dinamik PSO dengan pemodelan kinematika pada *Differential Drive Mobile Robot* (DDMR) yang bersifat *leader-follower*. Tujuan diterapkannya sistem kontrol dengan metode SLFT2 adalah agar *robot leader-follower* dapat mencapai target yang diharapkan, mampu mengendalikan kecepatan roda serta menjaga stabilitas kesalahan pergerakan yang minimum, sedangkan metode DPSO diterapkan untuk memecahkan masalah optimisasi dan mencari nilai terbaik secara otomatis dari rute tempuh robot *leader-follower*.

Sistem kontrol formasi untuk robot *leader-follower* dengan metode SLFT2-DPSO bekerja tidak secara bersamaan tetapi memiliki fungsinya masing-masing. SLFT2 memiliki fungsi untuk mengontrol pergerakan posisi dan orientasi robot *leader* saat bergerak dalam lingkungan, sedangkan SLFT2 bagi robot *follower* untuk menjaga jarak antara robot dengan robot dan robot dengan dinding penghalang. Jika sensor jarak aktif pada saat robot bergerak, maka dianggap terdapat halangan pada lingkungannya. Masukan dari sensor jarak akan

mengaktifkan pengendali SLFT2 dan robot *leader-follower* akan menghindari penghalang yang ada di lingkungan. Sedangkan pengendali DPSO akan mengontrol pergerakan robot *leader* berjalan menuju target. Dengan menggunakan teori optimisasi diharapkan robot dapat memilih rute terbaiknya dan dapat tetap menuju ke target. Pada bagian ini robot *leader* akan bergerak jika terdapat masukan dari sensor target. Sedangkan robot *follower* akan mengikuti robot *leader* dengan menjaga jarak pergerakan. Blok diagramnya dapat dilihat pada Gambar 3.23.



Gambar 3. 23. Sistem kontrol formasi dengan metode SLFT2-DPSO

Pada Gambar 3.23, dapat dilihat bagaimana jalannya pengontrolan pergerakan berdasarkan nilai referensi (posisi dan orientasi) yang diberikan pada robot *leader-follower*. Pada setiap pergerakan robot akan terdapat eror, yaitu selisih posisi referensi dan posisi aktual. Sehingga diperlukan sebuah sistem kontrol untuk mengatasinya. Aksi dari sistem kontrol adalah untuk memperkecil eror pergerakan, jika nilai dari posisi referensi bernilai sama atau mendekati posisi aktual maka dapat dikatakan bahwa sistem kontrol mampu meminimalisir nilai eror yang didapat dan menghasilkan pergerakan yang baik. Keluaran berupa kecepatan linier (v_f) dan kecepatan angular (ω_f) dimana keluaran ini akan diubah menjadi masukan dalam persamaan kinematika robot *leader*. Sedangkan lambang *integrator* merupakan pembalik persamaan perubahan posisi yang telah didapatkan.

Sistem kontrol formasi kombinasi metode SLFT2 dan DPO akan berjalan menggunakan 4 kondisi sesuai dengan blok diagram Gambar 3.23 yaitu:

1. Robot *leader-follower* akan menghindari halangan maupun dinding yang ada di depannya jika sensor jarak aktif dan mendeteksi halangan, sinyal dari sensor jarak akan mengaktifkan pengontrol SLFT2;
2. Robot *leader-follower* akan menuju ke target jika sensor jarak tidak aktif, dan sensor target memberikan indikasi target berada pada area pendeteksian sensor dan hal itu akan mengaktifkan pengontrol DPSO untuk mencari rute terbaik;
3. Jika kedua sensor aktif, dalam arti sensor jarak mendeteksi penghalang dan sensor jarak mendeteksi target, maka prioritas utama adalah menghindari halangan. Sehingga SLFT2 akan mengontrol pergerakan, dan pengontrol DPSO akan tidak aktif. Maka dalam algoritma SLFT2-DPSO akan sering terjadi interupsi kepada DPSO untuk menonaktifkan aksinya.
4. Robot *leader-follower* akan berhenti bergerak jika target telah tercapai, jika belum maka kedua sensor akan terus aktif, tetapi dengan prioritas adalah menghindari halangan.

3.7. Pengujian dan Validasi

Dalam perancangan sistem kontrol formasi pada robot *swarm*, untuk mencapai tingkat presisi, sistem harus mampu mengatasi rintangan yang tak terduga di lingkungan. Lingkungan dikondisikan dengan beberapa keadaan, seperti lingkungan yang telah sederhana, lingkungan yang tidak diketahui, serta pada lingkungan yang dinamik. Dalam menentukan parameter nilai fungsi keanggotaan pada sistem logika fuzzy diambil dengan menggunakan data fisik dari pengukuran sensor kamera.

3.7.1. Pengujian Sistem Pada Lingkungan Terstruktur

Data yang digunakan dalam penelitian ini terdiri dari informasi tentang rintangan yang ada lingkungan, yang dideteksi oleh sensor ultrasonik robot *swarm*. Informasi akan membentuk nilai masukan dari lingkungan termasuk koridor, kanan dinding dan kiri-dinding yang memungkinkan robot untuk berinteraksi dengan lingkungan. Data tersebut dikumpulkan dari 3 kondisi lingkungan yang berbeda.

Kondisi 1 : Lingkungan yang telah dikondisikan, robot *swarm* akan bergerak untuk mencapai target, pada kondisi ruangan tertutup dengan rintangan berupa objek benda, lorong sempit untuk melakukan berbagai gerakan: gerak maju dan mundur dengan berbagai kecepatan, belok kiri dan kanan dalam berbagai lingkungan dan sudut kendala yang berbeda.

Kondisi 2 : Lingkungan yang tidak diketahui, robot akan bergerak mencapai target, dengan kemampuan menghindari rintangan dan tabrakan antar robot. Jika tidak ada halangan di depan robot, robot akan bergerak ke target dengan kecepatan maksimum. Tetapi jika terdapat rintangan di depannya robot akan mendeteksi robot akan memperlambat kecepatan.

Kondisi 3 : Lingkungan yang berubah, robot akan bergerak mencapai target, dengan kemampuan menghindari rintangan dan tabrakan antar robot. Jika tidak ada halangan di depan robot, robot akan bergerak ke target dengan kecepatan maksimum dengan tetap menjaga jarak minimum dengan rintangan. Tetapi jika terdapat rintangan di depannya robot akan mendeteksi robot akan memperlambat kecepatan.

Dalam kondisi tersebut, perlu mengevaluasi ukuran lingkungan, perspektif di mana target yang terdeteksi, perilaku khas, sifat rintangan, dan konfigurasi lainnya dari rintangan.

3.7.2. Pengujian Sistem Pada Lingkungan Tidak Terstruktur

Lingkungan tidak terstruktur adalah suatu lingkungan kompleks dengan banyak dinding dan halangan. Halangan pada lingkungan tidak terstruktur dapat bersifat statis dan dinamis. Pada pengujian ini akan dilihat kinerja ketahanan robot dalam menghadapi gangguan yang ada pada lingkungan antara lain:

1. Perubahan kompleksitas lingkungan
2. Adanya kemungkinan kegagalan pada robot *leader*
3. Adanya pergerakan robot *follower* yang dapat dianggap sebagai halangan

3.8. Analisis Kinerja Sistem Kontrol Formasi

Untuk melakukan pengukuran dari kinerja sistem kontrol yang dirancang dilihat dari kemampuan sistem dalam bekerja sesuai kaidah yang telah ada, dan dapat beroperasi pada 3 lingkungan yang dikondisikan yaitu lingkungan sederhana, lingkungan kompleks dan lingkungan tidak terstruktur.

3.8.1. Trajectory

Pada bagian ini akan dibandingkan beberapa kinerja pergerakan robot *leader-follower* dalam berbagai kondisi lingkungan. Untuk melihat apakah sistem kontrol yang di pilih yaitu SLFT2 dan Dinamik PSO menghasilkan kinerja yang baik, maka digunakan kombinasi antara Sistem Logika *Fuzzy* Type-1 untuk pengontrolan pergerakan robot serta PSO dengan algoritma standar dan algoritma GPSO. Hal itu untuk melihat bagaimana kinerja sistem kontrol formasi yang telah diimplementasikan pada kinerja robot *leader-follower* apakah telah sesuai dengan yang diinginkan.

3.8.2. Beban Komputasi

Pada bagian ini akan dibandingkan berbagai metode, untuk melihat berapa besar beban komputasi. Parameter ini akan menjadi sesuatu nilai yang penting jika sistem kontrol formasi akan diimplementasikan pada platform tertanam. Kinerja yang diharapkan dari penelitian ini adalah beban komputasi yang relatif kecil dengan hasil pengontrolan yang baik. Dalam bagian ini akan dilihat bagaimana perubahan parameter lingkungan, metode pengontrolan, kompleksitas lingkungan dapat mempengaruhi beban komputasi.

3.8.3. Ketahanan

Dalam penelitian ini, sistem kontrol formasi harus memiliki kinerja yang baik karena pengontrol berurusan dengan beberapa variasi dan kesalahan selama

operasi seperti variasi pemodelan sistem, variasi gerak, perubahan lingkungan, kesalahan pengukuran dan lain-lain. Hal ini harus dipertimbangkan sebelum model pengendali dirancang sehingga menghasilkan kinerja yang baik. Penentuan besaran ketidakpastian dalam linguistik yang ada pada SLF harus benar-benar diperhatikan dalam perancangan sistem kontrol formasi.

3.8.4. Waktu Pemrosesan Dalam Mencapai Target

Waktu pemrosesan menjadi parameter penting dalam penelitian ini, karena untuk menghasilkan kinerja yang baik waktu yang diperlukan harus relatif cepat. Dalam hal ini robot *swarm* harus mampu bekerja dalam kondisi yang sederhana sampai kondisi yang tidak menentu. Khususnya dalam penelitian ini, dikarenakan sistem bekerja dalam platform sistem tertanam dapat mempengaruhi kecepatan pemrosesan jika tahapan yang diatur dan metode yang dipilih tidak sesuai. Diharapkan dengan metode yang dipilih, waktu pemrosesan dapat lebih cepat sehingga menghasilkan kinerja yang baik.

BAB IV HASIL DAN PEMBAHASAN

4.1. Pendahuluan

Bagian ini akan menjelaskan hasil-hasil yang telah dicapai dalam penelitian ini. Data-data yang telah dihasilkan merupakan data dasar yang harus diolah untuk menghasilkan analisis kinerja sistem kontrol formasi. Hasil yang didapatkan dibagi menjadi tiga bagian yaitu (1) pengontrolan sistem gerak robot *leader-follower* dalam formasi dengan menggunakan SLF; (2) optimisasi route robot *leader-follower* dalam mencapai target menggunakan dinamik PSO; dan (3) penggabungan kedua metode tersebut menjadi SLFT2 dan DPSO dalam pengontrolan formasi robot *leader-follower* dalam mencapai target.

4.2. Sistem Kontrol Pergerakan Robot

4.2.1. Metode SLFT1 dengan 2 Variabel Linguistik

Pada bagian ini menjelaskan tentang bagaimana menghitung nilai dari keluaran SLF yang digunakan oleh robot *swarm*. Metode pengujian menggunakan salah satu model *fuzzy inference* sistem yaitu model mamdani. Dari hasil pengujian berupa nilai defuzzifikasi dari variabel yang telah ditentukan. Pengujian ini untuk mengetahui *output fuzzy* yang dihitung berdasarkan dua variabel *fuzzy* yaitu jarak dan sudut. Pada sistem kontrol formasi model robot *leader-follower* akan dibandingkan 3 struktur *fuzzy* yaitu menggunakan SLFT1 dengan 2 variabel, SLFT1 dengan 3 variabel, dan SLFT2 dengan 2 variabel.

Nilai *default* parameter fungsi keanggotaan masukan menggunakan SLFT1 dengan 2 variabel dapat dilihat pada Tabel 4.1 sebagai berikut:

Tabel 4. 1. Linguistik fungsi masukan dengan 2 variabel

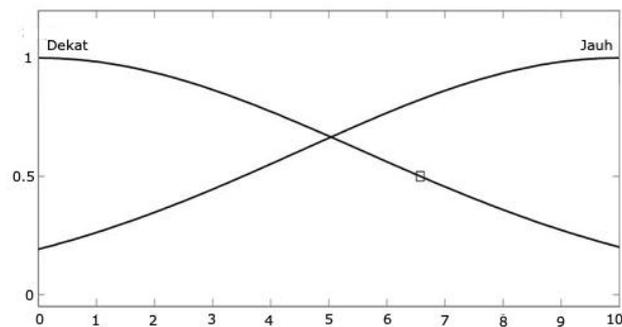
Variable linguistik	Parameter (σ, c)
Dekat	(5.5, 0)
Jauh	(5.5, 10)

Sedangkan nilai *default* parameter fungsi keanggotaan keluaran menggunakan nilai pada Tabel 4.2. sebagai berikut:

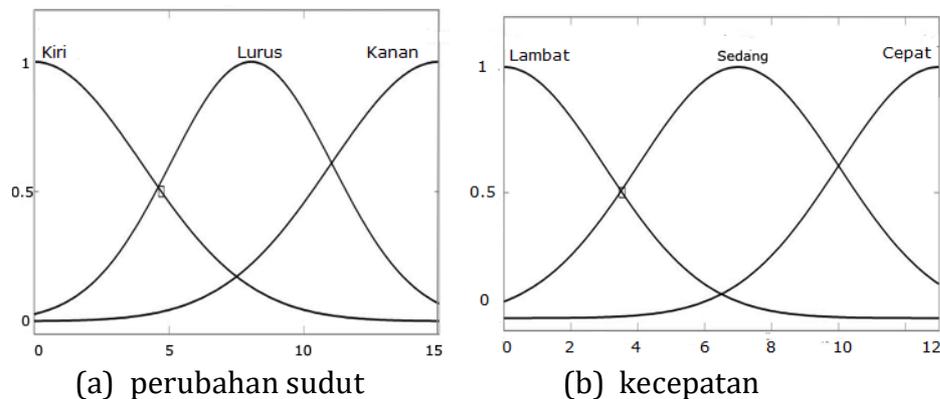
Tabel 4. 2. Linguistik fungsi keluaran SLFT 1 dengan 2 variabel

Keluaran	Variabel linguistik	Parameter (σ, c)
Sudut	Kiri	(4, 0)
	Lurus	(3, 8)
	Kanan	(4, 15)
Kecepatan	Lambat	(3, 0)
	Sedang	(3, 7)
	Cepat	(3, 13)

Setelah menentukan nilai-nilai *default*, kemudian SLFT1 dengan 2 variabel linguistik diterapkan pada perangkat lunak Gambar 4.1. dan Gambar 4.2 menunjukkan fungsi keanggotaan masukan dan keluaran sebagai berikut:



Gambar 4. 1. Fungsi keanggotaan masukan dengan 2 variabel linguistik



Gambar 4. 2. Fungsi keanggotaan keluaran dengan 2 variabel linguistik

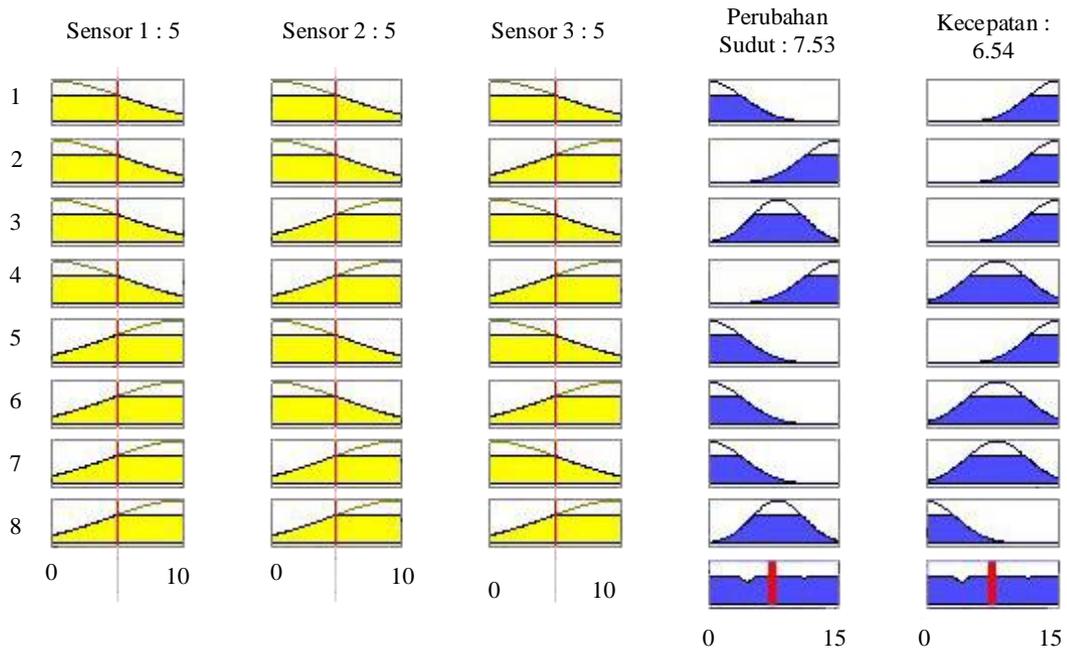
Semua kaidah sistem kontrol formasi dengan SLFT1 yang digunakan pada tiap-tiap pengujian dapat dilihat pada Tabel 4.3.

Tabel 4. 3. Basis kaidah SLFT1 dengan 2 variabel

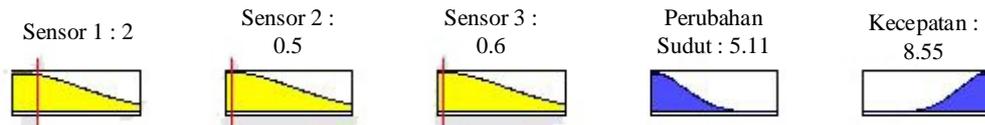
Sensor 0	Sensor1	Sensor2	Kecepatan	Perubahan Sudut
Dekat	Dekat	Dekat	Lambat	Kiri
Dekat	Dekat	Jauh	Lambat	Kanan
Dekat	Jauh	Dekat	Lambat	Lurus
Dekat	Jauh	Jauh	Sedang	Kanan
Jauh	Dekat	Dekat	Lambat	Kiri
Jauh	Dekat	Jauh	Sedang	Kiri
Jauh	Jauh	Dekat	Sedang	Kiri
Jauh	Jauh	Jauh	Cepat	Lurus

Jika dilakukan proses fuzzifikasi maka akan didapat perubahan nilai crisp ke nilai *fuzzy*. Dari seluruh kaidah yang ada, maka dapat diambil beberapa bagian kaidah yang aktif sesuai dengan kondisi pendeteksian sensor jarak. Mekanisme *inference* setiap basis kaidah keluaran yang aktif pada SLFT1 dengan 2 variabel linguistik dapat dilihat dari Gambar 4.3 dan keluarannya dapat dilihat pada Gambar 4.4.

Berikut ini akan diberikan diberikan satu contoh kasus penyelesaian dengan menggunakan algoritma SLFT1 dengan 2 variabel linguistik. Jika masukkan sensor 1 = 2 adalah kondisi masukkan sensor 2 = 0.5 adalah kondisi **Dekat**, dan masukkan sensor 3 = 0.6 adalah kondisi **Dekat**. Maka kaidah yang aktif sesuai dengan basis kaidah yang telah ditetapkan (Tabel 4.3) adalah sebagai berikut: **IF** sensor 1 = dekat **and** sensor 2 = dekat **and** sensor 3 = dekat **THEN** keluaran arah sudut senilai 5.11 dengan keluaran lurus **and** kecepatan senilai 8.55 dengan keluaran lambat. Hasil defuzzifikasi dapat dilihat pada Gambar 4.4.



Gambar 4. 3. Proses fuzzifikasi dengan 2 variabel linguistik



Gambar 4. 4. Hasil defuzzifikasi dengan 2 variabel linguistik

4.2.2. Algoritma SLFT1 dengan 3 Variabel Linguistik

Nilai *default* parameter fungsi keanggotaan masukan menggunakan nilai pada Tabel 4.4. sebagai berikut:

Tabel 4. 4. Linguistik fungsi keanggotaan masukan dengan 3 variabel

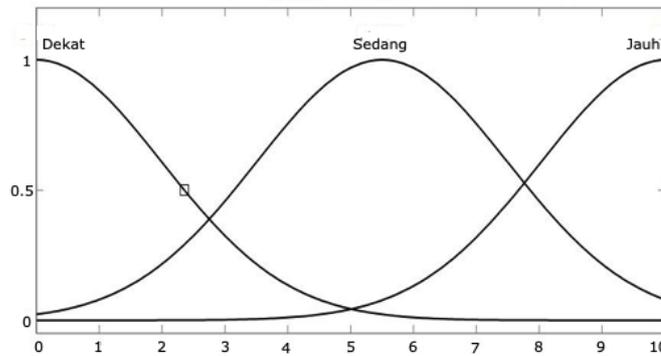
Variabel linguistik	Parameter (σ, c)
Dekat	(2, 0)
Sedang	(2, 5.5)
Jauh	(2, 10)

Sedangkan nilai *default* parameter fungsi keanggotaan keluaran menggunakan nilai pada Tabel 4.5 sebagai berikut:

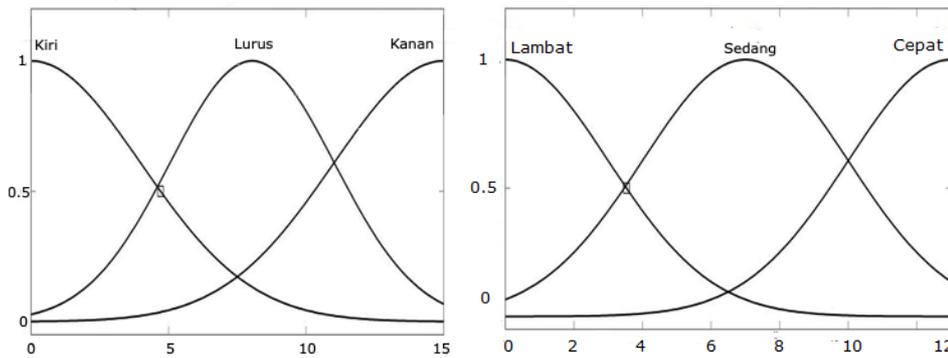
Tabel 4. 5. Linguistik fungsi keanggotaan keluaran dengan 3 variabel

Keluaran	Variable linguistik	Parameter (σ, c)
Sudut	Kiri	(4, 0)
	Lurus	(3, 8)
	Kanan	(4, 15)
Kecepatan	Lambat	(3, 0)
	Sedang	(3, 7)
	Cepat	(3, 13)

Setelah menentukan nilai-nilai default, kemudian SLFT1 dengan 3 linguistik diterapkan pada perangkat lunak dan dapat digambar sebagai fungsi masukan dan fungsi keluaran seperti Gambar 4.5 dan Gambar 4.6 sebagai berikut:



