

KENDALI TEKANAN UDARA BERBASIS FUZZY LOGIC DAN PID
PADA RUANG RUANG OPERASI RUMAH SAKIT

Tesis

Disusun Sebagai Syarat Untuk Memperoleh Gelar Magister Teknik

Pada Program Magister Teknik Elektro

Universitas Islam Sultan Agung Semarang



Susilo Ari Wibowo

MTE. 20601700013

PROGRAM STUDI MAGISTER TEKNIK ELEKTRO
UNIVERSITAS ISLAM SULTAN AGUNG SEMARANG

2021

TESIS

KENDALI TEKANAN UDARA BERBASIS FUZZY LOGIC DAN PID PADA RUANG RUANG OPERASI RUMAH SAKIT

Yang dipersiapkan dan disusun oleh

Susilo Ari Wibowo

MTE 20601700013

Telah dipertahankan didepan dewan

Penguji Pada tanggal : 17 Juni 2021

Susunan Dewan Penguji

Pembimbing Utama


Arief Marwanto, ST.M.Eng., Ph.D
NIDN.0628097501

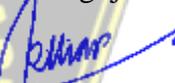
Ketua penguji


Dr. Ir.H. Muhamad Haddin, MT
NIDN. 0618066301

Pembimbing Pendamping


Ir. Suryani Alifah, MT., Ph.D
NIDN : 0625036901

Penguji I


Dr. Hj. Sri Artini Dwi P, M.Si
NIDN : 0620026501

Penguji II


Imam Much. Ibnu Subroto, ST, M.Sc.Ph.D
NIDN : 0613037301

Tesis ini telah diterima sebagai salah satu persyaratan untuk memperoleh gelar Megister Teknik

Tanggal, 17 Juni 2021

Ketua Program Studi Magister Teknik Elektro



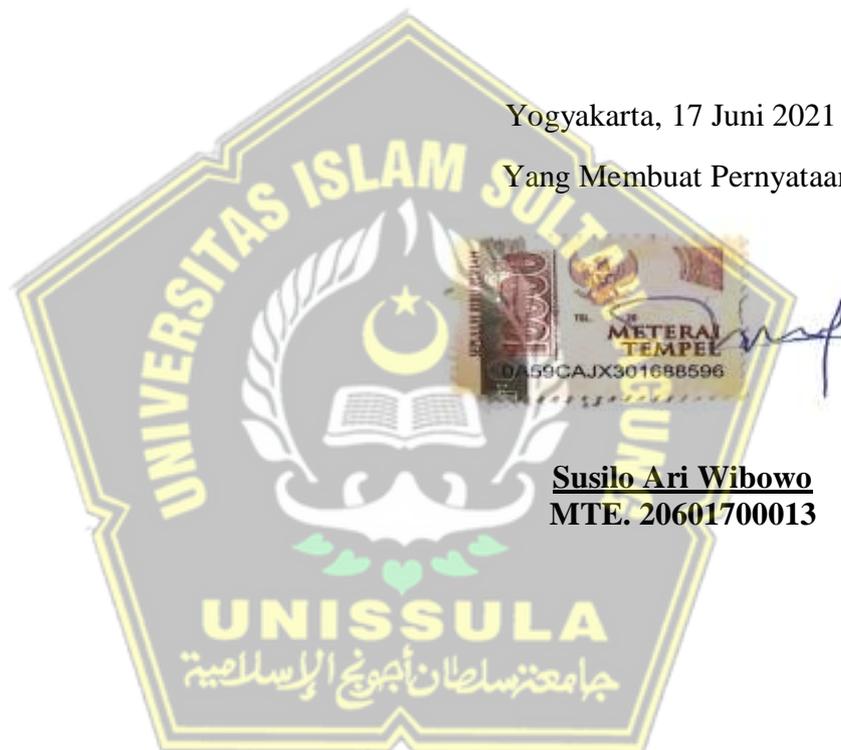

Arief Marwanto, ST., M.Eng., Ph.D
NIDN : 0628097501

LEMBAR PERNYATAAN

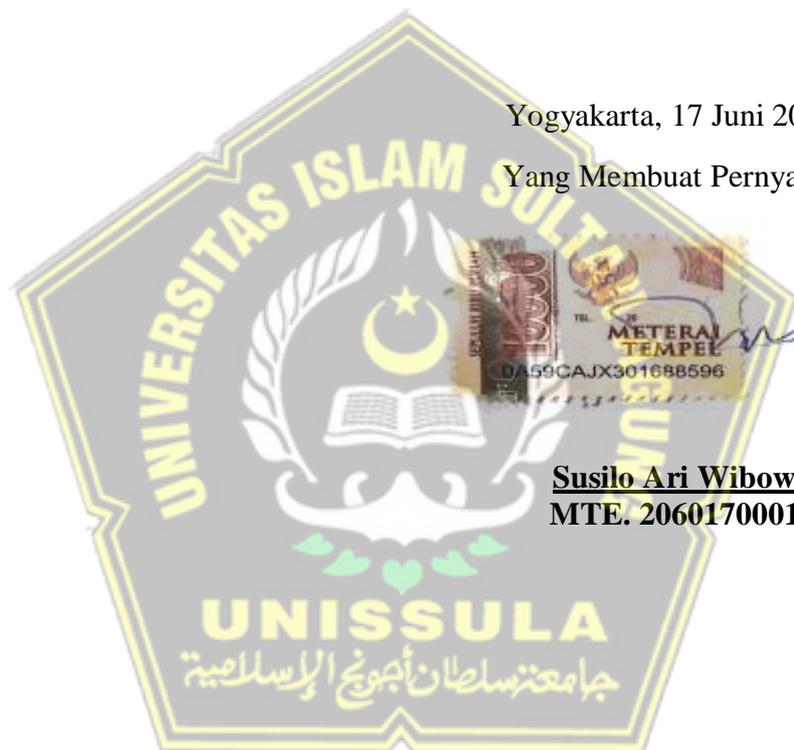
Dengan ini saya menyatakan bahwa tesis ini bukan merupakan hasil plagiat karya orang lain, melainkan hasil karya saya sendiri dan belum pernah diterbitkan oleh pihak manapun. Demikian pernyataan ini saya buat dengan sebenar-benarnya dan apabila di kemudian hari ada yang mengklaim bahwa karya ini adalah milik orang lain dan dibenarkan secara hukum, maka saya bersedia dituntut berdasarkan hukum yang berlaku di Indonesia.

Yogyakarta, 17 Juni 2021

Yang Membuat Pernyataan:



Susilo Ari Wibowo
MTE. 20601700013



KATA PENGANTAR

Segala puji dan syukur kehadirat Allah SWT yang telah melimpahkan rahmat, hidayah, dan karunia-Nya sehingga penulis dapat menyelesaikan tesis yang berjudul “ **KENDALI TEKANAN UDARA BERBASIS FUZZY LOGIC DAN PID PADA RUANG OPERASI RUMAH SAKIT** ” disusun dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana S2 pada Program Studi Magister Teknik Elektro Fakultas Teknologi Industri Universitas Islam Sultan Agung (UNISSULA) Semarang.

Dalam melakukan penelitian dan penyusunan tesis ini penulis telah mendapatkan banyak dukungan dan bantuan dari berbagai pihak. Penulis mengucapkan terimakasih yang tak terhingga kepada :

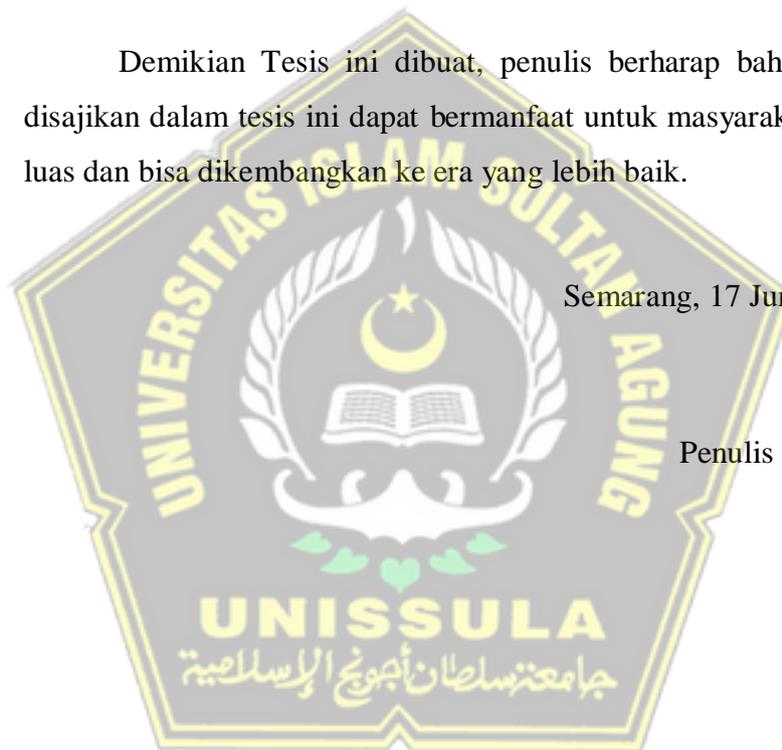
1. Bapak Drs. Bedjo Santoso, MT., Ph.D selaku Rektor UNISSULA
2. Ir. Suryani Alifah, MT., Ph.D selaku pembimbing pertama yang telah banyak memberikan masukan, saran, meluangkan waktu serta kemudahan kepada penulis dalam penyusunan tesis ini.
3. Arief Marwanto, ST., M.Eng., Ph.D selaku pembimbing kedua yang telah banyak memberikan masukan, saran, meluangkan waktu serta kemudahan kepada penulis dalam penyusunan ini.
4. Seluruh dosen Program Studi Magister Teknik Elektro Universitas Islam Sultan Agung yang tidak bisa penulis sebutkan semuanya, terima kasih telah memberikan ilmu dan pengalamannya kepada penulis, semoga ilmu yang bapak ibu berikan bermanfaat bagi semuanya dan semoga senantiasa mendapat pahala dari Allah SWT.
5. Ibunda tercinta Sunarti yang telah memberikan dukungan dan doa restu demi kesuksesan ananda.
6. Istri dan anakku tersayang yang selalu mensupport .

8. Teman-teman seperjuangan MTE Jogja (K u a t, D j o k o, T a t i a, L a t i f, M u t o h a r, I r f a n, d a n B r a m a) yang selalu memberikan dorongan, motivasi dan semangat selama penulis menyelesaikan tesis ini. Salam sukses buat kita semua
9. Segenap IPSRS RSU Islam Klaten, yang telah memberikan dorongan dan motivasi baik moril maupun materiil bagi penulis untuk menyelesaikan studi ini
10. Teman-teman seangkatan MTE Unisula 2017 yang tidak bisa disebutkan satu-satu.

Demikian Tesis ini dibuat, penulis berharap bahwa apa yang disajikan dalam tesis ini dapat bermanfaat untuk masyarakat yang lebih luas dan bisa dikembangkan ke era yang lebih baik.

Semarang, 17 Juni 2021

Penulis



DAFTAR ISI

JUDUL	i
LEMBAR PENGESAHAN.....	ii
LEMBAR PERNYATAAN	iii
KATA PENGANTAR	iv
DAFTAR ISI	vi
DAFTAR GAMBAR	ix
DAFTAR TABEL	xii
ABSTRAK.....	xiii
ABSTRACT.....	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang Penelitian	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan Penelitian	3
1.4 Batasan Penelitian.....	3
1.5 Manfaat Penelitian	3
1.6 Keaslian Penelitian.....	3
1.7 Kontribusi Tesis.....	4
BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI	5
2.1 Tinjauan Pustaka.....	5
2.2 Landasan Teori	5
2.2.1 Ruang Operasi	5
2.2.2 Bagian-Bagian kamar operasi	6
2.2.3 Bentuk dan ukuran.....	6
2.2.4 Suhu dan kelembaban	7
2.2.5 Logika Fuzzy	8
2.3 Defuzzifikasi.....	14
2.4 Kendali PID.....	15
2.5 Mikrokontroler.....	20
2.5.1 Gambaran Umum	20
2.5.2 Bus Alamat	21
2.5.3 Bus Data.....	21
2.5.4 Bus Kontrol	21
2.5.5 Memori.....	21
2.5.6 Input/Output	23
2.5.7 Mikrokontroler Arduino Uno	23

2.6 Kipas/fan	29
2.7 LCD.....	30
2.8 Power Supply.....	31
2.9 Driver Motor.....	32
2.10 Sensor mps20n0040d-d.....	33
2.11 Hx711	35
BAB III METODOLOGI.....	36
3.1 Desain Penelitian	36
3.1.1 Studi Pustaka.....	36
3.1.2 Analisa Kebutuhan	36
3.1.3 Perancangan	37
3.1.3.1 Parameter	37
3.1.3.2 Pemodelan Ruang Operasi.....	37
3.1.3.3 Aliran Udara.....	38
3.1.3.4 Hardware	39
3.1.3.5 Sistem Kendali	40
3.1.4 Software	42
3.1.5 Perancangan Input Output.....	43
3.1.6 Diagram Blok Sistem Hardware.....	44
3.2 Flowchart Sistem Fuzzy Logic	45
3.3 Desain Kendali tekanan positif ruangan berbasis fuzzy logic	46
3.3.1 Proses Fuzzyfikasi	47
3.3.2 Fuzzy Logic Untuk Sistem Kendali Tekanan Positif R. Operasi	49
3.3.3 Turunan Input Sensor Tekanan	73
3.4 PID	87
3.5 Pelaksanaan	89
3.6 Pengujian dan Analisa.....	89
BAB IV HASIL DAN PENGUJIAN.....	91
4.1 Pengujian Alat	91
4.1.1 Sensor Tekanan.....	91
4.1.2 Sistem Input.....	92
4.1.3 Sistem Penampil Data	95
4.1.4 Sistem Blower dan Exhaust.....	96
4.2 Analisis Metode	96
4.2.1 Fuzzy Logic Controller	96
4.2.2 PID Controller	110
4.2.2.1 Tuning system PID	110
4.2.2.2 Hasil analisis respon PID	113

BAB V: KESIMPULAN DAN SARAN	116
5.1 Kesimpulan.....	116
5.2 Saran.....	119
DAFTAR PUSTAKA	117
LAMPIRAN	120



DAFTAR GAMBAR

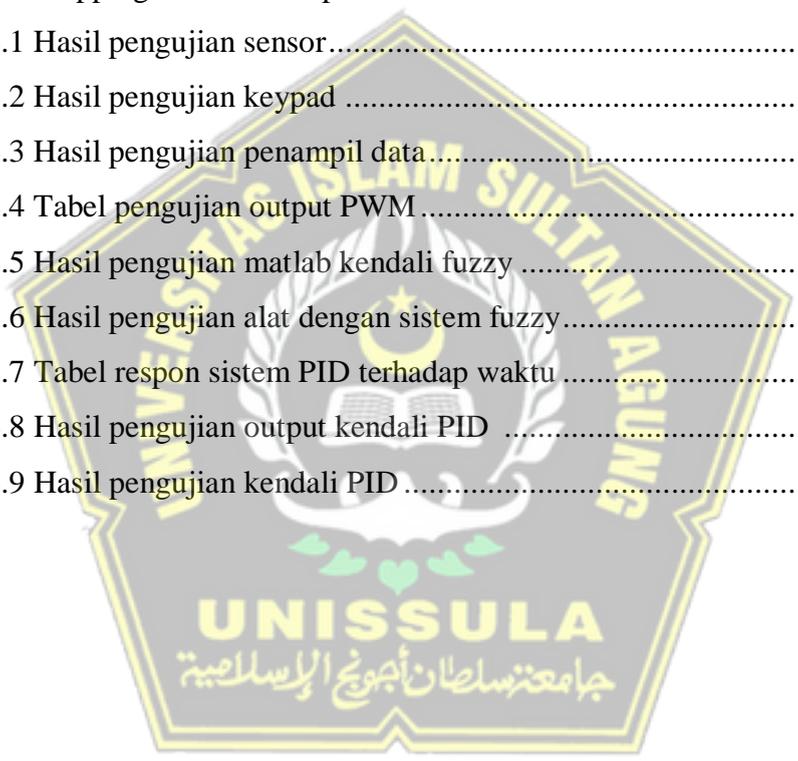
Gambar 2.1. Ruang Bedah.....	6
Gambar 2.2.Suhu dan kelembaban ruangan	7
Gambar 2.3. Proses Fuzzification.....	11
Gambar 2.4. Fungsi Keanggotaan pada masukan Error.....	12
Gambar 2.5. Fungsi Keanggotaan pada masukan DError.....	12
Gambar 2.6. Fungsi Keanggotaan pada Output.....	12
Gambar 2.7. Proses Rule Evaluation	13
Gambar 2.8. Proses Defuzzification	14
Gambar 2.9. Struktur dasar sistem control logika Fuzzy.....	15
Gambar 2.10. Blok Diagram Kontrol Logika Fuzzy	15
Gambar 2.11. Kendali PID	16
Gambar 2.12. PersamaanKendali PID	16
Gambar 2.13 Kurva dengan lonjakan 25%	17
Gambar 2.14. Respon tangga satuan sistem.....	17
Gambar 2.15. Kurva Respons berbentuk S	18
Gambar 2.16. Sistem untaian tertutup dengan alat kontrol proporsional	19
Gambar 2.17. Kurva respon sustain oscillation.....	20
Gambar 2.18. Modul Mikrokontroler Arduino.....	24
Gambar 2.19. Kipas Angin/ Fan.....	29
Gambar 2.20. Kipas Angin/ Fan.....	30
Gambar 2.21 LCD.....	31
Gambar 2.22. <i>Power Supply</i>	32
Gambar 2.23 Modul L298	33
Gambar 2.24 Sensor mps20n0040d-d.....	34
Gambar 2.25 Blok Diagram Sensor mps20n0040d-d.....	34
Gambar 2.26 Hx711	35
Gambar 3.1.Alur Kerja Penelitian	36
Gambar 3.2. Desain Ruang Operasi.....	38
Gambar 3.3. Bagan Aliran Udara	39
Gambar 3.4. Aliran Udara di Dalam Ruang.....	39
Gambar 3.5. Diagram Blok Keseluruhan Sistem Fuzzi	41

