TUGAS AKHIR - EE 184801

PERANCANGAN SISTEM KENDALI OTOMATIS PADA *AUTOMATED GUIDED VEHICLE* (*AGV*) MENGGUNAKAN KONTROLER LOGIKA *FUZZY* MAMDANI

Adi Novitarini Putri
NRP. 07111745000011

Dosen Pembimbing
Prof. Dr. Ir. Achmad Jazidie, M.Eng.
Imam Arifin, ST., MT.
Lam Siew Kei, B.ASc., M.Eng., Ph.D.
Mohammad Hafid, S.Pd.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2019

TUGAS AKHIR – TE 141599

PERANCANGAN SISTEM KENDALI OTOMATIS PADA *AUTOMATED GUIDED VEHICLE* (*AGV*) MENGGUNAKAN KONTROLER LOGIKA *FUZZY* MAMDANI

Adi Novitarini Putri
NRP. 07111745000011

Dosen Pembimbing
Prof. Dr. Ir. Achmad Jazidie,M.Eng.
Imam Arifin, ST., MT.
Lam Siew Kei,BAsc.,M.Eng.,PhD.
Mohammad Hafid,S.Pd.

JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2019

FINAL PROJECT – **TE 141599**

DESIGN AUTOPILOT OF AN AGV USING MAMDANI FUZZY LOGIC CONTROLLER

Adi Novitarini Putri
NRP. 07111745000011

Supervisor
Prof. Dr. Ir. Achmad Jazidie,M.Eng.
Imam Arifin, ST., MT.
Lam Siew Kei,BAsc.,M.Eng.,PhD.
Mohammad Hafid,S.Pd.

DEPARTMENT OF ELECTRICAL ENGINEERING
Faculty of Electrical Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2019

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul :

"Perancangan Sistem Kendali Otomatis pada *Automated Guided Vehicle* (*AGV*) menggunakan Kontroler Logika *Fuzzy* Mamdani"

adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 5 Juli 2019

Adi Novitarini Putri
NRP. 07111745000011

Halaman ini sengaja dikosongkan

PERANCANGAN SISTEM KENDALI OTOMATIS PADA *AUTOMATED GUIDED VEHICLE* (*AGV*) MENGGUNAKAN KONTROLER LOGIKA *FUZZY* MAMDANI

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan Untuk
Memperoleh Gelar Sarjana Teknik Elektro
Pada
Bidang Studi Teknik Sistem Pengaturan
Jurusan Teknik Elektro
Institut Teknologi Sepuluh Nopember

Menyetujui

Dosen Pembimbing I,

Prof. Dr. Ir. Achmad Jazidie, M.Eng.
Nip : 195902191986101001

Dosen Pembimbing II,

Imam Arifin, ST., MT.
Nip : 197302222002121001

Dosen Pembimbing III,

Lam Siew Kei, BAsc., M.Eng., PhD.
School of Computer Science and Engineering, Nanyang Technological University

Dosen Pembimbing IV,

Mohammad Hafid, S.Pd.
PT. Industrial Robotic Automation

KEMENTERIAN RISET, TEKNOLOGI, DAN PENDIDIKAN TINGGI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
FAKULTAS TEKNOLOGI ELEKTRO
DEPARTEMEN TEKNIK ELEKTRO

SURABAYA
JUNI, 2019

Halaman ini sengaja dikosongkan

PERANCANGAN SISTEM KENDALI OTOMATIS PADA *AUTOMATED GUIDED VEHICLE* (*AGV*) MENGGUNAKAN KONTROLER LOGIKA *FUZZY* MAMDANI

Adi Novitarini Putri
07111745000011

Dosen Pembimbing I	: Prof Dr. Ir. Achmad Jazidie,M.Eng
NIP	: 195902191986101001
Dosen Pembimbing II	: Imam Arifin, ST., MT.
NIP	: 197302222002121001
Dosen Pembimbing III	: Lam Siew Kei,BAsc.,M.Eng.,PhD. Nanyang Technological University, Singapore
Dosen Pembimbing IV	: Mohammad Hafid,S.Pd. PT. Industrial Robotics Automation

ABSTRAK

Automated Guided Vehicle (*AGV*) merupakan kendaraan tanpa awak yang digunakan untuk memindahkan material dari satu lokasi ke posisi tertentu. Tujuan dari penelitian ini adalah menghasilkan sebuah *control rule* yang dapat mengendalikan kecepatan robot agar mampu bergerak sesuai dengan lintasan yang diinginkan. *Control rule* Kanayama merupakan sebuah metode yang digunakan untuk mencari nilai kecepatan linier dan anguler pada *mobile robot*. Berdasarkan hasil pengujian sistem kendali otomatis menggunakan *control rule* Kanayama, ketika metode ini diterapkan pada sebuah lintasan berbentuk lingkaran, nilai RMSE yang dihasilkan sebesar 0.1305 pada sumbu X dan sumbu Y sebesar 0.0361. Kontroler logika *fuzzy* digunakan untuk menggantikan peran *control rule* Kanayama. Hasil penggunaan kontroler logika *fuzzy* untuk sistem KENDALI otomatis *AGV* adalah nilai RMSE pergerakan robot pada sumbu-X sebesar 0.2401 dan sumbu Y sebesar 0.2011.

Kata Kunci : *AGV*, KENDALI otomatis, perencanaan lintasan, kontroler logika *fuzzy*

Halaman ini sengaja dikosongkan

DESIGN AUTOPILOT FOR AUTOMATED GUIDED VEHICLE (AGV) USING FUZZY LOGIC CONTROLLER MAMDANI

Adi Novitarini Putri
07111745000011

Supervisor I	: Prof Dr. Ir. Achmad Jazidie,M.Eng
ID Number	: 195902191986101001
Supervisor II	: Imam Arifin, ST., MT.
ID Number	: 197302222002121001
Supervisor III	: Lam Siew Kei,BAsc.,M.Eng.,PhD. Nanyang Technological University, Singapore
Supervisor IV	: Mohammad Hafid,S.Pd. PT. Industrial Robotics Automation

ABSTRACT

Automated Guided Vehicle (AGV) is an unmanned vehicle used to move material from one location to a certain position. The generation trajectory is the process of creating a trajectory to connect two or more points on the desired trajectory. The purpose of this study is to produce a control rule that is able to make the robot move in accordance with the existing trajectory. Kanayama control rule is used to find the target speed value (linier and angular speed) but its use has an error value of 0.1305 on the X axis and Y axis of 0.0361. The use of fuzzy logic controllers is used to replace the role of Kanayama's control rule and the RMSE of robot movements on the X-axis is 0.2401 and the Y axis is 0.2011.

Keywords: *AGV, autopilot, tracking trajectory, fuzzy logic controller*

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji dan syukur kami panjatkan kehadirat Allah SWT, karena berkat rahmat dan karunia-Nya, penulis dapat menyelesaikan laporan penelitian dengan judul "**Perancangan Sistem Kendali Otomatis pada *Automated Guided Vehicle* (*AGV*) menggunakan Kontroler Logika *Fuzzy* Mamdani"** untuk memenuhi syarat kelulusan pada Bidang Studi Teknik Sistem Pengaturan, Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS) Surabaya.

Laporan penelitian ini dapat diselesaikan oleh penulis berkat bantuan, bimbingan, dan dukungan dari berbagai pihak. Penulis ingin mengucapkan terima kasih kepada orang tua, kakak, dan adik tercinta yang selalu memberi dukungan, semangat, dan doa untuk keberhasilan penulis. Bapak Prof. Dr. Ir. Achmad Jazidie,M.Eng. dan Bapak Imam Arifin, ST., MT., selaku dosen pembimbing atas bimbingan dan arahannya. Bapak Mohamad Abdul Hady, ST., MT. dan Bapak Ir. Ali Fatoni, MT. selaku pembimbing laboratorium yang telah mendidik penulis menjadi lebih baik. Dosen Bidang Studi Teknik Sistem Pengaturan dan Jurusan Teknik Elektro atas pendidikan dan ilmunya. Keluarga laboratorium AJ-104 dan CyPIRAL atas bantuan dan kerja samanya. Teman-teman dan semua pihak yang tidak dapat penulis sebutkan satu persatu.

Penulis berharap laporan ini dapat bermanfaat bagi pembaca pada umumnya dan penulis pada khususnya. Laporan ini masih jauh dari sempurna, sehingga penulis mengharapkan kritik dan saran dari pembaca yang bersifat membangun.

Surabaya, 23 Mei 2019

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

LEMBAR PENGESAHAN.. i
ABSTRAK..iii
ABSTRACT.. v
KATA PENGANTAR... vii
DAFTAR ISI.. ix
DAFTAR GAMBAR.. xi
DAFTAR TABEL.. xv
BAB I PENDAHULUAN
1.1 Latar belakang.. 1
1.2 Permasalahan 2
1.3 Tujuan 3
1.4 Batasan Masalah 3
1.5 Metodologi 3
1.6 Sistematika Penelitian 3
BAB II SISTEM KENDALI OTOMATIS PADA AUTOMATED GUIDED VEHICLE (AGV)
2.1 *Automated Guided Vehicle* (*AGV*) 5
2.2 Sistem KENDALI Otomatis 6
2.3 *Control Rule* Kanayama 6
2.4 Indeks Performansi Kesalahan 11
2.4.1 RMSE (Root Mean Square Error)............................. 11
2.4.2 ISE (Integral Square Error) 11
2.4.3 IAE (Integral Absolute Error) 12
2.5 Kontroler Logika *Fuzzy* 12
2.4.4 Fuzzifikasi... 13
2.4.5 *Fuzzy* Inference *Rules* 13
2.4.6 Defuzzifikasi 14
BAB III PERANCANGAN SISTEM KENDALI OTOMATIS
3.1 Perancangan Mekanik *AGV* 17
3.2 Model Kinematika *AGV*.. 20
3.3 Perancangan Sistem KENDALI Otomatis menggunakan *Control Rule* Kanayama... 22
3.4 Perancangan Sistem KENDALI Otomatis menggunakan Kontroler Logika *Fuzzy* 24
BAB IV PENGUJIAN DAN ANALISIS
4.1 Pengujian Sistem secara *Open Loop* 31
4.2 Pengujian *control rule* Kanayama.................................... 35
4.2.1 Pengujian parameter *control rule* Kanayama dengan .. 35

masukan sinyal unit-step (Kx=100, Ky=16, dan Kθ=2)........... 35
4.2.2 Pengujian parameter *control rule* Kanayama dengan... 37
masukan sinyal unit-step (Kx=100, Ky=16, dan Kθ=8)........... 37
4.2.3 Pengujian parameter *control rule* Kanayama dengan... 39
masukan sinyal unit-step (Kx=100, Ky=16, dan Kθ=16)......... 39
4.2.4 Pengujian parameter *control rule* Kanayama dengan... 43
Referensi Lintasan Lingkaran (Variasi Parameter Kx)............ 43
4.2.5 Pengujian parameter *control rule* Kanayama dengan... 45
Referensi Lintasan Lingkaran (Variasi Parameter Ky)............ 45
4.2.6 Pengujian parameter *control rule* Kanayama dengan... 48
Referensi Lintasan Lingkaran (Variasi Parameter Kθ)............ 48
4.3 Pengujian Lintasan Lingkaran menggunakan *Control Rule* Kanayama 51
4.4 Pengujian Kontroler Logika *Fuzzy*... 52
4.3.1. Variasi Jumlah Fungsi Keanggotaan 52
4.3.2. Variasi Kondisi Awal (*initial condition*) *AGV* 58
BAB V PENUTUP
5.1 Kesimpulan 65
5.2 Saran.. 65
DAFTAR PUSTAKA…………………………………………………67
LAMPIRAN………………………………………………………... 71
RIWAYAT PENULIS………………………………………………75

DAFTAR GAMBAR

Gambar 2.1 Postur referensi dan aktual 5
Gambar 2.2 Gambaran umum dari robot motion planning 6
Gambar 2.3 Representasi kesalahan posisi dan orientasi 7
Gambar 2.4 *Konsep Fuzzy Logic* 12
Gambar 2.5 Proses pengambilan keputusan metode Mamdani 13
Gambar 2.6 Metode defuzzifikasi aturan Mamdani 15
Gambar 3.1 Perancangan body *AGV* 17
Gambar 3.2 Perancangan steering *AGV* 18
Gambar 3.3 Mekanisme steering angle pada roda belakang 18
Gambar 3.4 Tampilan *AGV* tampak atas 19
Gambar 3.5 Tampilan *AGV* tampak samping 20
Gambar 3.6 Model kinematika *AGV* 20
Gambar 3.7 Diagram blok sistem KENDALI otomatis AGV menggunakan control rule Kanayama 22
Gambar 3.8 Diagram blok sistem KENDALI otomatis pada AGV menggunakan kontroler logika *Fuzzy* 24
Gambar 3.9 Fungsi keanggotaan 'R' dan 'θe' 25
Gambar 3.10 Diagram alir proses fuzzifikasi untuk kesalahan posisi (R) 26
Gambar 3.11 Diagram alir proses fuzzifikasi untuk kesalahan orientasi (θe) 27
Gambar 3.12 Sinyal k*control* kecepatan linier (v) dan anguler (ω) 29
Gambar 4.1 Diagram blok pengujian sistem secara *open loop* 32
Gambar 4.2 Respon posisi robot pada sumbu X dengan ζ=0.25 36
Gambar 4.3 Respon posisi robot pada sumbu Y dengan ζ=0.25 36
Gambar 4.4 Respon orientasi robot dengan ζ=0.25 37
Gambar 4.5 Respon posisi robot pada sumbu X dengan ζ=1 38
Gambar 4.6 Respon posisi robot pada sumbu Y dengan ζ=1 38
Gambar 4.7 Respon orientasi robot dengan ζ=1 39
Gambar 4.8 Respon posisi robot pada sumbu-X dengan ζ=2 40
Gambar 4.9 Respon posisi robot pada sumbu-Y dengan ζ=2 40
Gambar 4.10 Respon orientasi robot dengan ζ=2 41
Gambar 4.11 Respon posisi robot pada sumbu X dengan Variasi ζ 42
Gambar 4.12 Respon posisi robot pada sumbu Y dengan Variasi ζ 42
Gambar 4.13 Respon orientasi robot dengan Variasi ζ......................... 43
Gambar 4.14 Respon posisi robot pada sumbu X dengan Variasi Parameter Kx 43

Gambar 4.15 Respon posisi robot pada sumbu Y dengan Variasi Parameter Kx..45
Gambar 4.16 Respon posisi robot pada sumbu X dengan Variasi Parameter Ky..46
Gambar 4.17 Respon posisi robot pada sumbu Y dengan Variasi Parameter $K\boldsymbol{y}$ 48
Gambar 4.18 Respon posisi robot pada sumbu X dengan Variasi Parameter $K\theta$ 49
Gambar 4.19 Respon posisi robot pada sumbu Y dengan Variasi Parameter $K\theta$ 50
Gambar 4.20 Perbandingan antara postur referensi dan aktual dengan parameter *control rule* Kanayama Kx = 100 Ky=100 $K\theta$=1 51
Gambar 4.21 Pengujian Perancangan sistem kendali otomatis menggunakan KLF dengan tiga fungsi keanggotaan.............................52
Gambar 4.22 Fuzzifikasi R dengan lima fungsi keanggotaan...............53
Gambar 4.23 Fuzzifikasi θe dengan lima fungsi keanggotaan..............53
Gambar 4.24 Pengujian KLF dengan lima fungsi keanggotaan............54
Gambar 4.25 Fuzzifikasi R dengan sembilan fungsi keanggotaan 55
Gambar 4.26 Fuzzifikasi θe dengan sembilan fungsi keanggotaan 56
Gambar 4.27 Perancangan sistem kendali otomatismenggunakan KLF dengan sembilan fungsi keanggotaan...57
Gambar 4.28 Respon pergerakan robot pada sumbu X dengan variasi jumlah fungsi keanggotaan...57
Gambar 4.29 Respon pergerakan robot pada sumbu Y dengan variasi jumlah fungsi keanggotaan...58
Gambar 4.30 Respon pergerakan robot pada sumbu X menggunakan KLF dengan initial condition x=3 dan y=5 58
Gambar 4.31 Respon pergerakan robot pada sumbu Y menggunakan KLF dengan initial condition x=3 dan y=5 59
Gambar 4.32 Pengujian KLF dengan initial condition x=3 dan y=5 60
Gambar 4.33 Respon pergerakan robot pada sumbu X menggunakan KLF dengan initial condition x=-2.5 dan y=-3......................................60
Gambar 4.34 Respon pergerakan robot pada sumbu Y menggunakan KLF dengan initial condition x=-2.5 dan y=-3......................................61
Gambar 4.35 Pengujian KLF dengan initial condition x=-2.5 dan y=-3 61
Gambar 4.36 Respon pergerakan robot pada sumbu X menggunakan KLF dengan initial condition x=2 dan y=0.7 62

Gambar 4.37 Respon pergerakan robot pada sumbu Y menggunakan KLF dengan initial condition x=2 dan y=0.7 62
Gambar 4.38 Pengujian *path tracking* menggunakan KLF dengan initial condition x=2 dan y=0.7 63

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Tabel 3.2 Nomeclature dari kontroler logika *fuzzy* Mamdani 24
Tabel 3.3 *Rule* based untuk kecepatan linier 28
Tabel 3.4 *Rule* based untuk kecepatan anguler...................................... 28
Tabel 4.1 Pengujian Model Kinematika AGV (Percobaan 1) 32
Tabel 4.2 Pengujian Model Kinematika AGV (Percobaan 2) 33
Tabel 4.3 Pengujian Model Kinematika AGV (Percobaan 3) 33
Tabel 4.4 Pengujian Model Kinematika AGV (Percobaan 4) 34
Tabel 4.5 Pengujian Model Kinematika AGV (Percobaan 5) 34
Tabel 4.6 Indeks performansi sistem dengan variasi parameter konstanta Kx (pada sumbu X).. 44
Tabel 4.7 Indeks performansi sistem dengan variasi parameter konstanta Kx (pada sumbuY).. 45
Tabel 4.8 Indeks performansi sistem dengan variasi parameter konstanta $\boldsymbol{Ky}$ (pada sumbu X) 47
Tabel 4.9 Indeks performansi sistem dengan variasi parameter konstanta Ky (pada sumbu Y).. 48
Tabel 4.10 Indeks performansi sistem dengan variasi parameter konstanta $K\theta$ (pada sumbu X).. 49
Tabel 4.11 Indeks performansi sistem dengan variasi parameter konstanta $K\theta$ (pada sumbu Y).. 51
Tabel 4.12 *Rule based* untuk kecepatan linier 53
Tabel 4.13 *Rule* based untuk kecepatan anguler.................................... 54
Tabel 4.14 *Rule based* untuk kecepatan linier 55
Tabel 4.15 *Rule* based untuk kecepatan anguler.................................... 56
Tabel 4.16 Perbandingan jumlah fungsi keanggotaan........................... 58

Halaman ini sengaja dikosongkan

BAB I
PENDAHULUAN

Pada bab pendahuluan ini berisi mengenai latar belakang penelitian mengenai *AGV*. *AGV* banyak diterapkan karena memiliki kelebihan untuk meningkatkan efisiensi pada sistem logistik di industri. Permasalahan yang muncul dalam pergerakan *AGV* adalah kemampuannya untuk bergerak sesuai dengan lintasan yang diinginkan. Penelitian ini dibatasi oleh model kinematika dari robot berdasarkan konfigurasi jenis roda yang digunakan. Tujuan penelitian ini adalah menghasilkan sebuah metode untuk melakukan perancangan sistem KENDALI otomatis untuk pergerakan sebuah *AGV*.

1.1 Latar belakang

Sistem robotika telah mengambil peran penting dalam kehidupan manusia salah satunya untuk membantu dan menggantikan tenaga kerja di industri. Prioritas utama dalam pengelolaan pabrik adalah efisiensi agar biaya operasional yang dikeluarkan tetap minimal [1]. Salah satu peranan dari robot dapat ditemukan pada sistem logistik, contoh peranannya adalah *material handling*. Meskipun penggunaan konveyor banyak digunakan untuk memindahkan material namun beberapa industri besar mengandalkan kendaraan sebagai sistem transportasinya [2].

Automated Guided Vehicle (*AGV*) merupakan salah satu tipe *unmanned vehicle* yang dirancang untuk membawa material serta memindahkannya [3]. Kelebihan sistem ini dibandingkan dengan pemindahan barang secara manual adalah kemampuanya untuk mereduksi kesalahan yang disebabkan oleh manusia dan biaya produksi [4] sehingga banyak digunakan di industri saat ini. Transportasi konvensional yang terdapat di *warehouse* perlu ditingkatkan untuk mencapai efisiensi yang diinginkan sekaligus menjamin keselamatan ketika pengoperasian [5].