Gambar 4. 5. Fungsi keanggotaan masukan



(a) Perubahan sudut

(b) Kecepatan

Gambar 4. 6. Fungsi keanggotaan keluaran

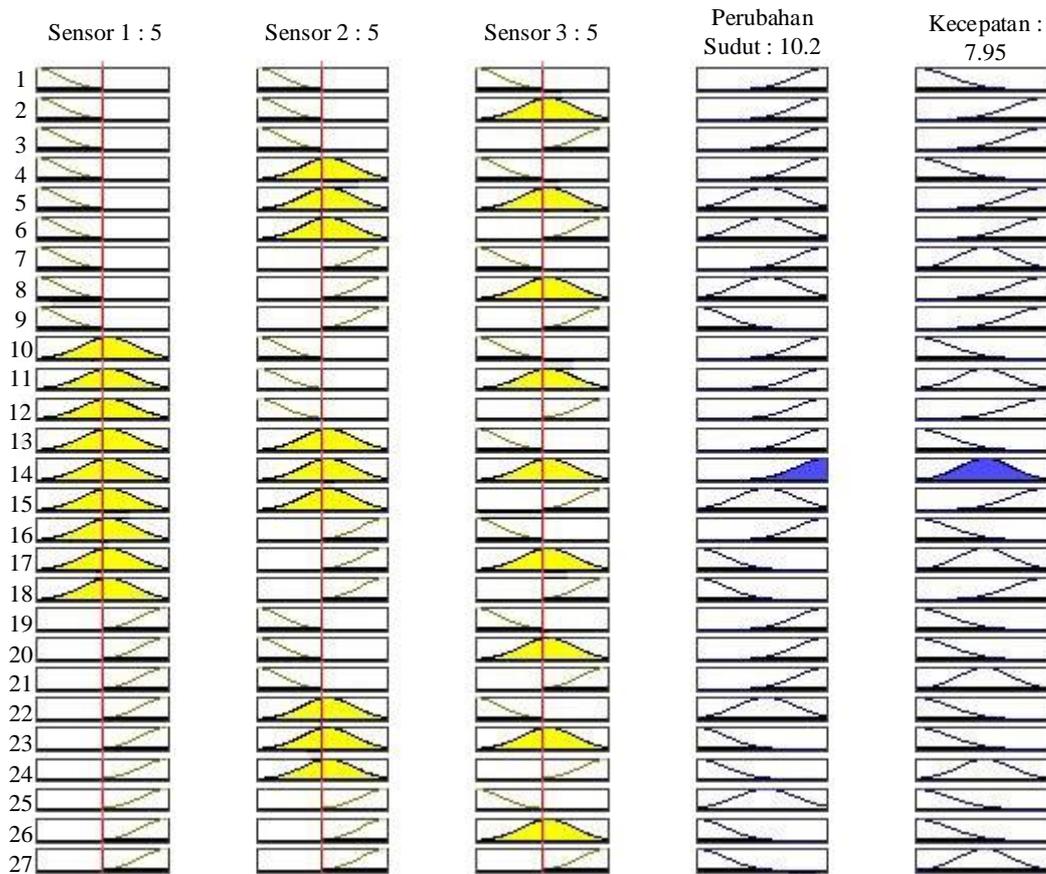
Basis kaidah untuk sistem kontrol pergerakan yang digunakan pada tiap-tiap pengujian dapat dilihat pada Tabel 4.6 berikut:

Tabel 4. 6. Basis kaidah dengan 3 variabel linguistik

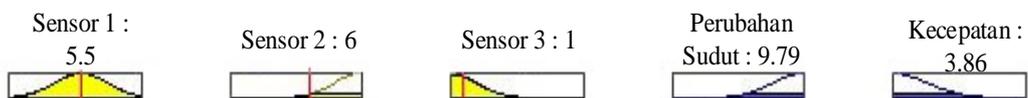
Sensor0	Sensor1	Sensor2	Kecepatan	Perubahan Sudut
dekat	dekat	dekat	lambat	kiri
dekat	dekat	sedang	lambat	kanan
dekat	dekat	jauh	lambat	kanan
dekat	sedang	dekat	lambat	kiri
dekat	sedang	sedang	sedang	kanan
dekat	sedang	jauh	sedang	kanan
dekat	jauh	dekat	lambat	lurus
dekat	jauh	sedang	sedang	kanan
dekat	jauh	jauh	cepat	kanan
sedang	dekat	dekat	lambat	kiri
sedang	dekat	sedang	lambat	lurus
sedang	dekat	jauh	lambat	kanan
sedang	sedang	dekat	lambat	kiri
sedang	sedang	sedang	sedang	lurus
sedang	sedang	jauh	sedang	kanan
sedang	jauh	dekat	lambat	kiri
sedang	jauh	sedang	cepat	lurus
sedang	jauh	jauh	cepat	kanan
jauh	dekat	dekat	lambat	kiri
jauh	dekat	sedang	lambat	kiri
jauh	dekat	jauh	lambat	lurus
jauh	sedang	dekat	sedang	kiri
jauh	sedang	sedang	cepat	kiri
jauh	sedang	jauh	cepat	lurus
jauh	jauh	dekat	sedang	kiri
jauh	jauh	sedang	cepat	kiri
jauh	jauh	jauh	cepat	lurus

Mekanisme mesin inferensi untuk setiap basis kaidah dari keluaran yang aktif pada SLFT1 dengan 3 variabel linguistik dapat dilihat dari Gambar 4.7 dan keluarannya dapat dilihat pada Gambar 4.8. Berikut diberikan satu contoh kasus dan penyelesaian menggunakan algoritma SLFT1 dengan 3 variabel linguistik.

Jika masukkan sensor 1 = 5.5 adalah kondisi **Sedang**, masukkan sensor 2 = 6 adalah kondisi **Jauh**, dan Masukkan sensor 3 = 1 adalah kondisi **Dekat**. Maka kaidah yang aktif sesuai dengan basis kaidah yang telah ditetapkan (Tabel 4.6) adalah sebagai berikut: **IF** sensor 1 = sedang **and** sensor 2 = jauh **and** sensor 3 = dekat **THEN** keluaran arah sudut senilai 3.86 dengan keluaran lurus **and** kecepatan senilai 9.79 dengan keluaran lambat. Hasil defuzzifikasi dapat dilihat pada Gambar 4.8.



Gambar 4. 7. Basis kaidah SLFT1



Gambar 4. 8. Hasil mekanisme mesin inference SLFT 1 dengan 3 variabel linguistik

4.2.3. Algoritma SLFT2

Nilai *default* parameter fungsi keanggotaan masukan menggunakan nilai pada Tabel 4.7 sebagai berikut:

Tabel 4. 7. Linguistik fungsi keanggotaan masukan

Variabel linguistik	Parameter (σ, c)
dekat	(0, 0.05, 0.5)
jauh	(0.95, 1, 0.05)

Sedangkan nilai *default* parameter fungsi keanggotaan keluaran menggunakan nilai pada Tabel 4.8 sebagai berikut:

Tabel 4. 8. Linguistik fungsi keanggotaan keluaran

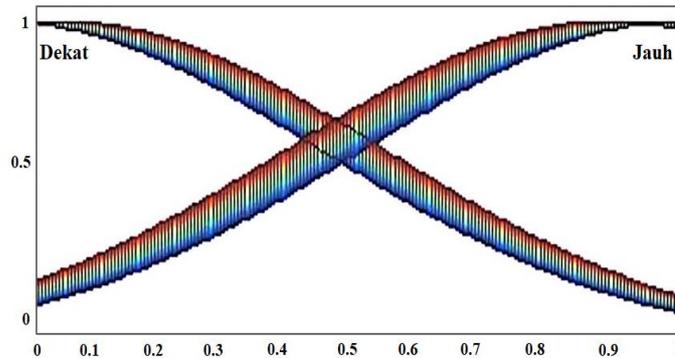
Keluaran	Variabel linguistik	Parameter (σ, c)
Sudut	kiri	(0, 0.05, 0.3)
	lurus	(0.45, 0.5, 0.3)
	kanan	(0.95, 1, 0.3)
Kecepatan	lambat	(-0.05, 0.05, 0.5)
	sedang	(0.45, 0.55, 0.5)
	cepat	(0.95, 1, 0.5)

Kaidah untuk sistem kontrol formasi yang digunakan pada tiap-tiap pengujian dapat dilihat seperti SLFT1 dengan 2 variabel pada Tabel 4.3. Setelah menentukan nilai-nilai *default*, kemudian algoritma SLFT2 diterapkan pada perangkat lunak dengan fungsi keanggotaan masukan fungsi *Gaussian* berikut,

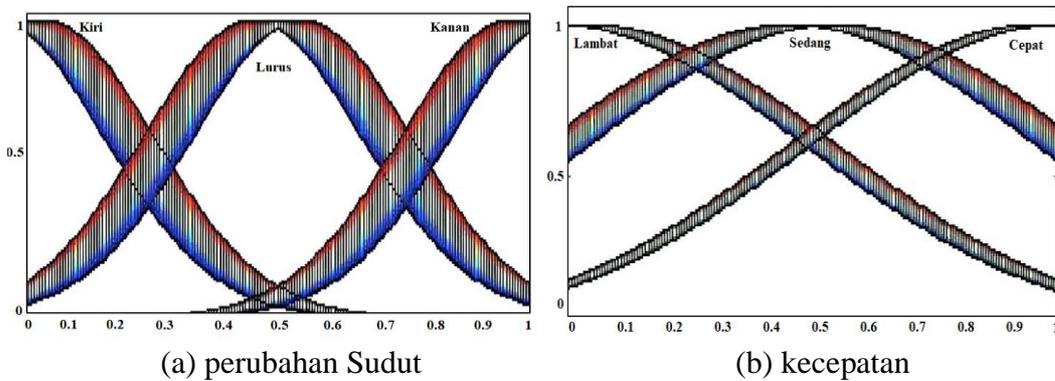
$$\bar{\mu}_x = \begin{cases} (c_1, \sigma; x), & x < c_1 \\ 1, & c_1 \leq x \leq c_2 \\ (c_2, \sigma; x), & x > c_2 \end{cases} \quad \underline{\mu}_x = \begin{cases} (c_2, \sigma; x), & x \leq \frac{c_1+c_2}{2} \\ (c_1, \sigma; x), & x > \frac{c_1+c_2}{2} \end{cases}$$

Dimana $(c_1, \sigma; x) = e^{-\frac{1}{2}\left(\frac{x+c_1}{\sigma}\right)^2}$ and $(c_2, \sigma; x) = e^{-\frac{1}{2}\left(\frac{x+c_2}{\sigma}\right)^2}$

Gambar 4.9 menunjukkan fungsi keanggotaan masukan dan Gambar 4.10 menunjukkan fungsi keanggotaan keluaran sebagai berikut:

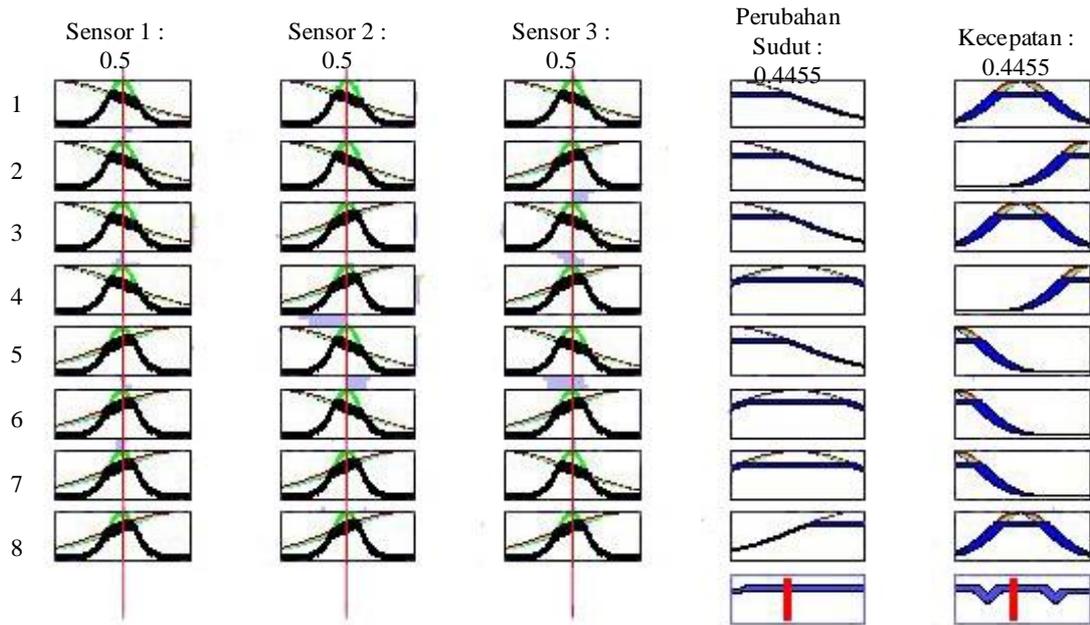


Gambar 4. 9. Fungsi Keanggotaan Masukan

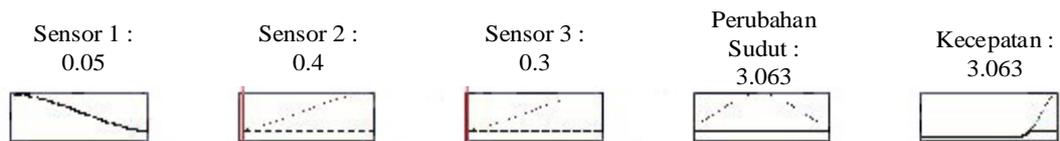


Gambar 4. 10 Fungsi Keanggotaan Keluaran

Pada proses inferensi maka dapat dilihat basis kaidah yang aktif untuk memproses pengontrolan output, pada Gambar 4.11. Dianggap masukan berasal dari sensor dan keluaran adalah pengontrolan gerak aktuator. Pada bagian ini akan diberikan satu contoh kasus penyelesaian dengan menggunakan algoritma SLFT 2 dalam bentuk simulasi pemrosesan *fuzzy* dengan 2 variabel linguistik sebagai berikut: Jika masukkan sensor 1 = 0.05 adalah kondisi **Dekat**, masukkan sensor 2 = 0.4 adalah kondisi **Jauh**, dan Masukkan sensor 3 = 0.3 adalah kondisi **Jauh**. Maka kaidah yang aktif berdasarkan basis kaidah yang telah ditetapkan (Tabel 4.3) adalah sebagai berikut: **IF** sensor 1 = dekat **and** sensor 2 = jauh **and** sensor 3 = jauh **THEN** keluaran arah sudut senilai 3.063 dengan keluaran sedang **and** kecepatan senilai 3.063 dengan keluaran kanan. Hasil *defuzzifikasi* dapat dilihat pada Gambar 4.12.



Gambar 4. 11. Proses fuzzifikasi algoritma SLFT2



Gambar 4. 12. Hasil defuzzifikasi algoritma SLFT 2

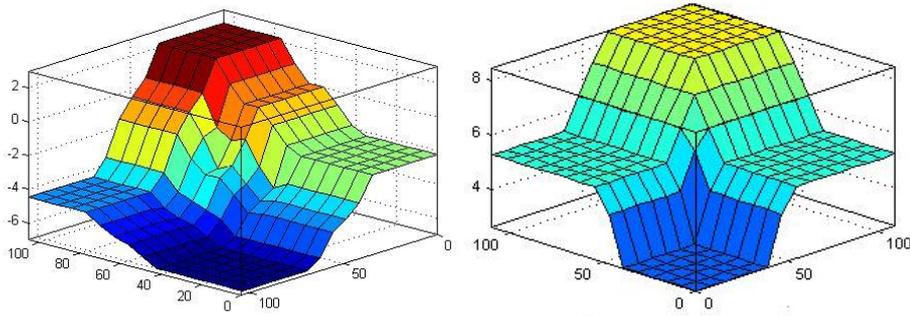
4.3. Kinerja Sistem Kontrol Formasi Dalam Mengatur Pergerakan

4.3.1. Analisis *Surfaces* SLF

Terlihat pada Gambar 4.13 (c) bahwa pengendali dengan SLFT2 interval memiliki permukaan yang halus hal ini ditandai dengan banyaknya lereng-lereng pada *surface*, dimana tiap-tiap lereng menggambarkan penurunan besar perubahan sudut yang bertahap, sedangkan pada *surface* dari SLFT1 dengan 8 kaidah (Gambar 4.13(b)), terlihat bahwa besarnya keluaran berupa perubahan sudut berubah-ubah dengan curam. Hal yang sama dimiliki oleh *surface* dari SLFT2 dengan 27 kaidah, walaupun memiliki lebih banyak lereng, tetapi perubahan sudutnya tetap curam pada beberapa titik seperti pada Gambar 4.13 (c).

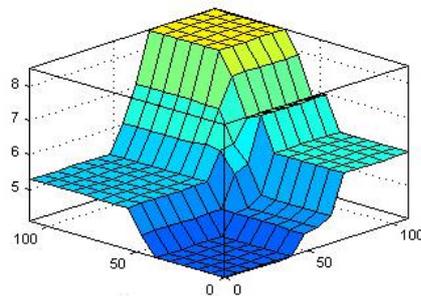
Permukaan yang lebih halus menandakan kemampuan SLFT2 mewakili lebih banyak kaidah, dan memberikan hasil kecepatan yang berubah secara bertahap, dibandingkan dengan SLFT1 dengan 8 kaidah dan 27 kaidah yang

memiliki lereng yang curam. Lereng-lereng yang curam pada *surface* menandakan hasil kecepatan yang berubah secara drastis saat melakukan aksi perubahan.



(a) SLFT2 dengan 8 kaidah

(b) SLFT1 dengan 8 kaidah



(c) SLFT1 dengan 27 kaidah

Gambar 4. 13. *Surfaces* SLFT1 dan SLFT2

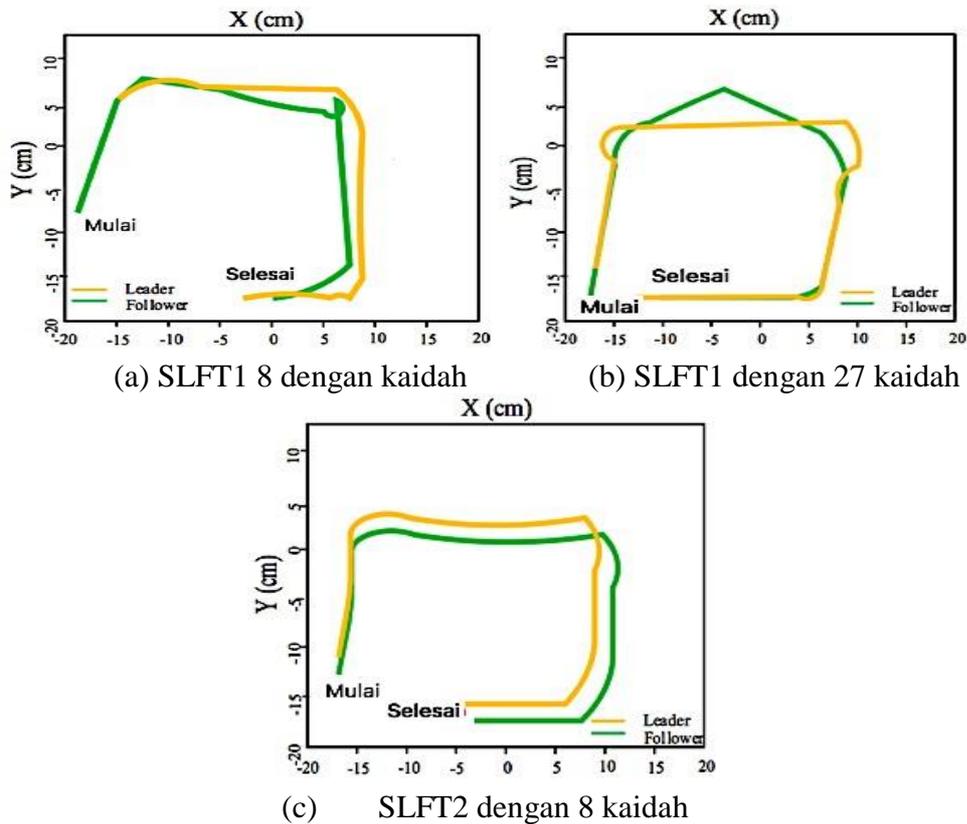
4.3.2. Pengontrolan Pergerakan Robot *Leader-Follower*

4.3.2.1. Lingkungan Sederhana

Untuk melihat efektif atau tidaknya sistem logika *fuzzy* yang digunakan untuk pengontrolan pergerakan robot *leader-follower*, maka sistem kontrol di uji dalam berbagai lingkungan. Pada penelitian ini metode yang diusulkan akan diuji dalam berbagai kondisi lingkungan yaitu lingkungan terbuka, lingkungan kompleks dan lingkungan tidak terstruktur. Dapat di lihat pada Gambar 4.14 (a) – (c) kinerja sistem kontrol pergerakan robot *leader-follower* dengan menggunakan SLFT1 dengan 8 kaidah dan 27 kaidah serta SLFT2 dengan 8 kaidah.

Hasil yang terlihat adalah bahwa kinerja sistem kontrol Logika *Fuzzy* terhadap pergerakan robot *leader-follower* sangat bervariasi. Dari ketiga gambar tersebut kinerja SLFT2 memperlihatkan hasil yang lebih baik dibandingkan kedua

hasil lainnya. Trajektori robot *leader* yang dihasilkan ketika menggunakan SLFT2 sangat halus, begitu juga robot *follower* mengikuti dengan pergerakan yang halus. Berbeda dengan SLFT1 baik 8 kaidah maupun 27 kaidah pergerakan robot *leader-follower* menghasilkan *trajectory* yang tidak halus, dan menghasilkan perubahan posisi dan orientasi yang tidak sama. Sehingga pergerakannya pun menjadi berbeda nilainya.



Gambar 4. 14. Kinerja sistem logika *fuzzy* untuk pengontrolan gerak dalam formasi pada lingkungan sederhana

Dari data yang didapatkan, dengan menggunakan SLFT1 dengan 8 kaidah membangkitkan 1931 data untuk robot *leader* dan 1945 data untuk robot *follower*. Sedangkan dengan menggunakan SLFT1 dengan 27 kaidah, data yang dibangkitkan untuk robot *leader* 1074 dan untuk robot *follower* 1077. Namun ketika SLFT2 yang dipergunakan, robot *leader* membangkitkan hanya 291 data dan robot *follower* 304 data. Dari data tersebut, dapat disimpulkan bahwa dalam menyelesaikan proses pengontrolan SLFT2 menghasilkan kinerja yang baik,

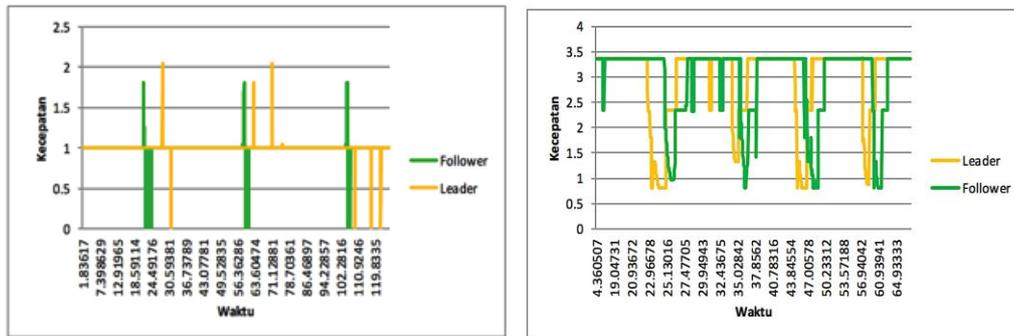
dalam hal pergerakan dan proses pengontrolan. Resume kinerja SLF dalam mengontrol pergerakan robot *leader-follower* dapat dilihat pada Tabel 4.9 berikut ini.