Gambar 3.6. Diagram Blok Keseluruhan Sistem Fuzzi	41
Gambar 3.7. Diagram Blok Sistem Kendali Fuzzy	41
Gambar 3.8. Diagram Blok Sistem Kendali PID.....	42
Gambar 3.9. Proses Pemrograman Mikrokontroler	43
Gambar 3.10. Wiring Diagram.....	44
Gambar 3.11. Flowchart sistem Fuzzy	46
Gambar 3.12. Desain model Fuzzy Logic.....	47
Gambar 3.13 Fungsi keanggotaan Sensor Tekanan.....	48
Gambar 3.14. Derajat Keanggotaan input simpangan negative besar	49
Gambar 3.15 Derajat Keanggotaan simpangan negative sedang	50
Gambar 3.16 Derajat Keanggotaan simpangan negative kecil	50
Gambar 3.17 Derajat Keanggotaan simpangan kondisi sesuai	51
Gambar 3.18 Derajat Keanggotaan simpangan positive kecil	51
Gambar 3.19 Derajat Keanggotaan simpangan positive sedang	52
Gambar 3.20 Derajat Keanggotaan simpangan positive besar	52
Gambar 3.21 Ruas Trapesium input simpangan negative besar.....	53
Gambar 3.22 Flowchart Fungsi Keanggotaan tekanan negative besar	54
Gambar 3.23 Ruas segitiga input simpangan negative sedang	56
Gambar 3.24 Flowchart Fungsi Keanggotaan tekanan negative sedang	57
Gambar 3.25 Ruas segitiga input simpangan negative kecil.....	59
Gambar 3.26 Flowchart Fungsi Keanggotaan tekanan negative kecil.....	60
Gambar 3.27 Ruas segitiga input simpangan kondisi sesuai	62
Gambar 3.28 Flowchart Fungsi Keanggotaan tekanan sesuai.....	63
Gambar 3.29 Ruas segitiga input simpangan positive kecil.....	65
Gambar 3.30 Flowchart Fungsi Keanggotaan tekanan positive kecil	66
Gambar 3.31 Ruas segitiga input simpangan positive sedang	68
Gambar 3.32 Flowchart Fungsi Keanggotaan tekanan positive sedang	69
Gambar 3.33 Ruas segitiga input simpangan positive besar	71
Gambar 3.34 Flowchart Fungsi Keanggotaan tekanan positive besar	72
Gambar 3.35 Himpunan turunan sensor tekanan.....	73
Gambar 3.36 Pemodelan Rules Fuzzy Logic	79
Gambar 3.37 Membership Output Blower	80

Gambar 3.38 Membership Output Exhaust.....	83
Gambar 3.39 Flowchart Sederhana Sistem PID	87
Gambar 3.40 Algoritma Sederhana Sistem PID.....	88
Gambar 3.41 Pemodelan PID pada kendali tekanan positive ruang.....	89
Gambar 4.1 Program pengetesan input keypad	93
Gambar 4.2 Rancangan sistem Fuzzy pada kendali tekanan positive ruang.....	97
Gambar 4.3 Output Fuzzy blower berhenti berputar	97
Gambar 4.4 Output Fuzzy blower berhenti pelan.....	98
Gambar 4.5 Output Fuzzy blower berhenti sedang	98
Gambar 4.6 Output Fuzzy blower berhenti cepat.....	99
Gambar 4.7 Output Fuzzy exhaust berhenti berputar	99
Gambar 4.8 Output Fuzzy exhaust berputar pelan	100
Gambar 4.9 Output Fuzzy exhaust berputar sedang	100
Gambar 4.10 Output Fuzzy exhaust berputar cepat.....	101
Gambar 4.11 Pemodelan ketika input bernilai sama dengan set poin	101
Gambar 4.12 Pemodelan ketika input bernilai 0	102
Gambar 4.13 Pemodelan ketika input bernilai 8	102
Gambar 4.14 Pemodelan ketika input bernilai 9	103
Gambar 4.15 Pemodelan ketika input bernilai 11	103
Gambar 4.16 Pemodelan ketika input bernilai 12	104
Gambar 4.17 Pemodelan ketika input bernilai 13	104
Gambar 4.18 Grafik nilai PWM Blower terhadap nilai input	105
Gambar 4.19 Grafik nilai PWM Exhaust terhadap nilai input	106
Gambar 4.20 Sinyal PWM dengan nilai PWM 39	107
Gambar 4.21 Sinyal PWM dengan nilai PWM 102	107
Gambar 4.22 Sinyal PWM dengan nilai PWM 153	107
Gambar 4.23 Sinyal PWM dengan nilai PWM 216	107
Gambar 4.24 Kurva respon dengan nilai P:2; I:5; dan D:1	110
Gambar 4.25 Kurva respon dengan nilai P:4; I:7; dan D:1	111
Gambar 4.26 Kurva respon dengan nilai P:6; I:5; dan D:1	111
Gambar 4.27 Kurva respon dengan nilai P:3; I:8; dan D:3	112
Gambar 4.28 Kurva respon dengan nilai P:2; I:10; dan D:1	112

DAFTAR TABEL

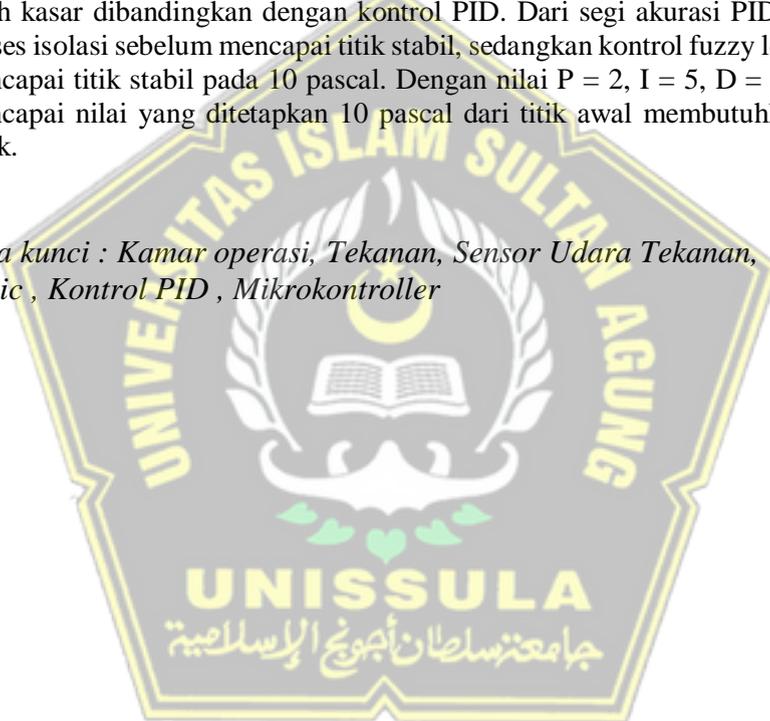
Tabel 2.1 Rules Kontrol Fuzzy.....	14
Tabel 2.2 Penalaan paramater PID dengan metode kurva reaksi	19
Tabel 2.3 Penalaan paramater PID dengan metode kurva reaksi	20
Tabel 3.1 Pin input output mikrokontroler	43
Tabel 3.2 Penalaran sistem input	73
Tabel 3.3 Penalaran sistem output blower	74
Tabel 3.4 Penalaran sistem output exhaust	74
Tabel 3.5 Mapping Nilai Proses pada Sistem PID	88
Tabel 4.1 Hasil pengujian sensor.....	91
Tabel 4.2 Hasil pengujian keypad	93
Tabel 4.3 Hasil pengujian penampil data.....	95
Tabel 4.4 Tabel pengujian output PWM.....	96
Tabel 4.5 Hasil pengujian matlab kendali fuzzy	108
Tabel 4.6 Hasil pengujian alat dengan sistem fuzzy.....	109
Tabel 4.7 Tabel respon sistem PID terhadap waktu.....	113
Tabel 4.8 Hasil pengujian output kendali PID	113
Tabel 4.9 Hasil pengujian kendali PID	114



ABSTRAK

Rumah sakit adalah bangunan yang penuh dengan sumber penyakit dan sumber infeksi, Oleh karena itu harus diperhatikan dan dikendalikan kemungkinan bahwa penyebaran infeksi terjadi melalui udara. Hal pertama yang harus diperhatikan adalah penerapan sistem tata kelola udara di gedung rumah sakit harus benar, terutama ruangan khusus seperti ruang operasi yang tekanannya harus dijaga agar selalu positif dan stabil. Dalam tulisan ini dua model pendekatan pertama dirancang dengan menggunakan sistem pengendali logika Fuzzy dan pendekatan dirancang menggunakan sistem kendali PID. Pengembangan sistem didukung oleh sensor tekanan mps20n0040d-d, mikrokontroler serta lcd karakter 20x4. Motor DC sebagai penggerak blower dan exhouser. Berdasarkan pengamatan terhadap 2 jenis kontrol yang diterapkan memiliki karakter yang berbeda dimana respon sistem logika fuzzy lebih kasar dibandingkan dengan kontrol PID. Dari segi akurasi PID membutuhkan proses isolasi sebelum mencapai titik stabil, sedangkan kontrol fuzzy lebih tepat untuk mencapai titik stabil pada 10 pascal. Dengan nilai $P = 2$, $I = 5$, $D = 1$, respon untuk mencapai nilai yang ditetapkan 10 pascal dari titik awal membutuhkan waktu 3,29 detik.

Kata kunci : Kamar operasi, Tekanan, Sensor Udara Tekanan, Kontrol Fuzzy Logic , Kontrol PID , Mikrokontroler



ABSTRACT

Hospitals are buildings full of sources of disease and resources infection. Therefore, everything must be considered and controlled the possibility that the spread of infection occurs mainly through the air. The first thing to pay attention to is the system of governance system application the air in the hospital building must be correct, especially special rooms such as the operating room where the pressure must be maintained so that it is always positive and stable. In this works, two approaches model has been proposed. First approach designed using fuzzy logic controller system and approach is designed using the PID control system. System development supported by mps20n0040d-d pressure sensor, microcontroller, 20x4 character LCD, as well. DC motor as the driver of the blower and exhaust. Based on observations of the 2 types of controls that are applied, they have different characters where the fuzzy logic system response is coarser than the PID control. In terms of accuracy, PID requires an oscillation process before reaching the stable point, while Fuzzy control is more precise in getting to the stable point at 10 Pascal. With a value of $P = 2$, $I = 5$, $d = 1$, the response to reach the set value of 10 pascals from the starting point takes 3.29 seconds.

Keywords: Operating Room, Pressure, Air Pressure Sensor, Fuzzy Logic Control, PID Controller, Microcontrol



BAB I

PENDAHULUAN

1.1. Latar Belakang

Dalam Undang-Undang No. 44 Tahun 2009 tentang Rumah Sakit pasal 10 ayat (2) menyebutkan, bangunan rumah sakit sebagaimana dimaksud pada ayat (1) bahwa persyaratan minimal bangunan rumah sakit diantaranya adalah harus memiliki ruang operasi. Bangunan rumah sakit khususnya ruang steril yang beresiko tinggi seperti ruang operasi dan ruang isolasi harus mendapatkan perhatian sangat khusus mulai dari perencanaan, pembangunan, pengoperasian dan pemeliharannya dikarenakan mempunyai standar keamanan yang tinggi terutama pada prasarana instalasi listrik medis, safety building, dan instalasi tata udara (heating, ventilation and air conditioning/HVAC) [1] sehingga bangunan rumah sakit mempunyai spesifikasi yang sangat berbeda dan tidak ditemui di bangunan gedung lain pada umumnya. Peraturan pedoman teknis prasarana sistem tata udara pada bangunan rumah sakit bahwa sistem tata udara di rumah sakit berfungsi untuk pengaturan temperatur, kelembaban udara relatif, sirkulasi udara, kebersihan udara dan tekanan udara ruangan dalam rangka mencegah penyebaran, berkembang biak dan tumbuh suburnya mikroorganisme, terutama di ruangan-ruangan khusus yang memiliki resiko tinggi.

Rumah sakit adalah tempat dimana orang yang sakit (dengan bermacam-macam penyakit) didiagnosa, diterapi, dirawat, dan dilakukan tindakan medik dimulai dari pemeriksaan biasa, pemeriksaan laboratorium, pemeriksaan dengan sinar radioaktif, pemeriksaan dengan ultrasonic, tindakan pembedahan ringan, tindakan pembedahan berat dan sebagainya. Dengan kondisi tersebut, maka salah satu faktor yang membedakan rumah sakit dengan bangunan gedung biasa terletak pada peralatan yang digunakan dan instalasi tata udaranya, jam kerja yang 24 jam sehari, 7 hari seminggu, berarti terus menerus membutuhkan pengkondisian yang dilakukan oleh sistem tata udara, mengingat rumah sakit bisa dikatakan sebagai pusat sumber dari berbagai jenis mikroorganisme yang bisa menimbulkan banyak masalah kesehatan baik kepada petugas, perawat, dokter serta pasiennya yang

berada di rumah sakit tersebut[2]. Untuk mencegah berkembang biak dan tumbuhsuburnya mikroorganismenya tersebut, terutama untuk ruangan-ruangan khusus seperti di ruang operasi dan ruang isolasi dimana diperlukan pengaturan:[3]

1. Temperatur ($20^{\circ}\text{C} \sim 24^{\circ}\text{C}$)
2. Kelembaban udara relatif (50%~60%)
3. Sistem filtrasi udara menggunakan (HEPA Filter) dan ventilasinya
4. Tekanan ruangan kamar operasi positif ($10 \text{ Pa} \sim 15 \text{ Pa}$)
5. Jumlah maksimal partikel debu $35.200 \text{ partikel}/\text{m}^3$ diameter $0,5\mu\text{m}$
6. Distribusi udara didalam ruangan (Air Change Hour 20 ACH).

Berdasarkan penelitian angka infeksi nosokomial terus meningkat sekitar 9,0% di rumah sakit seluruh dunia. Hasil survey point prevalensi dari 11 rumah sakit di DKI Jakarta yang dilakukan oleh Perdalim Jaya dan Rumah Sakit Penyakit Infeksi Prof. Dr. Sulianti Saroso Jakarta pada tahun 2003 didapatkan angka infeksi nosokomial untuk Infeksi Luka Operasi (ILO) 18,9% (Depkes RI, 2008). Prinsip konsistensi dan kontinuitas menjadi masalah pokok pada sistem tata udara yang diperlukan pada ruang-ruang steril, hal ini mengingat bahwa di kamar operasi rawan terjadinya infeksi karena ada tindakan-tindakan medis berupa pembedahan tubuh pasien. Adapun cara pemantauan ruang operasi agar tetap dalam kondisi steril adalah dengan mengukur tekanan udara pada ruang operasi menggunakan alat barometer dan harus menunjukkan nilai tekanan positif tujuannya agar udara luar tidak bisa masuk ke area kamar bedah, pengukuran suhu dengan alat thermometer agar kondisi ruangan nyaman untuk tindakan pembedahan. Pengukuran nilai kelembaban relatif (RH) menggunakan hygrometer tujuannya untuk mengurangi perkembangbiakan serta penyebaran jamur dan bakteri, untuk mengetahui tingkat kebersihan udara bisa diukur dengan alat airborne particular counter sehingga akan diketahui standar baku mutu partikular udara yang di iijinkan. [4]

1.1. Rumusan Masalah

Rumusan masalah yang didapat dari latar belakang adalah sebagai berikut :

1. Bagaimana merancang dan membuat system otomasi tekanan udara di ruang operasi agar selalu terjaga tingkat steril-nya.

2. Bagaimana data hasil pengukuran sensor digunakan sebagai decision support system pada kamar operasi, dengan logika Fuzzy dan PID?

1.2. Tujuan Penelitian

Adapun tujuan yang ingin dicapai adalah :

1. Membuat sistem otomasi tekanan udara ruang operasi menggunakan kendali logika Fuzzy.
2. Membuat sistem otomasi tekanan udara ruang operasi menggunakan kendali PID.
3. Membandingkan kinerja kendali Fuzzy dan kendali PID pada sistem otomasi tekanan udara ruang operasi.

1.3. Batasan Penelitian

Berikut batasan-batasan yang dilakukan terhadap penelitian ini :

1. Pembuatan prototipe hanya di tekankan pada pengaturan tekanan udara,
2. Menggunakan sensor mps20n0040d-d
3. Sistem mikrokontroler menggunakan Arduino- Uno

1.4. Manfaat Penelitian

Adapun manfaat yang diharapkan dalam penelitian ini yaitu :

1. Memberikan kepastian dan akurasi tekanan udara sehingga ruang operasi tidak terkontaminasi oleh bakteri dan virus dari luar.
2. Meningkatkan keamanan, kenyamanan petugas medis dan pasien.

1.5. Keaslian Penelitian

Berdasarkan penelusuran serta tinjauan pustaka yang telah dilakukan, belum ditemukan tulisan sejenis yang membahas tentang perlunya mempertahankan tekanan udara di ruang operasi.

Ada beberapa penelitian yang berkaitan, yaitu :

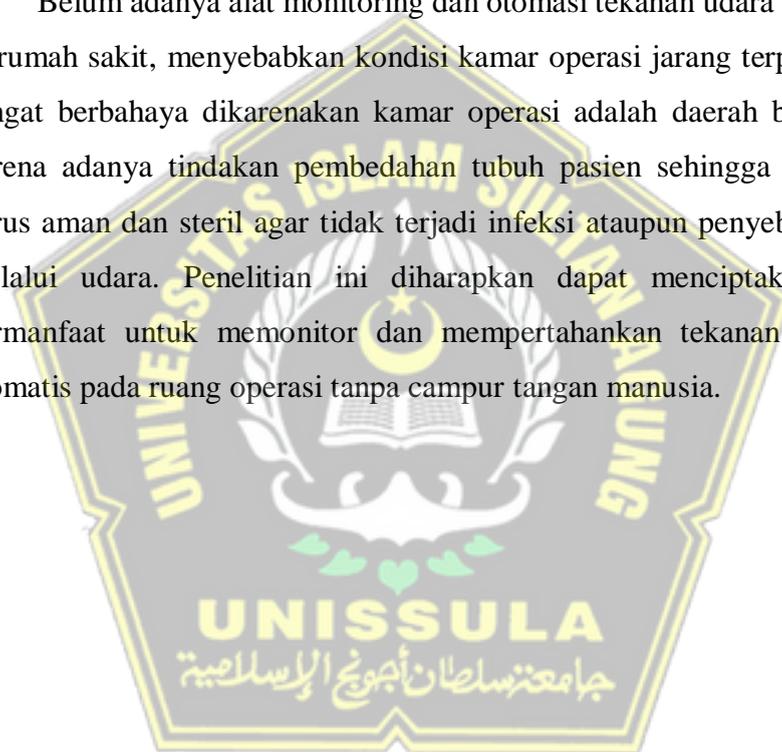
1. Penelitian Puput Dani Prasetyoadi tentang , Sistem Kendali Suhu dan Kelembaban Ruang Operasi Menggunakan Mikrokontroler Arduino dan Metode Logika Fuzzy, untuk sensor suhu dan kelembaban menggunakan DHT 11 diproses oleh mikrokontroler Arduino uno dan ditampilkan di LCD Display jika nilai diluar range maka akan terdengar bunyi alarm[2].
2. Faizal Wahab, Arif Sumardiono, Adnan Rafi Al tahtawi, Agus Faisal Aziz Mulyari Desain dan Purwarupa Fuzzy Logic Control Untuk

Pengendalian Suhu Ruangan. Dalam rancangan ini menggunakan dua sensor yang berbeda yaitu lm 35 dan ldr dengan tingkat kesalahan berturut – turut 1,31% dan 4,06 % [5].