Perkembangan *AGV* juga diikuti dengan kemajuan sistem navigasinya. Beberapa sistem navigasi yang dapat digunakan di *AGV* adalah *wiring*, *magnetic tape*, dan *laser triangulation*. Teknologi *wiring* telah ditemukan oleh Barrett Electronics pada tahun 1953. Pada

pertengahan tahun 1980 dikembangkannya navigasi menggunakan *magnetic tape*. KENDALIan pada akhir tahun 1980, teknologi laser mulai digunakan sebagai navigasi kendaraan tanpa awak. Penggunaan kabel sebagai sistem navigasi *AGV* dianggap kurang fleksibel karena memerlukan antenna pada setiap roda yang digunakan untuk mendeteksi sinyal dari *wire*. Namun, laser dapat menyebabkan pergerakan robot lebih lambat karena perlu memetakan keseluruhan ruangan. *Magnetic tape* sebagai sistem navigasi untuk *AGV* menghasilkan respon robot untuk bergerak menuju posisi yang diinginkna dapat dicapai dengan lebih cepat [6].

Berdasarkan beberapa penelitian yang ada, penggunaan kontroler PI, PD [7], dan PID yang dikembangkan dapat digunakan mengendalikan pergerakan robot. Kelemahan dari sistem ini hanya dapat diimplementasikan pada kondisi linier saja sedangkan *AGV* merupakan sistem yang bersifat nonlinier dan dinamik kompleks. *control rule* Kanayama yang diajukan pada [9] menggunakan posisi dan orientasi aktual dari robot yang dibandingkan dengan posisi dan orientasi referensi sehingga menghasilkan kesalahan posisi dan orientasi (kesalahan postur). Kesalahan postur tersebut digunakan sebagai masukan untuk *control rule* Kanayama untuk mengendalikan pergerakan robot.

Pada penelitian [10] kontroler logika *fuzzy* mulai digunakan untuk mengendalikan pergerakan *mobile robot* yang memiliki jenis roda omnidirectional. Penelitian ini berfokus pada perancangan kontroler logika *fuzzy* menggunakan aturan Mamdani untuk merancang sistem kendali otomatis pada pergerakan *AGV* yang memiliki konfigurasi roda berjenis *rear wheel*.

1.2 Permasalahan

AGV harus memiliki kapasitas untuk bergerak dari posisi awal menuju akhir selain itu untuk mengerjakan tugas tertentu *AGV* memerlukan lintasan yang halus agar mampu bekerja secara otomatis untuk bergerak sesuai dengan lintasan. Namun *AGV* merupakan sistem yang bersifat non-linier sehingga kontroler yang dirancang harus mampu memastikan kestabilannya.

1.3 Tujuan

Pada penelitian ini dilakukan untuk menghasilkan sebuah perancangan sistem kendali otomatis pada pergerakan *AGV*.

1.4 Batasan Masalah

Penelitian ini menggunakan model *mobile robot* dengan konfigurasi roda berjenis *rear-wheel*. Skema logika *fuzzy* yang digunakan adalah Mamdani dengann jenis fuzzifikasi menggunakan fungsi segitiga sama kaki.

1.5 Metodologi

Pada penelitian ini akan dilakukan beberapa tahap, diantaranya studi literatur mengenai permodelan kinematika, perancangan mekanik, perancangan *software*, pengujian dan analisis serta penulisan buku laporan akhir.

1.6 Sistematika Penelitian

Penulisan laporan penelitian ini dapat dijelaskan dengan sistematika sebagai berikut:

BAB 1 PENDAHULUAN

Bab ini membahas tentang latar belakang, permasalahan, batasn masalah, tujuan penelitian, metodologi, sistematika penulisan dan relevansi dari penelitian yang dilakukan.

BAB 2 SISTEM KENDALI OTOMATIS PADA *AGV*

Konsep dan teori yang mendasari penelitian ini meliputi deskripsi dari *AGV*, representasi posisi dan orientasi sebuah *mobile robot* pada *global frame*, indeks performansi kesalahan, *control rule* Kanayama meliputi pengaruh beberapa parameter yang digunakan pada metode ini terhadap respon pergerakan robot serta representasi kesalahan posisi dan orientasi sebuah *mobile robot*, dan kontroler logika *fuzzy* (KLF) dijelaskan pada bab ini.

BAB 3 PERANCANGAN SISTEM

Perancangan perangkat keras meliputi *body* serta kerangka *steering*, model kinematika *AGV*, sistem kendali otomatis *AGV* menggunakan *control rule* Kanayama, dan kontoler logika *fuzzy* merupakan bagian dari bab ini.

BAB 4   PENGUJIAN DAN ANALISIS

Bab ini membahas tentang validasi model kinematika dengan melakukan pengujian sistem secara *open loop* dan dibandingkan dengan pergerakan aktual dari *AGV* yang dibangun, hasil simulasi dari perancangan sistem KENDALI otomatis menggunakan *control rule* Kanayama dan logika *fuzzy* pada simulator *AGV* serta analisis performa sistem.

BAB 5   PENUTUP

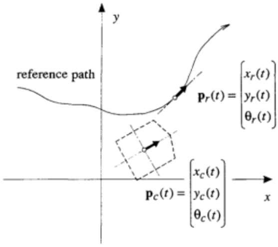
Bab 5 membahas tentang kesimpulan berkaitan dengan penelitian yang dilakukan.

BAB II
SISTEM KENDALI OTOMATIS PADA *AUTOMATED GUIDED VEHICLE* (*AGV*)

Bab II ini berisi mengenai beberapa teori penunjang yang dibutuhkan untuk melakukan perancangan sistem kendali otomatis pada pergerakan AGV. Deskripsi sebuah AGV, tahapan yang diperlukan dalam merancang pergerakan robot, *control rule* Kanayama serta kontroler logika *fuzzy* dapat ditemui pada bab ini.

2.1 *Automated Guided Vehicle* (*AGV*)

AGV merupakan sebuah kendaraan tanpa awak yang digunakan untuk memindahkan benda dari satu posisi ke posisi yang lainnya. Pada sebuah AGV, terdapat 4 komponen yakni yang pertama adalah kendaraan. Pada kendaraan tersebut terdapat dua sub system yakni *drive configuration* dan sistem navigasi. Bagian kedua adalaah *peripheral system* yakni berisi mekanisme pemindahan beban atau muatan atau sistem penggisian daya secara otomatis. On-Site system berisi aspek yang mempengaruhi pergerakan *AGV* misalnya jalan, gates (gerbang), lift, dan lain sebagainya. *Stationary control system* berfungsi sebagai pusat kendali dari *AGV* yang berisi administrasi permintaan, optimisasi jalur atau rute yang harus dilalui, serta visualisasinya (*Human Machine Interface*).



Gambar 2.1 Postur referensi dan aktual

AGV dapat disebut sebagai *mobile robot. Mobile robot* bergerak pada bidang Cartesin 2 dimensi dan memiliki tiga derajad kebebasan (*Degree*

of Freedom) dimana posisinya diwakili oleh postur p seperti pada persamaan (2.1) [9].

$$p = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \quad \ldots(2.1)$$

Dimana θ merupakan orientasi robot yang diukur dari sumbu x. Robot ini memiliki kapabilitas untuk bergerak pada bidang Cartesian 2D seperti pada Gambar 2.1.

2.2 Sistem Kendali Otomatis

Perancangan pergerakan robot memerlukan beberapa tahap, seperti pada Gambar 2.2 dimana tugas (*task*) yang harus dikerjakan oleh sebuah robot akan menjadi masukan pada proses *path planning* sehingga menghasilkan keluaran berupa *path* atau jalur. *Path* (jalur) dan *trajectory* (lintasan) merupakan dua hal yang berbeda. *Path* terdiri dari urutan titik (*waypoint*) yang harus diikuti oleh sebuah robot sedangkan lintasan merupakan sebuah *path* (jalur) ditambah dengan kecepatan dan percepatan pada masing-masing titik yang dituju [11].

Gambar 2.2 Gambaran umum dari robot motion planning

Lintasan yang harus diikuti robot merupakan sinyal masukan untuk proses trajectory *planning* yang menjadi fokus pada penelitian ini sehingga sinyal keluaran yang dihasilkan adalah lintasan (trajectory). *Robot controller* merupakan proses kompleks karena bertugas untuk mengendalikan pergerakan robot sesuai dengan lintasan yang diberikan. *Trajectory generation* merupakan proses yang mana membangkitkan lintasan untuk menghubungkan dua titik atau lebih. Selain itu, *trajectory generation* bertujuan untuk menghasilkan sinyal masukan pada sistem k*control* gerakan (*motion*) yang memastikan bahwa lintasan yang direncanakan dijalankan dengan 6isect.

2.3 *Control Rule* Kanayama

Kinematika merupakan ilmu yang mempelajari gerak benda tanpa mempersoalkan gaya yang menyebabkan pergerakannya [12].

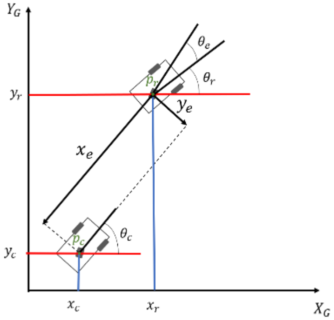
Pada penelitian [9] menghasilkan sebuah *control rule* yang menentukkan hasil kecepatan linier dan anguler dari robot.

$$q = \begin{bmatrix} v(p_e, q_r) \\ \omega(p_e, q_r) \end{bmatrix} = \begin{bmatrix} v_r cos\theta_e + K_x x_e \\ \omega_r + v_r \left(K_y y_e + K_\theta sin(\theta_e) \right) \end{bmatrix} \quad \ldots(2.2)$$

2.3.1 Kesalahan Postur [13]

Pergerakan robot menuju posisi dan orientasi referensi (x_r, y_r, θ_r) dibandingkan dengan posisi dan orientasi aktualnya (x_c, y_c, θ_c) dan menghasilkan kesalahan postur seperti pada Gambar 2.3.

$$\begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} x_r - x_c \\ y_r - y_c \\ \theta_r - \theta_c \end{bmatrix} \quad \ldots(2.3)$$



Gambar 2.3 Representasi kesalahan posisi dan orientasi

Dimana e_x merupakan kesalahan pada sumbu x, e_y merupakan kesalahan pada sumbu y dan e_θ merupakan kesalahan orientasi. Kesalahan postur tidak diberikan pada global frame tetapi nilai kesalahan pada local frame dari robot. Transformasi dibutuhkan untuk merubah matrix kesalahan postur ke dalam koordinat robot, menggunakan matriks putar sebagai berikut :

$$\begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = R(\theta) \begin{bmatrix} x_r - x_c \\ y_r - y_c \\ \theta_r - \theta_c \end{bmatrix} \quad \ldots(2.4)$$

$R(\theta)$ merupakan matriks rotasi terhadap sumbu Z

$$R(\theta) = \begin{bmatrix} cos(\theta) & sin(\theta) & 0 \\ -sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{...(2.5)}$$

Dimana $R(\theta)$ merupakan matriks putar terhadap sumbu Z.

$$p_e = \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} cos(\theta_c) & sin(\theta_c) & 0 \\ -sin(\theta_c) & cos(\theta_c) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x_c \\ y_r - y_c \\ \theta_r - \theta_c \end{bmatrix} \quad \text{...(2.6)}$$

Persamaan (3.8) dapat diselesaikan menjadi persamaan (3.9). Persamaan (3.9) merupakan kesalahan posisi dan orientasi.

$$\boldsymbol{p_e} = \begin{bmatrix} \boldsymbol{e_x} \\ \boldsymbol{e_y} \\ \boldsymbol{e_\theta} \end{bmatrix} = \begin{bmatrix} (\boldsymbol{x_r} - \boldsymbol{x_c})\boldsymbol{cos}(\boldsymbol{\theta_c}) + (\boldsymbol{y_r} - \boldsymbol{y_c})\boldsymbol{sin}(\boldsymbol{\theta_c}) \\ -(\boldsymbol{x_r} - \boldsymbol{x_c})\boldsymbol{sin}(\boldsymbol{\theta_c}) + (\boldsymbol{y_r} - \boldsymbol{y_c})\boldsymbol{cos}(\boldsymbol{\theta_c}) \\ \boldsymbol{\theta_r} - \boldsymbol{\theta_c} \end{bmatrix} \quad \text{...(2.7)}$$

Dinamika kesalahan postur didapatkan dengan melakukan diferensial pada Persamaan (2.7) sehingga menjadi Persamaan (2.8) (lihat Lampiran)

$$\begin{bmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_\theta \end{bmatrix} = \begin{bmatrix} v_r cos(\theta_e) - v_c + \omega_c y_e \\ -\omega_c x_e + v_r sin(\theta_e) \\ \omega_r - \omega_c \end{bmatrix} \quad \text{...(2.8)}$$

$\dot{p}_e$ dapat diasumsikan seperti pada Persamaan (2.9) dan diketahui jika $\dot{p}_e$ merupakan sebuah fungsi yang mengandung p_e, p_r, dan $q_r(v, \omega)$.

$$\dot{p}_e = \begin{bmatrix} v_r cos(\theta_e) - v_c + \omega_c y_e \\ -\omega_c x_e + v_r sin(\theta_e) \\ \omega_r - \omega_c \end{bmatrix} \quad \text{...(2.9)}$$

2.3.2 Pengaruh pada Variasi Parameter K_x, K_y, dan K_θ

Dinamika kesalahan postur didapatkan dengan melakukan diferensial pada Persamaan (2.9) sehingga menjadi Persamaan (2.10) (lihat Lampiran A)

$$\dot{p}_e = f(t, p_e) \begin{bmatrix} \left(\omega_r + v_r\left(K_x e_y + K_\theta sin(e_\theta)\right)\right) e_y - K_x e_x \\ -\left(\omega_r + v_r\left(K_x e_y + K_\theta sin(e_\theta)\right)\right) e_x + v_r sin(\theta_e) \\ -v_r\left(K_y e_y + K_\theta sin(e_\theta)\right) \end{bmatrix} \quad \text{...(2.10)}$$

Lineriasi persamaan (2.10) dengan cara membuat $\boldsymbol{p_e} = [\boldsymbol{e_x} \quad \boldsymbol{e_y} \quad \boldsymbol{e_\theta}]^{\boldsymbol{T}} = \boldsymbol{0}$ untuk menghasilkan nilai kesalahan postur sama dengan 0.

$$\left[\frac{\vartheta \dot{p}_e}{\vartheta p_e}\right]_{p_e=0} = \begin{bmatrix} \frac{\vartheta \dot{e}_x}{\vartheta e_x} & \frac{\vartheta \dot{e}_x}{\vartheta e_y} & \frac{\vartheta \dot{e}_x}{\vartheta e_\theta} \\ \frac{\vartheta \dot{e}_y}{\vartheta e_x} & \frac{\vartheta \dot{e}_y}{\vartheta e_y} & \frac{\vartheta \dot{e}_y}{\vartheta e_\theta} \\ \frac{\vartheta \dot{e}_\theta}{\vartheta e_x} & \frac{\vartheta \dot{e}_\theta}{\vartheta e_y} & \frac{\vartheta \dot{e}_\theta}{\vartheta e_\theta} \end{bmatrix} \quad \ldots(2.11)$$

Proses linierisasi (lihat pada Lampiran B) pada Persamaan (2.11) menghasilkan Persamaan (2.12) :

$$\dot{p}_e = Ap_e \quad \ldots(2.12)$$

Dimana matriks A berisi :

$$\left[\frac{\vartheta \dot{p}_e}{\vartheta p_e}\right]_{p_e=0} = \begin{bmatrix} \frac{\vartheta \dot{e}_x}{\vartheta e_x} & \frac{\vartheta \dot{e}_x}{\vartheta e_y} & \frac{\vartheta \dot{e}_x}{\vartheta e_\theta} \\ \frac{\vartheta \dot{e}_y}{\vartheta e_x} & \frac{\vartheta \dot{e}_y}{\vartheta e_y} & \frac{\vartheta \dot{e}_y}{\vartheta e_\theta} \\ \frac{\vartheta \dot{e}_\theta}{\vartheta e_x} & \frac{\vartheta \dot{e}_\theta}{\vartheta e_y} & \frac{\vartheta \dot{e}_\theta}{\vartheta e_\theta} \end{bmatrix} = \begin{bmatrix} -K_x & \omega_r & 0 \\ -\omega_r & 0 & v_r \\ 0 & -v_r K_y & -v_r K_\theta \end{bmatrix} \quad \ldots(2.13)$$

Substitusi Persamaan (3) ke dalam Persamaan kesalahan postur

$$p_e = R(\theta)(p_r - p_c) \quad \ldots(2.14)$$

Differensial dari Persamaan (2.14) :

$$\dot{p}_e = R(\theta)(\dot{p}_r - \dot{p}_c) + \dot{R}(\theta)(p_r - p_c) \quad \ldots(2.15)$$

Sehingga menghasilkan estimasi postur aktual seperti pada Persamaan (2.16):

$$\dot{p}_c = R(\theta)^{-1}\left(AR(\theta) - \dot{R}(\theta)\right)(p_c - p_r) + \dot{p}_r \quad \ldots(2.16)$$

Jika diasumsikan nilai $|e_\theta| < 1$ dan $|\dot{e}_\theta| < 1$ maka matriks $R(\theta)$ dan $R(\dot{\theta})$ merupakan matriks identitas dan matriks null sehingga :

$$\dot{p}_c = A(p_c - p_r) + \dot{p}_r \quad \ldots(2.17)$$

Jika respon yang diinginkan tidak memiliki osilasi namun respon tidak terlalu lambat maka pada penelitian ini mempertimbangkan pergerakan robot pada sumbu x dan dengan kecepatan linier robot bernilai konstan:

$$p_r(t) = \begin{bmatrix} x_r(t) \\ y_r(t) \\ \theta_r(t) \end{bmatrix} = \begin{bmatrix} v_r(t) \\ 0 \\ 0 \end{bmatrix} \text{ dan } q_r(t) = \begin{bmatrix} v_r(t) \\ \omega_r(t) \end{bmatrix} = \begin{bmatrix} v_r \\ 0 \end{bmatrix} \quad \ldots(2.18)$$

Nilai kesalahan orientasi beserta differensial kesalahan orientasi diasumsikan :

$$|e_\theta| \ll 1 \text{ dan } |\dot{e}_\theta| \ll 1 \quad \ldots(2.19)$$

Persamaan (6) dapat diselesaikan berdasarkan kondisi-kondisi pada Persamaan (7) dan (8) menjadi :

$$\dot{p}_c = \begin{bmatrix} -K_x & 0 & 0 \\ 0 & 0 & v_r \\ 0 & -v_r K_y & -v_r K_\theta \end{bmatrix} \begin{bmatrix} x_c - v_r(t) \\ y_c \\ \theta_c \end{bmatrix} + \dot{p}_r \qquad \ldots(2.20)$$

Ketika t=0 maka posisi aktual robot dapat diestimasi menggunakan persamaan :

$$x_c = \Delta x_c \qquad \ldots(2.21)$$

Berdasarkan Persamaan (2.20), nilai posisi aktual robot pada sumbu -X (x_c) dapat diselesaikan dengan Persamaan (2.22) sampai dengan Persamaan (2.26) :

$$\dot{x}_c = -K_x x_c + K_x v_t t + \dot{x}_r \qquad \ldots(2.22)$$

$$K_x x_c = K_x v_r t + \dot{x}_r - \dot{x}_c \qquad \ldots(2.23)$$

$$x_c = v_r t + \frac{1}{K_x}\dot{x}_r - \frac{1}{K_x}\dot{x}_c \qquad \ldots(2.24)$$

$$x_c = v_r t + \frac{1}{K_x}(\dot{x}_r - \dot{x}_c) \qquad \ldots(2.25)$$

$$x_c = v_r t + \frac{1}{K_x}(v_r - v_c) \qquad \ldots(2.26)$$

Sehingga nilai $\frac{1}{K_x}$ merupakan konstanta waktu. Pengaruh parameter K_y dan K_θ didaptkan dari penurunan rumus seperti pada Persamaan (2.27).

$$\dot{y}_c = v_r \theta_c \qquad \ldots(2.27)$$

$$u = v_r; du = \omega_r \qquad \ldots(2.28)$$

$$v = \theta_c; dv = \dot{\theta}_c = -v_r K_y y_c - v_r K_\theta \theta_c \qquad \ldots(2.29)$$

$$\ddot{y}_c = udv + vdu \qquad \ldots(2.30)$$

$$\ddot{y}_c = -{v_r}^2 K_y y_c - {v_r}^2 K_\theta \theta_c + \theta_c \omega_r \qquad \ldots(2.31)$$

$$\ddot{y}_c = -{v_r}^2 (K_y y_c + K_\theta \theta_c) \qquad \ldots(2.32)$$

Dari Persamaan (2.32) dapat diketahui bentuk differensialnya dapat ditulis sebagai berikut :

$$\ddot{y}_c + 2\zeta\omega_n \dot{y}_c + {\omega_n}^2 = 0 \qquad \ldots(2.33)$$

Sehingga nilai ζ dan ω_n bisa didapatkan dengan Persamaan (2.34) dan (2.35), dimana ζ disebut sebagai rasio redaman dan ω_n sebagai frekuensi alamiah tak teredam [14]

$$\zeta = \frac{K_\theta}{2\sqrt{K_y}} \quad \ldots(2.34)$$

$$\omega_n = v_r\sqrt{K_y} \quad \ldots(2.35)$$

2.4 Indeks Performansi Kesalahan

K*control* optimal tidak dapat didefinisikan dengan tepat. Solusi yang menurut sebuah masalah adalah kondisi optimal, mungkin bukan nilai yang optimal bagi permasalahan lain. Indeks performansi banyak digunakan oleh kalangan praktisi dan juga akademisi untuk membantu dalam menentukan kualitas sebuah sistem. Indeks performansi sendiri sebenarnya adalah fungsi hubungan dimana beberapa karakteristik sistem seperti kondisi optimal dari sistem didefinisikan. Indeks performansi secara umum dapat dirumuskan pada Persamaan (2.36) dimana J adalah indeks performansi dan e adalah kesalahan

2.4.1 RMSE (Root Mean Square Error)

Pada perhitungan kesalahan ini menggunakan hasil perbandingan nilai aktual dan referensi. RMSE adalah akar kuadrat rata-rata kuadrat kesalahan. Rumus RMSE adalah sebagai berikut:

$$\boldsymbol{RMSE = \frac{\sum_{n=1}^{n=k} y_r - y_n}{k}} \quad \ldots(2.36)$$

Dimana :

$\boldsymbol{y_n}$= Sinyal keluaran

$\boldsymbol{y_r}$= Sinyal masukan

K = Jumlah data

2.4.2 ISE (Integral Square Error)

Indeks performansi ISE ditunjukkan pada Persamaan (2.37). Indeks performansi ini berhubungan dengan *Mean Square Error* (MSE).

$$J = \int_0^\infty e^2(t)dt \quad \ldots(2.37)$$

ISE selain dapat digunakan untuk mendesain sebuah kontroler, juga dapat digunakan sebagai kriteria 11isecto mengenai seberapa baik sebuah sistem dengan kontroler dalam mengatasi pembebanan.