Tabel 4. 9. Perbandingan kinerja sistem kontrol formasi dengan perbedaan jumlah kaidah

Jenis Pengontrol pergerakan	Jumlah data yang dibangkitkan	Waktu pemrosesan <i>fuzzy</i> (detik)	Komputasi (<i>kbytes</i>)
<i>Robot Leader</i>			
SLFT1 (8 kaidah)	1931	127.38	100
SLFT1 (27 kaidah)	1074	68.3	60
SLFT2 (8 kaidah)	291	78.1	17
<i>Robot Follower</i>			
SLFT1 (8 kaidah)	1945	130.4	105
SLFT1 (27 kaidah)	1077	68.6	64
SLFT2 (8 kaidah)	304	80.3	16

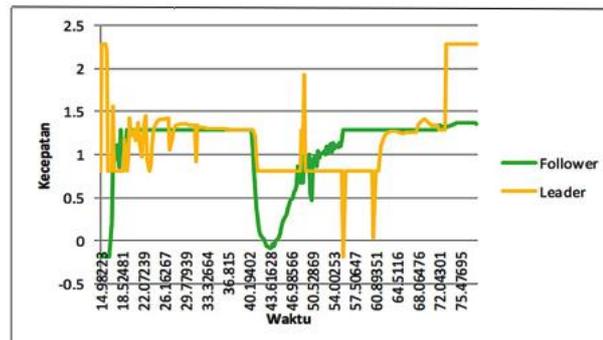
- Respon Waktu Perubahan Kecepatan

Sistem logika *fuzzy* dalam proses pengontrolannya mengatur kecepatan motor. Dalam model SLF pada penelitian ini, input berasal dari sensor dan output adalah hasil pengontrolan aktuator. Dari Gambar 4.14 (a), (b) dan (c) yaitu pergerakan robot *leader-follower* pada lingkungan sederhana maka di hasilkan respon waktu SLF saat melakukan proses perubahan kecepatan seperti pada Gambar 4.15 (a) (b) dan (c). Terlihat bahwa dengan SLFT1 dengan 8 kaidah, perubahan kecepatan saat robot melakukan pergerakan sangat kasar dengan pengaturan yang terlampau lebar (Gambar 4.15 (a)). Berbeda jika menggunakan SLT1 dengan 27 kaidah dan SLFT2 dengan 8 kaidah. Perubahan kecepatan lebih rapat dan halus, sehingga pengontrolan sistem gerak menghasilkan kinerja yang lebih baik. Jika dilihat waktu tempuh untuk mencapai target, SLFT1 dengan 8 kaidah dalam waktu 119.83 detik, SLFT1 dengan 27 kaidah dalam waktu 64.93 detik dan SLFT2 dengan 8 kaidah dalam waktu 75.47 detik. Lebih lambat dari SLFT1 dengan 27 kaidah, tetapi *trajectory* pergerakan lebih baik dalam hal kehalusan pergerakan robot.



(a) SLFT1 dengan 8 kaidah

(b) SLFT1 dengan 27 kaidah



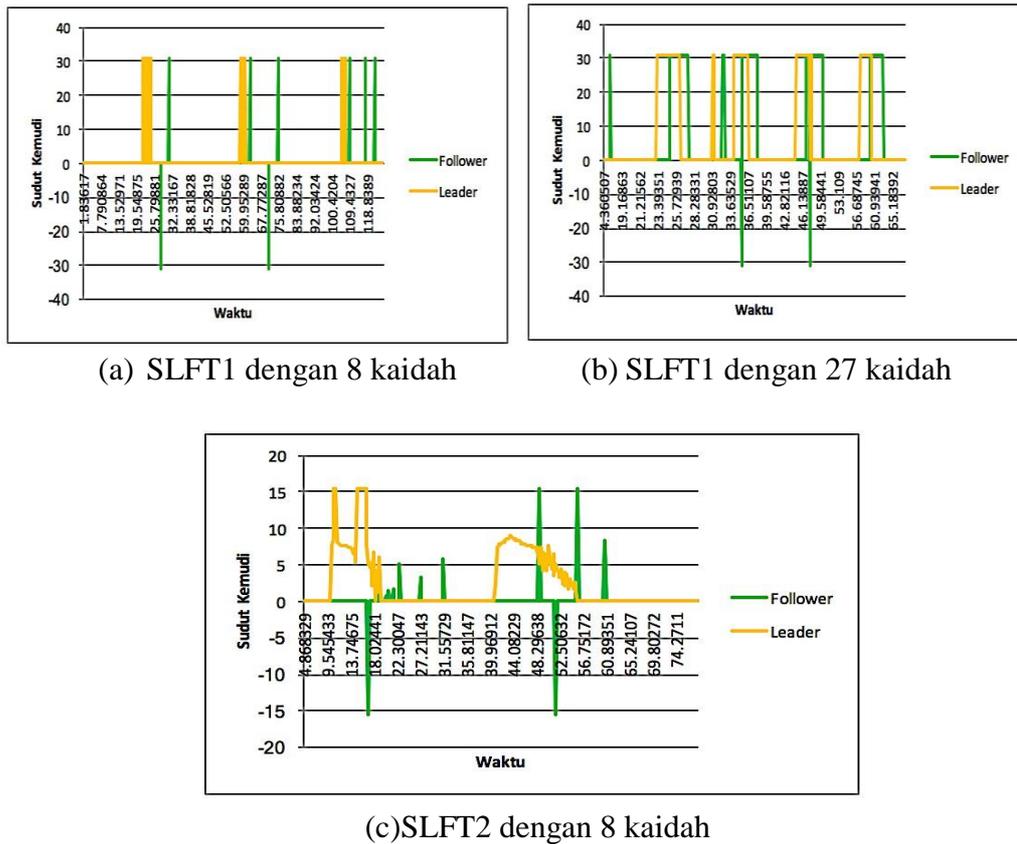
(c) SLFT2 dengan 8 kaidah

Gambar 4. 15. Respon waktu kecepatan menggunakan SLFT1 dan SLFT2 pada lingkungan sederhana

- Respon Waktu Perubahan Sudut Kemudi

Sistem logika *fuzzy* dalam mengatur sudut kemudi robot *leader* dan akan diikuti oleh robot *follower*. Dari Gambar 4.14 (a), (b) dan (c) yaitu pergerakan robot *leader-follower* pada lingkungan sederhana maka respon waktu yang dihasilkan saat melakukan proses perubahan sudut kemudi seperti pada Gambar 4.16 (a) (b) dan (c). Terlihat bahwa dengan SLFT1 dengan 8 kaidah, menghasilkan perubahan sudut kemudi robot sangat kasar dengan pengaturan yang terlampaui lebar (Gambar 4.16 (a)). Berbeda jika menggunakan SLT1 dengan 27 kaidah dan SLFT2 dengan 8 kaidah. Perubahan sudut kemudi lebih halus dengan perubahan sudut kemudi yang lebih sempit, sehingga pengontrolan sistem gerak menghasilkan kinerja yang lebih baik (Gambar 4.16 (b) dan (c)). Sudut maksimum dari sistem aktuator dengan SLFT1 dengan 8 kaidah maksimum 30° dengan perubahan yang sangat lebar dan kasar. Sedangkan menggunakan SLFT1 27 kaidah sudut kemudi tetap sama hanya saja pengaturannya lebih rapat dan

lebih halus. Sedangkan dengan SLFT2 dengan 8 kaidah perubahan sudut maksimum sebesar 15° lebih halus dari SLFT1 yaitu 30° . Hal tersebut menunjukkan bahwa robot *leader-follower* menggunakan SLFT2 pada lingkungan sederhana menghasilkan kinerja pergerakan yang baik, dengan perubahan kecepatan dan sudut kemudi robot yang lebih halus dibandingkan SLFT1.

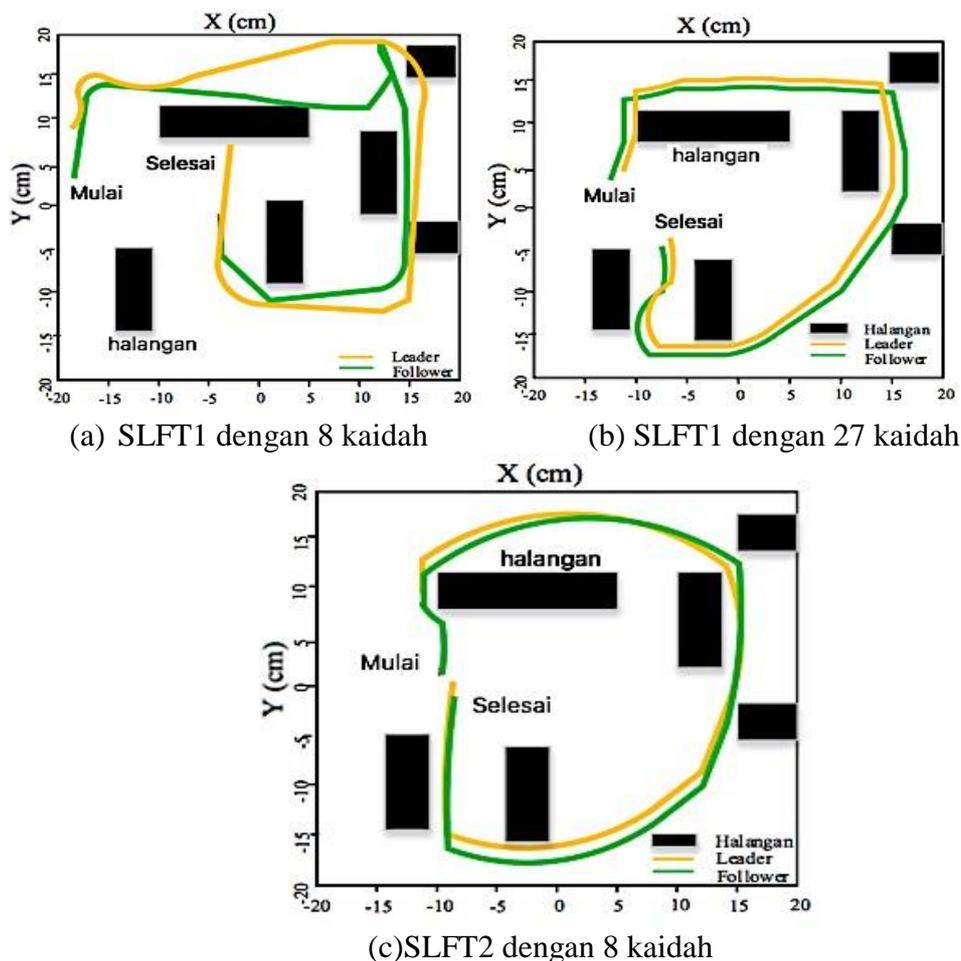


Gambar 4. 16. Respon waktu sudut kemudi robot menggunakan SLFT1 dan SLFT2 pada lingkungan sederhana

4.3.2.2. Lingkungan Kompleks 1

Pada penelitian ini sistem kontrol untuk menjamin pergerakan robot *leader-follower* tetap dalam formasi di uji pada lingkungan kompleks. Lingkungan yang dimaksud memiliki 6 penghalang yang tidak terstruktur. Dengan menggunakan SLF diharapkan robot tetap mampu bergerak dalam formasi dengan kinerja yang baik untuk mencapai target. Seperti pengujian sebelumnya SLF akan di rubah jumlah basis kaidahnya dan melihat respon yang dihasilkan oleh aktuator.

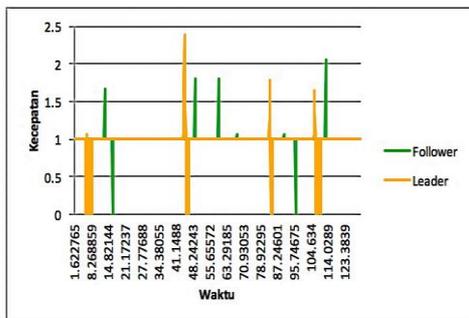
Dari 4.17 (a) – (c) menggambarkan pergerakan robot *leader-follower* pada lingkungan kompleks. Dengan menggunakan SLFT1 dengan 8 kaidah robot *leader* gagal menyelesaikan tugasnya dan robot *follower* juga mengalami hal yang sama menabrak dinding pembatas dan halangan (Gambar 4.17 (a)). Namun berbeda sistem kontrol menghasilkan kinerja yang lebih baik dengan menggunakan SLFT1 dengan 27 kaidah dan SLFT2 dengan 8 kaidah. Pergerakan robot *leader-follower* menjadi lebih baik tanpa terjadi tabrakan. Tetapi dengan menggunakan SLFT2 robot *leader* lebih menjaga jarak pergerakan lebih baik dibandingkan SLFT1 yang bergerak lebih mendekati dinding penghalang (Gambar 4.17 (b) dan (c)).



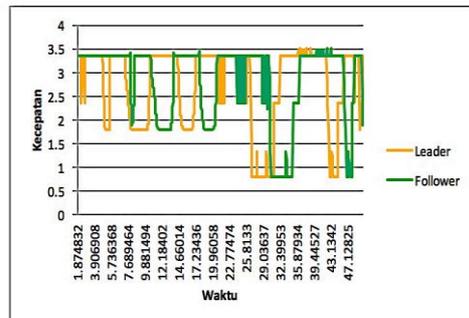
Gambar 4. 17. Kinerja sistem logika *fuzzy* untuk pengontrolan gerak dalam formasi pada lingkungan kompleks

- Respon Waktu Perubahan Kecepatan

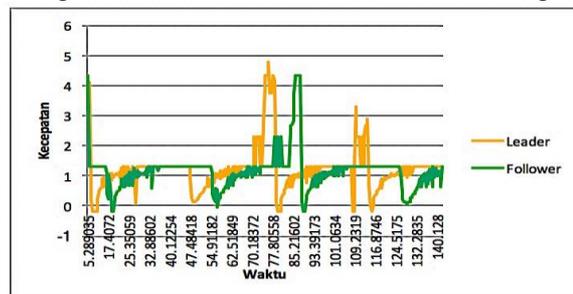
Sistem logika *fuzzy* dalam proses pengontrolannya mengatur kecepatan motor. Dalam model SLF pada penelitian ini, input berasal dari sensor dan output adalah hasil pengontrolan aktuator. Dari Gambar 4.17 (a), (b) dan (c) yaitu pergerakan robot *leader-follower* pada lingkungan sederhana maka di hasilkan respon waktu SLF saat melakukan proses perubahan kecepatan seperti pada Gambar 4.18 (a) (b) dan (c). Terlihat bahwa dengan SLFT1 dengan 8 kaidah, perubahan kecepatan saat robot melakukan pergerakan sangat kasar dengan pengaturan yang terlampaui lebar (Gambar 4.15 (a)). Berbeda jika menggunakan SLT1 dengan 27 kaidah dan SLFT2 dengan 8 kaidah. Perubahan kecepatan lebih halus, sehingga pengontrolan sistem gerak menghasilkan kinerja yang lebih baik. Jika dilihat waktu tempuh untuk mencapai target, SLFT1 dengan 8 kaidah dalam waktu 123.38 detik, SLFT1 dengan 27 kaidah dalam waktu 47.13 detik dan SLFT2 dengan 8 kaidah dalam waktu 140.13 detik. SLFT2 menghasilkan pergerakan yang lebih lambat dari SLFT1 dengan 27 kaidah, tetapi *trajectory* pergerakan lebih baik, karena SLFT1 dengan 27 kaidah robot *follower* tidak dapat menjaga jarak dengan dinding dan menabrak dinding pembatas. Sedangkan robot *follower* menabrak dinding ketika mendekati target.



(a) SLFT1 dengan 8 kaidah



(b) SLFT1 dengan 27 kaidah



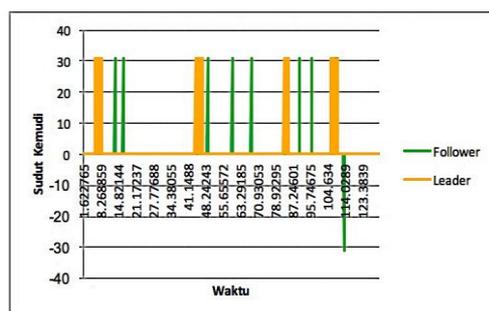
(c) SLFT2 dengan 8 kaidah

Gambar 4. 18. Respon waktu kecepatan menggunakan SLFT1 dan SLFT2 pada lingkungan kompleks

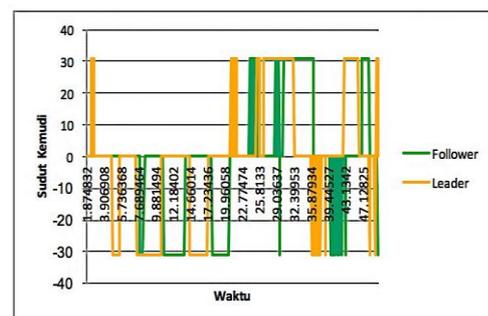
- Respon Waktu Perubahan Sudut Kemudi

Sudut kemudi robot *leader-follower* harus dikontrol untuk menghasilkan pergerakan yang baik dalam tujuan untuk menghindari halangan dan mencapai target. Dari Gambar 4.17 (a), (b) dan (c) yaitu pergerakan robot *leader-follower* pada lingkungan kompleks, sedangkan respon waktu yang dihasilkan saat melakukan proses perubahan sudut kemudi seperti pada Gambar 4.19 (a) (b) dan (c). Terlihat bahwa dengan SLFT1 dengan 8 kaidah, perubahan sudut kemudi robot sangat kasar dengan pengaturan yang terlampau lebar (Gambar 4.19 (a)). Respon perubahan sudut kemudi yang dihasilkan berbeda jika menggunakan SLFT1 dengan 27 kaidah dan SLFT2 dengan 8 kaidah. Perubahan sudut lebih rapat dan halus, sehingga pengontrolan sistem gerak menghasilkan kinerja yang lebih baik (Gambar 4.19 (b) dan (c)).

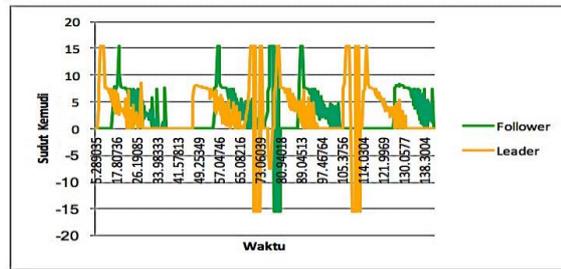
Sudut maksimum dari sistem aktuator dengan SLFT1 dengan 8 kaidah maksimum 30° dengan perubahan yang sangat lebar dan kasar. Sedangkan menggunakan SLFT1 27 kaidah sudut kemudi tetap sama hanya saja pengaturannya lebih rapat dan lebih halus. Sedangkan dengan SLFT2 dengan 8 kaidah sudut maksimum 15° dengan perubahan yang sangat rapat dan halus. Hal tersebut menunjukkan bahwa robot *leader-follower* menggunakan SLFT2 pada lingkungan kompleks menghasilkan kinerja pergerakan yang baik, dengan perubahan kecepatan dan sudut kemudi robot yang sangat halus.



(a) SLFT1 8 kaidah



(b) SLFT1 27 kaidah

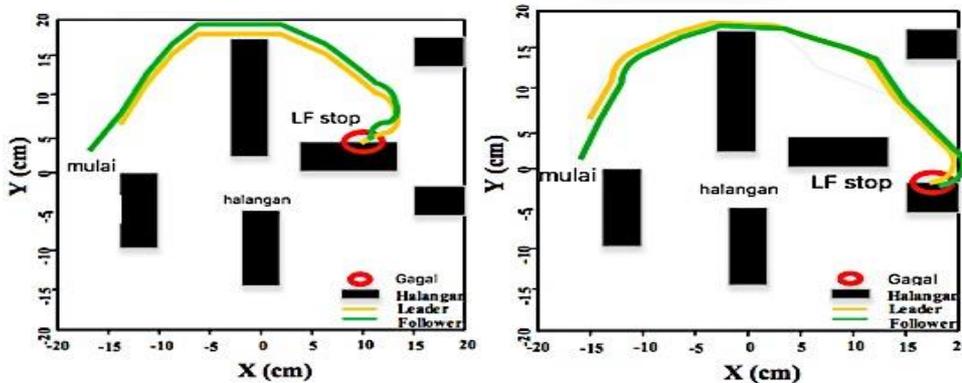


(c)SLFT2 8 kaidah

Gambar 4. 19. Respon waktu sudut kemudi menggunakan SLFT1 dan SLFT2 pada lingkungan kompleks

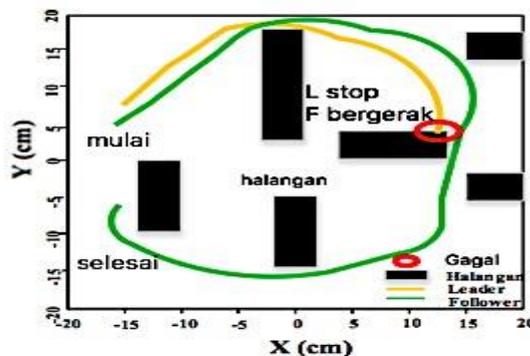
4.3.2.3. Lingkungan Kompleks 2

Pada bagian ini sistem kontrol formasi yang diusulkan diuji untuk menjamin pergerakan robot *leader-follower* pada lingkungan kompleks yang lainnya. Lingkungan yang dimaksud memiliki 6 penghalang yang tidak terstruktur dan diharapkan robot tetap mampu bergerak dalam formasi dengan kinerja yang baik untuk mencapai target. Seperti pengujian sebelumnya SLF akan di rubah jumlah kaidahnya, jenis pengontrolnya dan melihat respon yang dihasilkan oleh aktuator.