3. Yudha Dwi Aryandhi, Mozart Wilson Talakua, Penerapan Inferensi Fuzzy Untuk Pengendalian Suhu Ruangan Secara Otomatis Pada Air Conditioner (AC). Penelitian ini menitik beratkan pada suhu ideal untuk penyejuk ruangan dengan menghitung luas ruangan, jumlah orang dan besarnya kapasitas AC [6].

1.6. Kontribusi Tesis

Belum adanya alat monitoring dan otomasi tekanan udara kamar operasi di rumah sakit, menyebabkan kondisi kamar operasi jarang terpantau. Hal ini sangat berbahaya dikarenakan kamar operasi adalah daerah beresiko tinggi karena adanya tindakan pembedahan tubuh pasien sehingga kondisi ruang harus aman dan steril agar tidak terjadi infeksi ataupun penyebaran penyakit melalui udara. Penelitian ini diharapkan dapat menciptakan alat yang bermanfaat untuk memonitor dan mempertahankan tekanan udara secara otomatis pada ruang operasi tanpa campur tangan manusia.



BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1. Tinjauan Pustaka

Setelah peneliti melakukan telaah terhadap beberapa penelitian ada beberapa yang memiliki keterkaitan dengan penelitian yang peneliti lakukan. Penelitian pertama yang berhasil peneliti temukan adalah penelitian yang dilakukan oleh S. K. Putra, E. K. Simanjuntak, W. Hendradjit, and S. Hadisupadmo tahun 2015. Pada penelitian ini dibuat suatu perancangan sistem pengaturan mode pemilihan yaitu mode persiapan, mode operasi dan mode sanitasi sistem ini tidak mengatur otomatisasi tekanan udara ketika operator keluar masuk kamar operasi[7].

2.2. Landasan Teori

2.2.1. Ruang Operasi

Setiap bangunan ruang operasi rumah sakit merupakan tempat untuk melakukan tindakan pembedahan secara efektif ataupun akut yang membutuhkan kondisi steril dan kondisi khusus lainnya Berikut ini adalah bagian-bagian penting yang ada pada kamar bedah: (1) Daerah publik adalah daerah yang boleh dimasuki oleh semua orang tanpa syarat khusus, misalnya: kamar tunggu, gang, emperan depan komplek kamar operasi, (2) Daerah semi publik adalah daerah yang bias dimasuki oleh orang-orang tertentu saja seperti petugas dan biasanya diberi tulisan “DILARANG MASUK SELAIN PETUGAS” dan sudah ada pembatasan tentang jenis pakaian yang dikenakan oleh petugas (pakaian khusus kamar operasi) serta penggunaan alas kaki khusus didalam, (3) Daerah Aseptik adalah Daerah kamar bedah sendiri yang hanya bias dimasuki oleh orang yang langsung ada hubungan dengan kegiatan pembedahan. Umumnya daerah yang harus dijaga kesterilisasiannya. Daerah aseptik di bagi menjadi 3 bagian, yaitu : (1) Daerah Aseptik yaitu lapangan operasi, daerah tempat dilakukannya pembedahan, (2) Daerah Aseptik 1 yaitu daerah memakai gaun operasi, tempat duk/kain steril, tempat instrument dan tempat perawat instrument mengatur dan mempersiapkan alat, (3) Daerah Aseptik 2 yaitu tempat mencuci tangan, koridor penderita masuk, daerah sekitar ahli anesthesia [8].



Gambar 2.1. Ruang Bedah [9]

2.2.2 Bagian-Bagian kamar operasi

Kamar operasi terdiri dari beberapa ruang baik itu didalam kamar operasi maupun dilingkungan kamar operasi, antara lain : Kamar bedah, Kamar untuk mencuci tangan, Kamar untuk gudang alat-alat instrument, Kamar untuk sterilisasi, Kamar untuk ganti pakaian, Kamar laboratorium, Kamar arsip, Kamar Pulih Sadar (Recovery Room), Kamar gips, Kamar Istirahat, Kamar mandi (WC) dan Spoelhok (Tempat cuci alat), Kantor, Gudang, Kamar tunggu, Ruang Sterilisasi. Persyaratan Kamar Operasi antara lain : Kamar operasi yang baik harus memenuhi beberapa persyaratan sebagai berikut: Letak, letak kamar operasi berada ditengah-tengah rumah sakit, berdekatan dengan instalasi rawat darurat ICU dan Unit radiologi [9].

2.2.3 Bentuk dan ukuran

1. Bentuk
 - a. Kamar operasi tidak bersudut tajam. lantai, dinding. langit-langit berbentuk lengkung dan warna tidak mencolok
 - b. Lantai dan 2/3 dinding bagian bawah harus terbuat dari bahan yang keras, rata, kedapair, dan mudah dibersihkan dan tidak menampung debu.
2. Ukuran
 - a. Kamar operasi berukuran : 5,2 m x 5,6 m (29,1m²)
 - b. Kamar operasi yang nyaman diperlukan kira-kira diperlukan luas 40 m².

c. Kamar operasi untuk operasi besar diperlukan luas minimal 56 m² (7,2 m x 7,8 m).

3. Sistem Penerangan

Sistem Penerangan di dalam kamar operasi harus memakai lampu pijar putih dan mudah dibersihkan. sedangkan lampu operasi memiliki persyaratan khusus, yaitu arah dan fokusnya dapat diatur, tidak menimbulkan panas, cahayanya terang dan tidak menyilaukan serta tidak menimbulkan bayangan. pencahayaan 300-500 lux, meja operasi 10.000-20.000 lux [9].

4. Sistem Ventilasi

Sistem ventilasi di kamar bedah sebaiknya memakai sistem pengatur suhu sentral (AC Sentral) dan dapat diatur dengan alat kontrol yang memakai filter (Ultra Clean Laminar Air flow) dimana udara dipompakan kedalam kamar operasi dan udara dikamar operasi dihisap keluar.

2.2.4 Suhu dan kelembaban

Suhu dikamar operasi di daerah tropis sekitar 19⁰ – 22⁰ C, sedangkan di daerah sekitar 20⁰ – 24⁰ C dengan kelembaban 55% (50-60%).



Gambar 2.2.Suhu dan kelembaban ruangan [2]

2.2.5 Logika Fuzzy

Fuzzy set pertama kali diperkenalkan oleh Prof. Lotfi Zadeh pada tahun 1965, dalam papernya dipaparkan ide dasar fuzzy set yang meliputi inclusion, union, intersection, complement, relation, dan convexity. Ide tersebut terus dimatangkan oleh Zadeh dalam beberapa papernya [10].

Pelopori aplikasi fuzzy set dalam bidang kontrol, yang merupakan aplikasi pertama dan utama dari fuzzy set adalah Prof. Ebrahim Mamdani dan kawan-kawan dari Queen Mary College London. Aplikasinya meliputi proses pada tangki pencampur dan mesin uap, semuanya dalam skala lab. Penerapan kontrol logika fuzzy secara nyata di industri banyak dipelopori para ahli dari Jepang. Aplikasi fuzzy set hampir tak terbatas, misalnya kontrol proses, proses produksi, robotika, manajemen skala besar, teknik sipil, kimia, transportasi, kedokteran, maupun, ekonomi [11].

Perkembangan teori fuzzy dan penerapannya telah berlangsung sangat cepat. Banyaknya peminat dalam bidang ini tercermin dengan maraknya penerbitan jurnal- jurnal internasional dan seminar-seminar internasional yang sering dilakukan oleh institusi seperti IEEE dan IFAC, misalnya Suatu himpunan fuzzy A dalam semesta pembicaraan U didefinisikan sebagai kumpulan pasangan u dan fungsi keanggotaan $\mu_A(x)$. Derajat keanggotaan dari $\mu_A(x)$ mempunyai nilai interval $[0,1]$ untuk tiap nilai x dan U . Nilai fungsi keanggotaan menunjukkan tingkat keanggotaan elemen x dalam A , dapat dinotasikan sebagai [11]:

$$\mu_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases} \quad (2.1)$$

Secara umum himpunan fuzzy (fuzzy sets) dinotasikan sebagai berikut [11]:

$$A = \{(x, \mu_A(x)) \mid x \in U\} \quad (2.2)$$

Jika U kontinyu (misalnya $U = R$), A dapat dinyatakan sebagai [11]:

$$A = \int_u \mu_A(x) / x \quad (2.3)$$

Sedangkan bila U diskrit, maka A biasanya ditulis sebagai [11]:

$$A = \sum \mu_A(x) / x \quad (2.4)$$

atau

$$A = \mu_1(x) / x_1 + x_2 + \dots + \mu_n(x) / x_n$$

Notasi – notasi ‘+’ dan ‘/’ di atas bukan merupakan operasi aritmatik penjumlahan dan pembagian, melainkan menyatakan operator gabungan dan penghubung nilai elemen fuzzy dan derajat keanggotaan [11].

2.2.5.1 Operasi Dasar Fuzzy logic

Operasi – operasi logika harus ada pada himpunan fuzzy dan digunakan dalam *fuzzy reasoning* dan *fuzzy logic control*. Sedangkan pada fuzzy logic, operator tersebut harus ada pada fungsi keanggotaan yang sedang dioperasikan [11].

2.2.5.2 Complement (Not)

Komplemen dari A adalah *fuzzy sets* A dalam U yang fungsi keanggotaannya ditentukan sebagai berikut :

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad (2.5)$$

2.2.5.3 Union (Or)

Berdasarkan pada dua fuzzy sets A dan B pada semesta pembicaraan yang sama U , dinotasikan sebagai \cup . Sedangkan fungsi keanggotaannya ditanyakan sebagai [11]:

$$\mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)] \quad (2.6)$$

Penggunaan *max* dikarenakan elemen x harus berada pada salah satu himpunan (himpunan A dan himpunan B), sehingga ditampilkan derajat keanggotaan yang terbesar [11].

2.2.5.4 Intersection (And)

Sama dengan operator union. Operator intersection dapat dinotifikasikan sebagai $A \cap B$. Alasan menggunakan *min* adalah elemen x harus berada pada himpunan (himpunan A dan himpunan B) tersebut secara simultan dan deret keanggotaan kecil dinotasikan sebagai berikut [11]:

$$\mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)] \quad (2.7)$$

2.2.5.5 Operasi Fuzzy Controller

Ada beberapa tahap yang harus dilakukan dalam operasi controller fuzzy, yaitu:

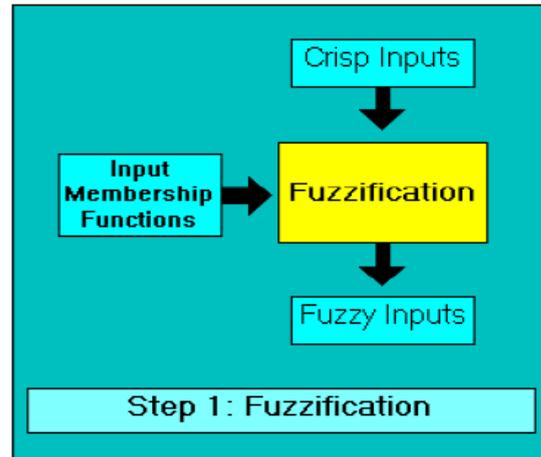
a. Fuzzifikasi

Variabel-variabel input dalam sistem control fuzzy umumnya dipetakan ke dalam himpunan fuzzy. Proses untuk mengkonversikan nilai input tegas (crisp) menjadi nilai fuzzy disebut dengan fuzzifikasi. Sebuah sistem control fuzzy juga memiliki beberapa tipe switch atau ON-OFF, Input yang menyatakan nilai input analog dan input switch tersebut selalu memiliki nilai kebenaran 1 atau 0 [12].

Skema yang cocok dengan hal tersebut adalah fungsi logic sederhana yang dapat memiliki suatu nilai atau nilai yang lainnya. Pemetaan variable-variabel ini input ke dalam fungsi keanggotaan dan nilai kebenaran, kemudian mikrokontroler membuat keputusan untuk mengambil tindakan berdasarkan sekumpulan aturan (rule base) [13].

Fuzzifikasi adalah proses perubahan *crisp input* (masukan yang berupa non-fuzzy) menjadi *fuzzy input*. Dan merupakan proses awal dan harus dilakukan dalam operasi *fuzzy controller* [13].

Fuzzifikasi dan fungsi keanggotaan mempunyai kaitan yang erat, karena fuzzifikasi ditentukan oleh fungsi keanggotaan. Kemudian pada proses fuzzifikasi ditentukan pula tingkat kuantisasi pada semesta pembicaraan. Sedangkan derajat keanggotaan dari *fuzzy input* berada pada interval [0,1] [13].



Gambar 2.3. Proses Fuzzification [13]

b. Fungsi Keanggotaan (Membership Function)

Pada proses fuzzifikasi dibutuhkan suatu besaran analog yang akan dijadikan masukan atau biasa disebut crisp input. Crisp input disini bernilai 255 dimana nantinya akan dibagi menjadi 2 yaitu crisp input yang bernilai positif terletak di sebelah kanan label Z (Zero) dan crisp input yang bernilai negatif terletak di sebelah kiri label Z (Zero). Lalu nilai crisp input ini yang akan dimasukkan pada batas scope/domain sehingga input tersebut dapat dinyatakan dengan label dari fungsi keanggotaan. Fungsi keanggotaan ini biasa disebut dengan fungsi keanggotaan input (membership function input), dari fungsi keanggotaan ini kita dapat mengetahui derajat keanggotaannya [14].

Fungsi keanggotaan yang direncanakan untuk masukan Error dan dError terdiri dari 7 label yaitu NB (*Negative Big*), NM, (*Negative Medium*) NS (*Negative Small*), Z (*Zero*), PS (*Positive Small*), PM (*Positive Medium*), PB (*Positive Big*) Dimana Error dan dError menggunakan rumus [14] :

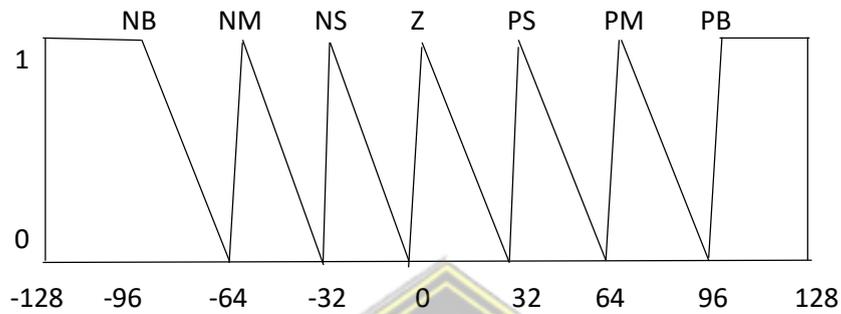
$$\text{Error} = \text{SP} - \text{PV} \dots \dots \quad (2.8)$$

Dimana SP = Set Point

PV=Point Value

$$\text{dError} = \text{Error}(n) - \text{Error}(n-1) \dots \dots \quad (2.9)$$

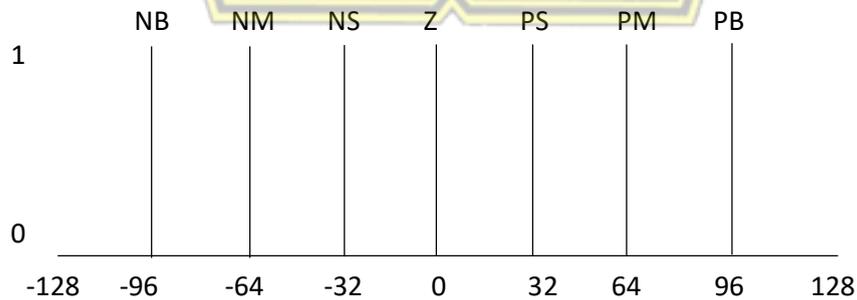
Sedangkan fungsi keanggotaan untuk keluaran yang berupa garis vertikal / (I / singleton) mempunyai label yang sama dengan masukan Error dan dError. Fungsi keanggotaan tersebut dapat dilihat pada gambar di bawah ini [14].



Gambar 2.4. Fungsi Keanggotaan pada masukan Error [14]



Gambar 2.5. Fungsi Keanggotaan pada masukan dError [14]



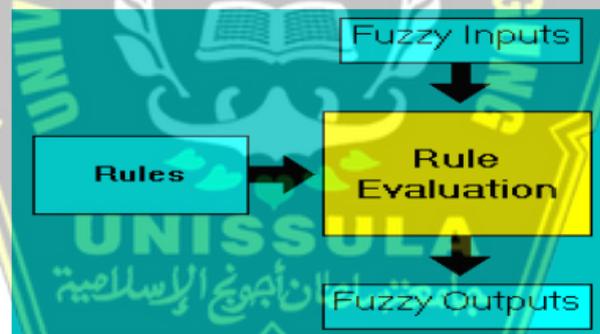
Gambar 2.6. Fungsi Keanggotaan pada Output [14]

c. Rule Evaluation (inference)

Masukan yang diberikan pada *rule evaluation* adalah *fuzzy input* yang merupakan hasil dari fuzzifikasi. Dan keluaran dari *rule evaluation* adalah *fuzzy output*. [15]

Dalam *rule evaluation* ada beberapa langkah yang harus dilakukan, yaitu :

1. Membuat *rule base* (aturan dasar) Format adalah sebagai berikut “IF *antecedent1* AND *antecedent2* AND....THEN *consequent1* AND....”
2. Menentukan harga kebenaran dari *antecedent*
Dari crisp input dapat ditentukan derajat kebenaran dari tiap-tiap *antecedent* dengan menggunakan transformasi fuzzifikasi.
3. Mencari bobot dari seluruh *rule*, yang sebanding dengan nilai minimum dari derajat kebenaran *antecedent*.
4. Menentukan fuzzy output dari tiap-tiap *consequent label* dengan mencari nilai maksimal dari bobot *rule* pada tiap-tiap *consequent label* [14].



Gambar 2.7. Proses Rule Evaluation (InnovativeElectronics.com)

Rule yang direncanakan terdapat 49 rule, dimana dalam penyusunan rule ini menggunakan metode Mac Vi Char Whelan. Sedangkan proses inference engine nanti menggunakan metode Max-Min.