2.4.3 IAE (Integral Absolute Error)

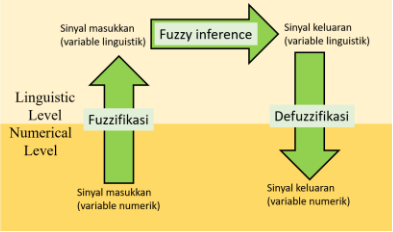
Indeks performansi IAE ditunjukkan pada Persamaan (2.38). IAE memiliki struktur kriteria yang paling mudah diaplikasikan dibandingkan dengan kriteria kesalahan lainnya, karena secara matematis indeks performansi tersebut mudah dioperasikan.

$$J = \int_{0}^{\infty} |e(t)| dt \tag{2.38}$$

Kriteria ini dapat menghilangkan kesalahan yang besar dan kesalahan yang kecil jika dibandingkan dengan ISE.

2.5 Kontroler Logika *Fuzzy*

Kontroler Logika *Fuzzy* (KLF) merupakan himpunan dari *linguistic control rules* yang berhubungan dengan *fuzzy implication* dan *fuzzy inference rules* [15]. Variabel 12isector1212 pada *fuzzy logic* adalah 12isector yang nilainya berupa kata-kata atau kalimat [16]. Misalnya 12isector usia, memiliki nilai berupa 12isector1212 seperti tua,muda, agak tua, agak muda, tua sekali, dan muda sekali bukan berupa bilangan seperti 21,43,53,70.



Gambar 2.4 *Konsep Fuzzy Logic*

Pada sistem pengaturan, KLF dibutuhkan ketika objek yang akan dikendali terlalu kompleks untuk dianalisis menggunakan metode konvensional atau sumber informasi yang tersedia bersifat kualitatif, tidak tepat, atau tidak pasti [15]. Konfigurasi KLF terdiri dari empat komponen

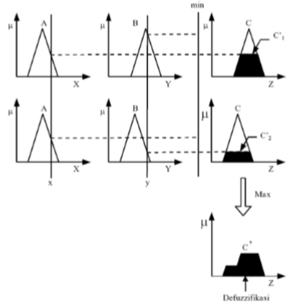
utama fuzzifikasi, *knowledge base*, *decision making logic* dan defuzzifikasi.

2.4.4 Fuzzifikasi

Fuzzifikasi merupakan proses untuk merubah sinyal masukan dalam bentuk *crisp* menjadi himpunan *fuzzy* yang dinyatakan dalam tingkat keanggotaan melalui fungsi keanggotaan yang didefinisikan. Dimana fungsi keanggotaan merupakan fungsi yang didefinisikan untuk suatu anggota himpunan *fuzzy* yang menggambarkan derajat kebenaran suatu kejadian. Umumnya bentuk atau pola nilai *crisp* dari fungsi keanggotaan berupa fungsi segitiga, gaussian, singleton, dan lain sebagainya.

2.4.5 *Fuzzy* Inference *Rules*

Knowledge base dari KLF terdiri dari dua komponen yang disebut *data base* dan *rule base* [15]. Data base berisi definisi perlu yang digunakan untuk mendeskripsikan *linguistic control rules*. Logika pengambilan keputusan (*fuzzy inference*) mengaplikasikan aturan-aturan *fuzzy* pada masukan *fuzzy* kendalian mengevaluasi setiap aturan. Prinsip logika *fuzzy* digunakan untuk mengkombinasi aturan-aturan JIKA-MAKA (IF-THEN) yang terdapat dalam basis aturan suatu pemetaan dari suatu himpunan *fuzzy* input himpunan *fuzzy* output. Logika pengambilan keputusan merupakan langkah kedua dalam pemrosesan logika *fuzzy*. Terdapat beberapa metode pengambilan keputusan dalam logika *fuzzy* diantaranya yaitu metode Mamdani.

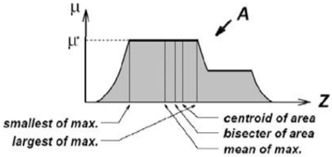


Gambar 2.5 Proses pengambilan keputusan metode Mamdani

Fungsi implikasi yang digunakan pada pengambilan keputusan dengan metode Mamdani dengan menggunakan MIN dan dalam melakukan komposisi dengan menggunakan MAX. Metode komposisi ini sering disebut MAX-MIN. Contoh dalam penggunaan pengambilan keputusan dengan metode Mamdani ditunjukkan pada Gambar 2.4 dengan memisalkan fungsi keanggotaan masukan dan keluaran menggunakan fungsi segitiga dan mempunyai dua aturan *fuzzy*, yaitu: − IF Kesalahan adalah Nol dan Beda kesalahan adalah Positif maka Keluaran adalah Positif. – IF Kesalahan adalah Nol dan Beda kesalahan adalah Nol maka Keluaran adalah Nol. Langkah pertama pengambilan keputusan metode Mamdani adalah melakukan proses fuzzifikasi untuk memetakan data tegas masukan kesalahan dan beda kesalahan data *fuzzy* sesuai dengan tipe dan bentuk fungsi keanggotaan. Langkah kedua adalah melakukan proses terhadap kedua data *fuzzy* tersebut dengan operator AND yang akan mengambil nilai paling minimal dari dua data tersebut. Langkah ketiga dengan implikasi MIN akan memotong fungsi keanggotaan keluaran setelah melalui operator AND sehingga didapatkan daerah *fuzzy*. Ketiga proses tersebut juga diterapkan pada aturanaturan *fuzzy* berikutnya. Setelah semua aturan *fuzzy* telah dieksekusi, dilakukan proses komposisi dengan metode MAX yaitu solusi himpunan *fuzzy* diperoleh dengan cara mengambil nilai maksimum aturan, kendalian menggunakannya untuk memodifikasi daerah *fuzzy*, dan mengaplikasikan ke output dengan menggunakan operator OR (union). Jika proposisi telah dievaluasi, maka output akan berisi suatu himpunan *fuzzy* yang merefleksikan kontribusi dari tiap-tiap proposisi. Setelah proses implikasi dan komposisi telah dilakukan maka proses selanjutnya adalah proses defuzzifikasi.

2.4.6 Defuzzifikasi

Masukan dari proses defuzzifikasi adalah suatu himpunan *fuzzy* yang diperoleh dari komposisi aturan-aturan *fuzzy*, sedangkan keluaran yang dihasilkan merupakan suatu bilangan pada *domain* himpunan *fuzzy* tersebut, sehingga jika diberikan suatu himpunan *fuzzy* dalam range tertentu, maka harus dapat diambil suatu nilai crisp tertentu sebagai keluarannya.



Gambar 2.6 Metode defuzzifikasi aturan Mamdani

Ada beberapa metode defuzzifikasi pada komposisi aturan Mamdani, diantaranya yaitu metode COA, 15isector, MOM, LOM, dan SOM seperti pada Gambar 2.5.

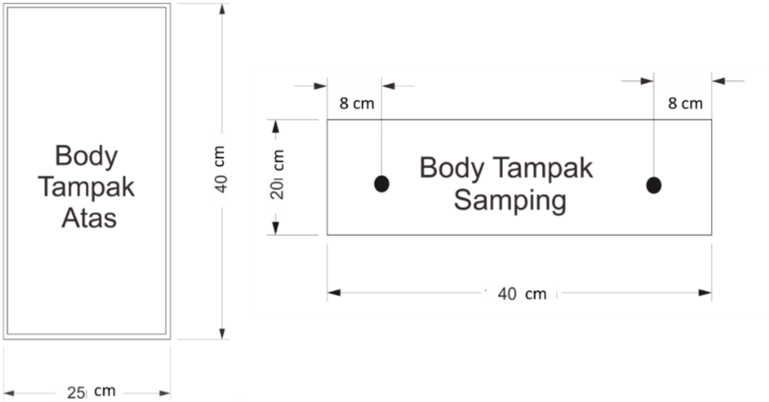
Halaman ini sengaja dikosongkan

BAB III
PERANCANGAN SISTEM KENDALI OTOMATIS

Bab ini berisi perancangan *AGV* meliputi perancangan dan implementasi perangkat keras, konfigurasi sistem beserta komponen yang digunakan, perancangan sistem kendali otomatis menggunakan *control rule* Kanayama dan kontroler logika *fuzzy*.

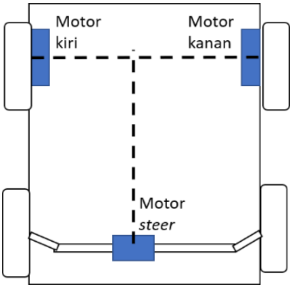
3.1 Perancangan Mekanik *AGV*

Gambar 3.1 merupakan perancangan body *AGV* menggunakan bahan aluminium, Gambar 1a merupakan gambar tampak atas dari body *AGV*. Panjang *AGV* adalah 40 cm dan lebarnya adalah 25 cm. Gambar 1b merupakan gambar tampak samping dari body *AGV*. Dimana pada bagian samping body akan diberi lubang berdiameter 2mm yang sesusai dengan diameter *shaft* yang ada pada motor DC.



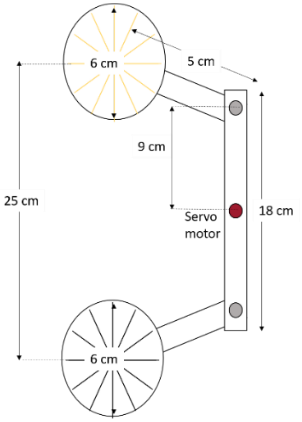
Gambar 3.1 Perancangan body *AGV*

Tipe roda pada *AGV* berjenis konfigurasi rear-wheel yang artinya, roda depan hanya berfungsi sebagai pengendali, sedangkan untuk penggeraknya atau *steering* dilakukan oleh roda belakang. Namun, karena lebih sukarnya pengendalian seperti ini maka metode Ackermann *steering* pun digunakan untuk diterapkan pada AGV seperti pada Gambar 3.2



Gambar 3.2 Perancangan steering *AGV*

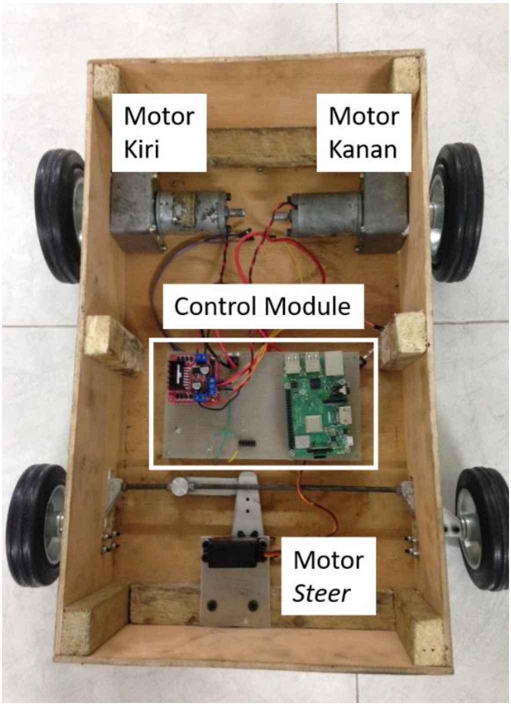
Gambar 3.2 merupakan mekanisme steering angle yang diimpleementasikan pada *AGV*. Bahan yang digunakan adalah alumunium yang dikendalian dibentuk seperti pada Gambar 3.3. Di bagian tengahnya akan diberi motor DC servo yang memiliki kapasitas untuk melakukan putaran sejauh 40^0 sampai dengan 70^0



Gambar 3.3 Mekanisme steering angle pada roda belakang

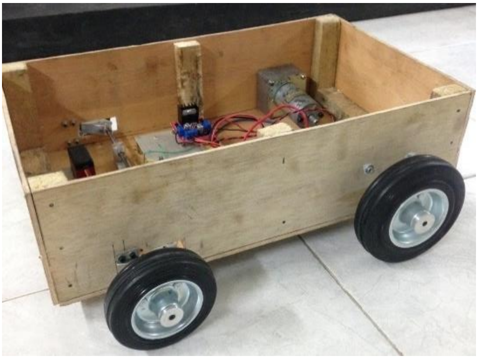
Pada Gambar 3.1 sampai dengan 3.3 merupakan perancangan *AGV* meliputi *chassis* dan *steer*. Perancangan kerangka *steer* dibuat menyerupai konsep kerja dari tuas (pengungkit). Servo motor yang diletakkan pada titik tengah rangka bertujuan sebagai penggerak, ketika servo bergerak ke kanan maka roda belakang juga ikut berbelok ka kanan

dikarenakan pergerakan as roda yang dikopel dengan servo motor. As roda yang digunakan untuk menyambung roda kanan dan kiri belakang berjenis besi ulir dengan diameter 8mm. Hasil implementasi dari perancangan tersebut dapat diamati pada Gambar 3.4



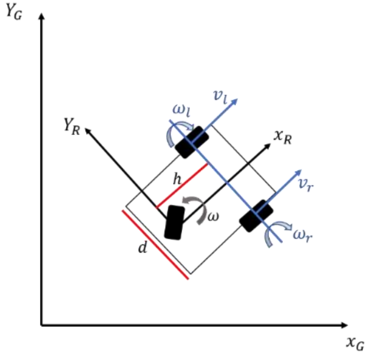
Gambar 3.4 Tampilan *AGV* tampak atas

Control module pada Gambar 3.4 berisi motor driver, rangkaian power, serta embedded system yang digunakan sebagai pusat pengendali. Motor kiri dan kanan yang digunakan berjenis motor DC gearbox. Kelebihan motor DC gearbox dibanding dengan jenis yang lain adalah putaran yang dihasilkan lambat dan torsinya besar namun untuk mengendalikan motor ini diperlukan *driver motor*. *Driver motor* memiliki kemampuan untuk merubah arah putar motor DC, mengatur kecepatan motor DC, serta motor steering yang digunakan merupakan motor servo.



Gambar 3.5 Tampilan *AGV* tampak samping

3.2 Model Kinematika *AGV*



Gambar 3.6 Model kinematika *AGV*

Gambar 3.6 merupakan geometri dari *mobile robot* yang digunakan untuk membangun model matematika dari *AGV*. Skema kinematika pada *mobile robot* yang digunakan pada penelitian ini terdiri dari dua buah roda yang berada pada satu sumbu dengan dua buah aktuator yang berbeda dengan sebuah *free wheel* dengan asumsi *body robot* bersifat *rigid* dan pergerakan terjadi tanpa ada *sliding* [17]. Namun, untuk mengetahui posisi *mobile robot* merupakan hal yang sukar karena *mobile robot* selalu bergerak terhadap lingkungannya sehingga perlu dilakukan estimasi posisi dengan cara mengintegrasikan gerakan robot setiap waktu [18]. Persamaan *nonholonomic constraint* pada robot mobile diformulasikan pada (3.1).

$$\dot{y}cos(\theta) - \dot{x}sin(\theta) = 0 \quad \ldots(3.1)$$

Dimana $\dot{x}$ dan $\dot{y}$ merupakan kecepatan robot terhadap sumbu X dan Y. Dari persamaan (3.1) dapat diketahui persamaan *forward kinematics* dari mobile base. *Forward kinematic* ini diperlukan untuk mengestimasi posisi dan orientasi robot pada bidang Cartesian [19]. Jarak antara roda kanan dan kiri dinotasikan sebagai d. Kecepatan linier pada roda penggerak sebelah kanan dan kiri adalah :

$$v_r = v + \omega \frac{d}{2} \quad \ldots(3.2)$$

$$v_l = v - \omega \frac{d}{2} \quad \ldots(3.3)$$

Dari Persamaan (3.2) dan (3.3) maka kecepatan anguler untuk roda kanan dan kiri dapat diformulasikan pada persamaan (3.4) dan (3.5) dengan r merupakan jari-jari roda.

$$v_r = r\omega_r \quad \ldots(3.4)$$

$$v_l = r\omega_l \quad \ldots(3.5)$$

Hubungan antara kecepatan linier dan anguler robot dengan kecepatan anguler roda kanan dan kiri dapat diformulasikan :

$$v = \frac{1}{2}(v_r + v_l) \quad \ldots(3.6)$$

$$\omega = \frac{r}{2d}(\omega_r - \omega_l) \quad \ldots(3.7)$$

Model kinematika *mobile robot* memiliki persamaan :

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} cos(\theta) & 0 \\ sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad \ldots(3.8)$$

Dari Persamaan (3.8) variabel kecepatan linier robot v dan kecepatan anguler robot ω diubah menjadi kecepatan anguler roda kanan ω_r dan roda kiri ω_l.

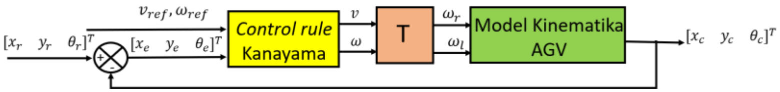
$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{1}{2}r & \frac{1}{2}r \\ \frac{r}{d} & -\frac{r}{d} \end{bmatrix} \begin{bmatrix} \omega_{right} \\ \omega_{left} \end{bmatrix} \quad \ldots(3.9)$$

Model kinematika *AGV* yang akan dibangun :

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix} = \begin{bmatrix}\frac{1}{2}rcos(\theta) & \frac{1}{2}rcos(\theta)\\ \frac{1}{2}rsin(\theta) & -\frac{1}{2}rsin(\theta)\\ \frac{r}{d} & -\frac{r}{d}\end{bmatrix}\begin{bmatrix}\omega_{right}\\ \omega_{left}\end{bmatrix} \quad \ldots(3.6)$$

3.3 Perancangan Sistem Kendali Otomatis menggunakan *Control Rule* Kanayama

Hal yang ingin dicapai pada perancangan sistem kendali otomatis menggunakan *control rule* Kanayama adalah nilai kesalahan posisi dan orientasi mendekati nol. Control *rule* Kanayama digunakan untuk menghasilkann kecepatan linier (v) dan anguler (ω) merupakan [20].



Gambar 3.7 Diagram blok sistem kendali otomatis AGV menggunakan control rule Kanayama

Gambar 3.7 merupakan diagram blok sistem *tracking generation* dengan menggunakkan *control rule* Kanayama dimana blok T berisi persamaan konversi dari kecepatan linier dan anguler robot menjadi kecepatan anguler roda kanan dan kiri seperti pada Persamaan (3.9) sedangkan nomenclaturenya berada pada Tabel 3.1, dimana posisi dan orientasi aktual (p_c). dibandingkan dengan posisi dan orientasi referensi (p_r).sehingga menghasilkan kesalahan posisi dan orientasi (p_e).