(a) SLFT2 dengan 8 kaidah

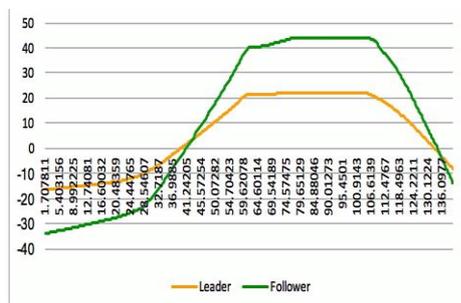
(b) SLFT2 dengan 27 kaidah



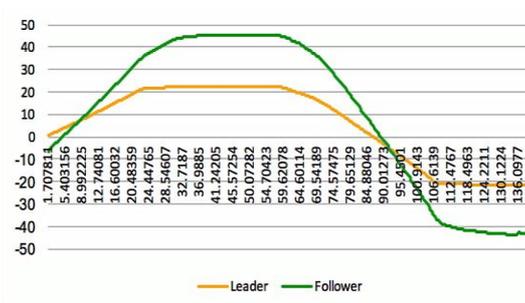
(c)SLFT2 dengan 8 kaidah

Gambar 4. 20. Kinerja sistem kontrol formasi pada lingkungan tidak terstruktur

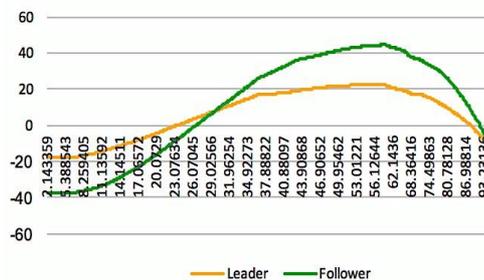
Dari Gambar 4.20 (a) – (c) menggambarkan pergerakan robot *leader-follower* pada lingkungan kompleks. Dengan menggunakan SLFT1 dengan 8 kaidah robot *leader* gagal menyelesaikan tugasnya dan robot *follower* juga mengalami hal yang sama menabrak dinding pembatas dan halangan dan tidak bisa melanjutkan aksinya (Gambar 4.20 (a)). Hal tersebut juga berlaku bagi sistem kontrol formasi dengan metode SLFT1 dengan 27 kaidah, robot *leader-follower* bergerak dan menabrak dinding serta penghalang dan berhenti bergerak. Sehingga dapat dikatakan sistem gagal melaksanakan tugasnya (Gambar 4.20 (b)). Namun berbeda sistem kontrol menggunakan SLFT2 dengan 8 kaidah. Robot *leader-follower*, hanya robot *leader* yang menabrak dinding tetapi robot *follower* mampu menghindari halangan dan mencapai target dengan pergerakan yang baik (Gambar 4.20 (c)). Pada bagian ini memperlihatkan dengan adanya fungsi keanggotaan bersifat *fuzzy* pada SLFT2, sistem kontrol dapat beraksi dengan mencari nilai terbaik dari output, dan dapat memperbaiki kondisi yang tidak terprediksi. Sehingga dapat dikatakan SLFT2 lebih *robust* (kokoh) dibandingkan dengan SLFT1 dalam kondisi lingkungan yang tidak terstruktur.



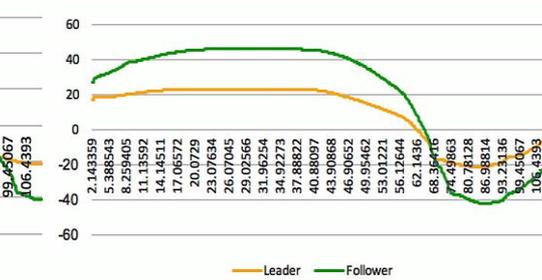
(a) posisi x(t) (SLFT1)



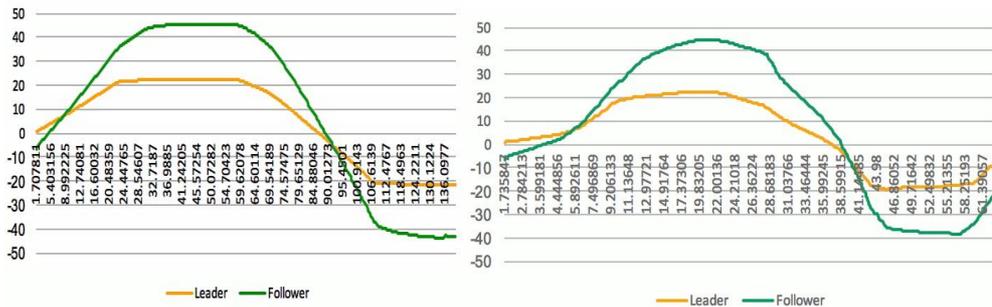
(b) posisi y(t) (SLFT1)



(c) posisi x(t) (SLFT2)



(d) posisi y(t) (SLFT2)

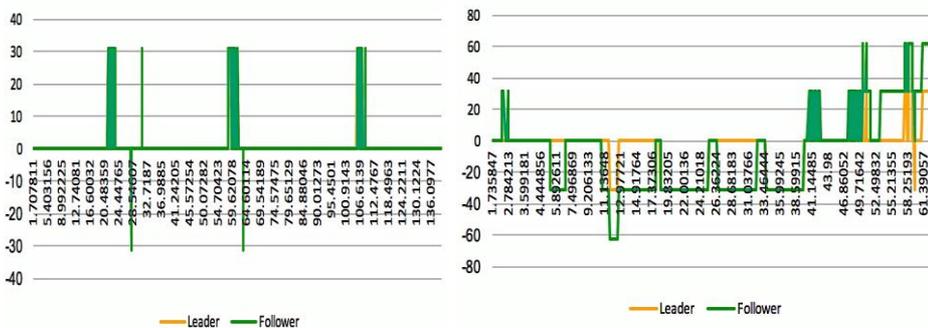


(e)posisi x(t) (SLFT1)

(f) posisi y(t) (SLFT1)

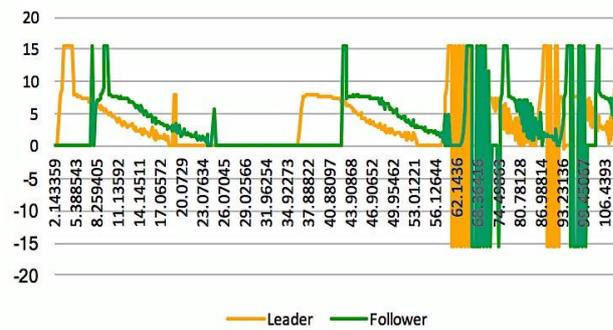
Gambar 4. 21. Kinerja posisi robot *leader-follower* dalam lingkungan kompleks 2 dengan SLFT1 8 kaidah, SLFT1 27 kaidah, dan SLFT2 8 kaidah

Kinerja lain yang dapat di analisis untuk melihat kekokohan dari sistem kontrol formasi SLFT2 adalah respon robot *leader-follower* terhadap posisi dan orientasi. Respon posisi menunjukkan perubahan dari posisi atau dapat dianggap kecepatan robot, sedangkan orientasi menunjukan perubahan sudut kemudi robot. Gambar 4.21. (a) – (f) memperlihatkan posisi perubahan x(t) dan y(t) dari robot *leader-follower*. Dengan menggunakan metode SLFT1 dengan 8 kaidah (Gambar 4.21(a)–(b)) dan metode SLFT1 dengan 27 kaidah (Gambar 4.21 (c)–(d)), perubahan posisi robot *leader* interval yang sangat lebar dan dengan jumlah data yang sama karena keduanya gagal bergerak dan berhenti di satu titik. Hal itu menunjukkan robot *leader-follower* gagal melaksanakan misinya. Untuk metode SLFT2 dengan 8 kaidah (Gambar 4.21 (e)–(f)), pergerakan perubahan posisi dengan respon yang lebih halus, interval yang kecil dan jumlah data yang dibangkitkan berbeda sekitar 136 data untuk robot *leader* karena gagal dan 61 data untuk robot *follower* karena terus bergerak menuju target.



(a) SLFT1 dengan 8 kaidah

(b) SLFT1 dengan 27 kaidah



(c) SLFT2 dengan 8 kaidah

Gambar 4. 22. Kinerja orientasi robot *leader-follower* dalam lingkungan kompleks 2

Untuk lebih memperjelas kinerja perubahan posisi $x(t)$ dan $y(t)$ robot *leader-follower*, maka akan di analisis dari perubahan orientasi robot atau sudut kemudi. Dari Gambar 4.22 (a) – (c) sangat terlihat perbedaannya. Dengan menggunakan SLFT 1 dengan 8 kaidah (Gambar 4.22 (a)) dan SLFT1 dengan 27 kaidah (Gambar 4.22 (b)) perubahan sudut kemudi sangat kasar, dengan interval perubahan yang sangat lebar. Sehingga terlihat seperti pengendalian dengan logika digital antara 0 (nol) dan 1 (satu). Pada metode SLFT 1 sudut kemudi robot *leader* tidak tampak di Gambar 4.22 (a) karena kegagalan misi akibat menabrak dinding. Sedangkan untuk metode SLFT2 sudut kemudi robot *leader-follower* muncul (Gambar 4.22 (b) hal tersebut dikarenakan perubahan aktuator saat robot *leader* gagal dan robot *follower* melanjutkan aksinya dianggap sama dalam perubahan sudut kemudinya. Sehingga data simulasi mengeluarkan hasil yang sama.

Selain sistem kontrol pergerakan pada lingkungan sederhana dan kompleks, kinerja sistem kontrol juga diuji untuk beberapa kondisi lingkungan lainnya. Namun tidak semua di analisis pada penelitian ini, yang tidak dijelaskan analisisnya akan diletakkan di dalam lampiran. Sedangkan resume yang ditunjukkan dalam Tabel 4.10 menyatakan kompilasi kinerja yang dihasilkan hanya 2 parameter yaitu waktu tempuh dan beban komputasi proses sistem kontrol *fuzzy*.

Dari semua pengujian pergerakan robot *leader-follower* pada berbagai kondisi lingkungan dengan menggunakan SLFT1 dan SLFT2, maka dapat disimpulkan seperti pada Tabel 4.10. Pengujian robot *leader-follower* menggunakan SLFT2 membutuhkan waktu yang lama dibanding SLFT1. Hal ini dikarenakan, SLFT2 melalui tahapan reduksi tipe. Tahapan tersebut membentuk banyak titik potong, sehingga keputusan kecepatan robot *leader-follower* mengalami perubahan secara bertahap. Sehingga kecepatan yang berubah secara bertahap membutuhkan waktu yang lama. Sedangkan, robot *leader-follower* dengan SLFT1 mengalami perubahan kecepatan robot dengan drastis. Sehingga waktu yang ditempuh semakin cepat. SLFT1 mengalami kegagalan dalam pencapaian target, hal ini terjadi karena kecepatan robot yang berubah secara cepat mempengaruhi kondisi robot untuk menabrak dinding penghalang.

Selain itu, SLFT2 memiliki fungsi keanggotaan berupa interval dan jumlah kaidah yang lebih sedikit daripada SLFT1. Hal ini dapat diartikan bahwa kaidah pada SLFT2 dapat mewakili banyaknya kaidah pada SLFT1. Pada Tabel 4.10 juga menunjukkan beban komputasi berdasarkan data yang dibangkitkan oleh SLFT2 lebih sedikit menggunakan memori dibandingkan SLFT1. Berdasarkan perbandingan kinerja kedua sistem kontrol formasi pada robot *leader-follower* maka SLFT2 menghasilkan pergerakan yang lebih halus dan lebih akurat dalam mengatasi ketidakpastian gerak robot *leader-follower* untuk pencapaian target.

Tabel 4. 10. Perbandingan kinerja sistem kontrol formasi dalam melakukan pergerakan di berbagai lingkungan pengujian

Bentuk Lingkungan	Jumlah data yang dibangkitkan		Waktu pemrosesan <i>fuzzy</i> (detik)	
	SLFT1 (27 kaidah)	SLFT2 (8 kaidah)	SLFT1 (27 kaidah)	SLFT2 (8 kaidah)
lingkaran	1169	486	63.9	111.7
oval	1636	427	57.7	165.5
oktagonal	360	305	20.7 (misi gagal)	93.5
Segiempat	1074	291	68.3	78.1
1 penghalang	520	485	22.8	85.8
2 penghalang	955	272	45 (misi gagal)	107.6
3 penghalang	1508	419	42.4 (misi gagal)	96.1
Beban komputasi	183 Kb	54 Kb	-	-

Dari berbagai macam validasi lingkungan yang telah dilakukan menunjukkan bahwa SLFT2 dengan fungsi keanggotaan interval mengungguli kinerja SLFT1, dalam hal kehalusan pergerakan, respon waktu, kekokohan sistem kontrol dan jumlah kaidah yang lebih minimal.

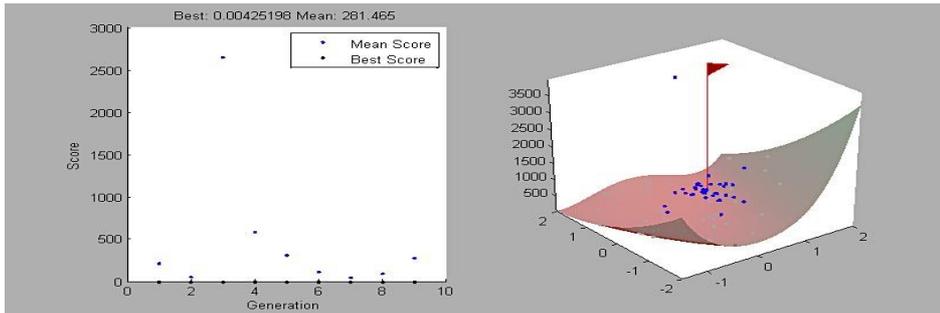
4.4. Sistem Kontrol Formasi Robot *Leader-Follower* Dalam Mencapai Target

4.4.1. Analisis *Surface* DPSO

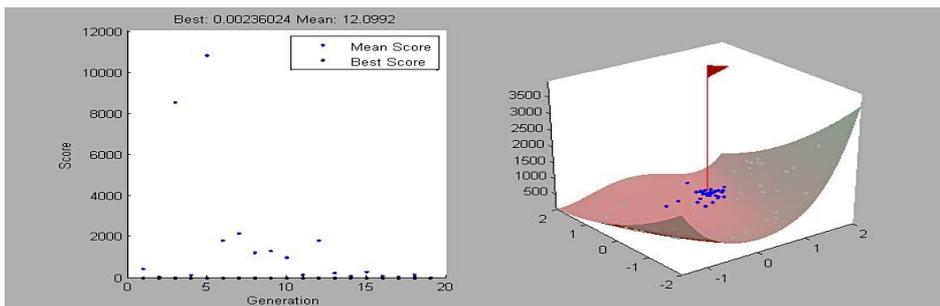
Di dalam metode DPSO yang diterapkan diberikan parameter pada tiap – tiap partikel dengan menentukan posisi (x_i) dan kecepatan awal (v_i) secara acak, evaluasi nilai *fitness* ($f(x_i)$), kemudian mengatur nilai *pbest* $p_i = x_i$, dan mengidentifikasi nilai *gbest* (p_g) untuk memperbarui posisi dan kecepatan pada partikel *swarm*.

Pada iterasi pertama ditentukan nilai sebanyak 10 x iterasi dalam penyebaran partikel *swarm* menuju ke target, iterasi kedua diberikan nilai sebanyak 20 x, iterasi ketiga sebanyak 30 x, iterasi keempat sebanyak 40 x, dan iterasi kelima sebanyak 50 x sampai dengan iterasi kesepuluh sebanyak 100 x iterasi. Hasil dari penyebaran partikel pada setiap iterasi dengan jumlah partikel yang sama akan ditunjukkan dalam fungsi plot grafik dan *surface* pada Gambar 4.23 untuk 10 kali iterasi sampai dengan 100 x iterasi.

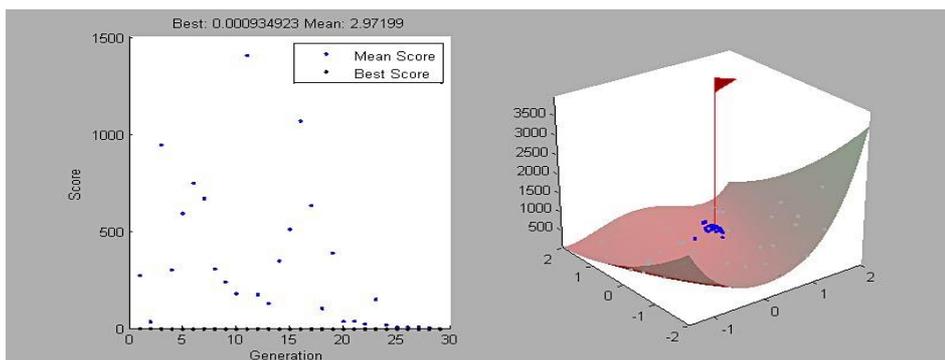
Dari perbandingan nilai iterasi yang berbeda tersebut, maka dapat disimpulkan semakin banyak nilai iterasi yang diberikan pada jumlah partikel *swarm* yang sama, maka penyebaran setiap partikel *swarm* akan semakin konvergen dalam menuju ke sumber target, dari data grafik tersebut maka dapat diperoleh nilai *Global Best* (*gbest*) dan nilai rata-rata (*Mean*) pada tiap – tiap iterasi dan pendekatan partikel terbaik pada posisi v . Resume nilai optimisasi atau nilai *minimum* yang diperoleh pada tiap – tiap iterasi yang diberikan untuk jumlah partikel *swarm* yang sama dapat dilihat pada Tabel 4.10.



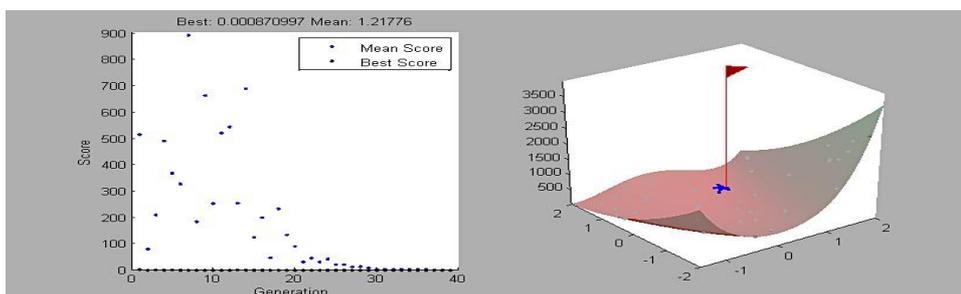
(a) penyebaran partikel 10 kali iterasi (b) *surface* hasil penyebaran partikel



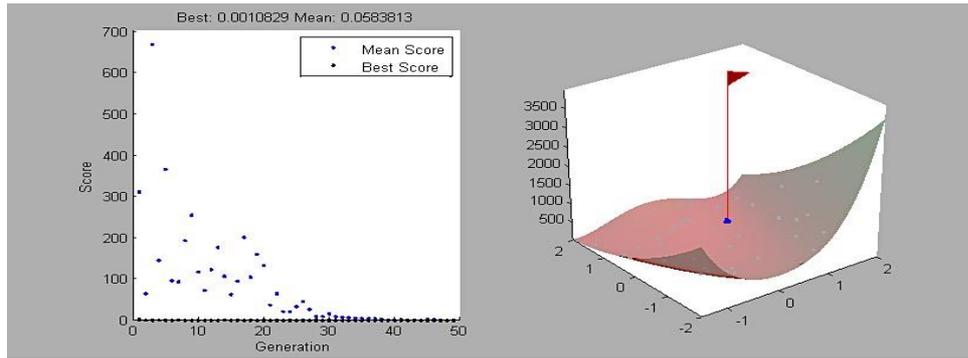
(c) Penyebaran partikel 20 kali iterasi (d) *surface* hasil penyebaran partikel



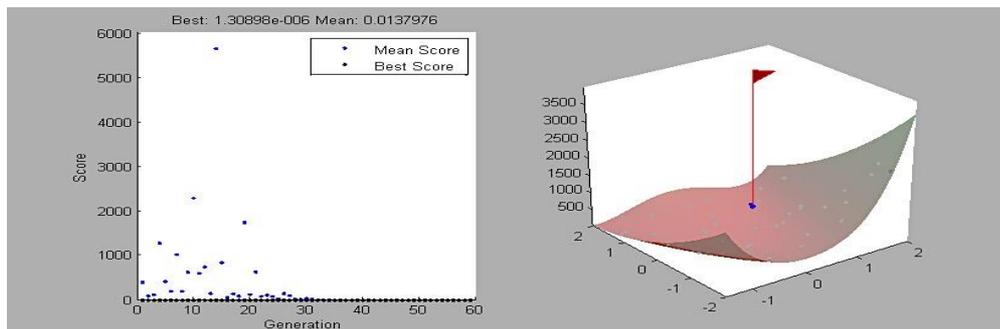
(e)Penyebaran partikel 30 kali iterasi (f) *surface* hasil penyebaran partikel



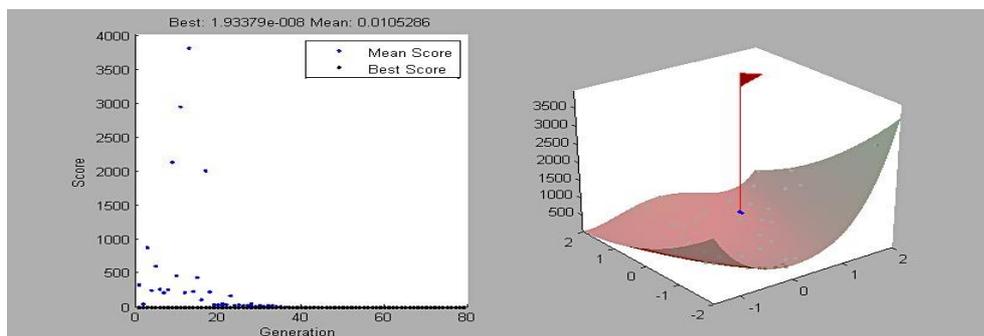
(g) penyebaran partikel 40 kali iterasi (h) *surface* hasil penyebaran partikel



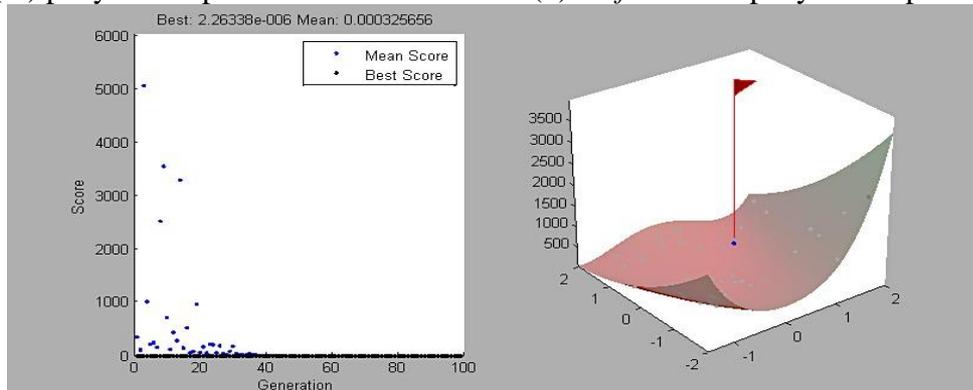
(i) penyebaran partikel 50 kali iterasi (j) *surface* hasil penyebaran partikel



(k) penyebaran partikel 60 kali iterasi (l) *surface* hasil penyebaran partikel



(m) penyebaran partikel 80 kali iterasi (n) *surface* hasil penyebaran partikel



(o) penyebaran partikel 100 kali iterasi (p) *surface* hasil penyebaran partikel

Gambar 4. 23. Simulasi *surface* dari algoritma PSO

Dari perbandingan nilai iterasi yang berbeda tersebut, maka dapat disimpulkan semakin banyak nilai iterasi yang diberikan pada jumlah partikel *swarm* yang sama, maka penyebaran setiap partikel *swarm* akan semakin konvergen dalam menuju ke sumber target, dari data grafik tersebut maka dapat diperoleh nilai *Global Best (gbest)* dan nilai rata-rata (*Mean*) pada setiap iterasi dan pendekatan partikel terbaik pada posisi *v*. Nilai optimisasi atau nilai *minimum* yang diperoleh pada setiap iterasi yang diberikan untuk jumlah partikel *swarm* yang sama dapat dilihat pada Tabel 4.11.