Tabel 2.1. Rules Kontrol Fuzzy [14]

		Error						
V		NB	NM	NS	Z	PS	PM	PB
dError	NB	NB	NB	NM	NM	NS	NS	Z
	NM	NB	NM	NM	NS	NS	Z	PS
	NS	NM	NM	NS	NS	Z	PS	PS
	Z	NM	NS	NS	Z	PS	PS	PM
	PS	NS	NS	Z	PS	PS	PM	PM
	PM	NS	Z	PS	PS	PM	PM	PB
	PB	Z	PS	PS	PM	PM	PB	PB

Keterangan :

NB : *Negative Big*

PB : *Positive Big*

NM : *Negative Medium*
Medium

PM : *Positive*

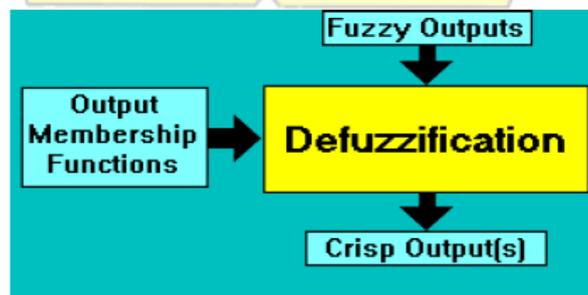
NS : *Negative Small*

PS : *Positive Small*

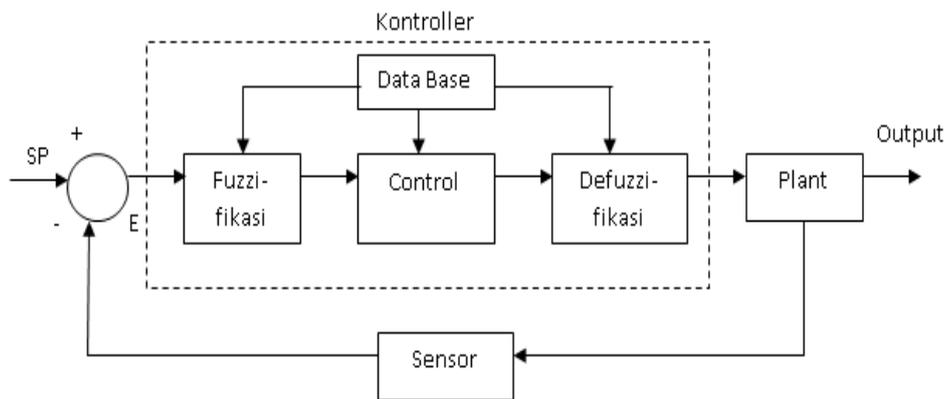
Z : *Zero*

2.3. Defuzzifikasi

Merupakan suatu proses akhir dalam operasi *fuzzy controller* yang merubah *fuzzy output* menjadi *crisp output* agar dapat dibaca oleh plant. Ada beberapa metode dalam proses defuzzifikasi, diantaranya adalah [14]:



Gambar 2.8. Proses Defuzzification [14]



Gambar 2.9. Struktur dasar sistem control logika Fuzzy [16]



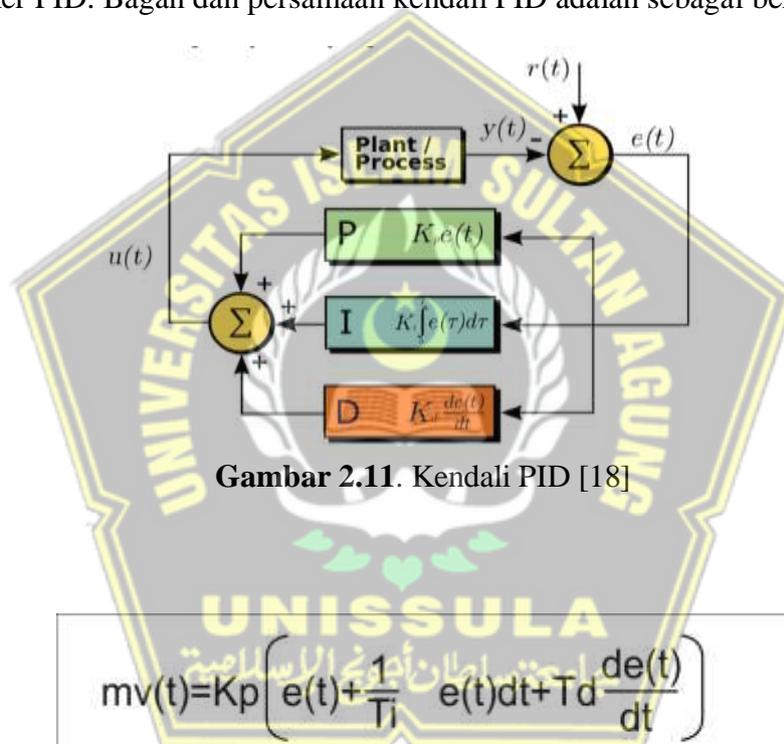
Gambar 2.10. Blok Diagram Kontrol Logika Fuzzy [16]

2.4 Kendali PID

Kendali algoritma PID merupakan kendali yang sederhana, satu persamaan, namun dapat digunakan untuk menghasilkan performa kendali yang relatif cukup baik pada berbagai proses. Fleksibilitas ini diperoleh melalui beberapa pengaturan atau penyetelan variabel, dimana nilai yang diperoleh dapat dipilih untuk memodifikasi perilaku dari sistem umpan balik.[17] Prosedur yang dipilih disebut dengan tuning (penyetelan), dan parameter yang diatur disebut penyetelan konstan. Penalaan parameter kontroler PID (Proporsional Integral Diferensial) selalu didasari atas tinjauan terhadap karakteristik yang diatur (Plant). Dengan demikian

betapapun rumitnya suatu plant, perilaku plant tersebut harus diketahui terlebih dahulu sebelum penalaan parameter PID itu dilakukan.

Proporsional (P), Integral (I) dan Derivatif (D) adalah tiga parameter utama dari kontroler PID. Nilai dari tiga parameter ini ditafsirkan dalam hal waktu, di mana 'P' tergantung pada error saat ini, 'I' pada akumulasi kesalahan sebelumnya dan 'D' adalah prediksi kesalahan masa depan, berdasarkan level saat perubahan. Dengan tuning tiga parameter dalam algoritma PID controller, kontroler dapat memberikan aksi kendali yang dirancang untuk kebutuhan proses tertentu. Nilai proporsional, integral dan derivatif dijumlahkan untuk menghitung keluaran dari kontroler PID. Bagan dan persamaan kendali PID adalah sebagai berikut [18]:



$$mv(t) = K_p \left(e(t) + \frac{1}{T_i} \int e(t) dt + T_d \frac{de(t)}{dt} \right)$$

Keterangan:

$mv(t)$ = output dari pengontrol PID atau *Manipulated Variable*

K_p = konstanta Proporsional

T_i = konstanta Integral

T_d = konstanta Derivatif

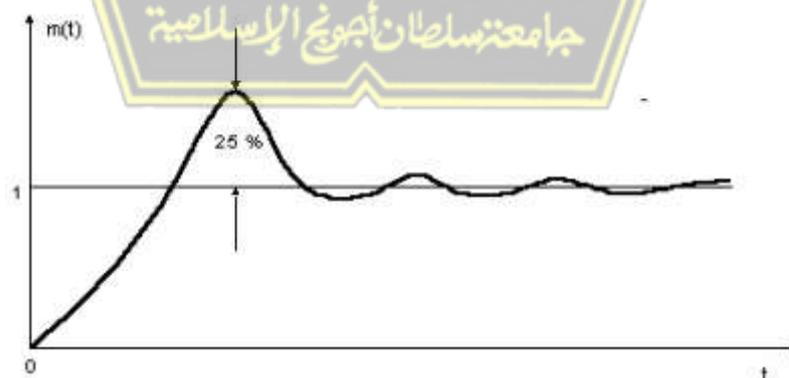
$e(t)$ = error (selisih antara set point dengan level aktual)

Gambar 2.12. Persamaan Kendali PID [18]

2.4.1 Kendali PID Menggunakan Metode Ziegler Nichols

Penalaan parameter controller PID selalu didasari atas tinjauan terhadap karakteristik yang diatur (Plant). Dengan demikian betapapun rumitnya suatu plant, perilaku plant tersebut harus diketahui terlebih dahulu sebelum penalaan parameter PID itu dilakukan. Karena penyusunan model matematik plant tidak mudah, maka dikembangkan suatu metode eksperimental. Metode ini didasarkan pada reaksi plant yang dikenai suatu perubahan. Dengan menggunakan metode itu model matematik perilaku plant tidak diperlukan lagi, karena dengan menggunakan data yang berupa kurva keluaran, penalaan kontroler PID telah dapat dilakukan. Penalaan bertujuan untuk mendapatkan kinerja sistem sesuai spesifikasi perancangan. Ogata menyatakan hal itu sebagai alat control (controller tuning)[19]. Metode pendekatan eksperimen yang telah banyak digunakan adalah Ziegler-Nichols. [20]

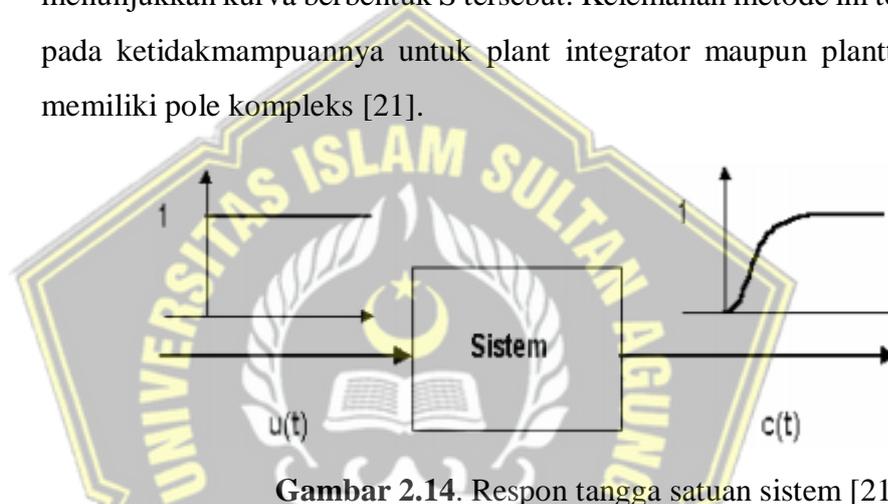
Ziegler-Nichols pertama kali memperkenalkan metodenya pada tahun 1942. Metode ini memiliki dua cara, metode osilasi dan kurva reaksi. Kedua metode ditujukan untuk menghasilkan respon sistem dengan lonjakan maksimum sebesar 25%. Gambar 2.13 memperlihatkan kurva dengan lonjakan 25%.



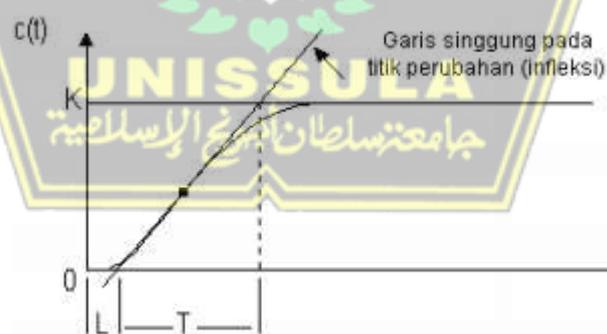
Gambar 2.13. Kurva respons tangga satuan yang memperlihatkan 25 % lonjakan maksimum [21]

a. Metode Kurva Rekasi

Metode ini didasarkan terhadap reaksi sistem untaian terbuka. Plant sebagai untaian terbuka dikenai sinyal fungsi tangga satuan (gambar 2.13). Kalau plant minimal tidak mengandung unsur integrator ataupun pole-pole kompleks, reaksi sistem akan berbentuk S. Gambar 2.14 menunjukkan kurva berbentuk S tersebut. Kelemahan metode ini terletak pada ketidakmampuannya untuk plant integrator maupun plant yang memiliki pole kompleks [21].



Gambar 2.14. Respon tangga satuan sistem [21]



Gambar 2.15. Kurva Respons berbentuk S [21]

Kurva berbentuk-s mempunyai dua konstanta, waktu mati (dead time) L dan waktu tunda T . Dari gambar 2.15 terlihat bahwa kurva reaksi berubah naik, setelah selang waktu L . Sedangkan waktu tunda menggambarkan perubahan kurva setelah mencapai 66% dari keadaan mantapnya. Pada kurva dibuat suatu garis yang bersinggungan dengan garis kurva. Garis singgung itu akan memotong dengan sumbu absis dan

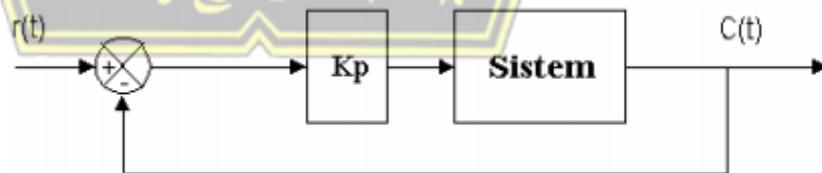
garis maksimum. Perpotongan garis singgung dengan sumbu absis merupakan ukuran waktu mati, dan perpotongan dengan garis maksimum merupakan waktu tunda yang diukur dari titik waktu L . Penalaan parameter PID didasarkan perolehan kedua konstanta itu. Zeigler dan Nichols melakukan eksperimen dan menyarankan parameter penyetelan nilai K_p , T_i , dan T_d dengan didasarkan pada kedua parameter tersebut. Tabel 1 merupakan rumusan penalaan parameter PID berdasarkan cara kurva reaksi [21].

Tabel 2.2 Penalaan parameter PID dengan metode kurva reaksi [21]

P	T/L	\sim	0
PI	$0,9 T/L$	$L/0,3$	0
PID	$1,2 T/L$	$2L$	$0,5L$

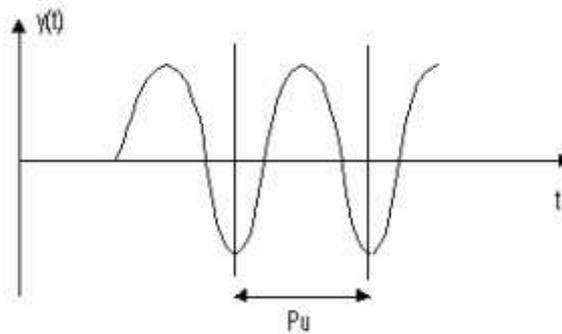
b. Metode Osilasi

Metode ini didasarkan pada reaksi sistem untaian tertutup. Plant disusun serial dengan kontroller PID. Semula parameter parameter integrator disetel tak berhingga dan parameter diferensial disetel nol ($T_i = \infty ; T_d = 0$). [22] Parameter proporsional kemudian dinaikkan bertahap. Mulai dari nol sampai mencapai harga yang mengakibatkan reaksi sistem berosilasi. Reaksi sistem harus berosilasi dengan magnitud tetap (Sustain oscillation) (Guterus, 1994, 9-9). Gambar 2.16 menunjukkan rangkaian untaian tertutup pada cara osilasi.



Gambar 2.16. Sistem untaian tertutup dengan alat kontrol proporsional [21]

Nilai penguatan proporsional pada saat sistem mencapai kondisi sustain oscillation disebut ultimate gain K_u . Periode dari sustained oscillation disebut ultimate period T_u (Perdikaris, 1991, 433). Gambar 15 menggambarkan kurva reaksi untaian tertutup ketika berosilasi [21].



Gambar 2.17. Kurva respon sustain oscillation [21]

Penalaan parameter PID didasarkan terhadap kedua konstanta hasil eksperimen, K_u dan P_u . Ziegler dan Nichols menyarankan nilai parameter K_p , T_i , dan T_d berdasarkan rumus yang diperlihatkan pada Tabel 2 [21].

Tabel 2.3. Penalaan paramater PID dengan metode kurva reaksi [21]

Type of Controller	K_p	T_i	T_d
P	$0,5K_{cr}$	∞	0
PI	$0,45K_{cr}$	$\frac{1}{1,2}P_{cr}$	0
PID	$0,6K_{cr}$	$0,5P_{cr}$	$0,125P_{cr}$

2.5. Mikrokontroler

Mikrokontroler dapat diartikan sebagai suatu komponen pengontrol dalam ukuran mikro yang tak ubahnya seperti sebuah Komputer dalam ukuran mini, yang terdiri dari sebuah *Central Processing Unit* (CPU), sebagai pengolah data, memori untuk penyimpanan data dan sarana *input/output* yang terangkum dalam sebuah chip. (Putra, Afgianto Eko. ,2006) [23].

2.5.1 Gambaran Umum

Unit pengolah pusat *Central Processing Unit* (CPU) mengendalikan seluruh operasi pada mikrokontroler. Unit ini terbagi atas dua bagian, yaitu unit pengendali {*Control Unit* ,CU} serta unit aritmatika dan logika *Arithmetic Logic Unit* (ALU).[24]

Fungsi utama unit pengendali adalah mengambil (*fetch*), mengkodekan (*decode*) dan melaksanakan urutan instruksi (*execute*) sebuah program yang tersimpan dalam memori. Unit pengendali juga menghasilkan dan mengatur sinyal pengendali yang diperlukan untuk sinkronisasi operasi, aliran dan instruksi

program serta mengendalikan aliran informasi pada bus data dan bus alamat yang kemudian dilanjutkan dengan menafsirkan dan mengatur sinyal yang terdapat pada bus pengendali. Unit aritmatika dan logika merupakan bagian yang berurusan dengan operasi aritmatika seperti penjumlahan, pengurangan, serta manipulasi data secara logika seperti operasi AND, OR dan perbandingan. Didalam CPU terdapat berbagai register, misalnya *register address counter* atau *instruction pointer*, yang bertugas menyimpan alamat data atau instruksi yang akan diambil selanjutnya. Selain itu terdapat pula *general purpose register* sebagai penyimpanan data sementara [23].

2.5.2. Bus Alamat

Bus Alamat berfungsi sebagai sejumlah lintasan saluran pengalamatan antara alat dengan sebuah komputer. Pengalamatan ini harus ditentukan terlebih dahulu untuk menghindari terjadinya kesalahan pengiriman sebuah instruksi dan terjadinya bentrok antara dua buah alat yang bekerja secara bersamaan [23].

2.5.3. Bus Data

Bus data merupakan sejumlah lintasan saluran keluar masuknya data dalam suatu mikrokontroler. Pada umumnya saluran data yang masuk sama dengan saluran data yang keluar.

2.5.4. Bus Kontrol

Bus kontrol atau bus pengendali ini berfungsi untuk menyerempakkan operasi mikrokontroler dengan operasi rangkaian luar.