Tabel 3.1 Nomeclature dari *tracjectory generation* menggunakan *control rule* Kanayama

Notasi	Deskripsi	Satuan
p_r	Posisi dan orientasi (pose) referensi $[x_r \quad y_r \quad \theta_r]^T$	$[m,m,rad]^T$
p_c	Posisi dan orientasi (pose) aktual $[x_c \quad y_c \quad \theta_c]^T$	$[m,m,rad]^T$
p_e	Kesalahan posisi dan orientasi	$[m,m,rad]^T$
v_{ref}	Kecepatan linier referensi	m/s
ω_{ref}	Kecepatan anguler referensi	rad/s
ω_r	Kecepatan anguler roda kanan	rad/s

ω_l	Kecepatan anguler roda kiri	rad/s

Pergerakan robot menuju posisi dan orientasi referensi (x_r, y_r, θ_r) dibandingkan dengan posisi dan orientasi aktualnya (x_c, y_c, θ_c) dan menghasilkan kesalahan postur.

$$\begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} x_r - x_c \\ y_r - y_c \\ \theta_r - \theta_c \end{bmatrix} \quad \ldots(3.7)$$

Dimana e_x merupakan kesalahan pada sumbu x, e_y merupakan kesalahan pada sumbu y dan e_θ merupakan kesalahan orientasi. Kesalahan postur tidak diberikan pada global frame tetapi nilai kesalahan pada local frame dari robot. Transformasi dibutuhkan untuk merubah matrix kesalahan postur ke dalam koordinat robot, menggunakan matriks putar sebagai berikut :

$$\begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = R(\theta) \begin{bmatrix} x_r - x_c \\ y_r - y_c \\ \theta_r - \theta_c \end{bmatrix} \quad \ldots(3.8)$$

Dimana $R(\theta)$ merupakan matriks putar terhadap sumbu Z.

$$p_e = \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} cos(\theta_c) & sin(\theta_c) & 0 \\ -sin(\theta_c) & cos(\theta_c) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x_c \\ y_r - y_c \\ \theta_r - \theta_c \end{bmatrix} \quad \ldots(3.9)$$

Persamaan (3.8) dapat diselesaikan menjadi persamaan (3.9). Persamaan (3.9) merupakan kesalahan posisi dan orientasi.

$$\boldsymbol{p_e} = \begin{bmatrix} \boldsymbol{e_x} \\ \boldsymbol{e_y} \\ \boldsymbol{e_\theta} \end{bmatrix} = \begin{bmatrix} (\boldsymbol{x_r} - \boldsymbol{x_c})\boldsymbol{cos}(\boldsymbol{\theta_c}) + (\boldsymbol{y_r} - \boldsymbol{y_c})\boldsymbol{sin}(\boldsymbol{\theta_c}) \\ -(\boldsymbol{x_r} - \boldsymbol{x_c})\boldsymbol{sin}(\boldsymbol{\theta_c}) + (\boldsymbol{y_r} - \boldsymbol{y_c})\boldsymbol{cos}(\boldsymbol{\theta_c}) \\ \boldsymbol{\theta_r} - \boldsymbol{\theta_c} \end{bmatrix} \quad \ldots(3.10)$$

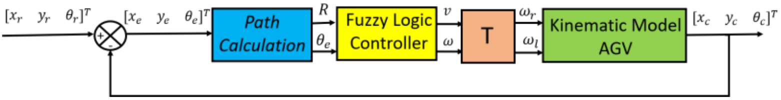
Blok pertama dari Gambar 3.7 berisi perencanaan lintasan. Di dalam blok ini berisi sebuah *control rule* Kanayama. Pada *control rule* Kanayama, sinyal masukan yang dibutuhkan adalah kecepatan referensi dan kesalahan *pose* (posisi dan orientasi) untuk menghasilkan kecepatan target yang dibutuhkan *AGV* dalam melakukan pergerakan. Hubungan antara kecepatan referensi dan kesalahan pose dapat diformulasikan pada persamaan (3.10).

$$q = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} v_r cos\theta_e + K_x x_e \\ \omega_r + v_r \left(K_y y_e + K_\theta sin(\theta_e)\right) \end{bmatrix} \quad \ldots(3.11)$$

Sinyal masukan sebuah *AGV* adalah kecepatan target yang terdiri dari v (kecepatan linier) dan ω (kecepatan anguler). Sinyal keluaran yang

dihasilkan *AGV* berupa kecepatan robot terhadap sumbu X dan Y serta kecepatan angulernya. Data posisi dan orientasi aktual *AGV* didapatkan dari hasil integral kecepatan linier dan anguler robot.

3.4 Perancangan Sistem Kendali Otomatis menggunakan Kontroler Logika *Fuzzy*



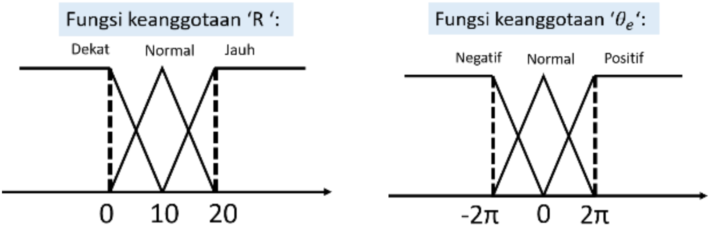
Gambar 3.8 Diagram blok sistem kendali otomatis pada AGV menggunakan kontroler logika *Fuzzy*

Gambar 3.8 merupakan diagram blok dari penelitian kali ini dimana blok T berisi persamaan konversi dari kecepatan linier dan anguler robot menjadi kecepatan anguler roda kanan dan kiri seperti pada Persamaan (3.9). Tabel 3.1 merupakan nomenclature dari variable yang terdapat pada Gambar 3.8. Posisi dan orientasi referensi dibandingkan dengan posisi dan orientasi aktual sehingga menghasilkan kesalahan posisi dan orientasi. Kesalahan posisi dan orientasi akan diproses untuk menghasilkan nilai R. $R = \sqrt{{x_e}^2 + {y_e}^2}$ dimana harga x_e dan y_e didapatkan dari Persamaan (3.9). Keluaran dari kontroler logika *fuzzy* Mamdani adalah kecepatan linier dan anguler robot. Hubungan antara kecepatan linier dan anguler robot dengan kecepatan anguler roda sebelah kanan dan kiri didapatkan dari Persamaan (3.2-3.5)

Tabel 3.2 Nomeclature dari kontroler logika *fuzzy* Mamdani

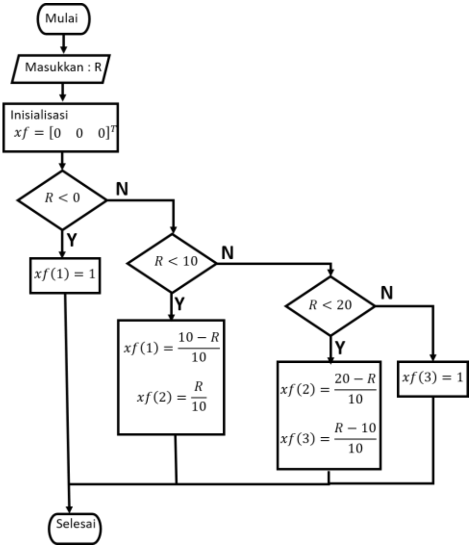
Notasi	Deskripsi	Satuan
p_r	Posisi dan orientasi (pose) referensi $[x_r \;\; y_r \;\; \theta_r]^T$	$[m,m,rad]^T$
p_c	Posisi dan orientasi (pose) aktual $[x_c \;\; y_c \;\; \theta_c]^T$	$[m,m,rad]^T$
p_e	Kesalahan posisi dan orientasi	$[m,m,rad]^T$
R	Kesalahan posisi	m
θ_e	Kesalahan orientasi	Radian
ω_r	Kecepatan anguler roda kanan	rad/s
ω_l	Kecepatan anguler roda kiri	rad/s

- Proses fuzzifikasi dengan tiga fungsi keanggotaan



Gambar 3.9 Fungsi keanggotaan 'R' dan 'θe'

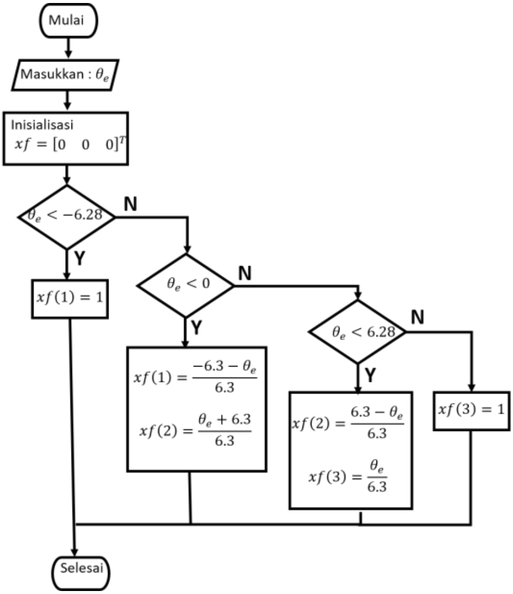
Fungsi keanggotaan untuk variable R dan θe digambarkan ke dalam bentuk segitiga dan trapesium seperti pada Gambar 3.9 dan memiliki tiga fungsi keanggotaan. Himpunan 'R' memiliki fungsi keanggotaan 'Dekat','Normal','Jauh', apabila terdapat kesalahan posisi kurang dari 0 cm maka masuk ke dalam kelas 'Dekat' atau jika kesalahan posisi berada pada jangkauan 0-10 cm maka R dapat masuk ke dalam dua kelas yakni 'Dekat' dan 'Normal' namun dengan tingkat keanggotaan yang berbeda. Begitu halnya apabila R berada pada antara 10–20cm maka R dapat masuk ke dalam dua kelas yakni 'Normal' dan 'Jauh' dengan tingkat keanggotaan yang berbeda. Apabila kesalahan posisi lebih dari 20 cm maka himpunan 'θe' memiliki fungsi keanggotaan 'Jauh'. Pada proses fuzzifikasi untuk kesalahan posisi dapat ditinjau pada Gambar 3.10. Dimana xf merupakan matriks berordo 3x1 yang berisi hasil fuzzifikasi, karena jumlah fungsi keanggotaan untuk kesalahan posisi sebanyak tiga dan memiliki rentang nilai seperti pada Gambar 3.2 sehingga perlu digunakannnya fungsi percabangan.



Gambar 3.10 Diagram alir proses fuzzifikasi untuk kesalahan posisi (R)

Kesalahan orientasi (θe) pada penelitian ini dibatasi sejauh -2π sampai dengan 2π (rad), apabila θe kurang dari -2π maka masuk ke dalam kelas ‘Negatif’. Jika kesalahan oritentasi berada pada *range* -2π sampai 0 radian maka θe akan masuk ke dalam kelas ‘Negatif’ dan ‘Normal’ dengan tingkat keanggotaan yang berbeda pula. Jika kesalahan oritentasi berada pada *range* 0 sampai 2π radian maka θe akan masuk ke dalam kelas ‘Normal’ dan ‘Positif’ dengan tingkat keanggotaan yang berbeda pula.

Pada proses fuzzifikasi untuk kesalahan posisi dapat ditinjau pada Gambar 3.11. Dimana xf merupakan matriks berordo 3x1 yang berisi hasil fuzzifikasi, karena jumlah fungsi keanggotaan untuk kesalahan posisi sebanyak tiga dan memiliki rentang nilai seperti pada Gambar 3.2 sehingga perlu digunakannnya fungsi percabangan.



Gambar 3.11 Diagram alir proses fuzzifikasi untuk kesalahan orientasi (θe)

- *Fuzzy* Inference *Rule* (FIS)

  Knowledge base dari KLF terdiri dari dua komponen yang disebut *data base* dan *rule base* [15]. Data base berisi definisi perlu yang digunakan untuk mendeskripsikan *linguistic control rules* seperti contohnya pada Tabel 3.1, himpunan $r = \{\text{'Dekat'}, \text{'Normal'}, \text{'Jauh'}\}$ dan himpunan sinyal keluaran v={'Dekat','Normal','Jauh'}. *Rule* based yang dikembangkan ini memiliki tujuan untuk menghasilkan kesalahan posisi mendekati 0 dengan melakukan mengendalikan kecepatan linier robot untuk bergerak menuju posisi yang diinginkan (referensi).

```
if kesalahan posisi dekat then kecepatan linier berhenti
if kesalahan posisi normal then kecepatan linier lambat
if kesalahan posisi jauh then kecepatan linier cepat
```

  Pada Tabel 3.2, himpunan e_θ={'Negatif','Sekitar nol','Positif'} dan himpunan sinyal keluaran $\omega = \{\text{'Positif'}, \text{'Off'}, \text{'Negatif'}\}$.

Tabel 3.3 *Rule* based untuk kecepatan linier

r	Dekat	Normal	Jauh
v	Off	Lambat	Cepat

Proses pembuatan *inference rule* diawali dengan inisiali *rule base* terlebih dahulu seperti pada Gambar 3.3 dan 3.4. Dimana *data base* untuk kesalahan posisi terdiri dari 'Dekat','Normal', dan 'Jauh' seperti pada Tabel 1 direpresentasikan menggunakan angka '1','2', dan '3'.

Tabel 3.4 *Rule* based untuk kecepatan anguler

e_θ	Negatif	Sekitar nol	Positif
w	Positif	Diam	Negatif

```
if kesalahan orientasi negatif then kecepatan anguler
bernilai positif
if kesalahan orientasi sekitar nol then kecepatan anguler
bernilai 0
if kesalahan orientasi positif then kecepatan anguler
bernilai negatif
```

Diagram alir dari proses *fuzzy inference system* dengan sinyal masukan berupa kesalahan posisi. Dimana jumlah *rule* base yang digunakan sebanyak tiga buah. Hal ini berdasarkan Tabel 1. Terdapat beberapa jenis metode yang dapat digunakan pada penyelesaian *fuzzy* inference, namun yang paling banyak digunakan adalah metode MAX-MIN dan MAX-DOT. Pada penelitian ini, digunakan metode MAX-MIN (Mamdani). Gambar 3.3 merupakan diagram alir untuk sinyal masukan berupa kesalahan posisi robot. Robot bergerak sesuai dengan kondisinya.

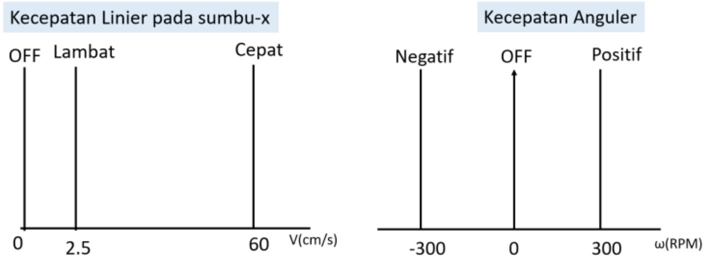
- Defuzzifikasi

  Metode defuzzifikasi yang digunakan pada penelitian ini adalah *Center of Area* (COA).

$$W = \frac{\sum_{i=1}^{n} \alpha_i M_i}{\sum_{i=1}^{n} \alpha_i A_i}$$

Dimana i merupakan urutan *rule*, n merupakan jumlah dari *fuzzy rule*, M merupakan keluaran (*consequent*) yang dihasilkan, α merupakan derajat

keanggotaan dari *rule* ke-*i* dan *A* merupakan luas dari keluaran tersebut. Fungsi untuk sinyal keluaran yang digunakanadalah *singleton* seperti pada Gambar 3.12 kecepatan linier yang dapat digunakan oleh *AGV* dalam pergerakannya adalah 0 cm/s, 2.5cm/s, dan 60 cm/s sedangkan batas kecepatan anguler yang dapat digunakan ada pada nilai -300 RPM, 0 RPM, dan 300 RPM.



Gambar 3.12 Sinyal k*control* kecepatan linier (v) dan anguler (ω)

Halaman ini sengaja dikosongkan

BAB IV
PENGUJIAN DAN ANALISIS

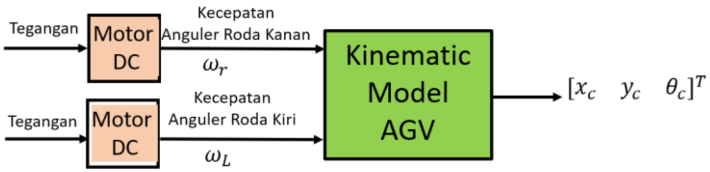
Pada bab ini membahas mengenai pengujian sistem dengan menggunakan *control rule* Kanayama dan kontroler logika *fuzzy*. Namun, sebelumnya pengujian Perancangan sistem kendali otomatisKanayama dilakukan dengan memberikan variasi nilai pada parameter $\boldsymbol{K_x}$, $\boldsymbol{K_y}$, dan $\boldsymbol{K_\theta}$ sedangkan pengujian kontroler logika *fuzzy* dilakukan dengan cara memberikan variasi jumlah fungsi keanggotaan dan variasi posisi awal robot yang berubah-ubah.

4.1 Pengujian Sistem secara *Open Loop*

Pengujian sistem secara *open loop* bertujuan untuk melakukan validasi model kinematika yang telah dirancang pada Persamaan (3.6). Pengujian model kinematika dilakukan dengan melakukan lima kali percobaan. Proses ini didapatkan dengan membandingkan antara posisi robot pada sumbu X melalui simulasi dan melalui pengukuran aktual dan ditinjau indeks performansinya menggunakan RMSE (seperti pada Persamaan 2.36). Dimana pengujian ini, pergerakan dilakukan dengan gerakan lintasan lurus yang artinya posisi robot terhadap sumbu Y sama dengan nol. Gambar 4.1 merupakan diagram blok dari pengujian sistem secara *open loop,* dimana masukan yang dibutuhkan motor DC adalah tegangan. Motor DC merupakan peralatan yang digunakan untuk merubah energi listrik menjadi mekanik. Dimana energi mekanik tersebut merupakan kecepatan anguler (ω) yang dihitung dalam radians/second (rad/s), *revolutions/second* (rps), atau *revolutions/minute* (rpm) [21]. Model matematis motor DC yang digunakan sebagai penggerak roda depan AGV seperti pada Persamaan (4.1) yang didapatkan dari proses identifikasi, simulasi dapat ditinjau pada Lampiran C.

$$G_m(s) = \frac{0.7121}{1.0481s + 1} \qquad \ldots(4.1)$$

Pengujian pada AGV dilakukan dengan memberikan variasi tegangan pada motor DC yang berfungsi sebagai roda penggerak. Pergerakan robot dibatasi dalam waktu 2 sekon, selama kurun waktu tersebut, perpindahan yang dialami AGV merepresentasikan posisi aktual terhadap sumbu X dari robot.



Gambar 4.1 Diagram blok pengujian sistem secara *open loop*

Percobaan 1

Pada percobaan pertama, diberikan variasi tegangan dari 3 sampai dengan 5 volt. Semakin besar tegangan yang diberikan kepada motor menyebabkan kecepatan anguler pada motor juga semakin besar sehingga menyebabkan pergerakan robot menjadi lebih jauh seperti pada Tabel 4.1.

Tabel 4.1 Pengujian Model Kinematika AGV (Percobaan 1)

Tegangan motor kanan	Tegangan motor kiri	Posisi pada sumbu-x (simulasi) (cm)	Posisi pada sumbu-x (real) (cm)
3	3	5.47	6.3
3.5	3.5	6.38	6.7
4	4	11.45	10.2
4.5	4.5	12.9	12.6
5	5	14.32	15

Pengujian ini bertujuan untuk melakukan validasi pada model kinematika terhadap *real plant AGV*, sehingga perlu dihitung indeks performansinya. Pada percobaan pertama ini, nilai RMSE adalah sebagai berikut :

$$RMSE = \sqrt{\frac{(5.47-6.3)^2+(6.38-6.7)^2+(11.45-10.2)^2+(12.9-12.6)^2+(14.32-15)^2}{5}}$$

$$RMSE = 0.7623$$

Percobaan 2

Pada percobaan pertama, diberikan variasi tegangan dari 3 sampai dengan 5 volt. Semakin besar tegangan yang diberikan kepada motor menyebabkan kecepatan anguler pada motor juga semakin besar sehingga menyebabkan pergerakan robot menjadi lebih jauh seperti pada Tabel 4.2.