Tabel 4. 11. Perbandingan hasil optimisasi berdasarkan jumlah iterasi

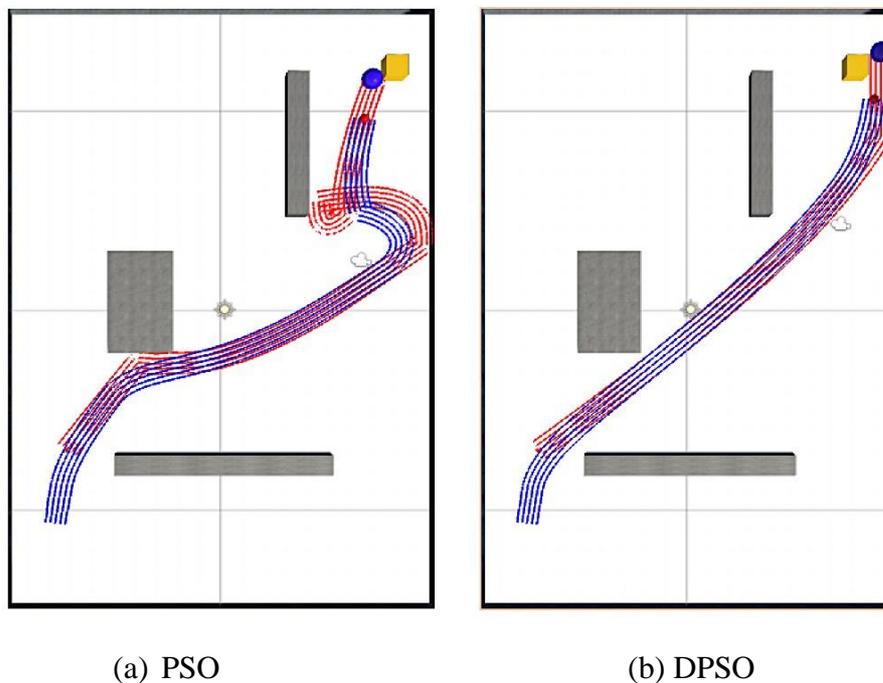
Jumlah Iterasi	<i>Final Best Point</i>	GBest	Nilai rata-rata
10	[1.0226 1.0512]	0,00425198	281,465
20	[0.95847 0.91614]	0,00236024	12,0992
30	[0.96712 0.93519]	0,000934923	2,97199
40	[0.97115 0.94286]	0,000870997	1,21776
50	[0.96712 0.93519]	0.0010829	0,0583813
60	[0.99886 0.99771]	1,30898e-006	0,0137976
70	[0.9973 0.9947]	1,53939e-007	0,0188771
80	[0.99987 0.99974]	1.93379e-008	0,0105286
90	[0.99986 0.99972]	2.142341e-008	0,0001335
100	[0.99999 0.99998]	2,26338e-006	0.000325656

Dari Tabel 4.11 hasil perbandingan optimisasi pada penyebaran partikel *swarm* pada setiap nilai *final best point* dan nilai rata-rata, maka dapat diperoleh kesimpulan bahwa semakin banyak iterasi yang diberikan maka akan semakin mendekati posisi terbaik dari target yang telah ditentukan.

4.4.2. Kinerja DPSO Pada Sistem Pendeteksi Target

4.4.2.1. Lingkungan Sederhana

Dalam penelitian ini telah dilakukan beberapa pengujian awal, untuk membuktikan bahwa metode yang di gunakan memiliki kinerja yang baik. Berikut ini telah di rancang algoritma DPSO dengan menggunakan fungsi inersia yang telah di hitung melalui berbagai percobaan. Pada penelitian ini disebut dengan dinamik PSO, karena mampu menghasilkan kinerja yang baik di berbagai kondisi lingkungan. Gambar 4.24 (a) dan (b) menunjukkan penggunaan PSO dalam mencapai target dengan menggunakan *original* PSO dan dinamik PSO (DPSO).



Gambar 4. 24. Kinerja robot *leader-follower* dalam mencapai target pada lingkungan sederhana dengan 3 halangan

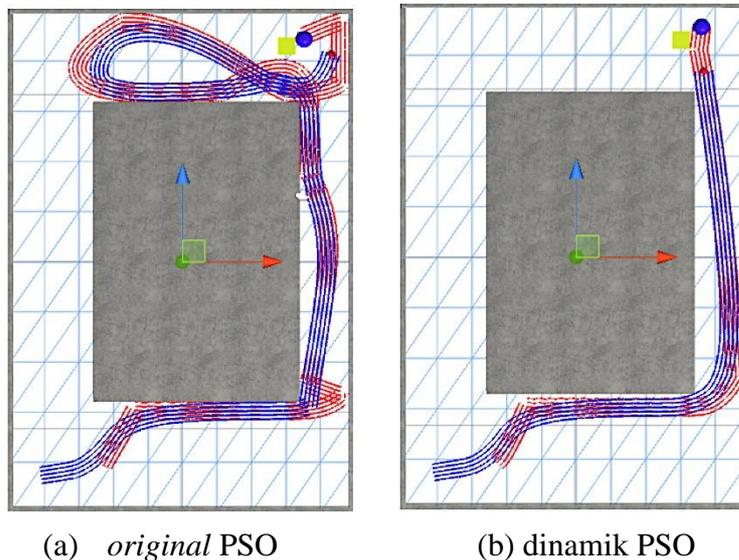
Pada Gambar 4.24 (a) dan (b) tersebut, robot *leader-follower* bergerak untuk mencapai target. Pada bagian ini, 2 jenis PSO digunakan untuk melihat kinerja pergerakan. Dari Hasil yang didapat, robot *leader-follower* dengan *original* PSO mampu mencapai target, tetapi *trajectory* yang dihasilkan tidak halus, waktu pemrosesan yang panjang, pembangkitan jumlah data yang besar untuk mencapai target. Dengan menggunakan PSO yang diusulkan, pergerakan robot *leader-follower* memiliki kinerja yang lebih baik, pergerakan lebih halus, waktu tempuh

yang pendek dan pembangkitan data yang cukup kecil. Resumennya adalah pada Tabel 4.12 sebagai berikut:

Tabel 4. 12. Kinerja PSO Dalam Mencapai Target Pada Sistem Kontrol Formasi

Keterangan	<i>Leader</i>			<i>Follower</i>		
	koordinat X	koordinat Z	waktu	koordinat X	koordinat Z	waktu
PSO						
Titik Mulai	-7.10	-6.90	1.52	-7.68	-10.59	1.53
Titik Berhenti	6.90	11.46	29.15	6.62	9.50	29.15
DPSO						
Titik Awal	-7.1	-6.9	1.66	-7.68	-10.59	1.67
Titik Berhenti	9.25	12.79	13.59	8.91	10.46	13.59

Selain lingkungan pengujian seperti pada Gambar 4.24, untuk penelitian ini diuji dengan menggunakan lingkungan yang berbentuk empat persegi panjang. Dari hasil pengujian yang telah dilakukan dihasilkan *trajectory* robot *leader-follower* seperti pada Gambar 4.25 sebagai berikut:



Gambar 4. 25. Kinerja robot *leader-follower* pada lingkungan segi empat

Dari pergerakan robot *leader-follower* pada Gambar 4.25 (a) dan (b), terlihat bahwa dari posisi awal robot telah bergerak mencari target dan berhasil menemukan target, keberhasilan mencapai ke target terlihat dari hasil *trajectory*

yang mengarah pada satu titik. Namun, berdasarkan dari Gambar 4.25 (a) dan (b) terlihat bahwa pergerakan robot dengan menggunakan algoritma DPSO dalam mencapai target lebih teratur dibandingkan dengan menggunakan algoritma standar PSO. Tabel 4.11 menjelaskan koordinat awal masing-masing robot pada saat mulai bergerak menuju target dan koordinat akhir pada saat robot telah mencapai target. Selain itu ditunjukkan juga waktu yang ditempuh masing-masing robot dalam mencapai target yang telah ditentukan.

Berdasarkan data Tabel 4.13 waktu yang ditempuh masing-masing robot dengan menggunakan algoritma PSO dari titik awal bergerak menuju target sampai robot berhenti pada saat telah mencapai target adalah ± 66 detik. Sedangkan waktu yang ditempuh masing-masing robot dengan menggunakan algoritma Dinamik PSO dari titik awal bergerak menuju target sampai robot berhenti pada saat telah mencapai target adalah ± 31 detik. Dimana waktu yang ditempuh untuk mencapai target dengan menggunakan algoritma dinamik PSO lebih cepat dibandingkan dengan algoritma PSO.

Tabel 4. 13. Titik koordinat awal dan akhir serta waktu yang ditempuh robot dalam mencapai target menggunakan PSO

Keterangan	<i>Leader</i>			<i>Follower</i>		
	koordinat X	koordinat Z	waktu	koordinat X	koordinat Z	waktu
PSO						
Titik Mulai	-4.5	-12.2	1.57	-8.15	-12.79	1.57
Titik Berhenti	7.13	13.32	66.24	8.70	12.46	66.26
DPSO						
Titik Awal	-4.5	-12.2	1.59	-8.15	-12.79	1.59
Titik Akhir	7.28	13.69	31.93	7.42	11.08	31.94

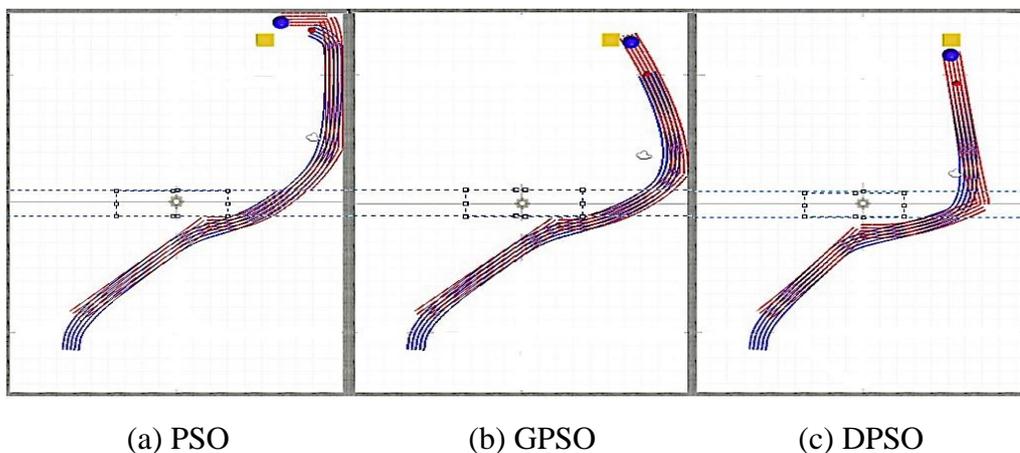
4.4.2.2. Lingkungan Tidak Terstruktur

Pada bagian ini kinerja PSO yang diimplementasikan pada robot *leader-follower* diuji dengan menggunakan halangan tidak terstruktur, yaitu (i) halangan yang bergerak secara dinamik tanpa halangan statis dan (ii) dengan 3 halangan statis dan halangan yang bergerak secara dinamik. Hasil pengujian dapat dilihat

pada Gambar 4.26 dan Gambar 4.27. Seperti halnya pengujian pada lingkungan yang hanya terdapat penghalang statis, pada bagian ini juga digunakan 3 algoritma optimisasi yaitu, orisinal *Particle Swarm Optimization* (PSO), *Gregarious Particle Swarm Optimization* (GPSO) dan dinamik *Particle Swarm Optimization* (DPSO).

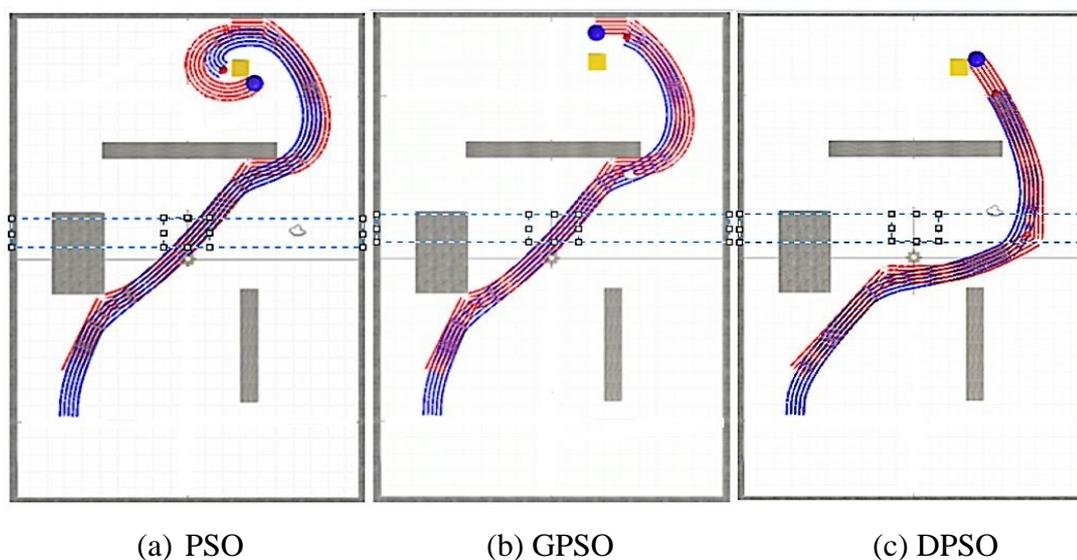
Pada GPSO, variabel acak seragam digantikan oleh nilai absolut dari variabel acak *Gaussian*. Dari sudut pandang statistik, ini memiliki efek mengubah *mean* menjadi 0,798 dan deviasi standar menjadi 0,603. Dengan meningkatkan standar deviasi dan probabilitas nilai-nilai yang besar, diharapkan robot dapat meningkatkan kemungkinan lolos dari lokal minimum. Dengan membandingkan ke 3 algoritma tersebut, untuk melihat bagaimana kinerja dari robot *leader-follower* dalam menyelesaikan tugas yang diberikan kepadanya tanpa mengalami kondisi lokal minimum.

Pada Gambar 4.26 (a) robot *leader-follower* dengan sistem kontrol *hybrid* SLFT2 dan PSO orisinal bergerak menghindari penghalang yang bergerak, tetapi mengalami tabrakan. Akibatnya perubahan posisi sangat jauh dan kembali menabrak dinding penghalang dan mencapai target. Hal tersebut juga diikuti oleh robot *follower*. Hal tersebut juga dialami oleh robot *leader-follower* dengan menggunakan GPSO, robot *leader* menabrak penghalang bergerak dan menabrak dinding pembatas (Gambar 4.26 (b)). Namun dengan menggunakan DPSO robot mampu menghindari halangan bergerak dengan cepat dan mencapai target tanpa menabrak dinding pembatas. Sehingga dengan menggunakan pendekatan SLFT2 dan DPSO kinerja kontrol formasi dapat memperbaiki pergerakan robot *leader-follower* dalam mencapai target (Gambar 4.26 (c)).



Gambar 4. 26.Optimisasi rute dengan halangan bergerak

Pada Gambar 4.27 (a)-(c), yaitu bagaimana DPSO juga menghasilkan kinerja pengontrolan formasi yang lebih baik dibandingkan GPSO dan PSO. Pada Gambar 4.27 (a) dan (b), terdapat halangan statis dan dinamik, dengan menggunakan PSO dan GPSO robot *leader-follower* tidak bisa menjaga formasinya. Dalam hal ini robot *leader* menabrak halangan yang bergerak dan halangan yang statis begitu juga robot *follower*. Tetapi hal itu tidak terjadi saat sistem kontrol yang digunakan DPSO, robot *leader* mampu menghindari kedua halangan dan robot *follower* juga menghasilkan pergerakan yang sama. Dengan menggunakan sistem kontrol formasi dengan optimisasi DPSO terjadi perbaikan pada kinerja pergerakan robot, yaitu robot *leader-follower* mampu menghindari halangan statis dan dinamik, mampu mencapai target dengan menghasilkan jarak terpendek dan menghasilkan pergerakan yang halus tanpa terjadi tabrakan (Gambar 4.27 (c)).



Gambar 4. 27.Optimisasi rute dengan halangan bergerak dan halangan tetap

Resume kinerja dengan 3 algoritma optimisasi adalah seperti terlihat pada Tabel 4.14 berikut. Dari Tabel 4.14 dan Tabel 4.15 dapat dilihat bahwa algoritma PSO, GPSO dan DPSO menghasilkan kinerja yang berbeda, tetapi DPSO lebih unggul dalam hal jumlah beban komputasi, waktu mencapai target dan data iterasi yang dibangkitkan baik itu pada lingkungan dinamik dan berubah-ubah maupun pada

lingkungan dengan halangan tetap dan halangan bergerak. Rata-rata optimisasi rute berhasil dilakukan oleh robot *leader* dan robot *follower* dengan seluruh metode PSO, GPSO dan DPSO, selain itu robot mampu mengikuti rute tersebut. Namun metode DPSO menghasilkan waktu selama 20 detik untuk lingkungan tetap dan bergerak dalam mencapai target, serta waktu selama 14.94 detik untuk lingkungan bergerak. Sehingga dapat disimpulkan optimisasi rute dengan DPSO mengungguli PSO orisinal GPSO.

Tabel 4. 14. Kinerja sistem kontrol pergerakan dan optimisasi rute pada halangan yang bergerak secara dinamis

Halangan dinamik	Algoritma					
	PSO		GPSO		DPSO	
	<i>Leader</i>	<i>Follower</i>	<i>Leader</i>	<i>Follower</i>	<i>Leader</i>	<i>Follower</i>
Beban komputasi (kb)	28.6	28.8	21.7	21.8	16.8	17
Waktu (detik)	23.32	23.32	16.84	16.84	14.94	14.94
Data (Iterasi)	1032	1032	771	771	599	599

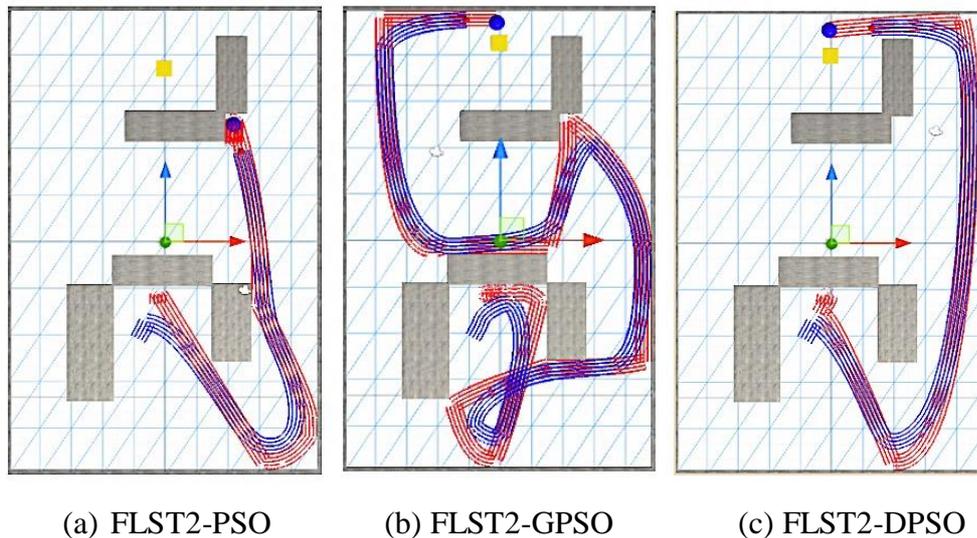
Tabel 4. 15. Kinerja sistem kontrol pergerakan dan optimisasi rute pada halangan statis dan dinamis

Halangan statis dan dinamis	Algoritma					
	PSO		GPSO		DPSO	
	<i>Leader</i>	<i>Follower</i>	<i>Leader</i>	<i>Follower</i>	<i>Leader</i>	<i>Follower</i>
Beban komputasi (kb)	28,7	28,6	22	21,2	20,1	20,1
Waktu (detik)	81,42	81,42	74	74	20,6	20,6
Data (Iterasi)	1035	1035	760	771	718	718

4.4.3. Kinerja Metode *Hybrid SLFT2* dan DPSO

Untuk melihat kinerja sistem kontrol yang telah dirancang, maka dilakukan pengujian pada lingkungan beberapa lingkungan yang tidak terstruktur yaitu lingkungan *U-Shape*, lingkungan kompleks 1, lingkungan kompleks 2, lingkungan berbentuk lingkaran dan lingkaran berbentuk oval. Seperti terlihat pada Gambar 4.28 (a) – (c) salah satu lingkungan yang sangat sulit yaitu jika berbentuk *U-*

Shape atau jebakan berbentuk U. Terlihat dari Gambar 4.28 (a) robot *leader-follower* tidak mampu mencapai target dan stop pada posisi sudut. Hal tersebut mengakibatkan kegagalan proses kontrol secara keseluruhan.