2.5.5. Memori

Didalam sebuah mikrokontroler terdapat suatu memori yang berfungsi untuk menyimpan data atau program. Ada beberapa jenis memori, diantaranya adalah RAM dan ROM. Ada beberapa tingkatan memori, diantaranya adalah register internal, memori utama dan memori masal. Register internal adalah memori di dalam ALU. Waktu akses register sangat cepat, umumnya kurang dari 100 ns. Memori utama adalah memori mn, yang ada pada suatu sistem. Waktu aksesnya lebih lambat dibandingkan register internal, yaitu antara 200 sampai 1000 ns. Memori masal dipakai untuk penyimpanan berkapasitas tinggi, biasanya berbentuk disket, pita magnetic atau kaset [23].

2.5.5.1.RAM

Random Access Memory (RAM) adalah memori yang dapat di baca atau di tulis. Data dalam RAM akan terhapus (bersifat *volatile*) bila catu daya dihilangkan. Karena sifat RAM yang *volatile* ini, maka program - program mikrokontroler/ mikroprosesor tidak disimpan dalam RAM. RAM hanya digunakan untuk menyimpan data sementara, yaitu data yang tidak begitu penting bila hilang akibat aliran daya listrik terputus [23].

Ada dua teknologi yang di pakai untuk membuat RAM, yaitu RAM static dan RAM dinamik. Dalam RAM statik, satu bit informasi di simpan dalam sebuah flip-flop. RAM statik ini tidak memerlukan penyegar dan penanganannya juga tidak terlalu rumit. Isi RAM tetap tersimpan selama daya di berikan. Dua contoh RAM statik adalah 6116 dan 6264 yang masing-masing berkapasitas 2 Kbyte dan 8 Kbyte. RAM dinamik menyimpan bit informasi sebagai muatan. Sel memori elementer di buat dari kapasitas gerbang substrat transistor MOS. Keuntungan RAM dinamik adalah sel-sel memori yang lebih kecil sehingga memerlukan tempat yang lebih sempit, sehingga kapasitas RAM dinamik menjadi lebih besar di bandingkan dengan RAM statik. Contoh RAM dinamik 4116 yang berkapasitas 16384 x 1 bit.

Kerugian penggunaan RAM dinamik adalah bertambahnya kerumitan pada papan memori karena diperlukannya rangkaian untuk proses penyegaran (*refresh*). Proses penyegaran untuk kapasitor ini dilakukan setiap 1 atau 2 milidetik. Dalam mikrokontroler STC 51 dengan nomor seri 12C5A60S2 terdapat 256 x 8 bit internal RAM

2.5.5.2.ROM

Read Only Memory (ROM) merupakan memori yang hanya dapat di baca. Data dalam ROM tidak akan hilang/terhapus meskipun catu daya listrik diputuskan (bersifat *non volatile*). Karena sifatnya yang demikian, maka ROM digunakan untuk menyimpan program.

Ada beberapa tipe ROM, diantaranya adalah ROM murni, PROM, EPROM dan EEPROM. ROM adalah memori yang sudah di program oleh pabrik. PROM dapat di program oleh pemakai, tetapi sekali di program tidak dapat di

program ulang. Alternatif lain adalah menggunakan EPROM (*Erasable programmable Read Only Memory*), yaitu PROM yang dapat di program ulang. Isi EPROM dihapus dengan menghadapkan jendela kaca di atas badan IC ke sinar *Ultra Violet (UV)* selama 5 sampai 10 menit. Isi EPROM setelah di hapus akan berlogika 1. Pemrograman EPROM adalah mengubah logika 1 menjadi logika 0.

2.5.6. Input/Output

Untuk melakukan hubungan dengan piranti di luar sistem, di butuhkan alat I/O (*Input/Output*). Sesuai dengan namanya, alat I/O dapat menerima data dari mikrokontroler/mikroprosesor dan dapat pula memberikan data ke mikrokontroler/mikroprosesor.

Ada dua macam perantara I/O yang di pakai, yaitu piranti untuk hubungan serial (UART) dan piranti untuk hubungan paralel (PIO). Pada mikrokontroler arduino kedua macam I/O tersebut sudah tersedia didalamnya. *Universal Asynchronous Receiver Transmitter (UART)* adalah perantara serial universal.[25]UART merupakan pengirim dan penerima tak serempak universal.Kerja UART adalah mengubah masukan serial menjadi keluaran paralel dan mengubah masukan paralel menjadi keluaran serial.*Parallel Input Output (PIO)* merupakan perantara untuk hubungan data dalam format paralel. PIO adalah alat yang dapat di program dan menyediakan perantara masukkan dan keluaran dasar untuk data paralel 8 bit. PIO ini di beri nama tersendiri oleh pabriknya, misalnya PIA oleh Motorola, PPI oleh Intel, PDC oleh Rockwell dan PIO oleh Zilog [23].

2.5.7 Mikrokontroler Arduino Uno

Arduino Uno dengan ATmega328P mempunyai 14 digital input/output (6 di antaranya dapat digunakan untuk PWM output), 6 analog input, 16 Mhz crystal oscillator, USB connection, power jack, ICSP header, dan reset button. Skema Arduino Uno ATmega328P didasarkan pada blog diagram dari mikrokontroler jenis AVR ATmega328. Komponen utama di dalam papan Arduino adalah sebuah mikrokontroler 8 bit dengan merek ATmega yang dibuat oleh perusahaan Atmel Corporation [26].



Gambar 2.18. Modul Mikrokontroler Arduino

Spesifikasi mikrokontroler Arduino serta fungsi pin ditunjukkan dalam keterangan dibawah ini.

Microcontroller	ATMega328
Operating Voltage	5V
Input Voltage (recommended)	7 - 12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 12 provide PWM output)
Analog Input Pins	6 (10 bits)
Total DC Output Current on all I/O lines	90 mA
DC Current for 3.3V Pin	50 mA
DC Current for 5V Pin	40 mA
32 KB all available for the user applications	
SRAM	2 KB EEPROM
1 KB Clock Speed	16 MHz

2.5.7.1. Power

ArduinoUno dapat diaktifkan melalui koneksi USB atau dengan power supply eksternal. Sumber daya akan dipilih secara otomatis. Eksternal (non-USB) daya dapat berasal baik itu dari adaptor AC-DC (dinding-kutit) atau baterai. Adaptor ini dapat dihubungkan dengan mencolokkan sebuah 2.1mm

steker pusat-positif ke colokan listrik board. Arahan dari baterai dapat dimasukkan ke dalam header pin Gnd dan Vin dari konektor DAYA [26].

Board dapat beroperasi pada persediaan eksternal 6 sampai 20 volt. Jika disertakan dengan kurang dari 7V, maka pin 5V hanya menyuplai kurang dari lima volt dan board mungkin tidak stabil. Jika menggunakan lebih dari 12V, tegangan regulator bisa panas dan merusak board. Kisaran yang disarankan adalah 7 sampai 12 volt. Power pada pin adalah sebagai berikut [26]:

1. VIN. Tegangan input ke papan Arduino saat menggunakan sumber daya eksternal (dibandingkan dengan 5 volt dari koneksi USB atau sumber daya lain). Anda dapat menyediakan tegangan melalui pin ini, atau jika menyediakan tegangan melalui colokan listrik, akses melalui pin ini (VIN)
2. 5V. Regulated power supply digunakan untuk daya mikrokontroler dan komponen lainnya di board. Ini dapat berasal baik dari VIN melalui regulator on-board, atau dipasok oleh USB atau suplai 5V regulated.
3. 3V3. Tegangan pasokan yang dihasilkan oleh chip FTDI on-board. Menarik arus maksimum 50 mA.
4. GND. Pin Ground.

2.5.7.2 Input dan Output

Masing-masing dari 14 digital pin pada Arduino Uno dapat digunakan sebagai input atau output_ menggunakan fungsi `pinMode()`, `digitalWrite()` dan `digitalRead()`. Mereka beroperasi pada tegangan 5 volt. Setiap pin dapat memberikan atau menerima maksimum 40 mA dan memiliki resistor pull-up internal yang (secara default terputus) dari 20-50 kOhms. Selain itu, beberapa pin memiliki fungsi khusus:

- Serial: 0 (RX) and 1 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) TTL data serial. Pin ini terhubung ke pin sesuai FTDI USB-to-TTL Serial chip.

- External Interrupts: 2 and 3. Pin ini dapat dikonfigurasi untuk memicu interrupt pada nilai yang rendah, tepi naik atau jatuh_ atau perubahan nilai. Lihat attachInterrupt () fungsi untuk rincian.
- PWM: 3, 5, 6, 9, 10, and 11. Menyediakan 8-bit PWM output dengan analogWrite () function.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Pin ini mendukung komunikasi SPI menggunakan perpustakaan SPI.
- LED: 13. Terdapat built-in LED terhubung ke pin digital 13. Ketika pin adalah nilai HIGH, LED menyala, saat pin adalah LOW, LED Mati. Arduino Uno memiliki 6 input analog, yang masing-masing menyediakan 10 bit resolusi (yaitu 1024 nilai yang berbeda). Secara default mereka mengukur dari tanah sampai 5 volt, meskipun sebenarnya bisa diganti batas atas jangkauan mereka menggunakan pin AREF dan analogReference () function. Selain itu, beberapa pin memiliki fungsi khusus:
 - I²C: analog input pins A4 (SDA) and A5 (SCL). Mendukung I²C (TWI) komunikasi menggunakan Wire Library. Ada beberapa pin lainnya pada board.
 - AREF. Tegangan referensi untuk input analog. Digunakan dengan analogReference ().
 - Reset. untuk me-reset mikrokontroler. Biasanya digunakan untuk menambahkan tombol reset untuk pelindung yang memblok salah satu di board.

2.5.7.3. Komunikasi

Arduino Uno memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, Arduino lain, atau mikrokontroler lainnya. ATmega328 menyediakan UART TTL (5V) komunikasi serial, yang tersedia di pin digital 0 (RX) dan 1 (TX). Sebuah FTDI FT232RL pada saluran forum ini komunikasi serial melalui USB dan driver FTDI (disertakan dengan versi Windows dari software Arduino) menyediakan port com virtual

untuk perangkat lunak pada komputer. [27] Perangkat lunak Arduino termasuk monitor seri yang memungkinkan data tekstual sederhana yang akan dikirim ke dan dari papan Arduino. RX dan TX LED di papan akan berkedip ketika data sedang dikirim melalui chip FTDI dan koneksi USB ke komputer (tetapi tidak untuk komunikasi serial pada pin 0 dan 1). ATmega328 juga mendukung I2C (TWI) dan komunikasi SPI. Software Arduino termasuk perpustakaan Kabel untuk mempermudah penggunaan bus I2C, lihat Dokumen untuk rincian. Untuk komunikasi SPI, menggunakan SPI Library.

2.5.7.4. USB Proteksi Arus Lebih

Arduino Uno memiliki *polyfuse reset* yang melindungi port USB komputer Anda dari arus pendek dan arus lebih. Meskipun kebanyakan komputer memberikan perlindungan internal mereka sendiri, sekering menyediakan lapisan tambahan perlindungan. Jika lebih dari 500 mA diterapkan ke port USB, sekering akan secara otomatis memutuskan sambungan sampai pendek atau kelebihan beban dihilangkan.

2.5.7.5. Pin input/output digital (0-13)

Berfungsi sebagai input atau output, dapat diatur oleh program. Khusus untuk 6 buah pin 3, 5, 6, 9, 10 dan 11, dapat juga berfungsi sebagai pin analog output dimana tegangan output-nya dapat diatur. Nilai sebuah pin output analog dapat deprogram antara 0 – 255, dimana hal itu mewakili nilai tegangan 0 – 5V.

- a. USB Berfungsi untuk Memuat program dari komputer ke dalam papan Komunikasi serial antara papan dan computer dan memberi daya listrik kepada papan
- b. Sambungan SV1
Sambungan atau *jumper* untuk memilih sumber daya papan, apakah dari sumber eksternal atau menggunakan USB. Sambungan ini tidak diperlukan lagi pada papan Arduino versi terakhir karena pemilihan sumber daya eksternal atau USB dilakukan secara otomatis.

c. Q1 – Kristal (*quartz crystal oscillator*)

Jika microcontroller dianggap sebagai sebuah otak, maka kristal adalah jantung-nya karena komponen ini menghasilkan detak-detak yang dikirim kepada microcontroller agar melakukan sebuah operasi untuk setiap detak-nya. Kristal ini dipilih yang berdetak 16 juta kali per detik (16MHz).

d. Tombol Reset S1

Untuk me-reset papan sehingga program akan mulai lagi dari awal. Perhatikan bahwa tombol reset ini bukan untuk menghapus program atau mengosongkan microcontroller.

e. *In-Circuit Serial Programming (ICSP)*

Port ICSP memungkinkan pengguna untuk memprogram microcontroller secara langsung, tanpa melalui bootloader. Umumnya pengguna Arduino tidak melakukan ini sehingga ICSP tidak terlalu dipakai walaupun disediakan.

f. IC 1 – Microcontroller Atmega

Komponen utama dari papan Arduino, di dalamnya terdapat CPU, ROM dan RAM.

g. X1 - sumber daya eksternal

Jika hendak disuplai dengan sumber daya eksternal, papan Arduino dapat diberikan tegangan DC antara 9-12V.

h. 6 pin input analog (0-5)

Pin ini sangat berguna untuk membaca tegangan yang dihasilkan oleh sensor analog, seperti sensor suhu. Program dapat membaca nilai sebuah pin input antara 0 – 1023, dimana hal itu mewakili nilai tegangan 0 – 5V.

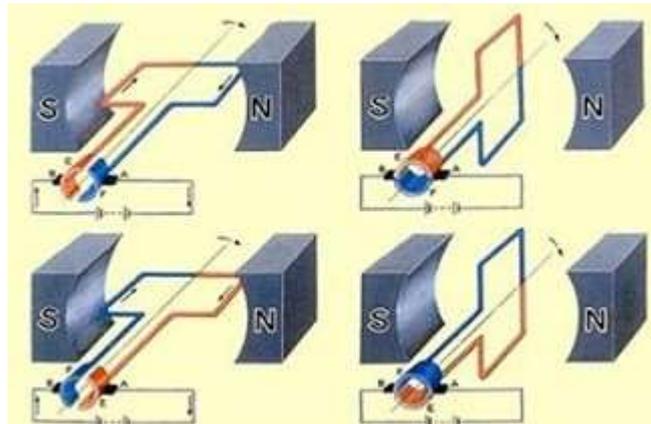
2.6 Kipas/fan

Kipas atau fan berfungsi untuk membuat tekanan positif didalam ruang operasi dengan cara memasukkan sejumlah udara dari ruang terbuka kedalam ruang operasi , Terdapat dua bagian utama pada sebuah Fan DC, yaitu Stator dan Rotor. Stator adalah bagian motor yang tidak berputar, bagian yang statis ini terdiri dari rangka dan kumparan medan. Sedangkan Rotor adalah bagian yang berputar, bagian Rotor ini terdiri dari kumparan Jangkar. Dua bagian utama ini dapat dibagi lagi menjadi beberapa komponen penting yaitu diantaranya adalah Yoke (kerangka magnet), Poles (kutub motor), Field winding (kumparan medan magnet), Armature Winding (Kumparan Jangkar), Commutator (Komutator) dan Brushes (kuas/sikat arang).



Gambar 2.19. Kipas Angin/ Fan

Pada prinsipnya motor listrik DC menggunakan fenomena elektromagnet untuk bergerak, ketika arus listrik diberikan ke kumparan, permukaan kumparan yang bersifat utara akan bergerak menghadap ke magnet yang berkutub selatan dan kumparan yang bersifat selatan akan bergerak menghadap ke utara magnet. Saat ini, karena kutub utara kumparan bertemu dengan kutub selatan magnet ataupun kutub selatan kumparan bertemu dengan kutub utara magnet maka akan terjadi saling tarik menarik yang menyebabkan pergerakan kumparan berhenti. [28]



Gambar 2.20. Kipas Angin/ Fan

Untuk menggerakannya lagi, tepat pada saat kutub kumparan berhadapan dengan kutub magnet, arah arus pada kumparan dibalik. Dengan demikian, kutub utara kumparan akan berubah menjadi kutub selatan dan kutub selatannya akan berubah menjadi kutub utara. Pada saat perubahan kutub tersebut terjadi, kutub selatan kumparan akan berhadapan dengan kutub selatan magnet dan kutub utara kumparan akan berhadapan dengan kutub utara magnet. Karena kutubnya sama, maka akan terjadi tolak menolak sehingga kumparan bergerak memutar hingga utara kumparan berhadapan dengan selatan magnet dan selatan kumparan berhadapan dengan utara magnet. Pada saat ini, arus yang mengalir ke kumparan dibalik lagi dan kumparan akan berputar lagi karena adanya perubahan kutub. Siklus ini akan berulang-ulang hingga arus listrik pada kumparan diputuskan.

2.7 LCD

LCD sebagai penampil, sehingga data suhu dan kelembapan dapat ditampilkan secara jelas dan kristis. Kemampuan dari LCD untuk menampilkan tidak hanya angka-angka, tetapi juga huruf-huruf, kata-kata dan semua sarana simbol, lebih bagus dan serbaguna daripada penampil-penampil menggunakan 7-segment LED (Light Emiting Diode) yang sudah umum. Modul LCD mempunyai *basic interface* yang cukup baik, yang mana sesuai dengan minimum system. Sesuai juga dengan keluarga mikrokontroler yang lain. Bentuk dan ukuran modul-modul berbasis karakter banyak ragamnya, salah satu variasi bentuk dan ukuran yang

tersedia dan dipergunakan pada peralatan ini adalah 20 x 4 karakter (panjang 20, baris 4, karakter 80) dan 16 pin, bentuk LCD diperlihatkan dalam gambar 2.21.



Gambar 2.21 LCD

2.8 Power Supply

Pengertian *power supply*, fungsi dan cara kerjanya adalah merupakan suatu komponen yang mempunyai fungsi sebagai pemberi suatu tegangan serta arus listrik kepada komponen - komponen komputer lainnya yang telah terpasang dengan baik pada *motherboard* atau papan induk, sedang tujuan awal dari penyaluran arus listrik ini adalah agar perangkat atau komponen - komponen komputer lainnya bisa berfungsi sebagaimana mestinya sesuai dengan tugasnya. [29] Arus listrik yang disalurkan oleh *power supply* ini merupakan arus listrik dengan jenis AC atau arus bolak balik, namun dengan kelebihanannya PSU ini dapat mengubah arus AC tersebut menjadi arus DC atau merupakan arus yang searah karena pada dasarnya semua komponen yang terdapat pada perangkat komputer hanya bisa melakukan pergerakan pada satu aliran listrik.