Tabel 4.2 Pengujian Model Kinematika AGV (Percobaan 2)

Tegangan motor kanan	**Tegangan motor kiri**	**Posisi pada sumbu-x (simulasi) (cm)**	**Posisi pada sumbu-x (real) (cm)**
3	3	5.47	6.4
3.5	3.5	6.38	6.7
4	4	11.45	9.6
4.5	4.5	12.9	13.1
5	5	14.32	14.9

Pengujian ini bertujuan untuk melakukan validasi pada model kinematika terhadap *real plant AGV*, sehingga perlu dihitung indeks performansinya. Pada percobaan kedua ini, nilai RMSE adalah sebagai berikut :

$$RMSE = \sqrt{\frac{(5.47-6.4)^2+(6.38-6.7)^2+(11.45-9.6)^2+(12.9-13.1)^2+(14.32-14.9)^2}{5}}$$

$$RMSE = 0.97634$$

Percobaan 3

Pada percobaan pertama, diberikan variasi tegangan dari 3 sampai dengan 5 volt. Semakin besar tegangan yang diberikan kepada motor menyebabkan kecepatan anguler pada motor juga semakin besar sehingga menyebabkan pergerakan robot menjadi lebih jauh seperti pada Tabel 4.3.

Tabel 4.3 Pengujian Model Kinematika AGV (Percobaan 3)

Tegangan motor kanan	**Tegangan motor kiri**	**Posisi pada sumbu-x (simulasi) (cm)**	**Posisi pada sumbu-x (real) (cm)**
3	3	5.47	6.7
3.5	3.5	6.38	7.2
4	4	11.45	12.2
4.5	4.5	12.9	12.9
5	5	14.32	14.25

Pengujian ini bertujuan untuk melakukan validasi pada model kinematika terhadap *real plant AGV*, sehingga perlu dihitung indeks performansinya. Pada percobaan ketiga ini, nilai RMSE adalah sebagai berikut :

$$RMSE = \sqrt{\frac{(5.47-6.7)^2+(6.38-7.2)^2+(11.45-12.2)^2+(12.9-12.9)^2+(14.32-14.25)^2}{5}}$$

$$RMSE = 0.741984$$

Percobaan 4

Pada percobaan pertama, diberikan variasi tegangan dari 3 sampai dengan 5 volt. Semakin besar tegangan yang diberikan kepada motor menyebabkan kecepatan anguler pada motor juga semakin besar sehingga menyebabkan pergerakan robot menjadi lebih jauh seperti pada Tabel 4.4.

Tabel 4.4 Pengujian Model Kinematika AGV (Percobaan 4)

Tegangan motor kanan	Tegangan motor kiri	Posisi pada sumbu-x (simulasi) (cm)	Posisi pada sumbu-x (real) (cm)
3	3	5.47	6.8
3.5	3.5	6.38	7.3
4	4	11.45	12.4
4.5	4.5	12.9	13
5	5	14.32	15.5

Pengujian ini bertujuan untuk melakukan validasi pada model kinematika terhadap *real plant AGV*, sehingga perlu dihitung indeks performansinya. Pada percobaan keempat ini, nilai RMSE adalah sebagai berikut :

$$RMSE = \sqrt{\frac{(5.47-6.8)^2+(6.38-7.3)^2+(11.45-12.4)^2+(12.9-13)^2+(14.32-15.5)^2}{5}}$$

$$RMSE = 0.9919$$

Percobaan 5

Pada percobaan pertama, diberikan variasi tegangan dari 3 sampai dengan 5 volt. Semakin besar tegangan yang diberikan kepada motor menyebabkan kecepatan anguler pada motor juga semakin besar sehingga menyebabkan pergerakan robot menjadi lebih jauh seperti pada Tabel 4.5.

Tabel 4.5 Pengujian Model Kinematika AGV (Percobaan 5)

Tegangan motor kanan	Tegangan motor kiri	Posisi pada sumbu-x (simulasi) (cm)	Posisi pada sumbu-x (real) (cm)
3	3	5.47	7.1
3.5	3.5	6.38	7.8
4	4	11.45	12.3

4.5	4.5	12.9	12.9
5	5	14.32	14.2

Pengujian ini bertujuan untuk melakukan validasi pada model kinematika terhadap *real plant AGV,* sehingga perlu dihitung indeks performansinya. Pada percobaan kelima ini, nilai RMSE adalah sebagai berikut :

$$RMSE = \sqrt{\frac{(5.47 - 7.1)^2 + (6.38 - 7.8)^2 + (11.45 - 12.3)^2 + (12.9 - 12.9)^2 + (14.32 - 14.2)^2}{5}}$$

$$RMSE = 1.0402$$

4.2 Pengujian *control rule* Kanayama

Control rule yang diusulkan oleh Kanayama mengandung beberapa variabel seperti kecepatan linier dan anguler referensi, kesalahan posisi dan orientasi, serta tiga konstanta K_x, K_y, dan K_θ. Dimana K_x, K_y, dan K_θ merupakan konstanta bernilai positif. Pengujian Perancangan sistem kendali otomatismeliputi pengujian variasi parameter K_x, K_y, dan K_θ.

4.2.1 Pengujian parameter *control rule* Kanayama dengan masukan sinyal unit-step (K_x=100, K_y=16, dan K_θ=2)

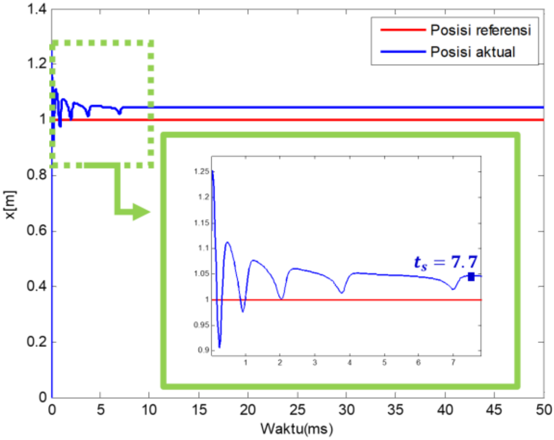
Pada pengujian ini dilakukan untuk mengetahui pengaruh parameter K_x, K_y, dan K_θ pada respon pergerakan robot pada sumbu X, Y, dan orientasinya. K_x merupakan konstanta waktu sehingga dipilih K_x=100 bertujuan untuk mempercepat pergerakan robot menuju posisi dan orientasi referensi. Berdasarkan Persamaan (2.34) dan (2.35) nilai parameter K_y dan K_θ dapat diperoleh dari nilai ζ serta ω_n. Pengujian kali ini menggunakan nilai $\zeta = 0.25$ dan $\omega_n = 20$. Maka nilai parameter K_y dan K_θ menjadi Persamaan (4.1) dan (4.2).

$$K_y = \frac{{\omega_n}^2}{{v_r}^2} = \frac{400}{25} = 16 \qquad \ldots(4.1)$$

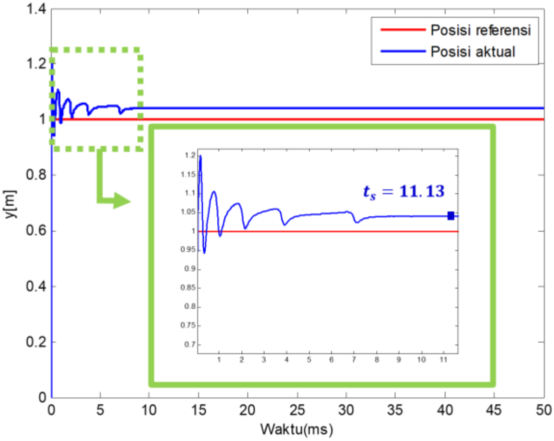
$$K_\theta = 2\,\zeta\sqrt{K_y} = 2(0.25)(4) = 2 \qquad \ldots(4.2)$$

Berdasarkan hasil pengujian dengan K_y dan K_θ seperti pada Persamaan (4.1) dan (4.2) ketika postur referensi yang harus dicapai robot adalah $p_r = [x_r \quad y_r \quad \theta_r]^T = [1m \quad 1m \quad 1rad]^T$, maka posisi aktual robot pada sumbu-X seperti pada Gambar 4.2 sumbu-Y seperti pada Gambar

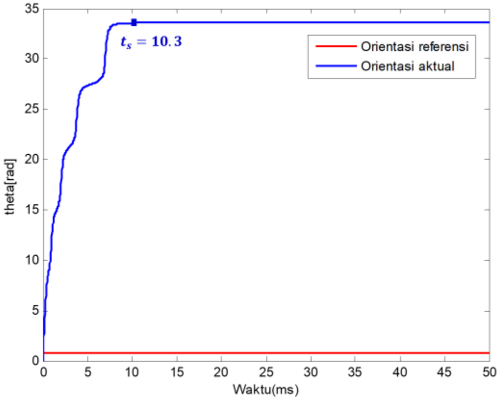
4.3, dan orientasinya pada Gambar 4.4. Hal ini dikarenakan, pada pengujian ini nilai $\zeta = 0.25$. Pada sistem orde kedua, ketika nilai $0 < \zeta < 1$ maka sistem tersebut masuk ke dalam kasus *underdamped*.



Gambar 4.2 Respon posisi robot pada sumbu X dengan ζ=0.25



Gambar 4.3 Respon posisi robot pada sumbu Y dengan ζ=0.25



Gambar 4.4 Respon orientasi robot dengan ζ=0.25

4.2.2 Pengujian parameter *control rule* Kanayama dengan masukan sinyal unit-step (K_x=100, K_y=16, dan K_θ=8)

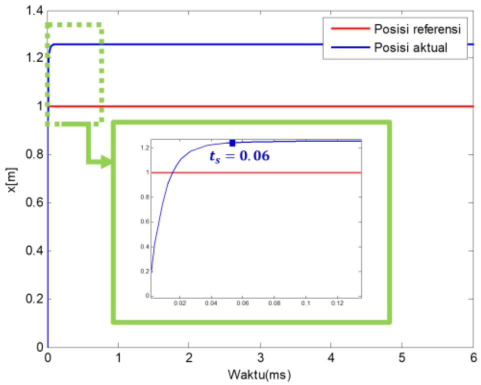
Pada pengujian ini dilakukan untuk mengetahui pengaruh parameter K_x, K_y, dan K_θ pada respon pergerakan robot pada sumbu X, Y, dan orientasinya. K_x merupakan konstanta waktu sehingga dipilih K_x=100 bertujuan untuk mempercepat pergerakan robot menuju posisi dan orientasi referensi. Berdasarkan Persamaan (2.34) dan (2.35) nilai parameter K_y dan K_θ dapat diperoleh dari nilai ζ serta ω_n. Pengujian kali ini menggunakan nilai $\zeta = 1$ dan $\omega_n = 20$. Maka nilai parameter K_y dan K_θ menjadi Persamaan (4.1) dan (4.2).

$$K_y = \frac{{\omega_n}^2}{{v_r}^2} = \frac{400}{25} = 16 \qquad \text{…(4.3)}$$

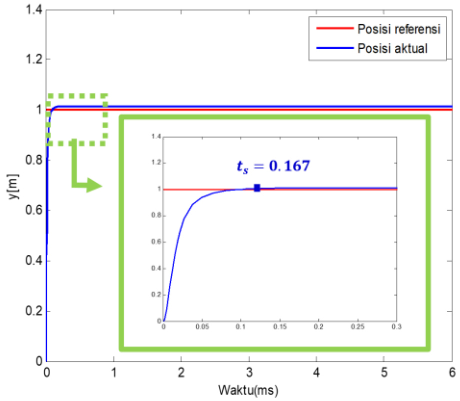
$$K_\theta = 2\,\zeta\sqrt{K_y} = 2(1)(4) = 8 \qquad \text{…(4.4)}$$

Berdasarkan hasil pengujian dengan K_y dan K_θ seperti pada Persamaan (4.3) dan (4.4) ketika postur referensi yang harus dicapai robot adalah $p_r = [x_r \quad y_r \quad \theta_r]^T = [1m \quad 1m \quad 1rad]^T$, maka posisi aktual robot pada sumbu-X seperti pada Gambar 4.5 sumbu-Y seperti pada Gambar 4.6, dan orientasinya pada Gambar 4.7. Hal ini dikarenakan, pada

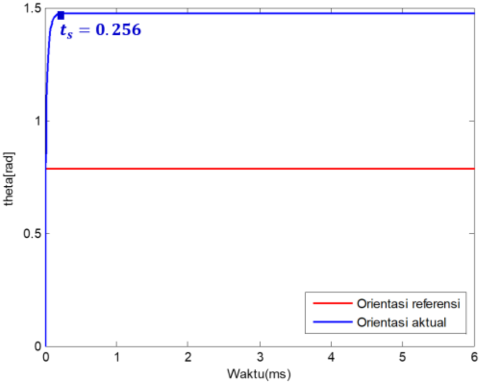
pengujian ini nilai $\zeta = 1$. Pada sistem orde kedua, ketika nilai $\zeta = 1$ maka sistem tersebut masuk ke dalam kasus *critically damped*.



Gambar 4.5 Respon posisi robot pada sumbu X dengan ζ=1



Gambar 4.6 Respon posisi robot pada sumbu Y dengan ζ=1



Gambar 4.7 Respon orientasi robot dengan ζ=1

4.2.3 Pengujian parameter *control rule* Kanayama dengan masukan sinyal unit-step (K_x=100, K_y=16, dan K_θ=16)

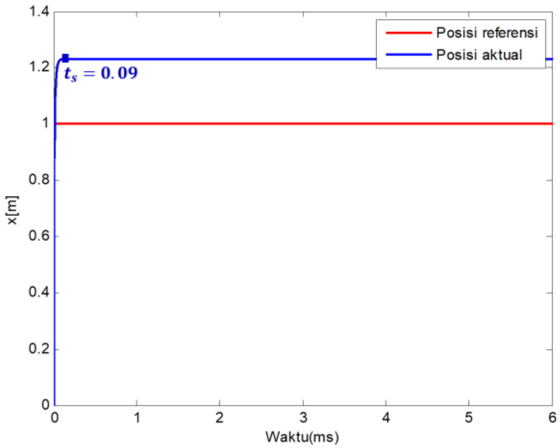
Pada pengujian ini dilakukan untuk mengetahui pengaruh parameter K_x, K_y, dan K_θ pada respon pergerakan robot pada sumbu X, Y, dan orientasinya. K_x merupakan konstanta waktu sehingga dipilih K_x=100 bertujuan untuk mempercepat pergerakan robot menuju posisi dan orientasi referensi. Berdasarkan Persamaan (2.34) dan (2.35) nilai parameter K_y dan K_θ dapat diperoleh dari nilai ζ serta ω_n. Pengujian kali ini menggunakan nilai $\zeta = 2$ dan $\omega_n = 20$. Maka nilai parameter K_y dan K_θ menjadi Persamaan (4.5) dan (4.6).

$$K_y = \frac{{\omega_n}^2}{{v_r}^2} = \frac{400}{25} = 16 \qquad \text{...(4.5)}$$

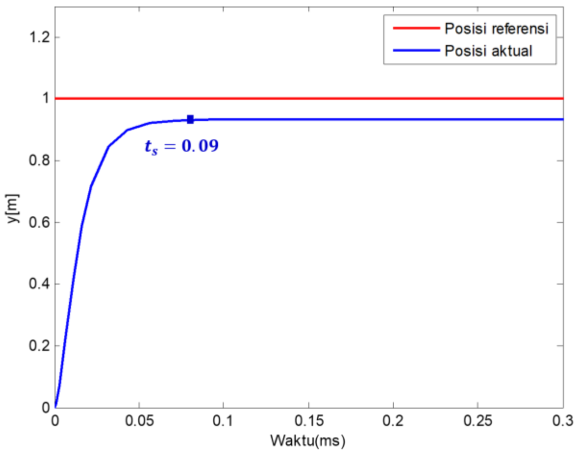
$$K_\theta = 2\,\zeta\sqrt{K_y} = 2(2)(4) = 16 \qquad \text{...(4.6)}$$

Berdasarkan hasil pengujian dengan K_y dan K_θ seperti pada Persamaan (4.5) dan (4.6) ketika postur referensi yang harus dicapai robot adalah $p_r = [x_r \quad y_r \quad \theta_r]^T = [1m \quad 1m \quad 1rad]^T$, maka posisi aktual robot pada sumbu-X seperti pada Gambar 4.8 sumbu-Y seperti pada Gambar 4.9, dan orientasinya pada Gambar 4.10. Hal ini dikarenakan,pada pengujian ini nilai $\zeta = 2$. Pada sistem orde kedua, ketika nilai $\zeta > 1$ maka sistem tersebut masuk ke dalam kasus *overdamped*. Pada sistem orde kedua yang bersifat *overdamped*, respon yang dihasilkan apabila diberi

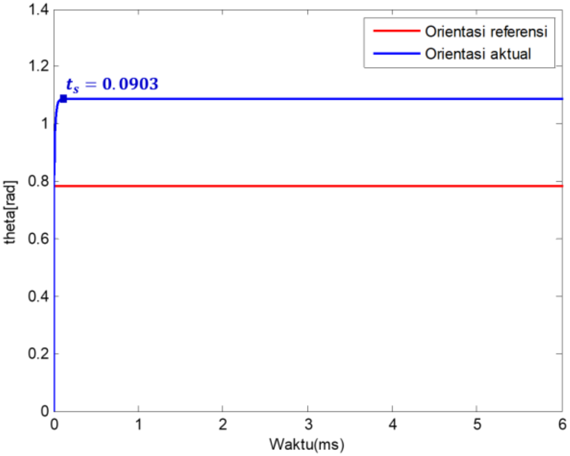
masukan berupa sinyal *unit-step* adalah tidak adanya osilasi seperti pada Gambar 4.8, 4.9, dan 4.10.



Gambar 4.8 Respon posisi robot pada sumbu-X dengan ζ=2

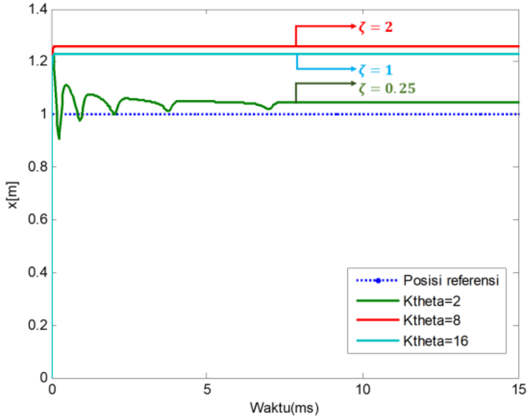


Gambar 4.9 Respon posisi robot pada sumbu-Y dengan ζ=2

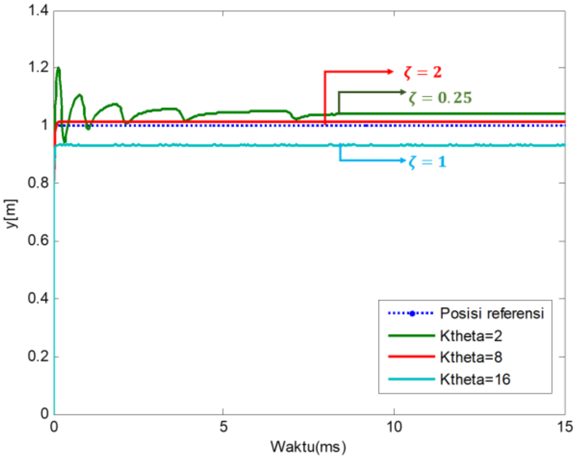


Gambar 4.10 Respon orientasi robot dengan ζ=2

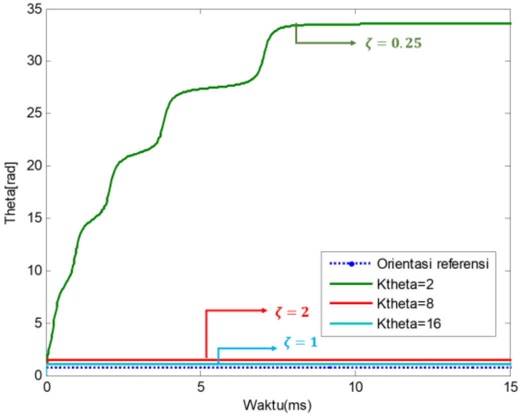
Pengujian berikutnya dilakukan untuk mengetahui nilai parameter K_x, K_y, dan K_θ mana yang paling mendekati postur referensi. Pada pengujian kali ini, nilai frekuensi alamiah tak teredam tidak diubah karena tujuan dari Perancangan sistem kendali otomatisadalah robot dapat bergerak sesuai lintasan yang diinginkan dengan lancar. Pengaruh parameter ω_n merupakan frekuensi yang menghasilkan osilasi ketika redaman menurun menuju nol [14] sehingga menyebabkan pergerakan robot menuju postur yang diinginkan menjadi lebih lama. Hasil pengujian pada Gambar 4.10 sampai dengan Gambar 4.13 merupakan respon posisi dan orientasi robot terhadap posisi dan orientasi referensi.