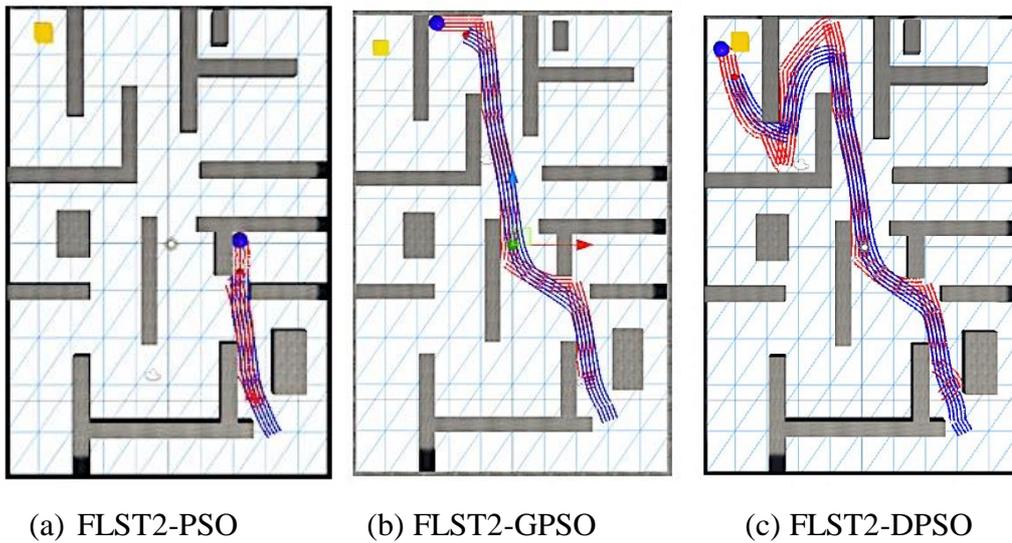


Gambar 4. 28.Kinerja sistem kontrol formasi untuk lingkungan *U-Shape*

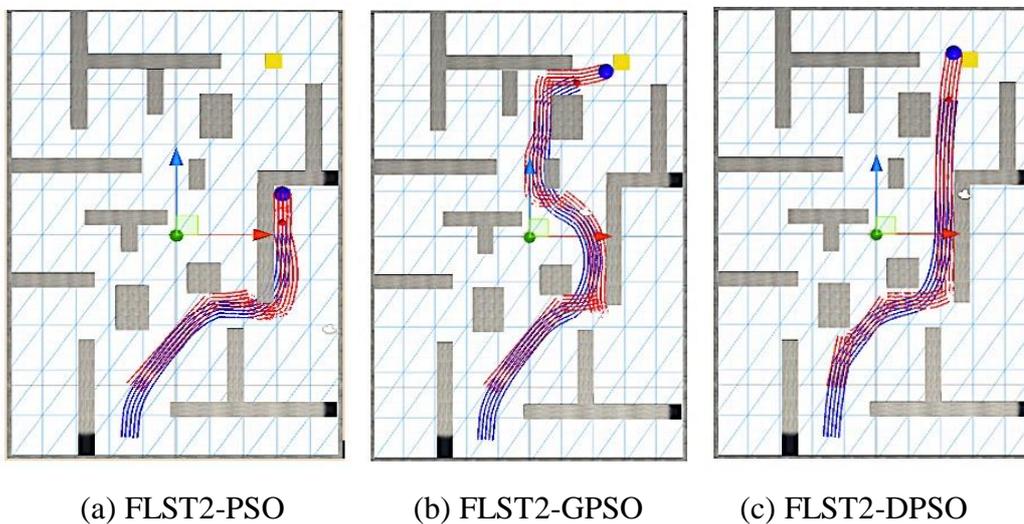
Dari Gambar 4.28 (a) menggunakan sistem kontrol formasi berbasis SLFT2 dan PSO menghasilkan 192 detik, beban komputasi 125 kb dan membangkitkan data sekitar 4607 iterasi. Namun dengan kinerja tersebut, robot *leader-follower* gagal memproses seluruh tugasnya, dan stop pada titik tertentu. Gambar 4.28 (b) sistem kontrol formasi dengan menggunakan SLFT2 dan GPSO menghasilkan 95 detik, beban komputasi 90.4 kb dan membangkitkan data sekitar 3142 iterasi. Dengan metode SLFT2 dan GPSO robot *leader-follower* mampu menyelesaikan tugasnya walaupun dengan rute yang sangat panjang dan menabrak dinding pembatas. Sedangkan yang terakhir adalah sistem kontrol formasi dengan SLFT2 dan DPSO (Gambar 4.28 (c)) menghasilkan kinerja penyelesaian tugas selama 40 detik, beban komputasi yang dihasilkan 46.4 Kb dan membangkitkan data sekitar 1670 iterasi. Kombinasi metode SLFT2 dan DPSO menghasilkan kinerja yang terbaik, robot *leader-follower* berada dalam formasi, rute yang dipilih pendek dan waktu pemrosesan yang cepat dan tidak menabrak dinding pembatas.

Kinerja sistem kontrol formasi juga diuji pada 2 lingkungan yang kompleks dengan banyak penghalang. Dengan menggunakan metode *hybrid* FLST2 dan

DPSO dihasilkan pergerakan robot *leader-follower* seperti Gambar 4.29 (a) – (c) dan Gambar 4.30 (a) – (c). Dalam kasus ini digunakan 3 sistem kontrol formasi yang lainnya sebagai pembanding yaitu FLST2-PSO, FLST2-GPSO dan FLST2-DPSO.



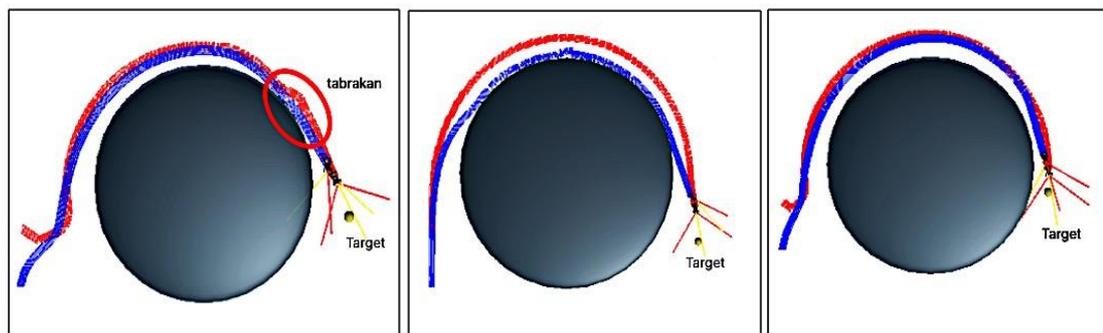
Gambar 4. 29.Kinerja sistem kontrol formasi untuk lingkungan kompleks 1



Gambar 4. 30. Kinerja sistem kontrol formasi untuk lingkungan kompleks 2

Diharapkan robot dapat bergerak mencapai target tanpa menabrak dinding pembatas yang harus dihindari adalah robot berhenti atau bahkan kehilangan target. Dari keseluruhan pengujian terlihat bahwa sistem kontrol formasi dengan

menggunakan FLST2-DPSO menghasilkan kinerja yang paling baik dalam hal, pergerakan dengan rute terpendek, proses yang cepat, tanpa tabrakan dan dapat menemukan target (Gambar 4.29 (c) dan 4.30 (c)). Namun dengan menggunakan FLST2-PSO robot *leader-follower* tidak mampu menyelesaikan tugas pencairan target, menabrak dinding pembatas dan berhenti (Gambar 4.29 (a) dan 4.30 (a)). Sedangkan jika menggunakan FLST2-GPSO robot *leader-follower* mampu menghindari tabrakan dan mencapai target tetapi melalui rute yang agak panjang (4.29 (b) dan 4.30 (b)). Sehingga rute yang di pilih menjadi tidak optimal.

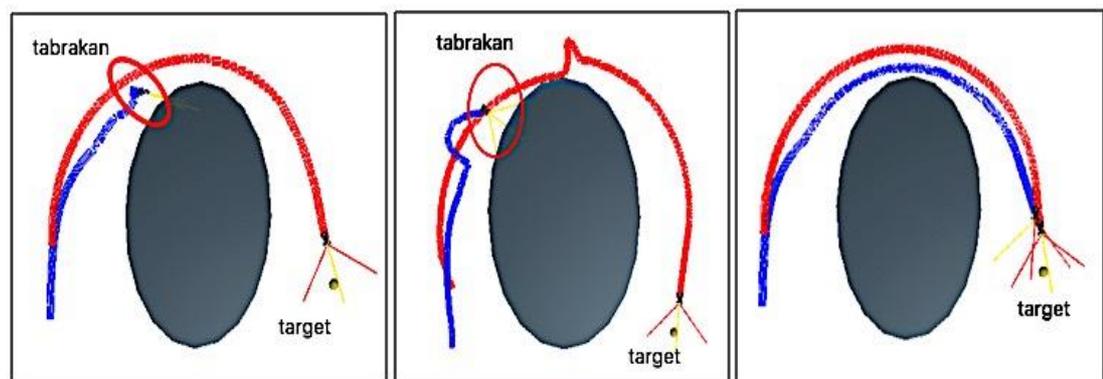


(a) SLFT2-PSO

(b) SLFT2-GPSO

(c) SLFT2-DPSO

Gambar 4. 31. Sistem kontrol formasi untuk lingkungan berbentuk lingkaran

(a) SLFT1-DPSO
(8 kaidah)(b) SLFT1-DPSO
(27 kaidah)(c) SLFT2-DPSO
(8 kaidah)

Gambar 4. 32. Sistem kontrol formasi untuk lingkungan berbentuk oval

Pada bagian ini sistem kontrol formasi dengan metode SLFT2-DPSO juga diuji pada lingkungan yang tidak terstruktur seperti bentuk lingkaran dan oval. Pada Gambar 4.31. dan Gambar 4.32 dapat dilihat pengujian sistem kontrol SLFT2-DPSO terhadap kehalusan pergerakan dan ketahanan terhadap lingkungan

tidak terstruktur. Dari hasil pengujian terlihat pada Gambar 4.31 (c), bahwa dengan menggunakan metode SLFT2-DPSO, robot *leader-follower* mampu melakukan pergerakan yang halus tanpa menabrak dinding pembatas. Dengan memasukkan nilai interval pada fungsi keanggotaan metode SLFT2, pengendali mampu mengatasi perubahan pendeteksian sensor secara lebih akurat dan memperbaiki perubahan kecepatan aktuator. Selain itu dengan penambahan DPSO untuk optimisasi rute, maka pengontrolan gerak robot berubah sesuai dengan bobot w masing-masing partikel. Hal tersebut tidak tercapai saat menggunakan metode SLFT2-PSO (Gambar 4.31 (a)) dan SLFT2-GPSO (Gambar 4.31 (b)), dimana robot *leader-follower* menabrak dinding pembatas.

Sedangkan pada Gambar 4.32, menunjukkan bahwa saat robot *leader-follower* bergerak pada lingkungan yang berbentuk oval, SLFT2-DPSO dengan 8 kaidah mampu melewati lintasan dengan pergerakan yang halus tanpa menabrak dinding pembatas. Namun tidak demikian dengan SLFT1-DPSO menggunakan 8 kaidah dan SLFT1-DPSO menggunakan 27 kaidah, robot *leader-follower* bertabrakan, dan hanya robot *leader* yang melanjutkan perjalanan menuju target, sedangkan robot *follower* berhenti dan tidak bisa melanjutkan perjalanannya menuju target. Sehingga dapat disimpulkan bahwa robot *leader-follower* dengan metode SLFT2-DPSO dengan jumlah kaidah yang lebih minimal memiliki kinerja yang lebih baik dalam hal kehalusan pergerakan, ketahanan terhadap perubahan lingkungan dan waktu tempuh yang lebih pendek dibandingkan metode SLFT1-DPSO.

Tabel 4. 16. Kinerja sistem kontrol *hybrid* untuk waktu tempuh

Keterangan	Waktu (detik)					
	SLFT2-PSO		SLFT2-GPSO		SLFT2-DPSO	
	<i>Leader</i>	<i>Follower</i>	<i>Leader</i>	<i>Follower</i>	<i>Leader</i>	<i>Follower</i>
<i>U-Shape</i>	108.27 (gagal)	108.28 (gagal)	191.6665	191.68	27.28	27.29
Lingkungan kompleks 1	263.21 (gagal)	263.23 (gagal)	247.6061 (gagal)	247.6183 (gagal)	43.67	43.68
Lingkungan kompleks 2	274.5 (gagal)	273.4 (gagal)	258.56	255.70	47.90	47.57

Tabel 4. 17. Kinerja sistem kontrol *hybrid* untuk beban komputasi

Keterangan	Beban Komputasi (kb)					
	SLFT2-PSO		SLFT2-GPSO		SLFT2-DPSO	
	<i>Leader</i>	<i>Follower</i>	<i>Leader</i>	<i>Follower</i>	<i>Leader</i>	<i>Follower</i>
<i>U-Shape</i>	109	108	145	146	27.8	27.2
Lingkungan kompleks 1	45.5	46.7	117	117.5	32.8	32.1
Lingkungan kompleks 2	193	199	127	118	57.9	57.4

Resume sistem kontrol yang diusulkan yaitu SLFT2-DPSO dibandingkan dengan SLFT1-GPSO dan SLFT2-PSO dapat dilihat pada Tabel 4.16 dan Tabel 4.17. Terlihat bahwa metode yang diusulkan yaitu SLFT2-DPSO lebih unggul dalam hal waktu pemrosesan, pembangkitan data beban komputasi, dan kehalusan pergerakan dalam mencapai target. Tabel 4.16 menggambarkan waktu tempuh yang dihasilkan oleh robot *Leader-Follower* dalam mencapai target di 3 lingkungan yaitu lingkungan *U-Shape* pada Gambar 4.25, lingkungan kompleks 1 pada Gambar 4.26 (a), (b) dan (c) dan lingkungan kompleks 2 pada Gambar 4.27 (a), (b) dan (c). Hasil yang di dapat adalah untuk lingkungan kompleks 1, robot *leader-follower* dengan sistem kontrol formasi dengan metode SLFT2-PSO dan SLFT2-GPSO gagal mencapai target, sedangkan sistem kontrol formasi dengan metode SLFT2-DPSO berhasil mencapai target. Hal ini dikarenakan DPSO mampu beradaptasi dengan perubahan kondisi lingkungan, sehingga robot tetap menghasilkan optimisasi rute tanpa terjebak di suatu titik tertentu dalam lingkungan. Pada lingkungan kompleks 2, hanya sistem kontrol formasi dengan metode SLFT2-PSO yang gagal, karena dari beberapa pustaka sebelumnya, bahwa algoritma PSO sering terjebak pada nilai optimum lokal. Sehingga dengan penggunaan bobot inersia yang berubah sesuai perubahan kondisi lingkungan, sistem kontrol formasi dengan metode SLFT2-DPSO berhasil mengungguli sistem kontrol lainnya tanpa terjebak di titik optimum lokal dan tanpa mengalami tabrakan. Sedangkan pada lingkungan *U-shape*, ketiga sistem kontrol mampu membawa robot *leader-follower* ke target, walaupun waktu tempuh sistem kontrol

SLFT2-PSO sekitar 108.27 detik dan SLFT2-GPSO sekitar 191.67 detik lebih lama dibandingkan dengan sistem SLFT2-DPSO hanya 27.28 detik

Sistem kontrol formasi dengan metode SLFT2-DPSO akan digunakan pada robot *swarm*, sehingga perlu dilihat beban komputasi dari metode yang digunakan. Dari Tabel 4.17 terlihat bahwa SLFT2-DPSO menghasilkan beban komputasi dalam mencapai target yang cukup kecil yaitu 27.8 kb untuk lingkungan *U-Shape*, 32.8 kb untuk lingkungan kompleks 1 dan 57.9 untuk lingkungan kompleks 2. Hal ini menunjukkan penggunaan SLFT2-DP SO secara umum dapat digunakan pada sistem berbasis mikrokontroler.

IV. 5. Ringkasan

Sistem kontrol formasi dengan metode logika *fuzzy* tipe-2 berhasil membantu mengatasi ketidakteelitian/ketidakkuratan sensor dan ketidakpastian aktuator yang ada di lingkungan waktu nyata, menghasilkan sistem reaktif yang baik yang sesuai untuk lingkungan yang kompleks, dengan penghalang statis maupun dinamis. Hal ini jelas menunjukkan kemampuan SLFT2 dan SLFT1 dalam menghindari penghalang, namun percobaan menunjukkan bahwa SLFT2 menghasilkan lintasan robot *leader-follower* yang lebih halus dengan osilasi dan penyimpangan yang lebih sedikit. Ditemukan juga bahwa bahkan dengan menggunakan basis kaidah yang sama yaitu 8 kaidah, SLFT2 telah mengungguli SLFT1. Sayangnya beban komputasi yang dihasilkan lebih besar karena perhitungan reduksi tipe. Dalam metode SLFT2, karena setiap perilaku dianggap sebagai basis aturan kecil, ini memberikan perhitungan yang lebih cepat dan kinerja yang dihasilkan lebih baik. Hal ini semakin di perjelas bahwa percobaan dengan menggunakan metode SLFT2 dengan kaidah yang lebih minimal, yaitu 8 kaidah telah berhasil mengatasi ketidakpastian yang ada di lingkungan yang tidak terstruktur.

Sistem kontrol *hybrid* SLFT2-DPSO telah dikembangkan dalam penelitian ini yang memfasilitasi pergerakan yang aman dan kokoh ke tujuan tertentu dalam lingkungan yang berubah-ubah dan dinamik. Untuk lebih spesifik, model sistem kontrol *hybrid* yang dirancang menggabungkan metode SLFT2 untuk menghindar halangan dan metode DPSO untuk mencapai target berdasarkan pergerakan reaktif

dalam lingkungan yang dinamis. Kekokohan dievaluasi berdasarkan masalah minimum lokal seperti lingkungan *U-shape*, lintasan kesalahan di lingkungan yang tidak terstruktur, dan deviasi dengan lintasan yang diinginkan karena variasi FOU. Selain itu, karena pada dasarnya menerapkan sistem kontrol formasi yang reaktif dalam pergerakan robot *leader-follower* menggunakan SLFT2-DPSO, itu menjadi bersifat adaptif dengan variasi kecil di lingkungan sekitarnya dan juga kuat untuk kebisingan sensor (*noise*).

Algoritma pengontrol SLFT2-DPSO yang diusulkan telah diverifikasi melalui eksperimen secara simulasi dan dibandingkan dengan sistem kontrol SLFT1-PSO, SLFT1-GPSO, SLFT1-DPSO, SLFT2-PSO dan SLFT2-GPSO. Namun PSO standar dan GPSO tidak mampu memberikan kinerja yang baik saat lingkungan menjadi sangat kompleks dan lingkungan *U-Shape*, dikarenakan bobot inersia perubahan partikel pada PSO standar adalah 1 dan pada GPSO adalah bernilai konstan tidak berubah secara adaptif. Berbeda dengan DPSO, algoritma ini terus mengukur perubahan sekecil apa pun dari pergerakan partikel dalam mencari solusinya dengan menggunakan pembobotan inersia w yang berubah secara linier dan adaptif. Hasil yang telah dicapai dengan menggunakan sistem kontrol yang diusulkan adalah robot *leader-follower* berhasil mencapai tujuan di lingkungan yang tidak terstruktur, mampu menangani ketidakpastian dalam pembacaan sensor, menghasilkan kinerja yang baik dalam melakukan tindakan pengontrolan di lingkungan yang tidak terstruktur dengan beban komputasi yang minimal, waktu mencapai target yang cepat dan rute tempuh yang efisien.

BAB V KESIMPULAN DAN SARAN

5.1. Kesimpulan

Bab ini menunjukkan bagaimana aspek paling relevan dari penelitian yang dikembangkan dan disajikan dapat berkontribusi pada teknologi robot *swarm* yang cerdas dengan teknik komputasi cerdas menggunakan ide dan strategi inovatif. Pada bagian ini, akan ditinjau apakah hasil yang diperoleh mencapai tujuan yang diusulkan. Ini menyoroti kesimpulan paling relevan yang timbul dari semua percobaan, di seluruh lingkungan pengujian yang bervariasi untuk memberikan pengembangan dan eksplorasi solusi dari metode yang diusulkan

Kesimpulannya, tujuan penelitian telah berhasil dicapai, telah diterapkan dan diverifikasi secara simulasi yaitu:

1. Sistem kontrol formasi yang dikembangkan dengan menggunakan kombinasi metode sistem logika *fuzzy* tipe-2 (SLFT2) dan *dinamik particle swarm optimization* (DPSO) mampu meningkatkan kinerja pergerakan dalam hal waktu pemrosesan, optimisasi rute dan komputasi yang minimal. Selain itu dengan menggunakan sistem kontrol SLFT2-DPSO robot *leader-follower* mampu menghindari tabrakan, dan dapat mencapai target yang telah ditetapkan
2. Algoritma SLFT2 telah berhasil dirancang dan telah diimplementasikan pada pengontrolan formasi robot *leader-follower* dan menghasilkan pembangkitan data yang kecil, waktu pemrosesan *fuzzy* yang lama dari *fuzzy* tipe-1, tetapi beban komputasi yang minimal. Walaupun waktu pemrosesan *fuzzy* tipe-2 lebih lama, tetap menghasilkan waktu pemrosesan keseluruhan yang cepat karena data yang dibangkitkan kecil, sehingga pergerakan yang dihasilkan lebih teliti.
3. Algoritma *dinamik particle swarm optimization* telah berhasil dirancang dan telah implementasikan pada robot *leader-follower* untuk optimisasi rute pencapaian target. Hasil yang didapat rute terpendek di perolehi saat DPSO digunakan dengan waktu pemrosesan yang cepat.

4. Proses kombinasi metode SLFT2 dan DPSO telah berhasil dilakukan, dan algoritma kombinasi kedua metode tersebut telah diimplementasikan pada robot *leader-follower*. Metode tersebut digunakan untuk dua kondisi yaitu untuk menghindari halangan dan melakukan optimisasi rute untuk mencapai target. Hasil yang diperoleh sangat memuaskan dengan kinerja yang baik dalam hal kehalusan pergerakan, respon yang cepat dan adaptif, rute tempuh yang cepat dan kokoh pada semua lingkungan pengujian.
5. Dari hasil penelitian yang telah dilakukan dapat dilihat kinerja sistem kontrol formasi dengan menggunakan SLFT2 dan DPSO dapat bekerja dengan baik di berbagai kondisi lingkungan dan dengan berbagai kondisi penghalang. Validasi dilakukan dengan berbagai metode yaitu SLFT1-PSO, SLFT1-Gregarious PSO, SLFT2-PSO dan SLFT2-Gregarious PSO. Dan hasil yang dicapai menunjukkan bahwa SLFT2-DPSO menghasilkan kinerja yang lebih baik dibandingkan metode lainnya dalam hal data yang dibangkitkan, beban komputasi, waktu tempuh dan kehalusan pergerakan dalam lingkungan.

5.2. Kontribusi Penelitian

Penelitian ini membuat sejumlah kontribusi dalam pengendali hibrida robot berkelompok *leader-follower* dalam melakukan pengontrolan formasi untuk mencapai target. Kontribusi spesifik dari penelitian ini dirangkum sebagai berikut:

1. Perancangan algoritma SLFT2 untuk pengontrolan gerak formasi robot *leader-follower* telah dilakukan dan menghasilkan kinerja yang memuaskan dibandingkan dengan SLFT1. Fungsi keanggotaan dalam bentuk interval pada SLFT2 berhasil meningkatkan ketelitian pengontrolan aktuator dibandingkan dengan fungsi keanggotaan berbentuk satu nilai pada SLFT1. Robot *leader-follower* mampu menjaga pergerakan dalam formasi. Selain itu dengan perubahan nilai kecepatan dan sudut kemudi yang lebih halus dan rapat membuat robot mampu menghindari tabrakan antara robot dan dinding pembatas dan kokoh terhadap berbagai kondisi lingkungan.
2. Perancangan algoritma Dinamik PSO dengan *auto-tuning* pada nilai *learning rate* (c_1, c_2) dan fungsi inersia w bervariasi dengan waktu telah berhasil dilakukan. Dari hasil perancangan didapatkan optimisasi rute dari robot

leader-follower dalam mencapai target. Hasil optimisasi dengan menggunakan pendekatan DPSO lebih baik dalam hal panjang rute tempuh, dan waktu pemrosesan dibandingkan dengan GPSO dan PSO.

3. Awalnya terdapat permasalahan pada pengontrolan robot *leader-follower* yaitu jika robot *leader* gagal, maka *follower* akan ikut gagal dan misi tidak berhasil di selesaikan. Dengan metode yang diusulkan yaitu SLFT2-DPSO permasalahan tersebut dapat diselesaikan, walaupun robot *leader* gagal dalam misi tersebut, tetapi robot *follower* tetap dapat melanjutkan pergerakan menuju target yang telah ditentukan.
4. Di seluruh kondisi lingkungan metode SLFT2-DPSO mampu menghasilkan kinerja yang baik, kokoh terhadap ketidakpastian lingkungan dan mampu mengatasi perubahan kondisi lingkungan yang dinamik. Telah dilakukan pengujian dalam berbagai lingkungan yaitu sederhana, kompleks dan tidak terstruktur, dengan penghalang diam dan bergerak. Namun sistem kontrol yang diusulkan SLFT2 dan DPSO menghasilkan kinerja terbaik dibandingkan dengan SLFT2 dan PSO serta SLFT2 dan GPSO.