Fungsi utama dari *power supply* adalah sebagai alat yang mampu memberikan sebuah suplai arus listrik kepada semua komponen komputer yang sudah terpasang dengan baik, dimana arus listrik yang dihasilkan merupakan arus AC dan selanjutnya akan dirubah menjadi arus DC. Yang perlu digaris bawahi adalah jika semua komponen *hardware* yang sudah terpasang pada komputer ini tidak bisa menerima arus listrik AC namun hanya bisa menerima aliran listrik dengan tipe DC . Gambar 2.22. adalah *power supply*. Untuk memberikan sumber DC bagi mikrokontroller dan komponen Elektronika seperti heater dan kipas.



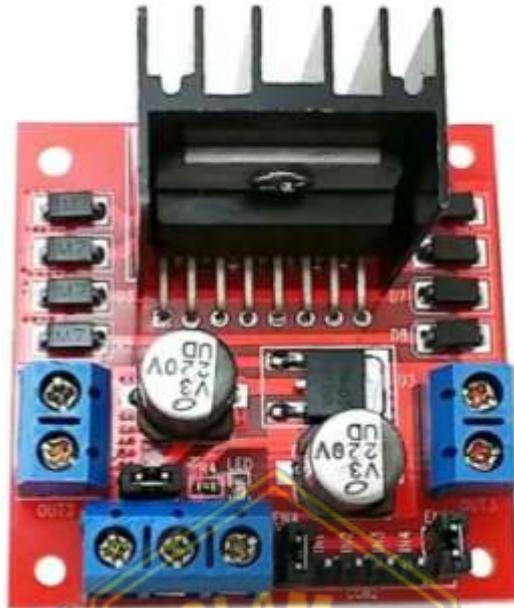
Gambar 2.22. *Power Supply*

2.9 Driver Motor

Pengaturan Motor DC Menggunakan Pulse-Width Modulation Pulse Width Modulation adalah suatu teknik manipulasi dalam pengemudi motor (atau perangkat elektronik berarus besar lainnya) yang menggunakan prinsip cut-off dan saturasi. Pulse-Width Modulation dapat juga diartikan sebagai sebuah teknik untuk membangkitkan sinyal keluaran yang periodenya berulang-ulang berupa high dan low yang dapat mengontrol durasi sinyal high dan low sesuai dengan yang diinginkan (Endra Pitowarno, 2006 : 90).

Metode PWM ini membutuhkan driver sebagai *manipulator* dari *plant* menuju *actuator* berupa motor DC.

Driver adalah rangkaian yang tersusun dari transistor yang digunakan untuk menggerakkan motor DC. Motor memang dapat berputar hanya dengan daya DC, tapi tidak bisa diatur tanpa menggunakan driver, maka diperlukan suatu rangkaian driver yang berfungsi untuk mengatur kerja dari motor. dapat dilihat driver motor yang digunakan sebagai berikut. Pada driver motor DC ini dapat mengeluarkan arus hingga 2A, dengan memiliki fungsi Pulse-Width Modulation. Tegangan sumber DC yang dapat diberikan antara 5 V- 35VDC, sedangkan tegangan input level 5VDC, driver motor ini menggunakan rangkaian full H-bridge dengan IC L298. Gambar 2.23 menunjukkan gambar fisik modul L298 H-Bridge Driver PWM.[30]



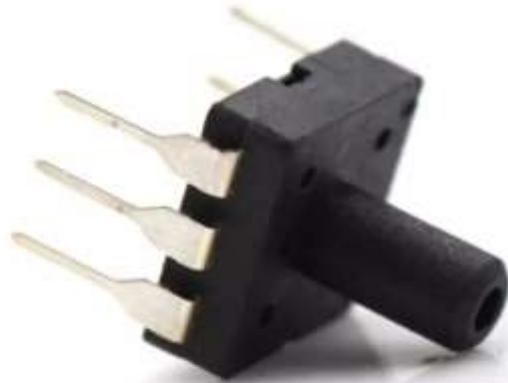
Gambar 2.23 Modul L298 H-Bridge Driver PWM.

Komponen utama driver motor di atas adalah transistor yang dipasang sesuai karakteristiknya dimana transistor yang digunakan pada rangkaian ini adalah Driver motor L298. Karakteristik Driver motor L298:

- RPWM (Forward Level atau Sinyal PWM Input, aktif HIGH)
- LPWM (Inversion Level atau Sinyal PWM Input, aktif HIGH)
- R_EN (Forward Drive Enable input, Aktif HIGH)
- L_EN (Reverse Drive Enable input, Aktif HIGH)
- VCC (+5v, connect to arduino) - GND(Ground)

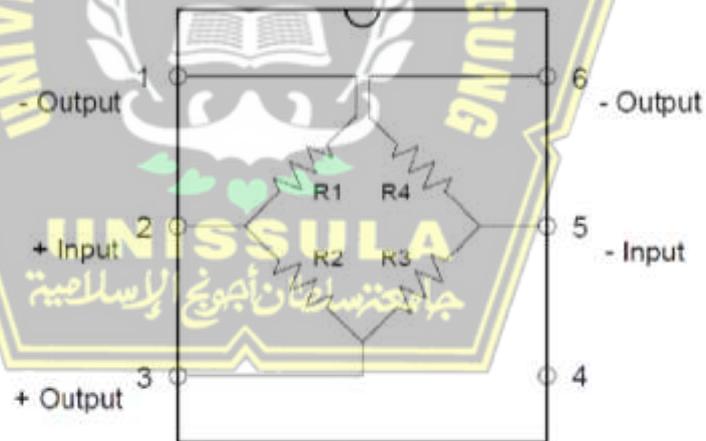
2.10 Sensor mps20n0040d-d

Perangkat ini adalah sensor tekanan atmosfer yang dapat digunakan dan dikombinasikan dengan mikrokontroler Arduino Uno. Penggunaan sensor ini dapat dijumpai pada industri otomotif seperti pada sensor pengukur tekanan ban, sensor MAP, pompa udara ban, dan sistem pengereman udara yang terdapat pada beberapa mobil, bus dan truk, dan pada industri kesehatan seperti pada aliran darah pada bagian pompa udara.



Gambar 2.24 Sensor mps20n0040d-d

Sensor ini memiliki kemampuan untuk mengukur tekanan udara, dengan rentang pengukuran antara 0 sampai dengan 40 kPa. Sumber tegangan yang digunakan adalah 5 VDC.

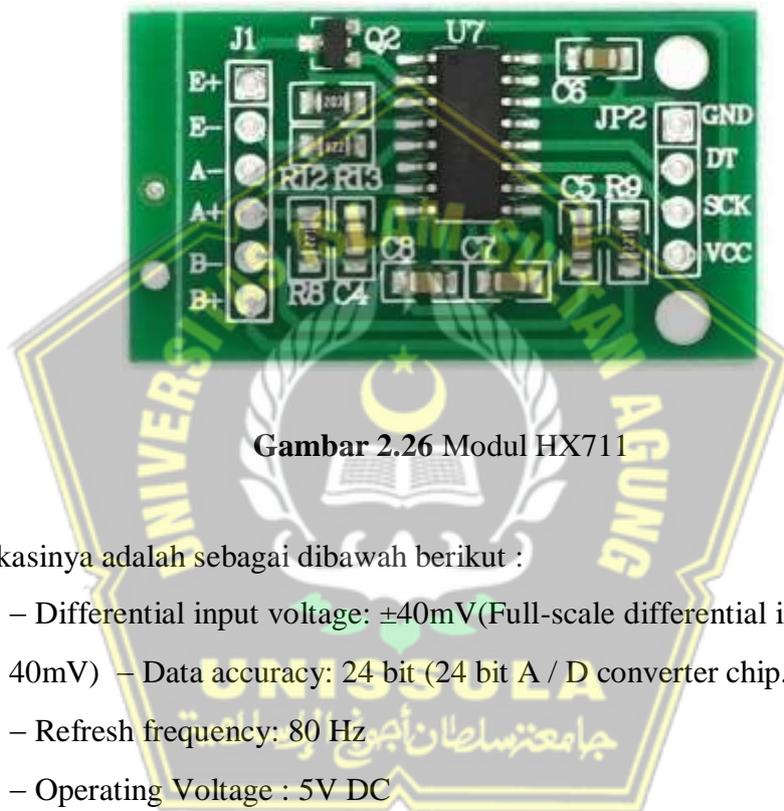


Gambar 2.25 Blok Diagram Sensor mps20n0040d-d

Prinsip kerja sensor mps20n0040d-d adalah seperti rangkaian jembatan wheatstone. Dimana di dalam sensor terdapat 4 buah resistor yang dirangkai sedemikian rupa, terdiri dari resistor tetap dan resistor *variable*. Hal ini seperti penggunaan konsep jembatan wheatstone pada loadcell atau timbangan.

2.11 Hx711

HX711 adalah sebuah IC yang memiliki kepresisian 24-bit analog to digital converter (ADC) yang didesain untuk sensor-sensor yang memiliki desain seperti jembatan whetstone. HX711 memiliki prinsip kerja mengkonversi perubahan yang terukur dalam perubahan resistansi dan 9 mengkonversinya ke dalam besaran tegangan melalui rangkaian yang ada. HX711 melakukan komunikasi dengan computer/mikrokontroller melalui komunikasi TTL232. Struktur yang sederhana, memiliki sensitivitas tinggi, dan mampu mengukur perubahan dengan cepat.[31]



Gambar 2.26 Modul HX711

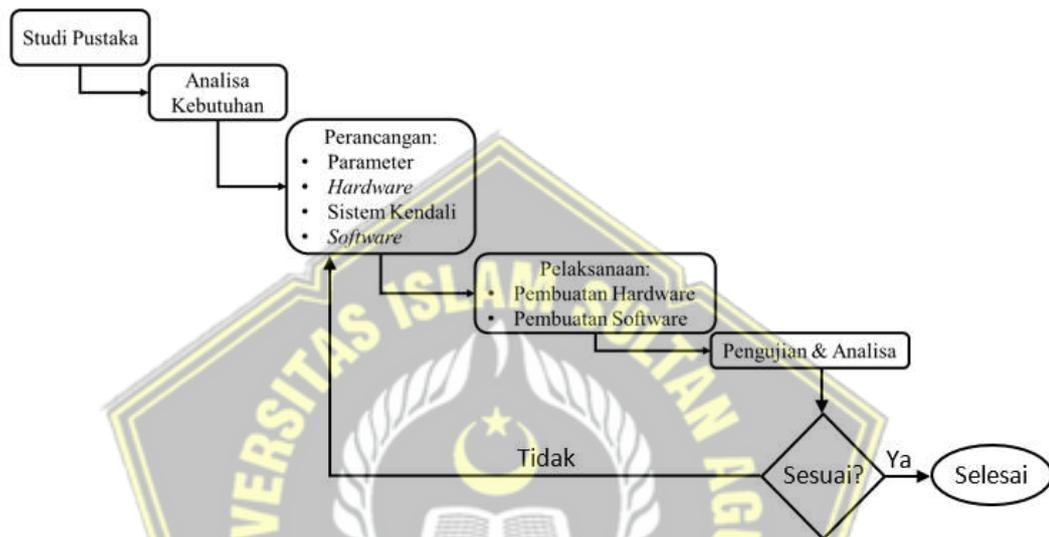
Spesifikasinya adalah sebagai dibawah berikut :

- Differential input voltage: $\pm 40\text{mV}$ (Full-scale differential input voltage $\pm 40\text{mV}$)
- Data accuracy: 24 bit (24 bit A / D converter chip.)
- Refresh frequency: 80 Hz
- Operating Voltage : 5V DC
- Operating current :

BAB III METODOLOGI

3.1. Desain Penelitian

Penelitian ini menggunakan model yang meliputi studi pustaka, analisis kebutuhan, perancangan parameter, perancangan arsitektur, perancangan alat, perancangan sistem, pelaksanaan, serta pengujian dan analisa. Alur kerja dalam penelitian ini dapat dilihat pada gambar 3.1.



Gambar 3.1. Alur Kerja Penelitian

3.1.1 Studi Pustaka

Hasil penelitian – penelitian terdahulu digunakan sebagai referensi dalam penelitian ini. Studi pustaka sangatlah penting dalam langkah awal melakukan penelitian karena akan memberikan gambaran awal tentang langkah - langkah yang akan dilakukan, serta mendapatkan pemahaman dalam proses penelitian, apa saja yang sudah dilakukan pendahulunya, serta akan mengetahui kekurangan dan kelebihan nya. Dari beberapa referensi yang didapat, metode logika Fuzzy dan PID dapat dianggap cukup efektif digunakan dalam proses pengendalian sistem tekanan udara di ruang operasi.

3.1.2 Analisa Kebutuhan

Analisa kebutuhan “Pemodelan Kendali Tekanan Udara Ruang Operasi di Rumah Sakit Menggunakan Logika Fuzzy dan PID” dilakukan untuk mendukung persiapan komponen yang akan digunakan. Komponen-komponen tersebut meliputi komponen elektronika, pemrograman, dan juga mekanik yang

akan diterapkan pada penelitian ini. Hal yang perlu menjadi perhatian khusus pada ruang operasi diantaranya adalah suhu ruangan, kelembaban ruangan, serta tekanan udara dalam ruang operasi. Beberapa hal tersebut sangat berpengaruh terhadap kondisi ideal ruang operasi. Kondisi yang ideal diharapkan dapat sesuai dengan kebutuhan kondisi ruang operasi, sehingga pada akhirnya angka kuman di ruang operasi sesuai dengan standar baku mutu yang telah ditetapkan.

3.1.3 Perancangan

3.1.3.1 Parameter

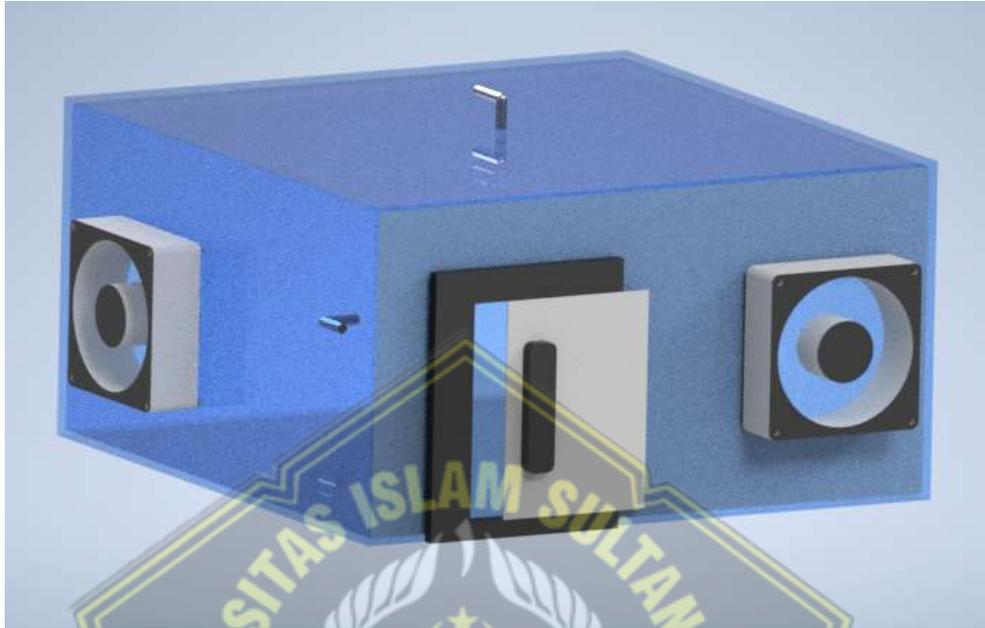
Sebelum merancang peralatan, dalam penelitian ini terlebih dahulu akan dilakukan penentuan parameter. Salah satu parameter yang harus diperhatikan dalam ruang operasi adalah tekanan udara. Pada penelitian ini pengendalian tekanan udara menjadi fokus utama. Hal ini dilakukan agar kondisi ideal ruang operasi tercapai dimana target tekanan udara di dalam ruang operasi adalah positif 10 Pascal.

Untuk mendapatkan kondisi ideal ruang operasi, diperlukan blower yang dapat terkontrol dengan sistem guna memberikan tekanan udara dari luar. Secara logika pada kondisi tekanan udara pada seting ruangan 10 pascal maka blower akan berhenti berputar dan akan kembali berputar ketika nilai tekanan udara dibawah 10 pascal. Ketika pintu terbuka maka akan terjadi penurunan tekanan pada ruang operasi sehingga sensor akan mendeteksi adanya penurunan tekanan udara sehingga sistem akan meningkatkan putaran blower untuk mengembalikan tekanan pada nilai 10 pascal. Dalam sistem yang dirancang, blower menggunakan 2 buah kipas yang dirangkai secara parallel. Jika terjadi nilai tekanan udara diatas 10 pascal maka exhaust akan berputar untuk mengurangi nilai tekanan udara di dalam ruang operasi. Prinsip kerja secara detail akan dijelaskan pada bab IV.