Gambar 4.11 Respon posisi robot pada sumbu X dengan Variasi ζ



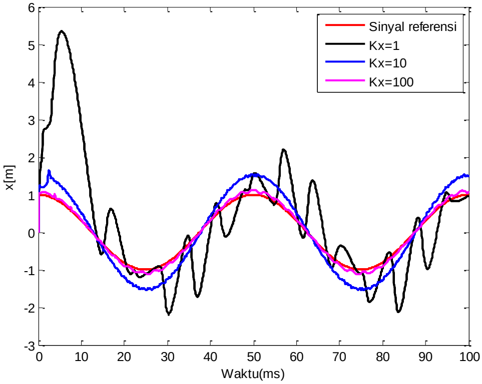
Gambar 4.12 Respon posisi robot pada sumbu Y dengan Variasi ζ



Gambar 4.13 Respon orientasi robot dengan Variasi ζ

4.2.4 Pengujian parameter *control rule* Kanayama dengan Referensi Lintasan Lingkaran (Variasi Parameter K_x)

Pada pengujian parameter K_x ini dilakukan tiga kali percobaan dengan tigakalia nilai K_x yang berbeda-beda namun nilai parameter lainnya (K_y dan K_θ bernilai tetap. Pengujian ini dilakukan untuk mengetahui pengaruh parameter K_x pada lintasan .



Gambar 4.14 Respon posisi robot pada sumbu X dengan Variasi Parameter K_x

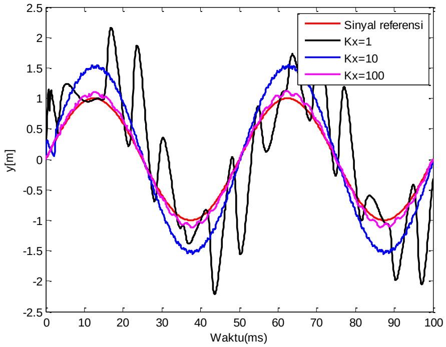
Gambar 4.14 merupakan grafik respon berupa posisi *AGV* pada sumbu X dengan nilai parameter K_x yang berbeda.Pada pengujian ini, sinyal referensi yang digunakan adalah sinusoidal. Grafik berwarna hitam merupakan respon posisi pada sumbu-X dengan nilai K_x sama dengan 1. Grafik berwarna biru merupakan respon posisi pada sumbu-X dengan nilai K_x sama dengan 10. Grafik berwarna magenta merupakan respon posisi pada sumbu-X dengan nilai K_x sama dengan 100 sedangkan sinyal referensi adalah yang berwarna merah. Sinyal masukan yang diberikan adalah sinyal cosinus. Seperti pada Persamaan (2.41) dan (2.42), nilai parameter $\frac{1}{K_x}$ merepresentasikan konstanta waktu, sehingga semakin besar nilai parameter K_x maka respon pergerakan robot akan semakin cepat menuju posisi referensi. Indeks performansi dari percobaan 1 ini dapat dilihat pada Tabel 4.6. RMSE, ISE, dan IAE merupakan metode yang digunakan untuk memvalidasi data hasil pengujian. Berdasarkan hasil gambar dan indeks performansi sistem, dapat diketahui jika semakin besar nilai K_x maka respon sistem yang dihasilkan semakin mendekati bentuk aslinya.

Tabel 4.6 Indeks performansi sistem dengan variasi parameter konstanta K_x (pada sumbu X)

Parameter K_x	**RMSE**	**ISE**	**IAE**
1	1.2455	1.6707e+003	837.8686
10	0.3707	179.0462	433.3028
100	0.0775	28.5297	284.3705

Gambar 4.15 merupakan grafik respon berupa posisi *AGV* pada sumbu Y dengan nilai parameter K_y yang berbeda.Pada pengujian ini, sinyal referensi yang digunakan adalah sinusoidal. Grafik berwarna hitam merupakan respon posisi pada sumbu-Y dengan nilai K_x sama dengan 1. Grafik berwarna biru merupakan respon posisi pada sumbu-X dengan nilai K_x sama dengan 10. Grafik berwarna magenta merupakan respon posisi pada sumbu-Y dengan nilai K_x sama dengan 100 sedangkan sinyal referensi adalah yang berwarna merah. Sinyal masukan yang diberikan adalah sinyal sinus. Sinyal masukan yang diberikan adalah sinyal cosinus.

Seperti pada Persamaan (2.41) dan (2.42), nilai parameter $\frac{1}{K_x}$ merepresentasikan konstanta waktu, sehingga semakin besar nilai parameter K_x maka respon pergerakan robot akan semakin cepat menuju posisi referensi. Indeks performansi dari percobaan 1 ini dapat dilihat pada Tabel 4.7.



Gambar 4.15 Respon posisi robot pada sumbu Y dengan Variasi Parameter K_x

Pada Tabel 4.7 RMSE, ISE, dan IAE merupakan metode yang digunakan untuk memvalidasi data hasil pengujian. Berdasarkan hasil gambar dan indeks performansi sistem, dapat diketahui jika semakin besar nilai K_x maka respon sistem yang dihasilkan semakin mendekati bentuk aslinya.

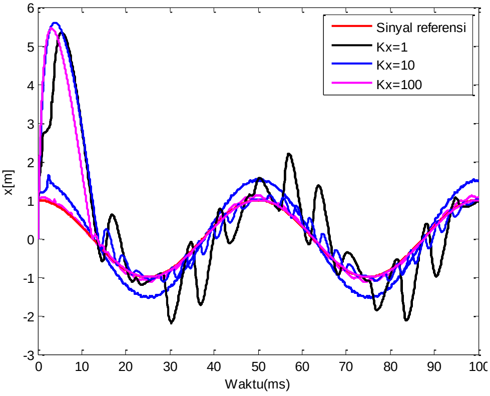
Tabel 4.7 Indeks performansi sistem dengan variasi parameter konstanta K_x (pada sumbuY)

Parameter K_x	**RMSE**	**ISE**	**IAE**
1	0.6888	511.1053	546.8081
10	0.3642	172.6049	427.0043
100	0.0674	21.5788	276.3455

4.2.5 Pengujian parameter *control rule* Kanayama dengan Referensi Lintasan Lingkaran (Variasi Parameter K_y)

Pada pengujian parameter K_y ini dilakukan tiga kali percobaan dengan tiga nilai K_y yang berbeda-beda namun nilai parameter lainnya (K_x dan K_θ) tetap. Pengujian ini dilakukan untuk mengetahui pengaruh parameter

K_y pada perencanaan lintasan. Gambar 4.16 merupakan grafik respon berupa posisi *AGV* pada sumbu-X dengan nilai parameter K_y yang berbeda.Pada pengujian ini, sinyal referensi yang digunakan adalah sinusoidal. Grafik berwarna merah merupakan respon posisi pada sumbu-X dengan nilai K_y sama dengan 0,1. Grafik berwarna hitam merupakan respon posisi pada sumbu-X dengan nilai K_y sama dengan 10. Grafik berwarna biru merupakan respon posisi pada sumbu-X dengan nilai K_y sama dengan 100. Sinyal masukan yang diberikan adalah sinyal cosinus. Seperti pada Persamaan (2.41), nilai parameter K_y memiliki hubungan dengan frekuensi alamiah tak teredam serta kecepatan referensi, sehingga semakin besar nilai parameter K_y maka respon pergerakan robot akan semakin cepat menuju posisi referensi karena nilai ω_n yang dihasilkan semakin besar yang artinya frekuensi yang dibutuhkan sistem untuk mengalami osilasi semakin besar pula. Indeks performansi dari percobaan 1 ini dapat dilihat pada Tabel 4.8.



Gambar 4.16 Respon posisi robot pada sumbu X dengan Variasi Parameter K_y

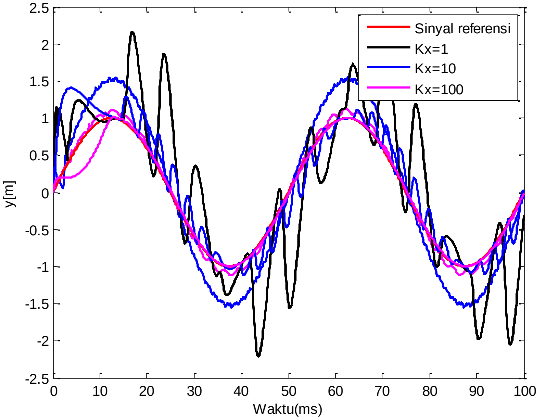
Gambar 4.17 merupakan grafik respon berupa posisi *AGV* pada sumbu Y dengan nilai parameter K_y yang berbeda. Pada pengujian ini, sinyal referensi yang digunakan adalah sinusoidal. Grafik berwarna hitam merupakan respon posisi pada sumbu-Y dengan nilai K_y sama dengan 1.

Grafik berwarna biru merupakan respon posisi pada sumbu-Y dengan nilai K_y sama dengan 10. Grafik berwarna magenta merupakan respon posisi pada sumbu-Y dengan nilai K_y sama dengan 100 sedangkan sinyal referensi adalah yang berwarna merah. Sinyal masukan yang diberikan adalah sinyal cosinus. Seperti pada Persamaan (2.41), nilai parameter K_y memiliki hubungan dengan frekuensi alamiah tak teredam serta kecepatan referensi, sehingga semakin besar nilai parameter K_y maka respon pergerakan robot akan semakin cepat menuju posisi referensi karena nilai ω_n yang dihasilkan semakin besar yang artinya frekuensi yang dibutuhkan sistem untuk mengalami osilasi semakin besar pula. Indeks performansi dari percobaan 1 ini dapat dilihat pada Tabel 4.8. Berdasarkan hasil gambar dan indeks performansi sistem, dapat diketahui jika semakin besar nilai K_y maka respon sistem yang dihasilkan semakin mdendekati bentuk aslinya.

Tabel 4.8 Indeks performansi sistem dengan variasi parameter konstanta $\boldsymbol{K_y}$ (pada sumbu X)

Parameter $\boldsymbol{K_y}$	**RMSE**	**ISE**	**IAE**
1	1.2455	1.6707e+003	837.8686
10	2.6044	1.0900e+004	2.8308e+003
100	3.3956	7.1879e+004	1.8764e+004

Pada Tabel 4.8 RMSE, ISE, dan IAE merupakan metode yang digunakan untuk memvalidasi data hasil pengujian. Berdasarkan hasil gambar dan indeks performansi sistem, dapat diketahui jika semakin besar nilai K_y maka respon sistem yang dihasilkan semakin mendekati bentuk aslinya.



Gambar 4.17 Respon posisi robot pada sumbu Y dengan Variasi Parameter K_y

Pada Tabel 4.9 RMSE, ISE, dan IAE merupakan hasil perhitungan indeks performansi yang digunakan untuk memvalidasi data hasil pengujian. Berdasarkan hasil gambar dan indeks performansi sistem, dapat diketahui jika semakin besar nilai K_y maka respon sistem yang dihasilkan semakin mendekati bentuk aslinya.

Tabel 4.9 Indeks performansi sistem dengan variasi parameter konstanta K_y (pada sumbu Y)

Parameter K_y	**RMSE**	**ISE**	**IAE**
1	0.6888	511.1053	546.8081
10	0.4879	382.4693	575.3492
100	0.2767	477.4526	1.4234e+003

4.2.6 Pengujian parameter *control rule* Kanayama dengan Referensi Lintasan Lingkaran (Variasi Parameter K_θ)

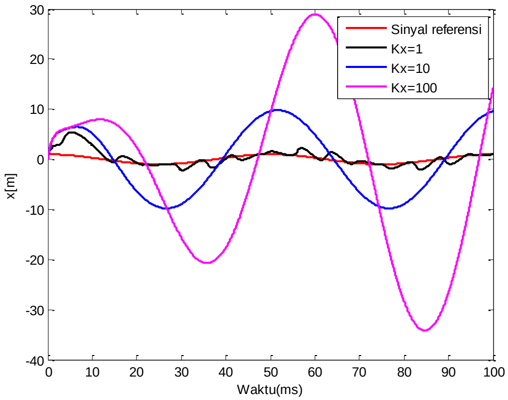
Pada pengujian parameter K_θ ini dilakukan tiga kali percobaan dengan tiga nilai K_θ yang berbeda-beda namun nilai parameter lainnya (K_x dan K_y) tetap. Pengujian ini dilakukan untuk mengetahui pengaruh parameter K_θ pada perencanaan lintasan. Gambar 4.18 merupakan grafik respon berupa posisi *AGV* pada sumbu X dengan nilai parameter K_θ yang berbeda. Pada pengujian ini, sinyal referensi yang digunakan adalah

sinusoidal. Grafik berwarna merah merupakan respon posisi pada sumbu-X dengan nilai K_θ sama dengan 0,1. Grafik berwarna hitam merupakan respon posisi pada sumbu-X dengan nilai K_θ sama dengan 10. Grafik berwarna biru merupakan respon posisi pada sumbu-X dengan nilai K_θ sama dengan 100. Seperti pada Persamaan (2.42), nilai parameter K_θ memiliki hubungan dengan rasio redaman (ζ) dan nilai parameter K_y, sehingga semakin besar nilai parameter K_θ maka respon pergerakan robot akan semakin berosilasi karena nilai ζ yang dihasilkan semakin besar. Indeks performansi dari percobaan 1 ini dapat dilihat pada Tabel 4.10.

Tabel 4.10 Indeks performansi sistem dengan variasi parameter konstanta K_θ (pada sumbu X)

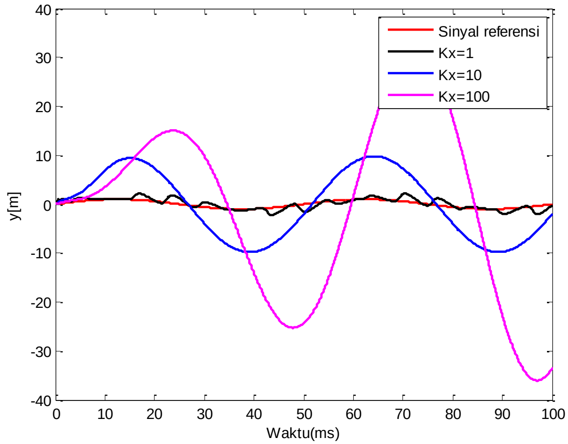
Parameter K_θ	**RMSE**	**ISE**	**IAE**
1	1.2455	1.6707e+003	837.8686
10	5.7755	3.7632e+004	5.8184e+003
100	17.2910	4.5626e+006	2.1859e+005

Pada Tabel 4.10 RMSE, ISE, dan IAE merupakan metode yang digunakan untuk memvalidasi data hasil pengujian. Berdasarkan hasil gambar dan indeks performansi sistem, dapat diketahui jika semakin besar nilai K_θ maka respon sistem yang dihasilkan semakin mendekati bentuk aslinya.



Gambar 4.18 Respon posisi robot pada sumbu X dengan Variasi Parameter K_θ

Gambar 4.19 merupakan grafik respon berupa posisi *AGV* pada sumbu Y dengan nilai parameter K_θ yang berbeda. Pada pengujian ini, sinyal referensi yang digunakan adalah sinusoidal. Grafik berwarna merah merupakan respon posisi pada sumbu-X dengan nilai K_θ sama dengan 0,1. Grafik berwarna hitam merupakan respon posisi pada sumbu-X dengan nilai K_θ sama dengan 10. Grafik berwarna biru merupakan respon posisi pada sumbu-X dengan nilai K_θ sama dengan 100. Seperti pada Persamaan (2.42), nilai parameter K_θ memiliki hubungan dengan rasio redaman (ζ) dan nilai parameter K_y, sehingga semakin besar nilai parameter K_θ maka respon pergerakan robot akan semakin berosilasi karena nilai ζ yang dihasilkan semakin besar. Indeks performansi dari percobaan 1 ini dapat dilihat pada Tabel 4.11.



Gambar 4.19 Respon posisi robot pada sumbu Y dengan Variasi Parameter K_θ

Pada Tabel 4.11 RMSE, ISE, dan IAE merupakan metode yang digunakan untuk memvalidasi data hasil pengujian. Berdasarkan hasil gambar dan indeks performansi sistem, dapat diketahui jika semakin besar nilai K_θ maka respon sistem yang dihasilkan semakin mendekati bentuk aslinya.

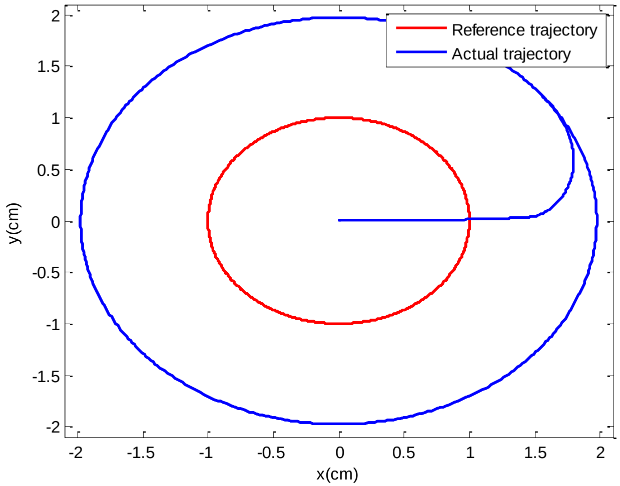
Tabel 4.11 Indeks performansi sistem dengan variasi parameter konstanta K_θ (pada sumbu Y)

Parameter K_θ	**RMSE**	**ISE**	**IAE**
1	0.6888	511.1053	546.8081
10	5.7803	3.7626e+004	5.6154e+003
100	18.2612	5.0899e+006	2.2785e+005

Berdasarkan data pengujian *control rule* Kanayama dengan variasi nilai konstanta K_x, K_y, dan K_θ didapatkan indeks performansi terkecil ketika konstanta K_x bernilai 100, K_y bernilai 100, dan K_θ bernilai 1.

4.3 Pengujian Lintasan Lingkaran menggunakan *Control Rule* Kanayama

Berdasarkan hasil pengujian pada subbab 4.1 maka didapatkan nilai parameter dengan indeks performansi paling kecil adalah $K_x = 100$ dan $K_y = 100$ serta $K_\theta = 1$. Hasil lintasan yang dihasilkan seperti pada Gambar 4.19.



Gambar 4.20 Perbandingan antara postur referensi dan aktual dengan parameter *control rule* Kanayama K_x = 100 K_y=100 K_θ=1

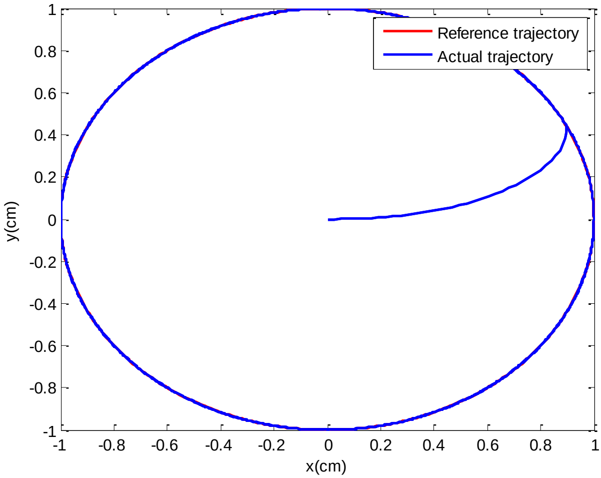
Gambar 4.20 menunjukkan respon yang dihasilkan mampu bergerak mengikuti lintasan berupa bentuk lingkaran yang diinginkan. Namun, masih tampak perbedaan yakni dengan RMSE sebesar sumbu-X 0.6777 dan sumbu-Y sebesar 0.6348.

4.4 Pengujian Kontroler Logika *Fuzzy*

Pengujian ini dilakukan untuk mengetahui hasil perancangan kontroler logika *fuzzy* merupakan solusi yang memiliki fungsi energi paling minimal. kendalian pengujian juga dilakukan untuk mengetahui apakah kontroler logika *fuzzy* yang dirancang mampu bekerja sesuai keinginan walaupun objek yang akan dikendalikan memiliki kondisi awal yang berbeda.

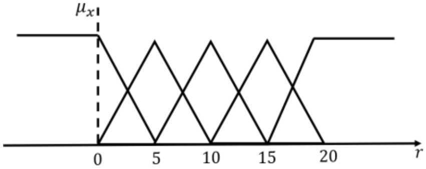
4.3.1. Variasi Jumlah Fungsi Keanggotaan

Pengujian dilakukan dengan menggunakan tiga fungsi keanggotaan yang berbeda jumlahnya. Pada pengujian ke I, digunakan tiga fungsi keanggotaan. Pengujian ke II menggunakan lima fungsi keanggotaan, sedangkan pada pengujian ke III menggunakan sembilan fungsi keanggotaan. Semakin banyak jumlah fungsi keanggotaan maka akan mempengaruhi performa kontroler.