5.3. Penelitian Selanjutnya

Dari hasil penelitian yang telah dilakukan maka didapat beberapa masalah yang dapat dijadikan penelitian lanjutan yaitu:

1. Metode sistem logika *fuzzy* tipe-2 dan DPSO pada penelitian ini masih dalam tahap simulasi, maka diperlukan suatu penelitian khusus jika algoritma yang telah dihasilkan akan di masukkan atau di instalasi ke dalam robot sesungguhnya. Disebabkan oleh robot *swarm* memiliki keterbatasan dalam hal beban komputasi dan waktu pemrosesan. Robot *swarm* dalam kondisi nyata memiliki perangkat sensor, processing dan aktuator secara *onboard* pada sistem mikrokontroler. Sehingga diperlukan sistem kontrol yang benar-benar memperhatikan keterbatasan perangkat keras.
2. Pada penelitian ini, SLFT2 menggunakan metode *iterative Karnik-Mendel* dengan hasil komputasi yang cukup besar, sehingga dapat dipertimbangkan untuk menggunakan algoritma *Wu-Mendel*. Selanjutnya

dapat dibandingkan kinerja pengontrolannya. Sehingga diharapkan dapat menurunkan waktu komputasi dari SLFT2.

3. Sistem kontrol dengan kombinasi metode SLFT2 dan DPSO yang diusulkan belum banyak dilakukan. Dalam penelitian ini hanya menggunakan pendekatan Robot *leader-follower*, selanjutnya untuk generalisasi dapat digunakan jenis robot *swarm* lainnya. Robot *swarm* lainnya dapat berbasiskan perilaku, yaitu sekelompok robot yang bergerak, dan pada awal pergerakan tidak menetapkan robot pemimpin.

DAFTAR PUSTAKA

- Abdalla, T. Y., and Hamzah, M. I. (2013). Trajectory Tracking Control for Mobile Robot using Wavelet Network, *74(3)*, 32–37.
- Abdessemed, F., Benmahammed, K., and Monacelli, E. (2004). A *fuzzy*-based reactive controller for a non-holonomic mobile robot. *Robotics and autonomous Systems*, *47(1)*, 31-46.
- Al-Hassan, W., Fayek, MB and Shaheen, SI. (2007). Psosa: An optimized particle swarm technique for solving the urban planning problem. In IEEE Computer Engineering and Systems, The International Conference on, pages 401–405.
- Antonelli, A., Arrichiello, F., and Chiaverini, S. (2010). Flocking for multi-robot systems via the Null-space-based behavioral control,” *Swarm Intell.*, vol. 4, no. 1, pp. 37–56.
- Asaro, P. (2012). On banning autonomous weapon systems : human rights , automation , and the dehumanization of lethal decision-making, *94(886)*, 687–709.
- Balch, T., and Arkin, R. C. (1998). Behavior-Based Formation Control for Multirobot Teams. *IEEE Transaction on Robotics and Automation*, *14(6)*, 926–939.
- Barnes, L., Fields, M., and Valavanis, K. (2007). Unmanned Ground Vehicle Swarm Formation Control Using Potential Fields. *Medierranean Conference Control and Automation*.
- Barca, J. C., and Sekercioglu, Y. A. (2013). Swarm robotics reviewed. *Robotica*, *31(3)*, 345-359.

- Bazoula, A., Djouadi, M. S., and Maaref, H. (2008). Formation control of multi-robots via fuzzy logic technique. *International Journal of Computers, Communications & Control*, 3(3).
- Basilico, N., Gatti, N., and Amigoni, F. (2009, May). Leader-follower strategies for robotic patrolling in environments with arbitrary topologies. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1* (pp. 57-64).
- Bayındır, L. “A review of swarm robotics tasks,” *Neurocomputing*, 2015.
- Benavidez, P., Jamshidi, M., and Chair, L. B. E. (2011). Mobile Robot Navigation and Target Tracking System, 299–304.
- Beard, R. W., Lawton, J., and Hadaegh, F. Y. (2001). A coordination architecture for spacecraft formation control. *IEEE Transactions on control systems technology*, 9(6), 777-790.
- Benbouabdallah, K., and Qi, Z. (2013). A Fuzzy Logic Behavior Architecture Controller for a Mobile Robot Path Planning in Multi-obstacles Environment. *Research Journal of Applied Sciences, Engineering and Technology*, 5(14), 3835–3842.
- Begnini, M., Bertol, D. W., and Martins, N. A. (2017). A robust adaptive fuzzy variable structure tracking control for the wheeled mobile robot: Simulation and experimental results. *Control Engineering Practice*, 64, 27-43.
- Beni, G. (2004). From swarm intelligence to swarm robotics. In *International Workshop on Swarm Robotics* (pp. 1-9). Springer, Berlin, Heidelberg.
- Benzerrouk, A., Adouane, L., Lequievre, L., & Martinet, P. (2010). Navigation of Multi-Robot Formation in Unstructured Environment Using Dynamical Virtual Structures, 5589–5594.

- Berg, J. Van Den, Ferguson, D., and Kuffner, J. (2006). Anytime Path Planning and Replanning in Dynamic Environments. *IEEE International Conference on Robotics and Automation*, (May), 2366–2371.
- Bradley, D., Brunton, A., Fiala, M., and Roth, G. (2005). Image- based navigation in real environments using panoramas,” in *Proceedings of the IEEE International Workshop on Haptic Audio Visual Environments and their Applications (HAVE '05)*, pp. 57– 59.
- Brambilla, M., Ferrante, E., Birattari, M., and Dorigo, M. (2013). Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 7(1),1-41.
- Brunete, A., Hernando, M., Gambao, E., and Torres, J. E. (2012). A behaviour-based control architecture for heterogeneous modular . *Robotics and Autonomous Systems*, 60, 1607–1624.
- Chen, G., Huang, X., Jia, J., and Min, Z. (2006). Natural exponential Inertia Weight strategy in particle swarm optimization. In *Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on*, volume 1, pages 3672–3675.
- Calli, B., Caarls, W., Jonker, P., and Wisse, M. (2012). Comparison of Extremum Seeking Control Algorithms for Robotic Applications. *IEEE International Conference Robot System*, 3195 – 3202.
- Carpin, S., and Pagello, E. (2009). An experimental study of distributed robot coordination. *Rob. Auton. Syst.*, vol. 57, no. 2, pp. 129–133.
- Castillo, O, et al. (2016). A comparative study of type-1 fuzzy logic systems, interval type-2 fuzzy logic systems and generalized type-2 fuzzy logic systems in control problems." *Information Sciences* 354: 257-274.

- Chern, C., Paul, S., Jacques, J., and Slotine, E. (2009). Region-based shape control for a swarm of robots. *Automatica*, 45(10), 2406–2411.
- Chen, Y.Q.C.Y.Q., and Wang, Z.W.Z. (2005). Formation control: a review and a new consideration. *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, no. 435, pp. 3664–3669.
- Chivarov, N., Paunski, Y., Angelov, G., Radev, D., Penkov, S., Vladimirov, V., Zahariev, R., Dimitrov, M.D.O., Ivanova, V., Shivarov, P.N., and Kopacek, T. (2012). Intelligent Modular Service Mobile Robot for Elderly Care. vol. 20, pp. 54–66.
- Consolini, L., Morbidi, F., Prattichizzo, D., and Tosques, M. (2006). On the Control of a Leader-Follower Formation of Nonholonomic Mobile Robots. *Proceeding of The 45 Th IEEE Conference on Decision and Control*, 5992–5997.
- Consolini, L., Morbidi, F., Prattichizzo, D., and Tosques, M. (2008). Leader – follower formation control of nonholonomic mobile robots with input constraints. *Autonomous*, 44, 1343–1349.
- Coupland, S. (2003). *Type-2 Fuzzy Control of a Mobile Robot*.
- Cui, R., Sam, S., Voon, B., How, E., and Sang, Y. (2010). Leader – follower formation control of underactuated autonomous underwater vehicles. *Ocean Engineering*, 37(17-18), 1491–1502.
- Dai, Y., and Lee, S. G. (2014). Formation Control of Mobile Robots with Obstacle Avoidance based on GOACM using Onboard Sensors. *International Journal of Control, Automation, and Systems*, 12, 1077–1089.

- Dang, Q.V., Nielsen, I., and G. Bocewicz. (2013). Modelling and scheduling autonomous mobile robot for a real-world industrial application,” *IFAC Proc. Vol.*, pp. 2098–2103.
- Das, A. K., Fierro, R., Kumar, V., Ostrowski, J. P., Spletzer, J., and Taylor, C. J. (2002). A Vision-Based Formation Control Framework. *IEEE Transaction on Robotics and Automation*, 18(5), 813–825.
- Defago, X., and Konagaya, A. (2002). Circle Formation for Oblivious Anonymous Mobile Robots with No Common Sense of Orientation.
- Desai, J. P., Ostrowski, J., and Kumar, V. (1998). Controlling formations of multiple mobile robots. *IEEE Transaction on Robotics and Automation*, 2863 – 2869.
- Desai, J. P., Ostrowski, J. P., and Kumar, V. (2001). Modeling and Control of Formations of Nonholonomic Mobile Robots. *IEEE Transaction on Robotics and Automation*, 17(6), 905–908.
- De La Cruz, C., and Carelli, R. (2008). Dynamic model based formation control and obstacle avoidance of multi-robot systems,” *Robotica*, vol. 26, no. 03.
- Diirr, H.-B., Stankovic, M. S., Dimarogonas, D. V., Ebenbauer, C., and Johansson, K. H. (2013). Obstacle Avoidance for an Extremum Seeking System using a Navigation Function, 4068 – 4073.
- Di Paola, D., Milella, A., Cicirelli, G., and Distanto, A. (2010). An Autonomous Mobile Robotic System for Surveillance of Indoor Environments,” vol. 7, no. 1, pp. 19–26.
- Dimarogonas, D. V, Tsiotras, P., and Kyriakopoulos, K. J. (2009). Leader-Follower Cooperative Attitude Control of Multiple Rigid Bodies. *Systems &*

Control Letters, 58(6), 429–435.

Dudek, G., Jnekin, M., Milios, E., and Wilkes, D. (1993). A Taxonomy for Swarm Robot. *IEEE*, 00(C), 441–447.

Eberhart, R. C., and Kennedy, J. (1995). A new optimizer using particle swarm theory. In *Proceedings of the sixth international symposium on micro machine and human science* (pp. 39–43), Nagoya, Japan. Piscataway: IEEE.

Eberhart, R. C., and Shi, Y. (2000). Comparing inertia weights and constriction factors in particle swarm optimization. In *Proceedings of the IEEE congress on evolutionary computation (CEC)* (pp. 84–88), San Diego, CA. Piscataway: IEEE.

Eberhart, R. C., and Shi, Y. (2001). Tracking and optimizing dynamic systems with particle swarms. In *Proceedings of the IEEE congress on evolutionary computation (CEC)* (pp. 94–100), Seoul, Korea. Piscataway: IEEE.

Egerstedt, M., and Hu, X. (2001). Formation Constrained Multi-Agent Control. *IEEE Transaction on Robotics and Automation*, 17(6), 947–951.

El-Desouky, and Hagrass, H. (2009). An Adaptive Type-2 Fuzzy based Agent for Multi Occupant Ambient Intelligent Environment. In *Proceeding of 5th International Conference on Intelligent Environment*.

Faigl, J., Krajník, T., Chudoba, J., Přeučil, L., and Saska, M. (2013). Low-cost embedded system for relative localization in robotic swarms. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on* (pp. 993–998).

Feng, Y., Teng, GF., Wang, A.X., and Yao., Y.M. (2007). Chaotic Inertia Weight in Particle Swarm Optimization”, In *Innovative Computing, Information and Control, ICICIC’07. Second International Conference on*, page 475.

- Gao, Y., An, X., and Liu., J (2008). A Particle Swarm Optimization Algorithm with Logarithm Decreasing Inertia Weight and Chaos Mutation, In Computational Intelligence and Security. International Conference on, volume 1, pages 61–65.
- Gamage, G. W., Mann, G. K. I., and Gosine, R. G. (2009). Formation Control of Multiple Nonholonomic Mobile Robots via Dynamic Feedback Linearization. *IEEE*.
- Garro, B. A., Sossa, H., and Vazquez, R. A. (2007). Evolving ant colony system for optimizing path planning in mobile robots. *Fourth Congress of Electronics, Robotics and Automotive Mechanics*, 444–449.
- Gazi, V., Marques, L., and Ordóñez, R. “Robot Swarms: Dynamics and Control,” pp. 1–32, 2014.
- Gaspar, J., Winters, N., and Santos-Victor, J. (2000). Vision-based navigation and environmental representations with an omnidirectional camera,” *IEEE Transactions on Robotics and Automation*, vol. 16, no. 6, pp. 890–898.
- Ghommam, J., Mehrjerdi, H., and Saad, M. (2011). Leader-Follower Formation Control of Nonholonomic Robots with Fuzzy Logic Based Approach for Obstacle Avoidance. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2340–2345.
- Gustavi, T., and Hu, X. (2008). Observer-Based Leader-Following Formation Control Using Onboard Sensor Information. *IEEE Transaction on Robotics and Automation*, 24(6), 292–297.
- Gu, D., and Hu, H., "Neural predictive control for a car-like mobile robot." *Robotics and Autonomous Systems* 39.2 (2002): 73-86.

- Handayani, A. S., Nurmaini, S., and Yani, I. (2015). Brief Review on Formation Control of Swarm Robot. *EECSI 2015*, (August), 19–20.
- Hagras, H., Callaghan, V., and Colley, M. (2004). Learning and adaptation of an intelligent mobile robot navigator operating in unstructured environment based on a novel online Fuzzy–Genetic system. *Fuzzy Sets and Systems*, *141*(1), 107-160.
- Hagras, H. A. (2004). A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots. *IEEE Transactions on fuzzy systems*, *12*(4), 524-539.
- Hernández-Martínez EG, and Aranda-Bricaire E. (2011). Convergence and collision avoidance in formation control: A survey of the artificial potential functions approach". In *Multi-Agent Systems-Modeling, Control, Programming, Simulations and Applications 2011*. InTech.
- Hoffmann, F. (2001). Evolutionary algorithms for fuzzy control system design. *Proceedings of the IEEE*, *89*(9), 1318-1333.
- Hassan, G. M., and Yahya, K. M. (2006). Leader-Follower Approach using Full-State Linearization via Dynamic Feedback. *ICET Second International Conference on Emerging Technologies*, (November), 297–305.
- He, L.-L., and LOU, X.-C. (2013). Study on the Formation Control Methods for Multi-Agent Based on Geometric, (1), 699–701.
- Huang, Y., Zhang, Y., Zhao, Q., and Cheng, J. (2015). Research on Multi-robot Cooperation Based on Leader-follower Method. *International Conference on Intelligent Systems Research and Mechantronics Engineering*, (Isrme), 1257–1265.

- Jamshidi, M., Vadiiee, N., and Ross, T. (1993). *Fuzzy logic and control: software and hardware applications* (Vol. 2). Pearson Education.
- John, R. (1998). Type 2 fuzzy sets: An appraisal of theory and applications. *International Journal Uncertainty, Fuzziness, Knowledge Based System*, 6(6):563–576.
- Juang, C-F, Jeng, T-L and Chang, Y-C. (2016). An Interpretable Fuzzy System Learned Through Online Rule Generation and Multi-objective ACO With a Mobile Robot Control Application." *IEEE transactions on cybernetics* 46.12: 2706-2718.
- Kang, J. W., Kim, S. J., Chung, M. J., Myung, H., Park, J. H., and Bang, S. W. (2007). Path Planning for Complete and Efficient Coverage Operation of Mobile Robots. *IEEE International Conference on Mechatronics and Automation*, 2126–2131.
- Karnik, N. N., Mendel, J. M., and Liang, Q. (1999). Type-2 Fuzzy Logic Systems. *IEEE Transaction on Fuzzy Systems*, 7(6), 643–658.
- Karnik, N.N., and Mendel, J.M. (1998). An introduction to type-2 fuzzy logic systems. USC report, <http://sipi.usc.edu/~mendel/report>.
- Karnik, N.N. and Mendel, J.M. (1999). Application of type-2 fuzzy logic system to forecasting of time-series. *Information Sciences*, 120: 89-111.
- Karnik, N.N., and Mendel, J.M. (2001). Operations on type-2 fuzzy sets. *Fuzzy Sets and Systems*, 122:327–348.
- Karnik, N.N. and Mendel, J.M. (2001a). Centroid of a type-2 fuzzy set. *Information Sciences*, 132:195-220.

- Karnik, N.N., and Mendel, J.M. (2002). Type-2 fuzzy logic systems. *IEEE Transactions on Fuzzy Systems*, 7(6):643-658
- Kennedy, J., Eberhart, R. C., and Shi, Y. (2001). *Swarm intelligence*. San Francisco: Kaufmann.
- Kessentini, S., and Barchiesi., D. (2015). Particle Swarm Optimization with Adaptive Inertia Weight. *International Journal of Machine Learning and Computing* 5(5):368-373.
- Kowdiki, K. H., Barai, R. K., and Bhattacharya, S. (2012). Leader-Follower Formation Control Using Artificial Potential Functions : A Kinematic Approach. *IEEE International Conference On Advances in Engineering, Science and Management*, 500 –505.
- Klir, G., and Yuan, B. (1995). *Fuzzy sets and fuzzy logic* (Vol. 4). New Jersey: Prentice hall.
- Kumra, S., Saxena, R., and Mehta, S. “An Extensive Review on Swarm Robotics,” pp. 140–145, 2009.
- Langer, R. A., Coelho, L. S., and Oliveira, G. H. C. (2007). K-Bug, A New Bug Approach for Mobile Robot’s Path Planning. *IEEE Multi-Conference on Systems and Control*, (October), 1–3.
- Lawton, J. R. T., Beard, R. W., and Young, B. J. (2003). A Decentralized Approach to Formation Maneuvers. *IEEE Transaction on Robotics and Automation*, 19(6), 933–941.
- Lei, B., and Li, W. (2007). A Fuzzy Behaviours Fusion Algorithm for Mobile Robot Real-time Path Planning in Unknown Environment. *IEEE International Conference On Integration Technology*, 173–178.

- Lewis, M. A., and Tan, K. (1997). High Precision Formation Control of Mobile Robots Using Virtual Structures. *Autonomous Robots* 4, 403, 387–403.
- Li, B., and Zhang, C. (2006). Adaptive fuzzy control for mobile robot obstacle avoidance based on virtual line path tracking. *Proceeding of the 2006 IEEE International Conference on Robotics and Biometrics*, (2), 1454–1458.
- Li, C., Li, Y., Zhou, F., and Fan, C. (2007). Multi-objective Path Planning for the Mobile Robot. *Proceedings of the IEEE International Conference on Automation and Logistics*, 2248–2252.
- Li, X., and Xiao, J. (2005). Robot Formation Control in Leader-Follower Motion Using Direct Lyapunov Method, *10*(3), 244–250.
- Li, X., Xiao, J., and Cai, Z. (2005). Backstepping Based Multiple Mobile Robots Formation, 887 – 892.
- Liang, Q., and Mendel, J. M. (2000). Interval Type-2 Fuzzy Logic Systems : Theory and Design. *IEEE Transaction on Fuzzy System*, 8(5 october), 535–550.
- Liang, Q., Karnik, N.N. and Mendel, J.M. (2000). Connection admission control in ATM networks using survey-based type-2 fuzzy logic systems. *IEEE Transaction System Man and Cybernetic*, 30:329–339.
- Liang, Q. and Mendel. J.M. (2000a) Interval type-2 fuzzy logic systems: theory and design. *IEEE Trans. on Fuzzy Systems*, 8:535-550.
- Liang, Q. and Mendel, J.M. (2000b). Equalization of nonlinear time-varying channels using type-2 fuzzy adaptive filters. *IEEE Transactions on Fuzzy Systems*, 8: 551-563.

- Liang, Q. and Mendel, J.M. (2001). Mpeg vbr video traffic modeling and classification using fuzzy technique. *IEEE Transactions on Fuzzy Systems*, 9: 183-193.
- Loria, A., Dasdemir, J., and Jarquin, N. A. (2015). Leader–Follower Formation and Tracking Control of Mobile Robots Along Straight Paths. *IEEE Transaction on Control Systems Technology*, 1–6.
- Malik, R.F., Rahman, T.A., Hashim, S.Z.M., and Ngah, R. (2007) New Particle Swarm Optimizer with Sigmoid Increasing Inertia Weight. *International Journal of Computer Science and Security (IJCSS)*, 1(2):35.
- Mariottini, G. L., Morbidi, F., Prattichizzo, D., Pappas, G. J., and Daniilidis, K. (2007). Leader-Follower Formations : Uncalibrated Vision-Based Localization and Control. *IEEE International Conference on Robotics and Automation*, (April), 10–14.
- Mataric, M. J. (1983). Behavior-Based Control : Examples from Navigation , Learning , and Group Behavior Maja J Mataric Abstract 1 Introduction 2 Deliberative and Reactive Approaches, 1 –15.
- Marino, A. Parker, L.E., Antonelli, G., and Caccavale, F. A Decentralized Architecture for Multi-Robot Systems Based on the Null-Space-Behavioral Control with Application to Multi-Robot Border Patrolling, 2012.
- Mendel, J.M. (2000). Uncertainty, fuzzy logic, and signal processing. *Signal Proc. J.*, 80: 913–933.
- Mendel, J. M. (2001). *Uncertain rule-based fuzzy logic systems: introduction and new directions* (pp. 131-184). Upper Saddle River: Prentice Hall PTR.

- Mendel, J.M., and Wu, H. (2002). Uncertainty versus choice in rule-based fuzzy logic systems. *Proceedings of IEEE on Fuzzy Systems*, 1336-1342.
- Mendel, J. M., and John, R. B. (2002). Type-2 fuzzy sets made simple. *IEEE Transactions on fuzzy systems*, 10(2), 117-127.
- Mendel, J. M., and John, R. B. (2002). Type-2 fuzzy sets made simple. *IEEE Transactions on fuzzy systems*, 10(2), 117-127.
- Mendel, J. M. (2001). *Uncertain rule-based fuzzy logic systems: introduction and new directions* (pp. 131-184). Upper Saddle River: Prentice Hall PTR.
- Mendel, J. M. (2007). Type-2 Fuzzy Sets and Systems: An Overview. *IEEE Computational Intelligence Magazine*, 2(February), 20 – 29.
- Mendel, J. M. (2009). Type-2 Fuzzy Sets and Systems : How to Learn About Them. *IEEE SMC eNewsletter*, (27 June), 1–8.
- Mendel, J. M. (2007). Type-2 Fuzzy Sets and Systems: An Overview. *IEEE Computational Intelligence Magazine*, 2(February), 20 – 29.
- Mendel, J. M. (2009). Type-2 Fuzzy Sets and Systems : How to Learn About Them. *IEEE SMC eNewsletter*, (27 June), 1–8.
- Melin, P., and Castillo, O. (2004). A new method for adaptive control of non-linear plants using type-2 fuzzy logic and neural networks. *International Journal of General Systems*, 33(2):289-304.
- Mies, G. Military robots of the present and the future. (2010). *Aarms*, vol. 9, no. 1, pp. 125–137.