3.1.3.2 Pemodelan Ruang Operasi

Aturan umum dimensi kamar bedah adalah minimal 29,1 – 37,16 meter persegi (5,6m x 5,6 m) dan maksimal 56 – 60 meter persegi (7,2 m x 7,8 m), sedangkan tinggi langit-langit (plafon) 3 m - 3,65m. Besar kecilnya ukurnya ini bisa disesuaikan dengan kebutuhan misalkan berkaitan dengan dimensi alat-alat yang akan diletakkan di dalam kamar operasi. Di dalam

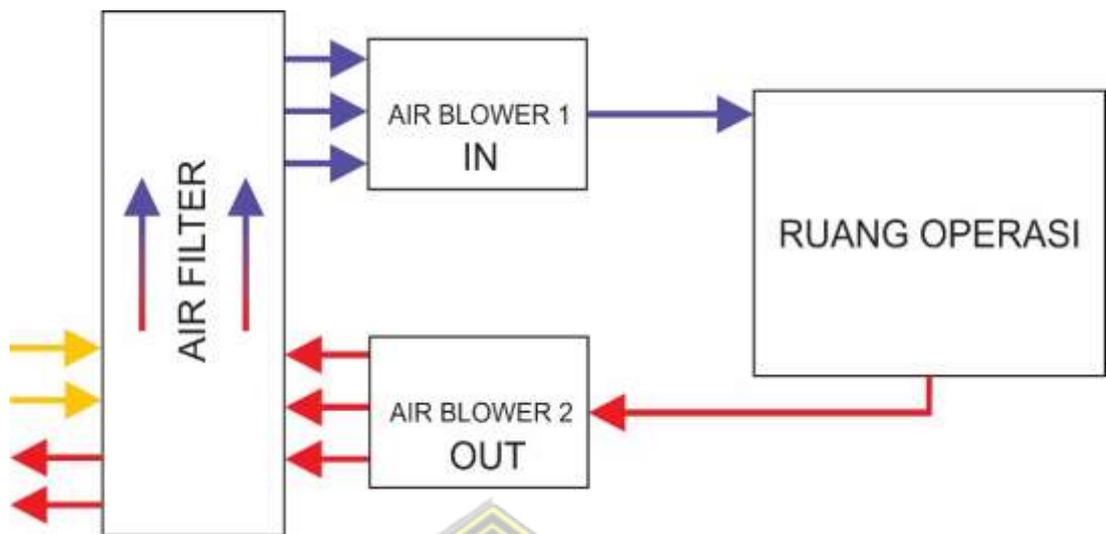
penelitian ini ruang operasi akan dimodelkan dengan miniatur dengan skala 1:200 sehingga ukuran yang dibuat adalah 31 cm x 31 cm dan tinggi 16 cm dengan desain sebagai berikut:



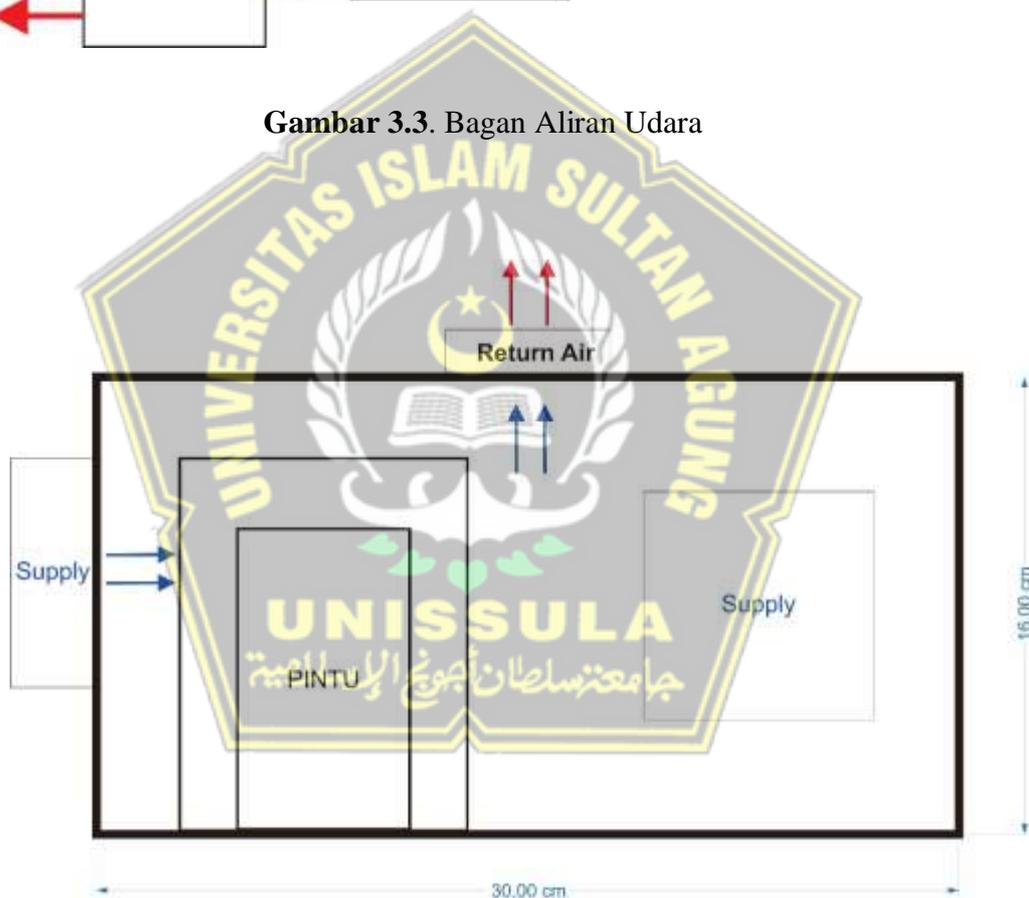
Gambar 3.2. Desain Pemodelan Ruang Operasi

3.1.3.3 Aliran Udara

Pengendalian aliran udara merupakan tumpuan dalam menjaga tekanan di dalam ruang operasi. Pengaturan aliran udara ini dengan mengendalikan udara yang masuk maupun yang keluar dari ruang operasi. Dimana udara bersih dimasukkan untuk menjaga tekanan tetap terjaga dan udara dikeluarkan melalui exhaust untuk di sirkulasi maupun dibuang keluar sistem jika berlebih. Desain aliran udara yang direncanakan adalah sebagai berikut:



Gambar 3.3. Bagan Aliran Udara



Gambar 3.4. Aliran Udara di Dalam Ruang

3.1.3.4 Hardware

Berdasarkan Analisa kebutuhan, komponen yang akan digunakan pada “Pemodelan Kendali Tekanan Udara Ruang Operasi di Rumah Sakit Menggunakan Logika Fuzzy” adalah sebagai berikut :

1. Fan / blower sebagai pendorong tekanan udara pada ruang operasi.

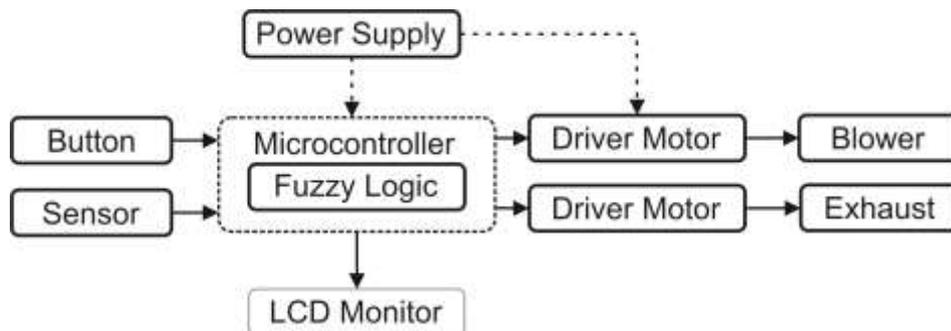
2. Motor Exhaust sebagai pembuang tekanan udara.
3. Sensor tekanan udara MPS sebagai pendeteksi tekanan udara di dalam ruang.
4. Mikrokontroler sebagai pusat kendali sistem.
5. Keypad / button sebagai input.
6. LCD 16 x 2 , digunakan sebagai layar display.
7. Power Suplay 12 Volt, digunakan untuk suplay tegangan pada Arduino dan fan.
8. Arduino IDE sebagai software pendukung pengembangan perangkat lunak sistem pada mikrokontroler.
9. Prototipe Ruang operasi berupa chamber akrilik.

Peralatan pendukung yang digunakan dalam proses penelitian ini adalah sebagai berikut :

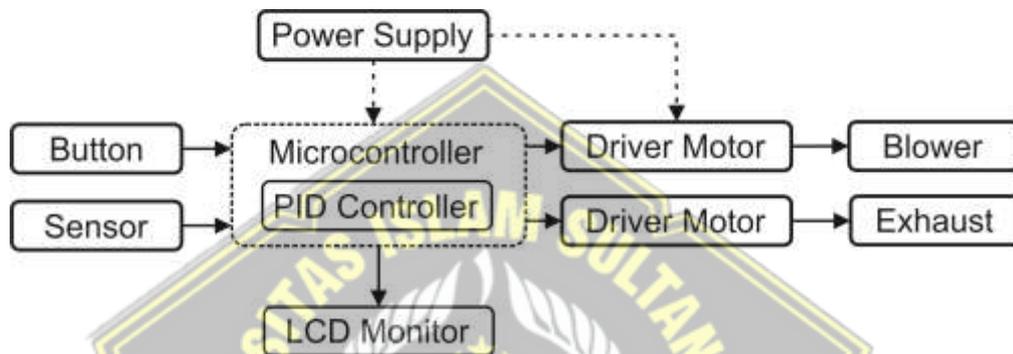
1. Laptop, digunakan sebagai pendukung pengembangan perangkat lunak sistem dan penulisan ilmiah.
2. Multimeter, digunakan untuk mengukur parameter, tegangan, dan tahanan.
3. Maghnehelic untuk mengukur tekanan udara .
4. Peralatan – peralatan pendukung lain, seperti solder, timah, palu , tool kit lem bakar, dan isolasi.

3.1.3.5 Perancangan Sistem Kendali

Sebagian besar dalam penelitian terdahulu adalah pengendalian suhu atau kelembaban ruang. Sehingga dianggap perlu untuk dikembangkan dalam penelitian ini, alat yang digunakan untuk mengendalikan tekanan udara agar selalu stabil. Diagram blok keseluruhan sistem yang dibuat dalam penelitian ini dapat dilihat pada gambar 3.5 dan gambar 3.6

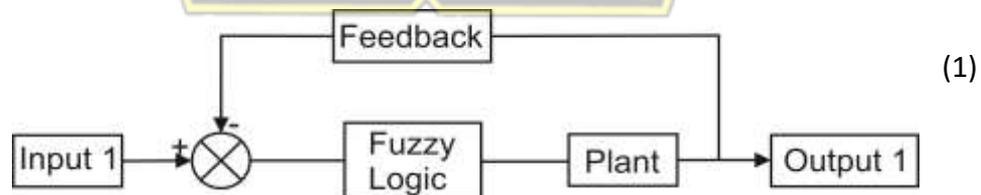


Gambar 3.5. Diagram Blok Keseluruhan Sistem Fuzzi

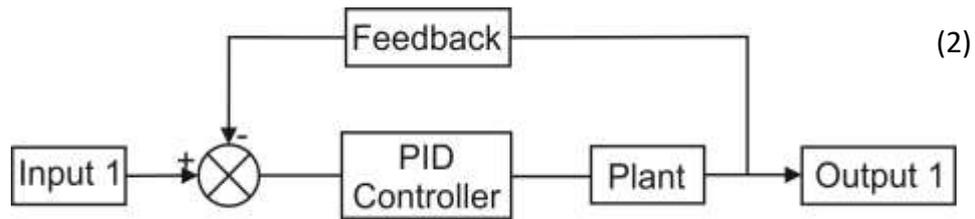


Gambar 3.6. Diagram Blok Keseluruhan Sistem PID

Berdasarkan pada Gambar 3.5 dan Gambar 3.6, dapat dilihat bahwa terdapat 2 metode kendali yang ditanamkan pada mikrokontroler. Hal ini diterapkan guna mendukung proses pengendalian Blower dan Exhaust pada ruang operasi dan dapat dilihat perbedaan karakter pada kedua kendali tersebut. Adapun penjabaran sistem kendali dapat dilihat pada gambar 3.7 dan gambar 3.8.



Gambar 3.7. Diagram Blok Sistem Kendali Fuzzy



Gambar 3.8. Diagram Blok Sistem Kendali PID

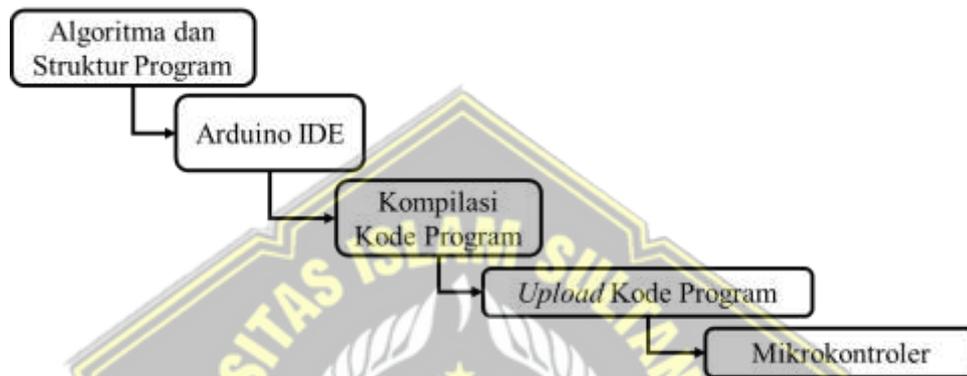
Perancangan sistem kendali dalam penelitian ini didesain dalam beberapa bagian yaitu sensor tekanan udara, sensor pintu, pemrosesan data sistem kendali logika Fuzzy dan sistem kendali PID pada mikrokontroler, serta display dan gerak fan baik blower maupun exhaust. Obyek yang di ambil dalam penelitian ini merupakan data tekanan udara. Selanjutnya data yang masuk berupa sinyal analog selanjutnya diubah menjadi sinyal digital. Setelah itu, data tekanan udara akan diolah menggunakan kendali PID dan Logika Fuzzy. Hasil dari sistem kendali dalam penelitian ini diharapkan dapat mendukung pengaturan tekanan udara yang dihasilkan dari fan secara otomatis berdasarkan setpoint yang telah ditentukan. Hasil dari pemrosesan tersebut akan ditampilkan dalam display.

3.1.4 Software

Mikrokontroler digunakan sebagai kendali utama sistem dalam penelitian ini. Dalam perancangan perangkat lunak sistem yang ditanam pada mikrokontroler, penentuan algoritma dan struktur program sangatlah diperlukan. Penelitian ini berfokus pada stabilisasi sistem tekanan positif pada ruang operasi. Prinsip kerja dari sistem penelitian ini dimulai dari pembacaan data sensor tekanan yang dapat terpengaruh dari terbuka / tertutupnya pintu ruang operasi. Dari data tekanan udara yang didapatkan, selanjutnya data yang diperoleh dari sensor tekanan diolah melalui proses fuzifikasi, rules, evaluasi aturan dan defuzifikasi, serta digunakan sebagai acuan nilai masukan untuk kendali PID. Selanjutnya, nilai keluaran dari defuzifikasi yang berupa nilai PWM dikirimkan menuju dirver motor guna mengatur putaran fan. Sistem juga dilengkapi dengan LCD untuk memonitor kondisi tekanan udara dan status pintu serta prosentasi kerja / putaran motor.

Setelah algoritma dan struktur program terancang dengan baik, tahapan selanjutnya adalah pembuatan kode program (*source code*) menggunakan

Arduino IDE. Arduino IDE merupakan salah satu perangkat lunak open source yang digunakan untuk mendukung pengembangan kode program dalam bahasa C untuk keluarga mikrokontroler AVR. Setelah kode program dibuat sesuai dengan algoritma dan struktur program yang telah dirancang, tahapan berikutnya adalah *me-upload* kode program ke dalam mikrokontroler dengan cara memasukkan file *.ino kedalam mikrokontroler dengan fitur yang telah disediakan Arduino IDE. Proses pemrograman mikrokontroler dapat dilihat pada Gambar 3.9



Gambar 3.9. Proses Pemrograman Mikrokontroler

3.1.5 Perancangan Input – Output

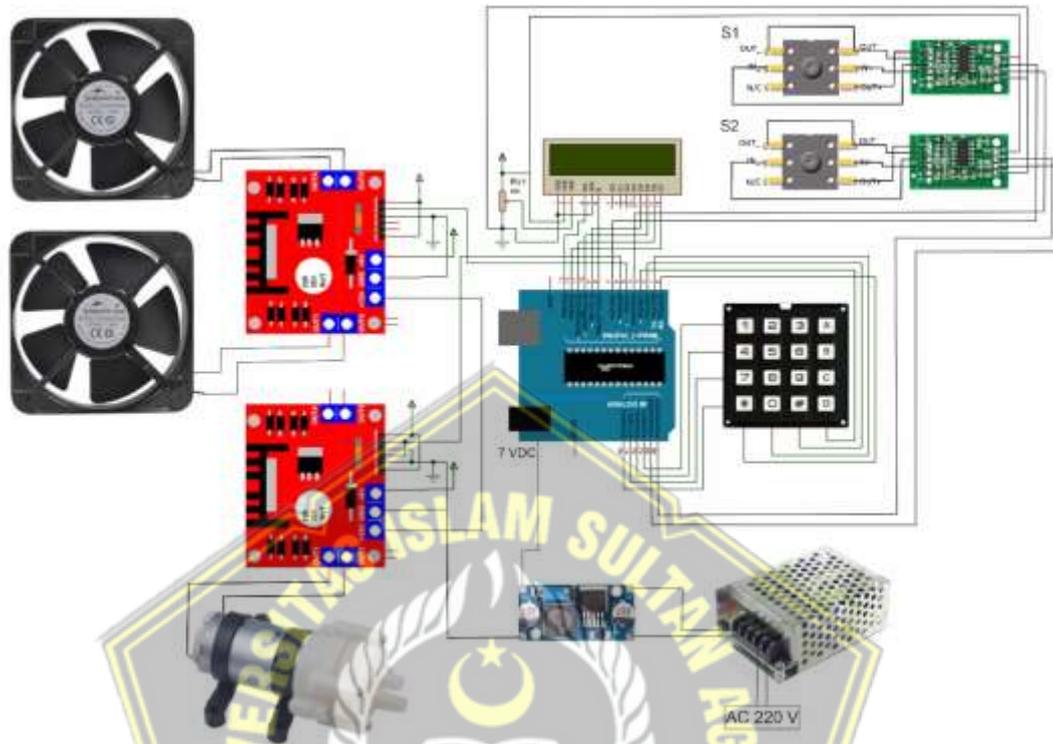
Pengendali utama dalam penelitian ini adalah ATmega 328, berikut adalah table perancangan input output mikrokontroler ATmega 328 yang digunakan dalam sistem pengendali tekanan ruang operasi ini:

Tabel 3.1 Pin input output mikrokontroler

No.	PIN	Keterangan
1	0 - 3	Pin kolom keypad
2	A0 – A3	Piw row keypad
3	6	Blower
4	7	Exhaust
5	4 dan 7	S1
6	A4 dan A5	S2
7	8	EN LCD
8	13	RW LCD
9	9 - 12	Data LCD

3.1.6 Diagram Blok Seluruh Sistem Hardware

Berikut ini adalah diagram blok hardware sistem kendali tekanan ruang operasi berbasis Mikrokontroler Atmega328.