Gambar 4.21 Pengujian Perancangan sistem kendali otomatismenggunakan KLF dengan tiga fungsi keanggotaan

Pada Gambar 4.21 gambar grafik berwarna merah tidak terlihat. Hal ini muncul karena respon yang dihasilkan menyerupai dengan lintasan referensi. Pengujian pertama ini dilakukan dengan menggunakan tiga fungsi keanggotaan, tiga variabel, dan tiga inference *rule*. Fungsi keanggotaan pada sumbu X berisi seperti pada Tabel 3.6. Pengujian dilakukan pada kontroler logika *fuzzy* dan menghasilkan respon seperti pada Gambar 4.21, dimana nilai RMSE pergerakan robot pada sumbu-X adalah 0.0758 dan sumbu Y sebesar 0.0167.

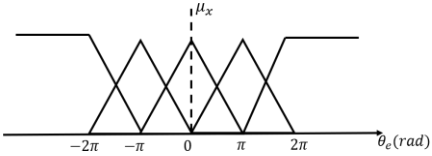


Gambar 4.22 Fuzzifikasi R dengan lima fungsi keanggotaan

Pengujian selanjutnya menggunakan lima fungsi keanggotaan dan lime inference *rule*. Dimana fungsi keanggotaan yang digunakan seperti pada Gambar 4.22 yakni fungsi segitiga. Jika kesalahan posisi yang dihasilkan oleh robot berada pada jarak 0-5 cm maka R masuk ke dalam kelas ‘Dekat’, kelas ‘Agak dekat’ didapatkan jika kesalahan posisi yang dihasilkan robot berada pada jarak 0-10 cm, kelas ’Normal’ untuk kesalahan posisi sebesar 5-15cm, jarak kesalahan posisi 10-20 cm masuk ke dalam kelas ‘Agak jauh’ , dan kesalahan posisi dengan jarak lebih dari 20 cm akan masuk ke dalam kelas ‘Jauh’.

Tabel 4.12 *Rule based* untuk kecepatan linier

$\boldsymbol{r}$	Dekat	Agak dekat	Normal	Agak Jauh	Jauh
$\boldsymbol{v}$	Off	Lambat	Agak lambat	Agak cepat	Cepat



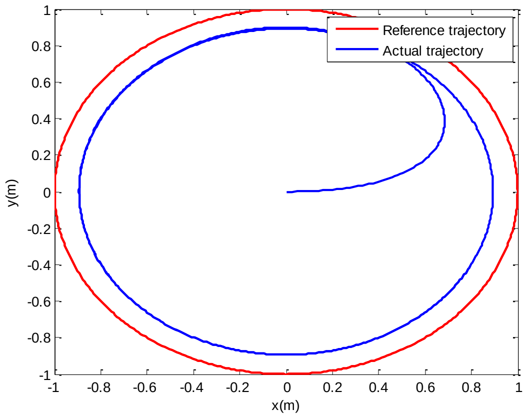
Gambar 4.23 Fuzzifikasi θe dengan lima fungsi keanggotaan

Jika kesalahan orientasi yang dihasilkan robot kurang dari -2π rad maka θe masuk ke dalam kelas ‘negatif besar’, kelas ‘Negatif kecil’ didapatkan jika kesalahan orientasi yang dihasilkan robot berada pada jarak -2π sampai 0 rad, kelas ’sekitar nol’ untuk kesalahan orientasi sebesar –π sampai dengan π rad, jarak kesalahan orientasi 0 sampai 2π rad masuk ke dalam kelas ‘Positif kecil’ , dan kesalahan orientasi dengan jarak lebih dari 2π rad akan masuk ke dalam kelas ‘Positif besar’.

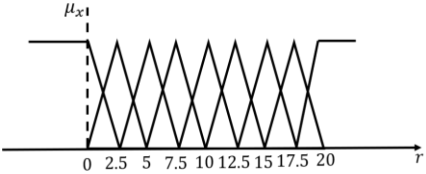
Tabel 4.13 *Rule* based untuk kecepatan anguler

θ_e	Negatif besar	Negatif kecil	Sekitar nol	Positif kecil	Positif besar
ω	Positif besar	Positif kecil	Off	Negatif kecil	Negatif besar

Pada Gambar 4.24 merupakan respon dari pergerakan robot pada lintasan berupa lingkaran. Pengujian kedua ini dilakukan dengan menggunakan lima fungsi keanggotaan seperti pada Gambar 4.22 dan 4.23 serta inference *rule* seperti pada Tabel 4.7 dan 4.8. Pengujian dilakukan pada kontroler logika *fuzzy* memiliki nilai RMSE pergerakan robot pada sumbu-X adalah 0.0758 dan sumbu Y sebesar 0.0167.



Gambar 4.24 Pengujian KLF dengan lima fungsi keanggotaan



Gambar 4.25 Fuzzifikasi R dengan sembilan fungsi keanggotaan

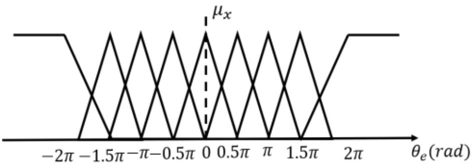
Pengujian selanjutnya menggunakan sembilan fungsi keanggotaan dan sembilan inference *rule*. Dimana fungsi keanggotaan yang digunakan seperti pada Gambar 4.25 yakni fungsi segitiga. Jika kesalahan posisi yang dihasilkan oleh robot berada pada jarak 0-2.5 cm maka R masuk ke dalam kelas 'Sangat dekat sekali', kelas 'Sangat dekat' didapatkan jika kesalahan posisi yang dihasilkan robot berada pada jarak 0-5 cm, kelas 'Agak dekat' untuk kesalahan posisi sebesar 2.5 – 7.5 cm, jarak kesalahan posisi 5-10 cm masuk ke dalam kelas 'Dekat' , dan kesalahan posisi dengan jarak 7.5 – 12.5 cm akan masuk ke dalam kelas 'Normal' sedangkan untuk R dengan jarak sebesar 10 – 15 cm masuk ke dalam kelas 'Agak jauh', kesalahan posisi R sebesar 12.5 – 17.5 cm termasuk ke dalam kelas 'Jauh', kelas 'Sangat jauh' berisi kesalahan posisi dengan jarak 15-20 cm, jika kesalahan posisi berada lebih dari 20 cm maka masuk ke dalam kelas 'Sangat jauh sekali'.

Tabel 4.14 *Rule based* untuk kecepatan linier

r	v
Sangat dekat sekali	Off
Sangat dekat	Sangat lambat sekali
Agak dekat	Lambat
Dekat	Agak lambat
Normal	Normal
Agak jauh	Agak cepat
Jauh	Cepat
Sangat jauh	Cepat sekali
Sangat jauh sekali	Sangat cepat sekali

Pengujian selanjutnya menggunakan sembilan fungsi keanggotaan dan sembilan inference *rule*. Dimana fungsi keanggotaan yang digunakan seperti pada Gambar 4.26 yakni fungsi segitiga. Jika kesalahan orientassi

yang dihasilkan oleh robot berada pada jarak kurang dari -2π rad maka R masuk ke dalam kelas ‘Negatif besar’, kelas ‘Negatif agak besar’ didapatkan jika kesalahan posisi yang dihasilkan robot berada pada jarak -2π sampai –π rad, kelas ’Negatif’ untuk kesalahan posisi sebesar -1.5π sampai dengan -0.5π rad, ,kelas ’Negatif kecil’ berisi kesalahan posisi dengan jarak – π sampai dengan 0 rad, jarak kesalahan posisi –0.5π sampai dengan 0.5π rad masuk ke dalam kelas ‘Sekitar nol’ , dan kesalahan orientasi sebesar 0 sampai dengan π rad akan masuk ke dalam kelas ‘Positif kecil’ sedangkan untuk θe dengan jarak sebesar 0.5π sampai 1.5π rad masuk ke dalam kelas ‘Positif’, kesalahan posisi θe sebesar π sampai dengan 2π rad termasuk ke dalam kelas ’Positif agak besar’, jika kesalahan posisi berada lebih dari 2π maka masuk ke dalam kelas ’Positif besar’.



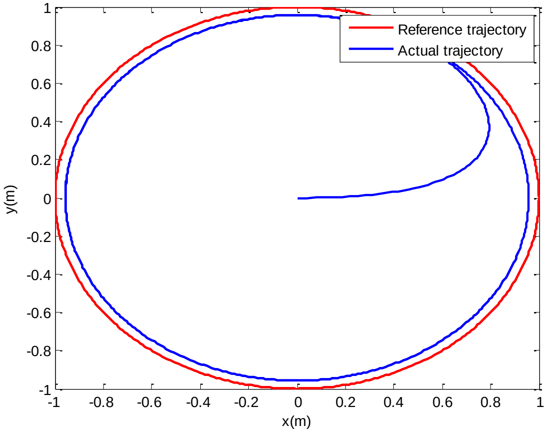
Gambar 4.26 Fuzzifikasi θe dengan sembilan fungsi keanggotaan

Tabel 4.15 *Rule* based untuk kecepatan anguler

θ_e	ω
Negatif besar	Positif kecil
Negatif agak besar	Positif
Negatif	Positif agak besar
Negatif kecil	Positif besar
Sekitar nol	Off
Positif kecil	Negatif besar
Positif	Negatif agak besar
Positif agak besar	Negatif
Positif besar	Negatif kecil

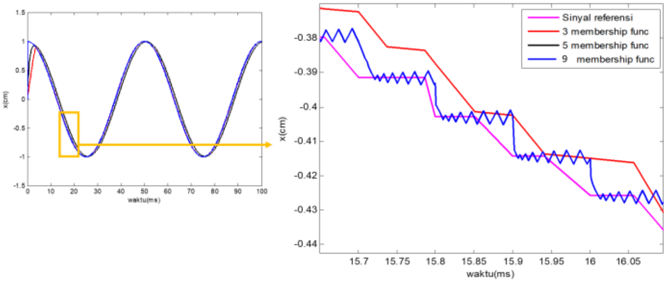
Pada Gambar 4.27 merupakan respon dari pergerakan robot pada lintasan berupa lingkaran. Pengujian kedua ini dilakukan dengan menggunakan lima fungsi keanggotaan seperti pada Gambar 4.25 dan 4.26 serta inference *rule* seperti pada Tabel 4.14 dan 4.15. Pengujian dilakukan pada

kontroler logika *fuzzy* memiliki nilai RMSE pergerakan robot pada sumbu-X adalah 0.2401 dan sumbu Y sebesar 0.2011.

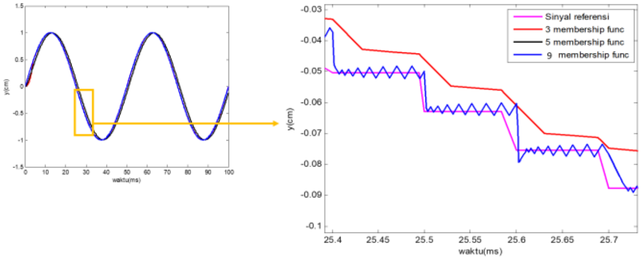


Gambar 4.27 Perancangan sistem kendali otomatismenggunakan KLF dengan sembilan fungsi keanggotaan

Pengujian mengenai jumlah fungsi keanggotaan, semakin banyak jumlahnya menyebabkan semakin banyak proses evaluasi nilai kesalahan posisi dan orientasi yang dilakukan. Sehingga menyebabkan sistem mengalami osilasi seperti pada Gambar 4.28 dan 4.29.



Gambar 4.28 Respon pergerakan robot pada sumbu X dengan variasi jumlah fungsi keanggotaan



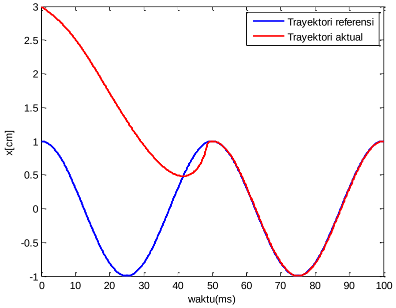
Gambar 4.29 Respon pergerakan robot pada sumbu Y dengan variasi jumlah fungsi keanggotaan

Tabel 4.16 Perbandingan jumlah fungsi keanggotaan

Jumlah *membership function*	**RMSE lintasan pada sumbu X**	**RMSE lintasan pada sumbu Y**
3	0.0758	0.0167
5	0.3451	0.3053
10	0.2401	0.2011

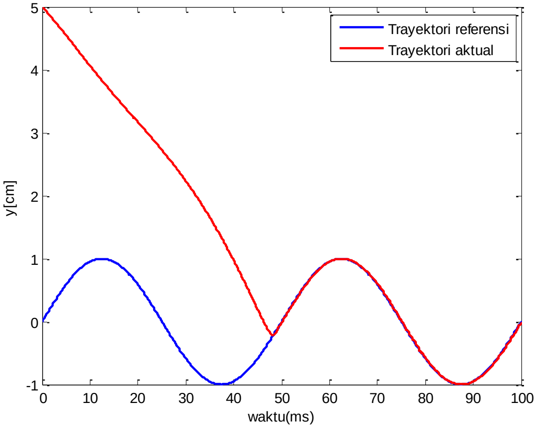
4.3.2. Variasi Kondisi Awal (*initial condition*) *AGV*

Pengujian selanjutnya dilakukan dengan mengubah posisi awal dari *AGV*. Tiga posisi awal diberikan kepada robot dan yang ingin ditinjau adalah kemampuan *AGV* untuk bergerak sesuai dengan lintasan yang diinginkan meskipun berasal dari posisi awal yang berubah-ubah. Hal ini memiliki tujuan untuk menguji kemampuan kontroler logika *fuzzy* yang diajukan mampu bekerja pada kondisi awal yang berbeda.



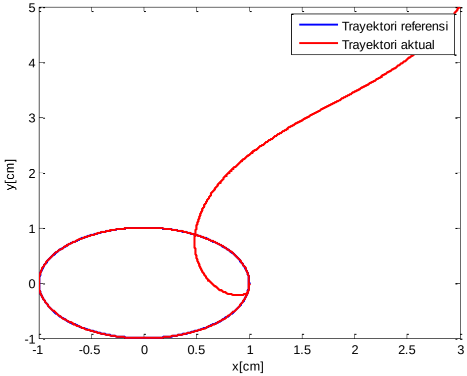
Gambar 4.30 Respon pergerakan robot pada sumbu X menggunakan KLF dengan initial condition x=3 dan y=5

Pengujian dilakukan pada kontroler logika *fuzzy* dengan *initial condition* posisi x_c=3 dan y_c=5. Pada Gambar 4.30, posisi awal robot berada pada titik x=3cm sehingga menghasilkan grafik berwarna merah, pada saat t=50 ms respon pergerakan robot mengikuti referensi yang diberikan hal ini disebabkan inference *rule* yang diimplementasikan seperti pada Tabel 3.3 dan 3.4. Respon pergerakan robot pada sumbu Y pada pengujian ini posisi awal robot berada pada y=5cm sehingga menghasilkan respon seperti pada Gambar 4.31. Pada saat t=50 ms respon pergerakan robot mengikuti referensi yang diberikan hal ini disebabkan inference *rule* yang diimplementasikan seperti pada Tabel 3.3 dan 3.4.



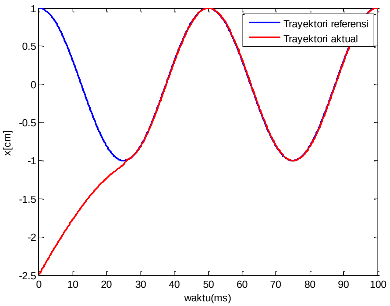
Gambar 4.31 Respon pergerakan robot pada sumbu Y menggunakan KLF dengan initial condition x=3 dan y=5

Dapat dilihat pada Gambar 4.32 bahwa garis berwarna merah merupakan lintasan referensi sedangkan garis berarna biru merupakan lintasan aktual, dapat diamati meskipun posisi awal robot tidak berada pada titik pusat gravitasi (0,0) dimana nilai RMSE pergerakan robot pada sumbu-X adalah 1.2016 dan sumbu Y sebesar 1.8602.

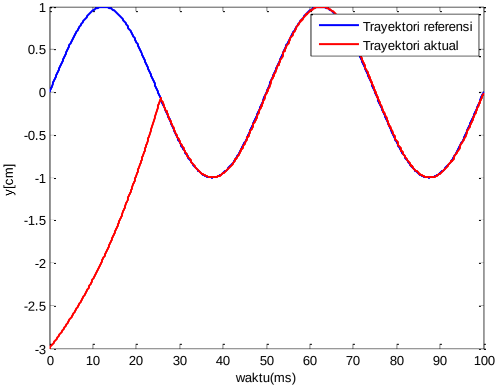


Gambar 4.32 Pengujian KLF dengan initial condition x=3 dan y=5

Pengujian dilakukan pada kontroler logika *fuzzy* dengan *initial condition* posisi x_c=-2.5 dan y_c=3. Pada Gambar 4.33, posisi awal robot berada pada titik x=-2.5cm sehingga menghasilkan grafik berwarna merah, pada saat t=30 ms respon pergerakan robot mengikuti referensi yang diberikan hal ini disebabkan inference *rule* yang diimplementasikan seperti pada Tabel 3.3 dan 3.4. Respon pergerakan robot pada sumbu Y pada pengujian ini posisi awal robot berada pada y=3cm sehingga menghasilkan respon seperti pada Gambar 4.34. Pada saat t=30 ms respon pergerakan robot mengikuti referensi yang diberikan hal ini disebabkan inference *rule* yang diimplementasikan seperti pada Tabel 3.3 dan 3.4.

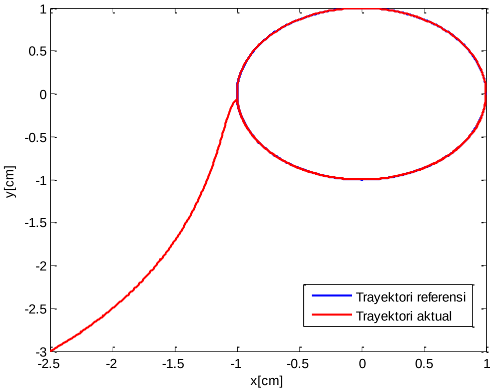


Gambar 4.33 Respon pergerakan robot pada sumbu X menggunakan KLF dengan initial condition x=-2.5 dan y=-3

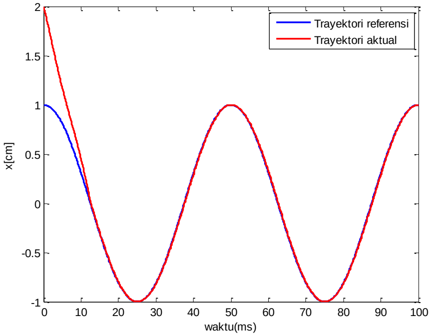


Gambar 4.34 Respon pergerakan robot pada sumbu Y menggunakan KLF dengan initial condition x=-2.5 dan y=-3

Pengujian dilakukan pada kontroler logika *fuzzy* dengan *initial condition* posisi x_c=-25 dan y_c=-3 dan menghasilkan respon seperti pada Gambar 5.6, dimana nilai RMSE pergerakan robot pada sumbu-X adalah 0.9728 dan sumbu Y sebesar 1.2481. Dari Gambar 4.35 dapat diketahui grafik berwarna merah menunjukkan lintasan referensi sedangkan grafik berwarna biru merupakan lintasan aktual.

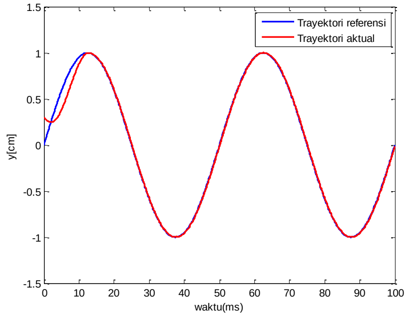


Gambar 4.35 Pengujian KLF dengan initial condition x=-2.5 dan y=-3



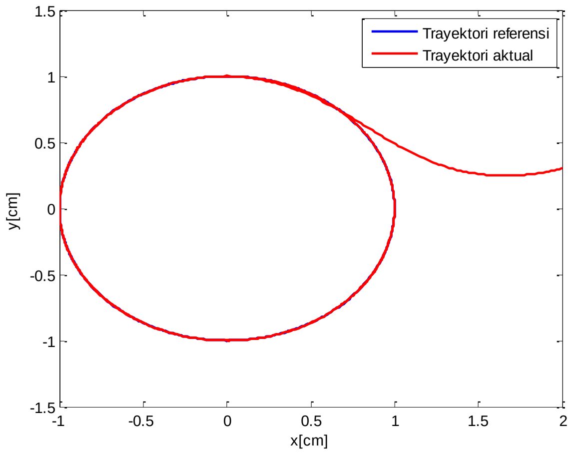
Gambar 4.36 Respon pergerakan robot pada sumbu X menggunakan KLF dengan initial condition x=2 dan y=0.7

Pengujian dilakukan pada kontroler logika *fuzzy* dengan *initial condition* posisi x_c=2 dan y_c=0.7. Pada Gambar 4.36, posisi awal robot berada pada titik x=2cm sehingga menghasilkan grafik berwarna merah, pada saat t=11 ms respon pergerakan robot mengikuti referensi yang diberikan hal ini disebabkan inference *rule* yang diimplementasikan seperti pada Tabel 3.3 dan 3.4. Respon pergerakan robot pada sumbu Y pada pengujian ini posisi awal robot berada pada y=0.7cm sehingga menghasilkan respon seperti pada Gambar 4.37. Pada saat t=11 ms respon pergerakan robot mengikuti referensi yang diberikan hal ini disebabkan inference *rule* yang diimplementasikan seperti pada Tabel 3.3 dan 3.4.