- Mohan, Y., and Ponnambalam, S. G. (2009). An Extensive Review of Research in Swarm Robotics. *IEEE*, 140–145.
- Mondada, F., Pettinaro, G.C., Guignard, A., Kwee, I.W., Floreano, D., Deneubourg, .L., Nolfi, S., Gambardella, L.M. and Dorigo, M. (2004). SWARM-BOT: A new distributed robotic concept. *Autonomous robots*, 17(2-3), pp.193-221.
- Ni, W., and Cheng, D. (2010). Leader- Following Consensus of Multi Agent Systems Under Fixed and Switching Topologies. *Systems & Control Letters*, 59(3-4), 209–217.
- Ni, J., Wu, L., Fan, X., and Yang, S. X. (2016). Bioinspired intelligent algorithm and its applications for mobile robot control: a survey. *Computational intelligence and neuroscience*, 2016, 1.
- Nurmaini, S., and Hashim, S. Z. M. (2008). An Embedded Fuzzy Type-2 Controller Based Sensor Behavior for Mobile Robot. *Eighth International Conference on Intelligent Systems Design and Applications An*, 29–34.
- Nurmaini, S., and Primanita, A. (2012). Modeling of Mobile Robot System with Control Strategy Based on Type-2 Fuzzy Logic. *International Journal of Information and Communication Tehnology Research*, 2(3), 235–242.
- Nurmaini, S., and Tutuko, B. (2011). A New Control Architecture in Mobile Robot Navigation based on IT2Neuro-Fuzzy Controller. *International Conference on Electrical Engineering and Informatics*, (July).
- Nurmaini, S and Tutuko, B. Motion coordination for swarm robots. (2014). *Proc. - 2014 Int. Conf. ICT Smart Soc. Smart Syst. Platf. Dev. City Soc. GoeSmart* pp. 312–315, 2014.

- Nurmaini, S., Zaiton, S., and Firnando, R. (2014). Cooperative Avoidance Control-based Interval Fuzzy Kohonen Networks Algorithm in Simple Swarm Robots. *Telkomnika*, 12(4).
- Olfati Saber, R.M., and Murray, R. Graph rigidity and distributed formation stabilization of multi vehicle systems.” p. Proceedings of the 41st IEEE Conference on Decision, 2002.
- Palm, R., and Bouguerra, A. (2013). Particle Swarm Optimization of Potential Fields for Obstacle Avoidance, 117 – 123.
- Payá, L., A. Gil, and O. Reinoso. A State-of-the-Art Review on Mapping and Localization of Mobile Robots Using Omnidirectional Vision. (2017). *Sensors. Journal of Sensors 2017* (2017).
- Pedersen, S.M., Fountas, S., Have, H., and Blackmore, B.S. (2006). Agricultural robots - System analysis and economic feasibility,” *Precis. Agric.*, vol. 7, no. 4, pp. 295–308.
- Peng, Z., Wen, G., and Rahmani, A. (2013). Leader-Follower Formation Control of Multiple nonholonomic Robots Based on Backstepping, 211–216.
- Peng, Z., Wen, G., Rahmani, A., and Yu, Y. (2013). Leader – follower formation control of nonholonomic mobile robots based on a bioinspired neurodynamic based approach. *Robotics and Autonomous Systems*, 61(9), 988–996.
- Park, Bong Seok, et al. (2009). Adaptive neural sliding mode control of nonholonomic wheeled mobile robots with model uncertainty." *IEEE Transactions on Control Systems Technology* 17.1: 207-214.
- Pimenta, L. C. A., Pereira, G. A. S., Michael, N., Mesquita, R. C., Bosque, M. M., Chaimowicz, L., and Kumar, V. (2013). Swarm Coordination Based on

Smoothed Particle Hydrodynamics Technique, 29(2), 383–399.

Pinciroli, C., Trianni, V., O’Grady, R., Pini, G., Brutschy, A., Brambilla, M., ... and Birattari, M. (2012). ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm intelligence*, 6(4), 271-295.

Pugh, J., and Martinoli, A. (2007). Inspiring and modelling multi-robot search with particle swarm optimization. In *Swarm Intelligence Symposium, 2007. SIS 2007. IEEE* (pp. 332-339). IEEE.

Rao, A. M., Ramji, K., Rao, B. S. S., Vasu, V., and Puneeth, C. (2017). Navigation of non-holonomic mobile robot using neuro-fuzzy logic with integrated safe boundary algorithm. *International Journal of Automation and Computing*, 14(3), 285-294.

Ren, W and Randal, B. (2004). Decentralized Scheme for Space Craft Formation Flying via the Virtual Structure Approach.

Rigatos, G. G. (2008). Distributed gradient and particle swarm optimization for multi-robot motion planning. *Robotica*, 26, 357–370.

Rm, K. C., Singaperumal, M., and Nagarajan, T. (2012). Behavior Based Multi Robot Formations with Active Obstacle Avoidance based on Switching Control Strategy, 440, 6630–6635.

Saffiotti, A. (1997). The uses of fuzzy logic in autonomous robot navigation. *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, 1(4), 180-197.

Sahin, E., Labella, T. H., Trianni, V., Deneubourg, J.-L., Rasse, P., Floreano, D., ... Dorigo, M. (2002). SWARM-BOT : Pattern Formation in a Swarm of Self-Assembling Mobile Robots.

- Sahin, E., Spears, W.M. and Windfield, A.F.T. *Swarm Robotics*, no. October. 2007.
- Sakurama, K., and Nakano, K. (2008). Leader-Following Formation Navigation for Multiple Robots with Collision Avoidance. *Proceeding of The 17 Th World Congress The International Federation of Automatic Control*, 12099–12104.
- Shao, J., Xie, G., Yu, J., and Wang, L. (2005). Leader-following Formation Control of Multiple Mobile Robots. *Proceeding of the 2005 IEEE International Symposium on Intelligent Control*, 808–813.
- Sharkey, A. J. C. (2006). *The Application of Swarm Intelligence to Collective Robots. in Advances in Applied Artificial Intelligence.*
- Sharkey, A. J. C. (2007). Swarm robotics and minimalism. *Connection Science*, 19(January 2015), 37–41.
- Soorki, M. N., Talebi, H., and Nikravesh, S. K. (2011). Robust Leader-following Formation Control of Multiple Mobile Robots using Lyapunov Redesign. *IEEE*, 277–282.
- Sugihara, K., and Suzuki, I. (1990). Distributed Motion Coordination of Multiple Mobile Robots. *IEEE*, (1), 138–143.
- Sun, D., Kleiner, A., and Nebel, B. “Behavior based Multi Robot Collision Avoidance,” pp. 1668–1673, 2014.
- Tan, K., and Lewis, M. A. (1996). Virtual Structures for High-Precision Cooperative Mobile Robotic Control *. *Proc IROS 96*, 132–139.

- Tan, Y., and Zheng, Z. Y. (2013). Research advance in swarm robotics. *Defence Technology*, 9(1), 18-39.
- Tanner, H. G., and Piovesan, J. L. (2010). Randomized Receding Horizon Navigation. *IEEE Transaction on AutomatiC Control*, 55(11), 2640–2644.
- Tardos, J.D., Neira, J., Newman, P.M. and Leonard, J.J. “Robust mapping and localization in indoor environments using sonar data,” *International Journal of Robotics Research*, vol. 21, no. 4, pp. 311–330, 2002.
- Tian, J., Gao, M., and Lu, E. (2007). Dynamic Collision Avoidance Path Planning for Mobile Robot Based on Multi-sensor Data Fusion by Support Vector Machine. *IEEE International Conference on Mechatronics and Automation*, 2779–2783.
- Tran, V., and Lee, S., “A Stable Formation Control Using Approximation of Translational and Angular Accelerations,” vol. 8, no. 1, pp. 65–75, 2011.
- Wagner, C., and Hagra, H. (2007). A Genetic Algorithm Based Architecture for Evolving Type-2 Fuzzy Logic Controllers for Real World Autonomous Mobile Robots. *IEEE*.
- Wang, J.-L., and Wu, H.-N. (2012). Leader-following formation control of multi-agent systems under fixed and switching topologies. *International Journal of Control*, (March 2015), 37–41.
- Wijk, O., and Christensen, H. “Localization and navigation of a mobile robot using natural point landmarks extracted from sonar data,” *Robotics and Autonomous Systems*, vol. 31, no. 1-2, pp. 31–42, 2000.

- Wu, D. (2013). Approaches for reducing the computational cost of interval type-2 fuzzy logic systems: overview and comparisons. *IEEE Transactions on Fuzzy Systems*, 21(1), 80-99.
- Wu, H., and Mendel, J.M. (2001). Introduction to uncertainty bounds and their use in the design of interval type-2 fuzzy logic systems. *IEEE International Conference on Fuzzy Systems*, USA, 3:662-665.
- Xu, D., Zhang, X., Zhu, Z., Chen, C., and Yang, P. “Behavior-based formation control of swarm robots,” *Math. Probl. Eng.*, vol. 2014, 2014.
- Yogeswaran, M and Ponnambalam, S. “Swarm Robotics: An Extensive Research Review,” *World Congr. Nat. Biol. Inspired Comput. NaBIC .*, pp. 140–145, 2009
- Yu, C., and Zhang, D. “A new 3D map reconstruction based mobile robot navigation,” in *Proceedings of the 8th International Conference on Signal Processing*, vol. 4, IEEE, Beijing, China, November 2006.
- Zadeh, L. A. (1965). Fuzzy Sets * -. *Information and Control*, 353, 338–353.
- Zadeh, L.A. (1975). The concept of a linguistic variable and its application to approximate reasoning-1. *Information Science*, 8:199-249.
- Zhang, and Ziyang. (2008). Research on Hierarchical Fuzzy Behavior Learning of Autonomous Robot. *International Conference on Internet in Science and Engineering*, 43 – 46.
- Zhang, L., and Ghosh, B.K. “Line segment based map building and localization using 2D laser range nder,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '00)*, vol. 3, pp. 2538–2543, April 2000.

Zhu, Q., Avidan, S., Yeh, M.C and Cheng, K.T. (2006). Fast human detection using a cascade of histograms of oriented gradients. in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 1491–1498.

Zu, L., Yang, P., Chen, L., Zhang, X., and Tian, Y. (2008). Obstacle Avoidance of Multi Mobile Robots Based on Behavior Decomposition Reinforcement Learning, 1018–1023.

LAMPIRAN – 1

**SURAT KETERANGAN PERBAIKAN UJIAN KOMPREHENSIF
DISERTASI**

SURAT KETERANGAN PERBAIKAN
UJIAN KOMPREHENSIF DISERTASI (TEK011317)

Dosen Penguji Ujian Komprehensif Disertasi Bidang Ilmu Teknik Informatika Fakultas Teknik Universitas Sriwijaya :

Nama : Prof. Ir. Subriyer Nasir, MS, Ph.D

NIP : 196009091987031004

Menyatakan bahwa mahasiswa sebagai berikut :

Nama : Bambang Tutuko

NIM : 03043681621003

BKU : Teknik Informatika

Judul : Sistem Kontrol Formasi Berdasarkan Kombinasi

Metoda Logika Fuzzy Tipe-2 dan Dinamik Particle

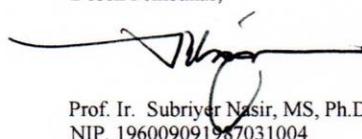
swarm Optimization

Telah melakukan perbaikan Ujian Komprehensif Disertasi sesuai dengan yang telah disarankan oleh Dosen penguji pada berita acara Ujian Komprehensif Disertasi.

Demikian surat keterangan ini dibuat untuk dipergunakan sebagaimana mestinya.

Palembang, Juli 2019

Dosen Pembahas,



Prof. Ir. Subriyer Nasir, MS, Ph.D
NIP. 196009091987031004

SURAT KETERANGAN PERBAIKAN
UJIAN KOMPREHENSIF DISERTASI (TEK011317)

Dosen Penguji Ujian Komprehensif Disertasi Program Studi Ilmu Teknik Bidang Ilmu
Teknik Informatika Program Doktor Fakultas Teknik Universitas Sriwijaya :

Nama : Prof. Dr. Ghazali bin Sulong (MSU)

NIP : --

Menyatakan bahwa mahasiswa sebagai berikut :

Nama : Bambang Tutuko

NIM : 03043681621003

BKU : Teknik Informatika

Judul : Sistem Kontrol Formasi Berdasarkan Kombinasi
Metoda Logika Fuzzy Tipe-2 dan Dinamik Particle
swarm Optimization

Telah melakukan perbaikan Ujian Komprehensif Disertasi sesuai dengan yang telah
disarankan oleh Dosen penguji pada berita acara Ujian Komprehensif Disertasi.

Demikian surat keterangan ini dibuat untuk dipergunakan sebagaimana mustinya.

Palembang, Juli 2019
Dosen Pembahas.



Prof. Dr. Ghazali bin Sulong (MSU)

SURAT KETERANGAN PERBAIKAN
UJIAN KOMPREHENSIF DISERTASI (TEK011317)

Dosen Penguji Ujian Komprehensif Disertasi Bidang Ilmu Teknik Informatika Fakultas Teknik Universitas Sriwijaya :

Nama : Reza Firsandaya Malik, ST, MT, PhD

NIP : 197604252010121001

Menyatakan bahwa mahasiswa sebagai berikut :

Nama : Bambang Tutuko

NIM : 03043681621003

BKU : Teknik Informatika

Judul : Sistem Kontrol Formasi Berdasarkan Kombinasi

Metoda Logika Fuzzy Tipe-2 dan Dinamik Particle

swarm Optimization

Telah melakukan perbaikan Ujian Komprehensif Disertasi sesuai dengan yang telah disarankan oleh Dosen penguji pada berita acara Ujian Komprehensif Disertasi.

Demikian surat keterangan ini dibuat untuk dipergunakan sebagaimana mestinya.

Palembang, Juli 2019

Dosen Pembahas,



Reza Firsandaya Malik, ST, MT, PhD
NIP.197604252010121001

SURAT KETERANGAN PERBAIKAN
UJIAN KOMPREHENSIF DISERTASI (TEK011317)

Dosen Penguji Ujian Komprehensif Disertasi Bidang Ilmu Teknik Informatika Fakultas Teknik Universitas Sriwijaya :

Nama : Dr. Yusuf Hartono, M.Sc

NIP : 196411161990031002

Menyatakan bahwa mahasiswa sebagai berikut :

Nama : Bambang Tutuko

NIM : 03043681621003

BKU : Teknik Informatika

Judul : Sistem Kontrol Formasi Berdasarkan Kombinasi

Metoda Logika Fuzzy Tipe-2 dan Dinamik Particle

swarm Optimization

Telah melakukan perbaikan Ujian Komprehensif Disertasi sesuai dengan yang telah disarankan oleh Dosen penguji pada berita acara Ujian Komprehensif Disertasi.

Demikian surat keterangan ini dibuat untuk dipergunakan sebagaimana mestinya.

Palembang, Juli 2019
Dosen Pembahas,



Dr. Yusuf Hartono, M.Sc
NIP. 196411161990031002

SURAT KETERANGAN PERBAIKAN
UJIAN KOMPREHENSIF DISERTASI (TEK011317)

Dosen Penguji Ujian Komprehensif Disertasi Bidang Ilmu Teknik Informatika Fakultas Teknik Universitas Sriwijaya :

Nama : Prof. Dr. Ir. Nukman, MT
NIP : 195903211987031001

Menyatakan bahwa mahasiswa sebagai berikut :

Nama : Bambang Tutuko
NIM : 03043681621003
BKU : Teknik Informatika
Judul : Sistem Kontrol Formasi Berdasarkan Kombinasi
Metoda Logika Fuzzy Tipe-2 dan Dinamik Particle
swarm Optimization

Telah melakukan perbaikan Ujian Komprehensif Disertasi sesuai dengan yang telah disarankan oleh Dosen penguji pada berita acara Ujian Komprehensif Disertasi.

Demikian surat keterangan ini dibuat untuk dipergunakan sebagaimana mestinya.

Palembang, Juli 2019
Dosen Pembahas,



Prof. Dr. Ir. Nukman, MT
NIP. 195903211987031001

LAMPIRAN – 2

**SURAT KETERANGAN PERBAIKAN SIDANG TERBUKA PROMOSI
DOKTOR**

**SURAT KETERANGAN PERBAIKAN
SIDANG TERBUKA PROMOSI DOKTOR**

Ketua Penguji Sidang Terbuka Promosi Doktor Bidang Ilmu Teknik Informatika
Fakultas Teknik Universitas Sriwijaya :

Nama : Prof. Dr. Ir. Anis Saggaff, MSCE

NIP : 196210281989031002

Menyatakan bahwa mahasiswa sebagai berikut :

Nama : Bambang Tutuko

NIM : 03043681621003

BKU : Teknik Informatika

Judul : Sistem Kontrol Formasi Berdasarkan Kombinasi

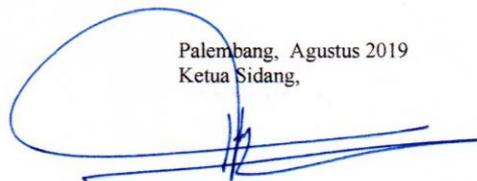
Metoda Logika Fuzzy Tipe-2 dan Dinamik Particle

swarm Optimization

Telah melakukan perbaikan Disertasi sesuai dengan yang telah disarankan oleh Dosen
penguji pada berita acara Ujian Sidang Terbuka Promosi Doktor.

Demikian surat keterangan ini dibuat untuk dipergunakan sebagaimana mustinya.

Palembang, Agustus 2019
Ketua Sidang,



Prof. Dr. Ir. Anis Saggaff, MSCE
NIP. 196210281989031002

SURAT KETERANGAN PERBAIKAN
UJIAN KOMPREHENSIF DISERTASI (TEK011317)

Dosen Penguji Ujian Komprehensif Disertasi Program Studi Ilmu Teknik Bidang Ilmu
Teknik Informatika Program Doktor Fakultas Teknik Universitas Sriwijaya :

Nama : Prof. Dr. Ghazali bin Sulong (MSU)

NIP : --

Menyatakan bahwa mahasiswa sebagai berikut :

Nama : Bambang Tutuko

NIM : 03043681621003

BKU : Teknik Informatika

Judul : Sistem Kontrol Formasi Berdasarkan Kombinasi

Metoda Logika Fuzzy Tipe-2 dan Dinamik Particle
swarm Optimization

Telah melakukan perbaikan Ujian Komprehensif Disertasi sesuai dengan yang telah
disarankan oleh Dosen penguji pada berita acara Ujian Komprehensif Disertasi.

Demikian surat keterangan ini dibuat untuk dipergunakan sebagaimana mustinya.

Palembang, Agustus 2019
Dosen Pembahas,



Prof. Dr. Ghazali bin Sulong (MSU)

SURAT KETERANGAN PERBAIKAN
SIDANG TERBUKA PROMOSI DOKTOR

Dosen Penguji Sidang Terbuka Promosi Doktor Bidang Ilmu Teknik Informatika
Fakultas Teknik Universitas Sriwijaya :

Nama : Prof. Ir. Subriyer Nasir, MS, Ph.D

NIP : 196009091987031004

Menyatakan bahwa mahasiswa sebagai berikut :

Nama : Bambang Tutuko

NIM : 03043681621003

BKU : Teknik Informatika

Judul : Sistem Kontrol Formasi Berdasarkan Kombinasi

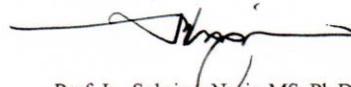
Metoda Logika Fuzzy Tipe-2 dan Dinamik Particle

swarm Optimization

Telah melakukan perbaikan Disertasi sesuai dengan yang telah disarankan oleh dosen
penguji pada berita acara Ujian Sidang Terbuka Promosi Doktor.

Demikian surat keterangan ini dibuat untuk dipergunakan sebagaimana mestinya.

Palembang, Agustus 2019
Dosen Pembahas,



Prof. Ir. Subriyer Nasir, MS, Ph.D
NIP. 196009091987031004

SURAT KETERANGAN PERBAIKAN
SIDANG TERBUKA PROMOSI DOKTOR

Dosen Penguji Sidang Terbuka Promosi Doktor Bidang Ilmu Teknik Informatika
Fakultas Teknik Universitas Sriwijaya :

Nama : Reza Firsandaya Malik, ST, MT, PhD

NIP : 197604252010121001

Menyatakan bahwa mahasiswa sebagai berikut :

Nama : Bambang Tutuko

NIM : 03043681621003

BKU : Teknik Informatika

Judul : Sistem Kontrol Formasi Berdasarkan Kombinasi

Metoda Logika Fuzzy Tipe-2 dan Dinamik Particle

swarm Optimization

Telah melakukan perbaikan Disertasi sesuai dengan yang telah disarankan oleh dosen penguji pada berita acara Ujian Sidang Terbuka Promosi Doktor.

Demikian surat keterangan ini dibuat untuk dipergunakan sebagaimana mestinya.

Palembang, Agustus 2019
Dosen Pembahas,



Reza Firsandaya Malik, ST, MT, PhD
NIP.197604252010121001

SURAT KETERANGAN PERBAIKAN
SIDANG TERBUKA PROMOSI DOKTOR

Dosen Penguji Sidang Terbuka Promosi Doktor Bidang Ilmu Teknik Informatika
Fakultas Teknik Universitas Sriwijaya :

Nama : Dr. Yusuf Hartono, M.Sc

NIP : 196411161990031002

Menyatakan bahwa mahasiswa sebagai berikut :

Nama : Bambang Tutuko

NIM : 03043681621003

BKU : Teknik Informatika

Judul : Sistem Kontrol Formasi Berdasarkan Kombinasi

Metoda Logika Fuzzy Tipe-2 dan Dinamik Particle
swarm Optimization

Telah melakukan perbaikan Disertasi sesuai dengan yang telah disarankan oleh dosen
penguji pada berita acara Ujian Sidang Terbuka Promosi Doktor.

Demikian surat keterangan ini dibuat untuk dipergunakan sebagaimana mestinya.

Palembang, Agustus 2019
Dosen Pembahas,



Dr. Yusuf Hartono, M.Sc
NIP. 196411161990031002

SURAT KETERANGAN PERBAIKAN
SIDANG TERBUKA PROMOSI DOKTOR

Dosen Penguji Sidang Terbuka Promosi Doktor Bidang Ilmu Teknik Informatika
Fakultas Teknik Universitas Sriwijaya :

Nama : Prof. Dr. Ir. Nukman, MT

NIP : 195903211987031001

Menyatakan bahwa mahasiswa sebagai berikut :

Nama : Bambang Tutuko

NIM : 03043681621003

BKU : Teknik Informatika

Judul : Sistem Kontrol Formasi Berdasarkan Kombinasi
Metoda Logika Fuzzy Tipe-2 dan Dinamik Particle
swarm Optimization

Telah melakukan perbaikan Disertasi sesuai dengan yang telah disarankan oleh dosen
penguji pada berita acara Ujian Sidang Terbuka Promosi Doktor.

Demikian surat keterangan ini dibuat untuk dipergunakan sebagaimana mestinya.

Palembang, Agustus 2019
Dosen Pembahas,



Prof. Dr. Ir. Nukman, MT
NIP. 195903211987031001