Gambar 4.37 Respon pergerakan robot pada sumbu Y menggunakan KLF dengan initial condition x=2 dan y=0.7

Pengujian dilakukan pada kontroler logika *fuzzy* dengan *initial condition* posisi x_c=2 dan y_c=0.3 dan menghasilkan respon seperti pada Gambar 4.38, dimana nilai RMSE pergerakan robot pada sumbu-X adalah 0.1379 dan sumbu Y sebesar 0.045.



Gambar 4.38 Pengujian *path tracking* menggunakan KLF dengan initial condition x=2 dan y=0.7

Halaman ini sengaja dikosongkan

BAB V
PENUTUP

5.1 Kesimpulan

Penggunaan Perancangan sistem kendali otomatisKanayama dengan nilai parameter K_x=100, K_y=100, dan K_θ=1 memiliki nilai RMSE sebesar pergerakan robot pada sumbu-X adalah 0.1305 dan sumbu Y sebesar 0.0361. Penggunaan kontroler logika *fuzzy* untuk trajectory tracking menghasilkan RMSE untuk pergerakan robot pada sumbu-X adalah 0.0758 dan sumbu Y sebesar 0.0167

5.2 Saran

Pada penelitian selanjutnya dapat mengembangkan *robot controller* sehingga pergerakan robot keseluruhan dapat diimplementasikan.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

[1] Hamed Fazlollahtabar, Mohammad Saidi-M. Autonomous Guided Vehicle (Methods and Models for Optimal Planning), Springer International , 2015.

[2] Lothar Schulze,Sebastian Behling, dan Stefan Buhrs "Automated Guided Vehicle Systems: a Driver for Increased Busines Performance," in International MultiConference of Engineers and Computer Scientist , HongKong , 2008.

[3] Martin Kajan, "Control of Automated Guided Vehicle with PLC SIMATIC ET200S CPU," American Journal of Mechanical Engineering , vol. 7, pp. 343-348, 2013.

[4] Gabriel Fedorko, Stanislav Honus, dan Rland Salai "Comparison of the Traditional and Autonomous AGV Systems," in Matec Web of Conferences , 2017.

[5] Tua Agustinus Tamba,Bong Hee, dan Keum Shik H. "a Path Following Control of an Unmanned Autonomous Forklift," International Journal of Control, Automation, and Systems , vol. 7, pp. 113-122, 2009.

[6] C. Russel, Writer, AGV Navigation: Pros and Cons. [Performance]. Mitsubishi Heavy Industries, 2018.

[7] Rachid, A. Kamga "Speed Steering Angle and Path Tracking Controls for a Tricycle Robot," in IEEE Transactions on Control Systems Technology , 2002.

[8] K. R. Sarath Kodagoda, W. S. Wijesoma, dan E.K. Teoh "Fuzzy Speed and Steering Control of an AGV," IEEE Transactions on Control System Technology , pp. 112-120, 2002.

[9] Yutaka Kanayama, Yoshihiko Kimura, Fumio Miyazaki, Tetsuo N. "A Stable Tracking Control Method," 1993.

[10] Mohammed Rabeea Hashim Al-Dahhan, Dr. Mochammad M. Ali "Path Tracking Control of a Mobile Robot using Fuzzy Logic," in 13th International Multi-Conference on Systems, Signals & Devices , 2016 .

[11] Vlavac Hlavac, "Robot Trajectory Generation," Czech Technical University, Prague.

[12] Mark W. Spong dan M. Vidyasagar, Robot Dynamic and Control.

[13] Sazo Blažic, "On Periodic Control Laws for Mobile Robots," IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, vol. 61, pp. 3660-3670, 2014.

[14] K. Ogata, "Transient-Response Analysis," in Modern Control Engineering, Prentice-Hall Inc, 1997, p. 146.

[15] C. C. Lee, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller - Part I," IEEE Transactons on Systems and Cybernetics, vol. 20, no. 2, pp. 404-435, 1990.

[16] L. A. Zadeh, "The Concept of a Linguistic Variable and Its Application to Approximate Reasoning-I," American Elsevier Publishing Company,Inc. , America, 1975.

[17] Ramzy S. Ali, Ammar A. Aldair, Ali K. A. "Design an Optimal PID Controller using Artificial Bee Colony and Genetic Algorithm for Autnoomous Mobile Robot," International Journal of Computer Applications, vol. 100, no. 16, pp. 8-16, 2014.

[18] Roland Siegwart, Ilah R. Nourbakhsh, David Scaramuza "Mobile Kinematics," in Introduction to Autonomous Mobile Robots, Cambridge , The MIT Press, 2004, pp. 57-58.

[19] R. Fierro, F. L. Lewis "Control of a Nonholonomic Mobile Robot: Backsteeping Kinematics into Dynamics," in Conference on Decision and Control, New Orleans, Los Angeles , 1995.

[20] D. Anesh, "Tracking Controller for Mobile Robot," in Interntaional Conference on Computing, Electronics, and Electrical Technologies , 2012.

[21] Mechanical Engineering MIT, "Understanding D.C. Motor Characteristics," 1999. [Online]. Available: http://lancet.mit.edu/motors/motors3.html. [Accessed 3 July 2019].

[22] Jun Yan, Michael Ryan, James Power"Fuzzy Logic Controllers (FLC)-design issues," in Using Fuzzy Logic, Hertfordshire, Prentice Hall International, 1994, p. 45.

[23] P. V. Nagy, "Trajectory Tracking Control for Industrial robotss," Journal of Mechanical Working Technoogy, vol. 20, pp. 273-281, 1989.

[24] N. Uddin, "Trajectory Tracking Control System Design For Autonomous Two-Wheeled Robot," Jurnal INFOTEL, pp. 90-97, 2018.

Halaman ini sengaja dikosongkan

LAMPIRAN

A Perhitungan diferensial kesalahan postur pada *mobile robot*

$$P_e = \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} cos(\theta_c) & sin(\theta_c) & 0 \\ -sin(\theta_c) & cos(\theta_c) & 0 \\ 0 & 0 & 1 \end{bmatrix} (q_r - q_d)$$

$$\begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} cos(\theta_c) & sin(\theta_c) & 0 \\ -sin(\theta_c) & cos(\theta_c) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x_c \\ y_r - y_c \\ \theta_r - \theta_c \end{bmatrix}$$

$$\begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} (x_r - x_c)cos(\theta_c) + (y_r - y_c)sin(\theta_c) \\ -(x_r - x_c)sin(\theta_c) + (y_r - y_c)cos(\theta_c) \\ \theta_r - \theta_c \end{bmatrix}$$

$$e_x = x_r cos(\theta_c) - x_c cos(\theta_c) + y_r sin(\theta_c) - y_c sin(\theta_c)$$

$$\dot{e_x} = \dot{x_r} cos(\theta_c) + x_r \omega_c sin(\theta_c) - \dot{x_c} cos(\theta_c) - x_c \omega_c sin(\theta_c) + \dot{y_r} sin(\theta_c) + y_r \omega_c cos(\theta_c) - \dot{y_c} cos(\theta_c) - y_c \omega_c sin(\theta_c)$$

$$\dot{e_x} = \dot{x_r} cos(\theta_c) - \dot{x_c} cos(\theta_c) + x_r \omega_c sin(\theta_c) - x_c \omega_c sin(\theta_c) + \dot{y_r} sin(\theta_c) - \dot{y_r} sin(\theta_c) + y_r \omega_c cos(\theta_c) - y_c \omega_c cos(\theta_c)$$

$$\dot{e_x} = \dot{x_r} cos(\theta_c) - \dot{x_c} cos(\theta_c) + \dot{y_r} sin(\theta_c) - \dot{y_r} sin(\theta_c) + x_r \omega_c sin(\theta_c) - x_c \omega_c sin(\theta_c) + y_r \omega_c cos(\theta_c) - y_c \omega_c cos(\theta_c)$$

$$\dot{e_x} = \dot{x_r} cos(\theta_c) - \dot{x_c} cos(\theta_c) + \dot{y_r} sin(\theta_c) - \dot{y_r} sin(\theta_c) + \omega_c \{x_r sin(\theta_c) - x_c sin(\theta_c) + y_r cos(\theta_c) - y_c cos(\theta_c)\}$$

$$\dot{e_x} = \dot{x_r} cos(\theta_c) - \dot{x_c} cos(\theta_c) + \dot{y_r} sin(\theta_c) - \dot{y_r} sin(\theta_c) + \omega_c \{(x_r - x_c) sin(\theta_c) + (y_r - y_c) cos(\theta_c)\}$$

$$\dot{e_x} = \dot{x_r} cos(\theta_c) - \dot{x_c} cos(\theta_c) + \dot{y_r} sin(\theta_c) - \dot{y_r} sin(\theta_c) + \omega_c y_e$$

$$\dot{e_x} = \dot{x_r} cos(\theta_c) - v_c cos(\theta_c) cos(\theta_c) + \dot{y_r} sin(\theta_c) - v_c sin(\theta_c) sin(\theta_c) + \omega_c y_e$$

$$\dot{e_x} = \dot{x_r} cos(\theta_c) + \dot{y_r} sin(\theta_c) - v_c \left(cos^2(\theta_c) + sin^2(\theta_c)\right) + \omega_c y_e$$

$$\dot{e_x} = \dot{x_r} cos(\theta_c) + \dot{y_r} sin(\theta_c) - v_c + \omega_c y_e$$

$$\dot{e_x} = v_r cos(\theta_r) cos(\theta_c) + v_r sin(\theta_r) sin(\theta_c) - v_c + \omega_c y_e$$

$$\dot{e_x} = v_r cos(\theta_r - \theta_c) - v_c + \omega_c y_e$$

$$\dot{e_x} = v_r cos(\theta_e) - v_c + \omega_c y_e$$

$$e_y = -x_r sin(\theta_c) + x_c sin(\theta_c) + y_r cos(\theta_c) - y_c cos(\theta_c)$$

$$\dot{e_y} = -\dot{x}_r sin(\theta_c) - x_r\omega_c cos(\theta_c) + \dot{x}_c sin(\theta_c) + x_c\omega_c cos(\theta_c) + \dot{y}_r cos(\theta_c) - y_r\omega_c sin(\theta_c) - \dot{y}_c cos(\theta_c) + y_c\omega_c sin(\theta_c)$$

$$\dot{e_y} = -\dot{x}_r sin(\theta_c) + \dot{x}_c sin(\theta_c) + \dot{y}_r cos(\theta_c) - \dot{y}_c cos(\theta_c) - x_r\omega_c cos(\theta_c) + x_c\omega_c cos(\theta_c) - y_r\omega_c sin(\theta_c) + y_c\omega_c sin(\theta_c)$$

$$\dot{e_y} = -\dot{x}_r sin(\theta_c) + \dot{x}_c sin(\theta_c) + \dot{y}_r cos(\theta_c) - \dot{y}_c cos(\theta_c) - (x_r + x_c)\omega_c cos(\theta_c) - (y_r + y_c)\omega_c sin(\theta_c)$$

$$\dot{e_y} = -\dot{x}_r sin(\theta_c) + \dot{x}_c sin(\theta_c) + \dot{y}_r cos(\theta_c) - \dot{y}_c cos(\theta_c) - \omega_c\{(x_r + x_c)cos(\theta_c) + (y_r + y_c)sin(\theta_c)\}$$

$$\dot{e_y} = -\dot{x}_r sin(\theta_c) + \dot{x}_c sin(\theta_c) + \dot{y}_r cos(\theta_c) - \dot{y}_c cos(\theta_c) - \omega_c x_e$$

$$\dot{e_y} = -\dot{x}_r sin(\theta_c) + \dot{y}_r cos(\theta_c) + \dot{x}_c sin(\theta_c) - \dot{y}_c cos(\theta_c) - \omega_c x_e$$

$$\dot{e_y} = -\dot{x}_r sin(\theta_c) + \dot{y}_r cos(\theta_c) + v_c cos(\theta_c) sin(\theta_c) - v_c sin(\theta_c) cos(\theta_c) - \omega_c x_e$$

$$\dot{e_y} = -\dot{x}_r sin(\theta_c) + \dot{y}_r cos(\theta_c) - \omega_c x_e$$

$$\dot{e_y} = -\omega_c \dot{x_e} - \dot{x}_r sin(\theta_r - \theta_e) + \dot{y}_r cos(\theta_r - \theta_e)$$

$$\dot{e_y} = -\omega_c \dot{x_e} - \dot{x}_r\big(sin(\theta_r)cos(\theta_e) - cos(\theta_r)sin(\theta_e)\big) + \dot{y}_r\big(cos(\theta_r)cos(\theta_e) + sin(\theta_r)sin(\theta_e)\big)$$

$$\dot{e_y} = -\omega_c \dot{x}_e - \dot{x}_r\ sin(\theta_r)cos(\theta_e) + \dot{x}_r cos(\theta_r)sin(\theta_e) + \dot{y}_r\ cos(\theta_r)cos(\theta_e) + \dot{y}_r sin(\theta_r)sin(\theta_e)$$

$$\dot{e_y} = -\omega_c \dot{x}_e + sin(\theta_e)\big(\dot{x}_r cos(\theta_r) + \dot{y}_r sin(\theta_r)\big) + cos(\theta_e)\big(\dot{y}_r\ cos(\theta_r) - \dot{x}_r\ sin(\theta_r)\big)$$

$$\dot{e_y} = -\omega_c \dot{x}_e + sin(\theta_e)\big(v_r cos^2(\theta_r) + v_r sin^2(\theta_r)\big) + cos(\theta_e)\big(v_r sin\ (\theta_r)cos(\theta_r) - v_r cos(\theta_r)sin(\theta_r)\big)$$

$$\dot{e_y} = -\omega_c \dot{x}_e + sin(\theta_e)\left(v_r\big(cos^2(\theta_r) + sin^2(\theta_r)\big)\right)$$

$$\dot{e}_y = -\omega_c x_e + v_r sin(\theta_e)$$

$$\dot{\theta}_e = \dot{\theta}_r - \dot{\theta}_c = \omega_r - \omega_c$$

B Penurunan rumus kesalahan postur

$$\frac{\vartheta \dot{e}_x}{\vartheta e_x} = \frac{\vartheta\big(\omega_r e_y + v_r K_x {e_y}^2 + e_y K_\theta sin(e_\theta) - K_x e_x\big)}{\vartheta e_x} = -K_x$$

$$\frac{\vartheta \dot{e}_x}{\vartheta e_y} = \frac{\vartheta\left(\omega_r e_y + v_r K_x {e_y}^2 + e_y K_\theta sin(e_\theta) - K_x e_x\right)}{\vartheta e_y}$$

$$= \omega_r + 2K_x e_y + K_\theta sin(e_\theta)$$

$$\frac{\vartheta \dot{e}_x}{\vartheta e_y} = \omega_r$$

$$\frac{\vartheta \dot{e}_x}{\vartheta e_\theta} = \frac{\vartheta\left(\omega_r e_y + v_r K_x {e_y}^2 + e_y K_\theta sin(e_\theta) - K_x e_x\right)}{\vartheta e_\theta} = 0$$

$$\frac{\vartheta \dot{e}_y}{\vartheta e_x} = \frac{\vartheta\left(-\omega_r e_x + v_r e_x K_x e_y + v_r e_x K_\theta sin(e_\theta) + v_r sin(\theta_e)\right)}{\vartheta e_x}$$

$$= -\omega_r + v_r K_x e_y + v_r K_\theta sin(e_\theta)$$

$$\frac{\vartheta \dot{e}_y}{\vartheta e_x} = -\omega_r$$

$$\frac{\vartheta \dot{e}_y}{\vartheta e_y} = \frac{\vartheta\left(-\omega_r e_x + v_r e_x K_x e_y + v_r e_x K_\theta sin(e_\theta) + v_r sin(\theta_e)\right)}{\vartheta e_y} = v_r e_x K_x$$

$$= 0$$

$$\frac{\vartheta \dot{e}_y}{\vartheta e_\theta} = \frac{\vartheta\left(-\omega_r e_x + v_r e_x K_x e_y + v_r e_x K_\theta sin(e_\theta) + v_r sin(\theta_e)\right)}{\vartheta e_\theta}$$

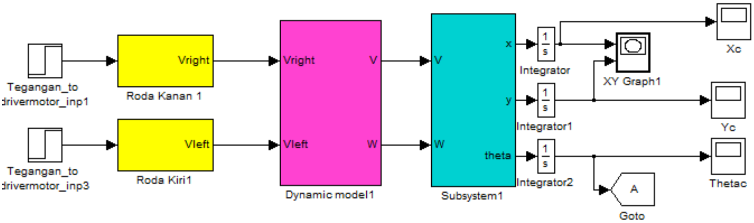
$$= v_r cos(\theta_e) = v_r$$

$$\frac{\vartheta \dot{e}_\theta}{\vartheta e_x} = \frac{-v_r K_y e_y - v_r K_\theta sin(e_\theta)}{\vartheta e_x} = 0$$

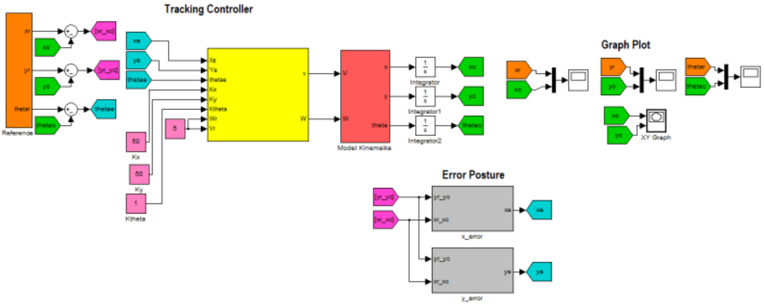
$$\frac{\vartheta \dot{e}_\theta}{\vartheta e_y} = \frac{-v_r K_y e_y - v_r K_\theta sin(e_\theta)}{\vartheta e_y} = -v_r K_y$$

$$\frac{\vartheta \dot{e}_\theta}{\vartheta e_\theta} = \frac{-v_r K_y e_y - v_r K_\theta sin(e_\theta)}{\vartheta e_\theta} = -v_r K_\theta cos(e_\theta) = -v_r K_\theta$$

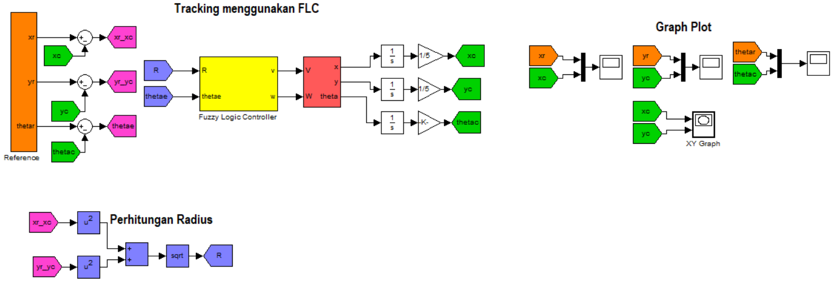
C Diagram SIMULINK Pengujian Model Kinematika



D Diagram SIMULINK Perancangan sistem kendali otomatis Kanayama



E Diagram SIMULINK Kontroler Logika *Fuzzy*



RIWAYAT PENULIS

Penulis bernama Adi Novitarini Putri, merupakan mahasiswi S1 Lintas Jalur Departemen Teknik Elektro ITS (bidang studi Teknik Sistem Pengaturan), Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember Surabaya. Penulis lahir di Surabaya, pada tanggal 14 November 1995. Penulis merupakan anak kedua dari tiga bersaudara dengan seorang saudari kembar. Sebelum melanjutkan perkuliahan di jenjang Sarjana, penulis berkuliah di Teknik Elektro Otomasi (bidang studi *Computer Control*), Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya (2014-2017) dan menghasilkan sebuah proyek akhir berjudul “Perancangan kinematika balik (*inverse kinematic*) pada Robot Manipulator 4DoF (*Degree of Freedom*)“.

Email : adinovitarini@gmail.com
No. Telp : 087855120583

Halaman ini sengaja dikosongkan