Gambar 3.10. Wiring Diagram

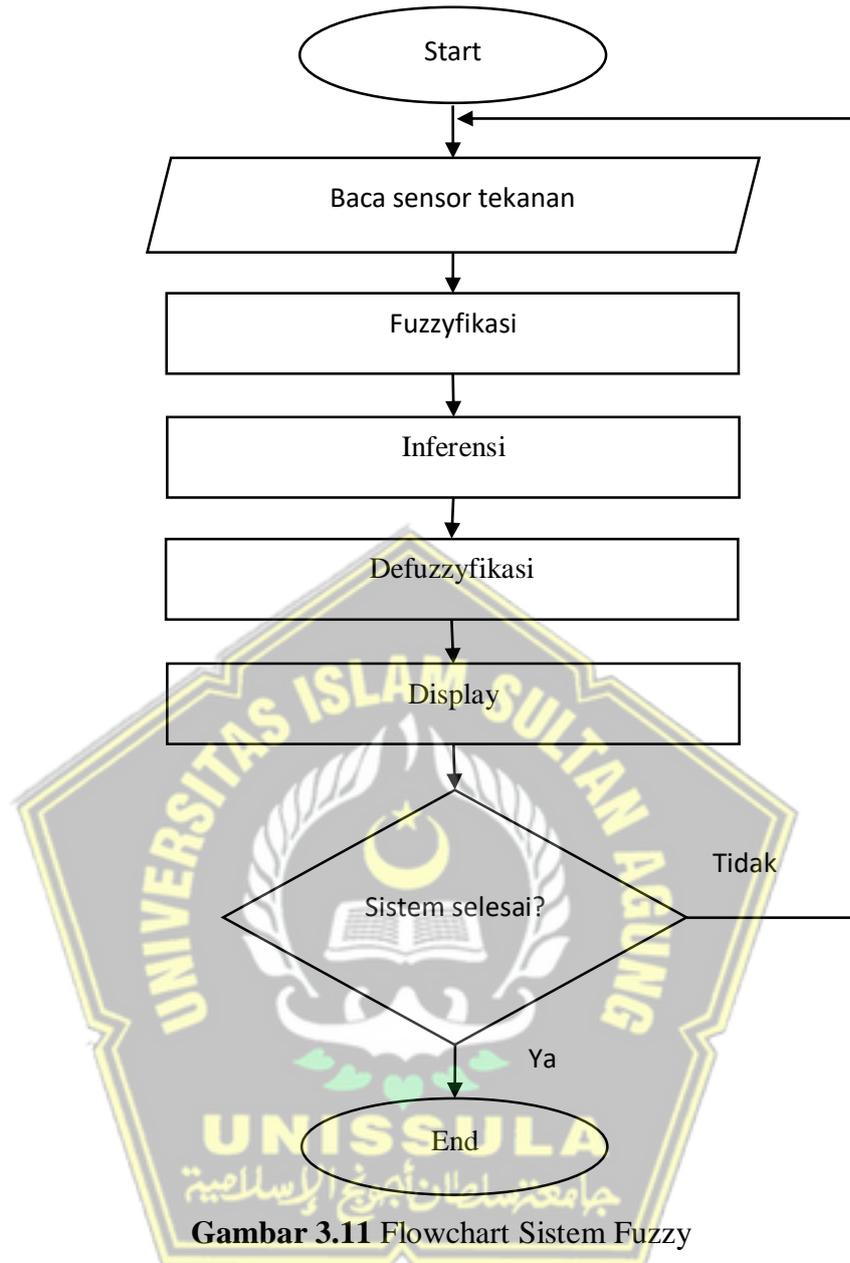
Penjelasan wiring diagram sistem kendali:

1. Power supply menggunakan power supply 12 V 3 A, nilai tegangan dan arus ini digunakan untuk memberi supply pada sistem, di dalam sistem nilai 12 volt akan diturunkan menjadi 5 volt terutama untuk sumber tegangan bagi perangkat mikrokontroler dan input output langsungnya. Sedangkan tegangan pure 12 volt digunakan untuk menyalakan blower dan exhaust.
2. Sensor tekanan yang digunakan adalah sensor tekanan dengan tipe mps20n0040d-d. Sensor ini adalah sensor tekanan yang memiliki rating pengukuran dari 0 sampai 40 KPa. Memanfaatkan cara kerja rangkaian adjustable resistor yang terdapat di dalam sensor, dimana nilai resistansi dari resistor ini akan berubah seiring dengan nilai tekanan yang diberikan.

3. Sensor tekanan dalam sistem yang dibangun ini membutuhkan konverter dari analog ke digital. Konverter yang digunakan adalah modul HX711. HX711 adalah sebuah IC yang memiliki kepresisian 24-bit analog to digital converter (ADC) yang didesain untuk sensor-sensor yang memiliki desain seperti jembatan whetstone.
4. LCD 20 x 4 pada rangkaian sistem ini berfungsi sebagai penampil informasi. Informasi yang ditampilkan adalah nilai pengukuran sensor, nilai PID yang dimasukkan dan nilai-nilai berkaitan dengan sisten Fuzzy serta nilai aktuasi untuk pengendalian motor blower dan exhaust.
5. Driver motor L298, memiliki fungsi sebagai jembatan aktuasi antara mikrokontroler dan motor blower maupun exhaust, fungsi utama driver L298 adalah men drive level tegangan 0 sampai dengan 12 volt dengan menggunakan tegangan 0 sampai dengan 5 volt.
6. Motor blower, memiliki fungsi untuk menyuplai udara bertekanan ke dalam miniatur ruangan yang telah dibuat. Motor ini memiliki level tegangan untuk putaran maksimal pada 12 volt.
7. Motor exhaust, digunakan untuk membuang udara dari dalam ruang dari miniatur ruang ketika tekanan terlalu tinggi. Level tegangan dari motor exhaust ini juga 12 volt.
8. ATmega 328, merupakan mikrokontroler yang digunakan menjadi pengendali utama dalam sistem yang dibuat. Segala pemrosesan input, PID, Fuzzy, serta mengeluarkan manipulated signal dilakukan oleh mikrokontroler ini. Basic dari sistem yang digunakan adalah board Arduino UNO.

3.2 Flowchart Sistem Fuzzy Logic

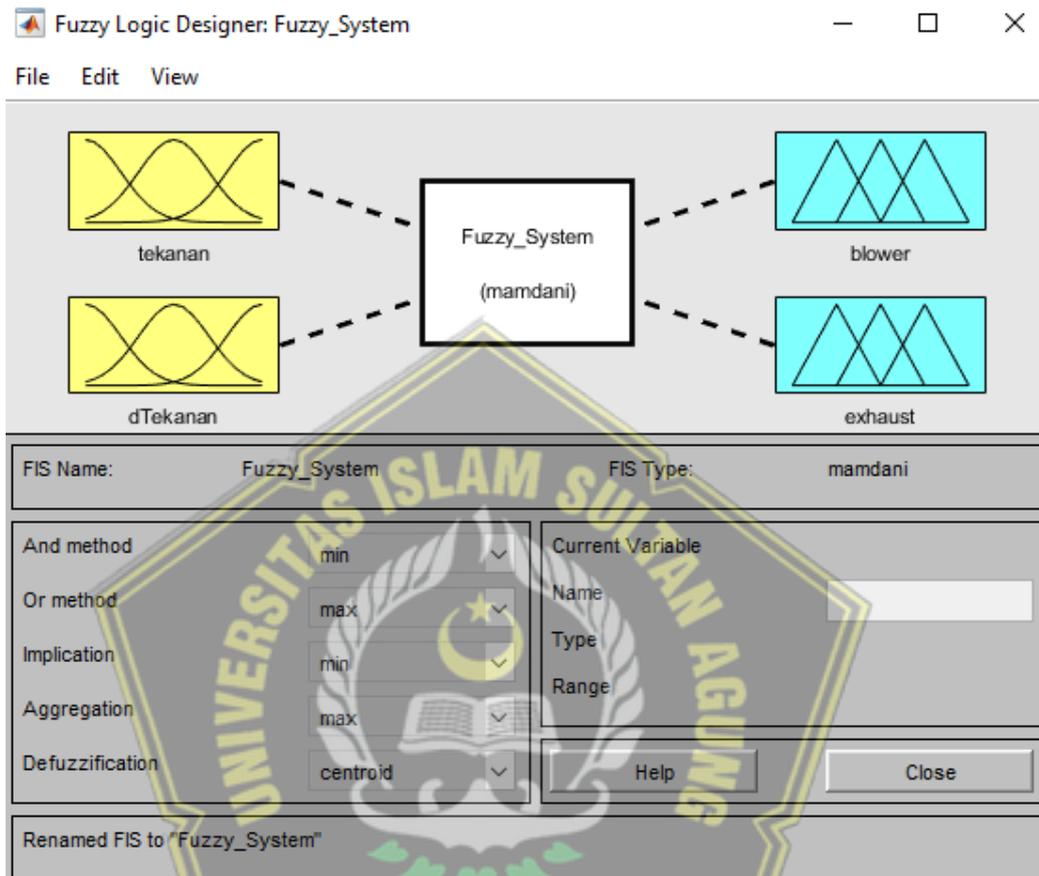
Untuk menjalankan sistem diperlukan perancangan kendali fuzzy yang ditanamkan didalam mikrokontroler atmega 328. Berikut adalah diagram flowchart sistem yang dibangun dan merupakan alur sistem pemrosesan input sampai dengan keluarnya output:



Flowchart diatas memperlihatkan alur sistem, dimana parameter awal diambil dari sensor tekanan. Dari nilai sensor tekanan ini akan dilakukan proses Fuzzyfikasi sehingga menghasilkan himpunan yang selanjutnya akan dilakukan proses penalaran oleh sistem dengan berpedoman pada Rules yang sudah ditentukan. Setelah proses penalaran dilakukan maka proses selanjutnya adalah proses Defuzzyfikasi yang menghasilkan nilai output.

3.3 Desain Kendali tekanan positif ruangan berbasis fuzzy logic

Dalam perancangan software perlu dibuat pemodelan untuk dapat mempermudah menganalisa respon sistem sebelum algoritma ditanamkan pada mikrokontroler. Pemodelan dalam penelitian ini menggunakan software matlab, dan untuk desain pemodelan Fuzzy Logic diperlihatkan pada gambar 3.12 berikut ini.



Gambar 3.12 Desain model Fuzzy Logic untuk kendali tekanan positif ruang operasi

Di dalam sistem ini algoritma yang digunakan adalah algoritma Fuzzy Mamdani dimana memiliki satu buah input yang berasal dari sensor tekanan dan memiliki dua output yaitu blower dan exhaust. Metode algoritma menggunakan metode Max-min dan defuzzifikasi menggunakan metode COA (Center of area). Untuk menghasilkan kinerja sistem seperti yang diinginkan maka perlu direncanakan Membership Function, Fuzzification dan defusifikasi seperti berikut:

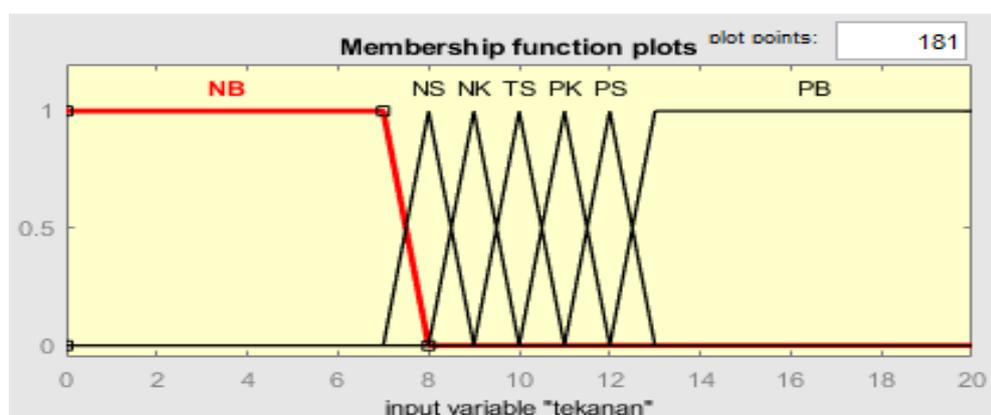
3.3.1 Proses Fuzzyfikasi

Setelah menginisialisasi sistem untuk persiapan proses kerja mikrokontroler, tahap berikutnya adalah membaca sensor tekanan yang hasilnya dimasukkan ke input mikrokontroler ATmega328 yang selanjutnya digunakan

sebagai input proses fuzzyfikasi. Proses fuzzyfikasi yang dilakukan adalah dengan menggunakan nilai pembacaan sensor tekanan.

Error yang terjadi akibat tidak sesuainya nilai tekanan yang terukur sensor dengan nilai tekanan yang diinginkan akan diperbaiki dengan menggunakan algoritma Fuzzy. Kondisi error sensor tekanan ada 7 kondisi. yaitu TS (error tekanan sesuai), NK (error simpangan tekanan negative kecil), NS (error simpangan negative sedang), NB (error simpangan tekanan negative besar), PK (error simpangan tekanan positif kecil), PS (error simpangan tekanan positif sedang), PB (error simpangan tekanan positif besar).

Kondisi TS adalah kondisi dimana nilai pengukuran sensor tekanan sesuai dengan nilai tekanan yang diinginkan. Kondisi NK adalah kondisi dimana nilai tekanan terukur oleh sensor dibawah dari nilai tekanan yang diinginkan tetapi dengan selisih 0 sampai dengan 2 pascal. Kondisi NS adalah kondisi dimana nilai tekanan terukur oleh sensor dibawah dari nilai tekanan yang diinginkan tetapi dengan selisih 1 sampai dengan 3 pascal. Kondisi NB adalah kondisi dimana nilai tekanan terukur oleh sensor dibawah dari nilai tekanan yang diinginkan tetapi dengan selisih lebih dari 2 pascal. Kondisi PK adalah kondisi dimana nilai tekanan terukur oleh sensor diatas dari nilai tekanan yang diinginkan tetapi dengan selisih antara 0 sampai dengan 2 pascal. Kondisi PS adalah kondisi dimana nilai tekanan terukur oleh sensor diatas dari nilai tekanan yang diinginkan tetapi dengan selisih antara 1 sampai dengan 3 pascal. Kondisi PB adalah kondisi dimana nilai tekanan terukur oleh sensor diatas dari nilai tekanan yang diinginkan tetapi dengan selisih lebih dari 2 pascal. Berikut adalah membership function dan desain model fuzzifikasi:



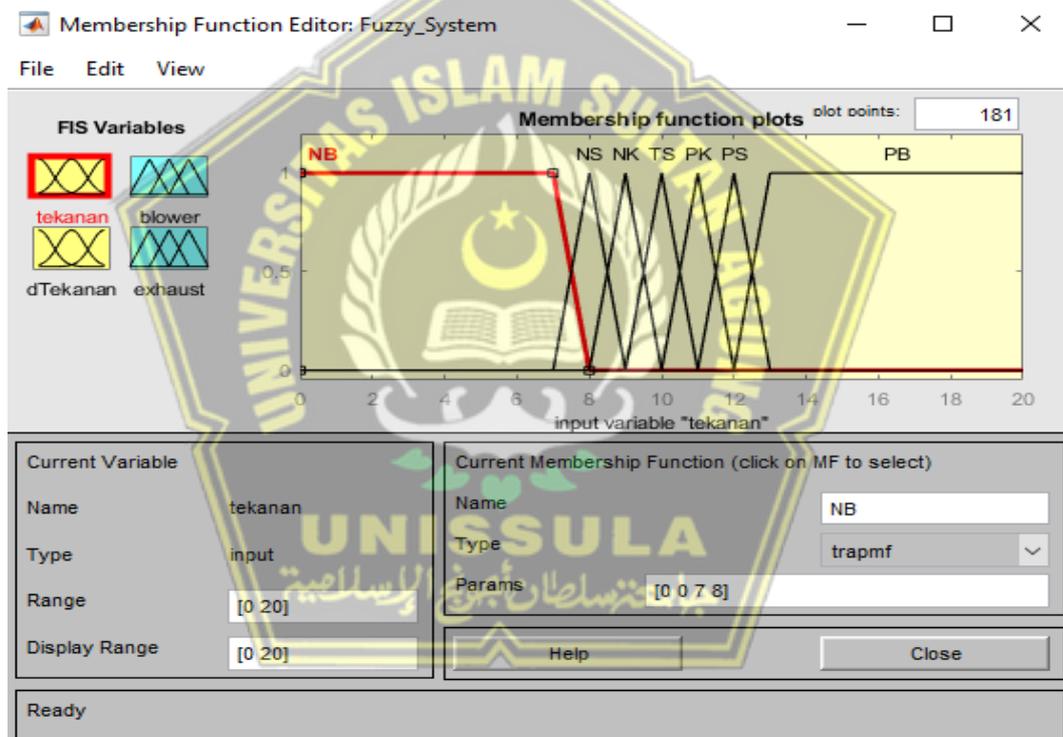
Gambar 3.13 Fungsi keanggotaan Sensor tekanan

Keterangan:

NB : Negative Besar	0 – 8 Pascal
NS : Negative Sedang	7 – 9 Pascal
NK : Negative Kecil	8 – 10 Pascal
TS : Tekanan Sesuai	9 – 11 Pascal
PK : Positive Kecil	10 – 12 Pascal
PS : Positive Sedang	11 – 13 Pascal
PB : Postive Besar	12 – 20 Pascal

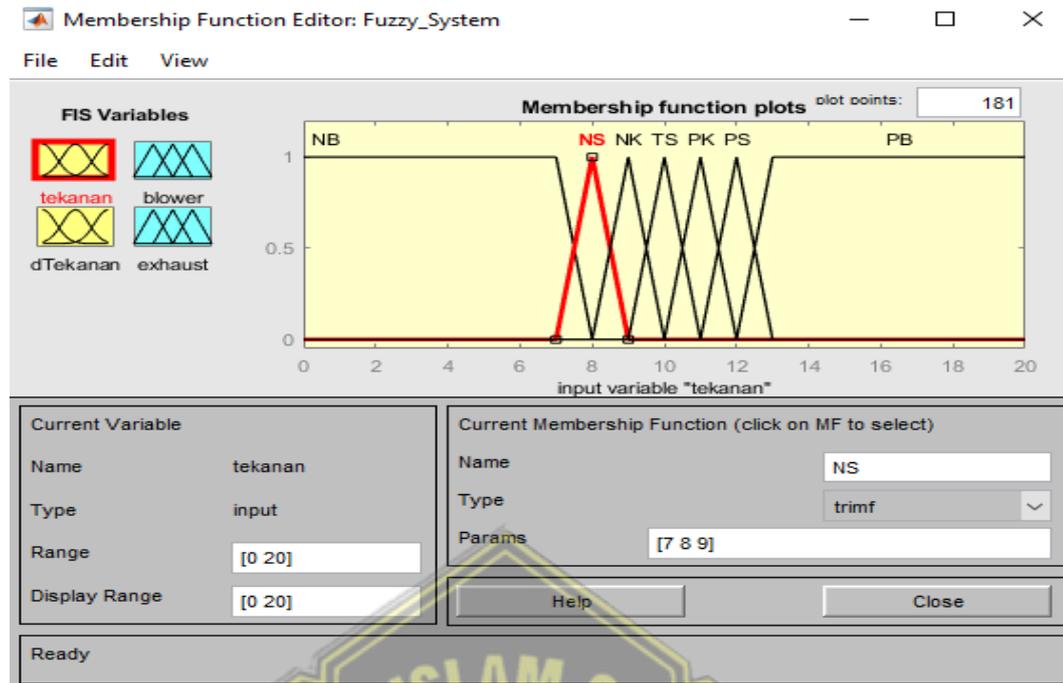
3.3.2 Fuzzy Logic untuk sistem kendali tekanan positif ruang operasi

1. Sensor input parameter membership function

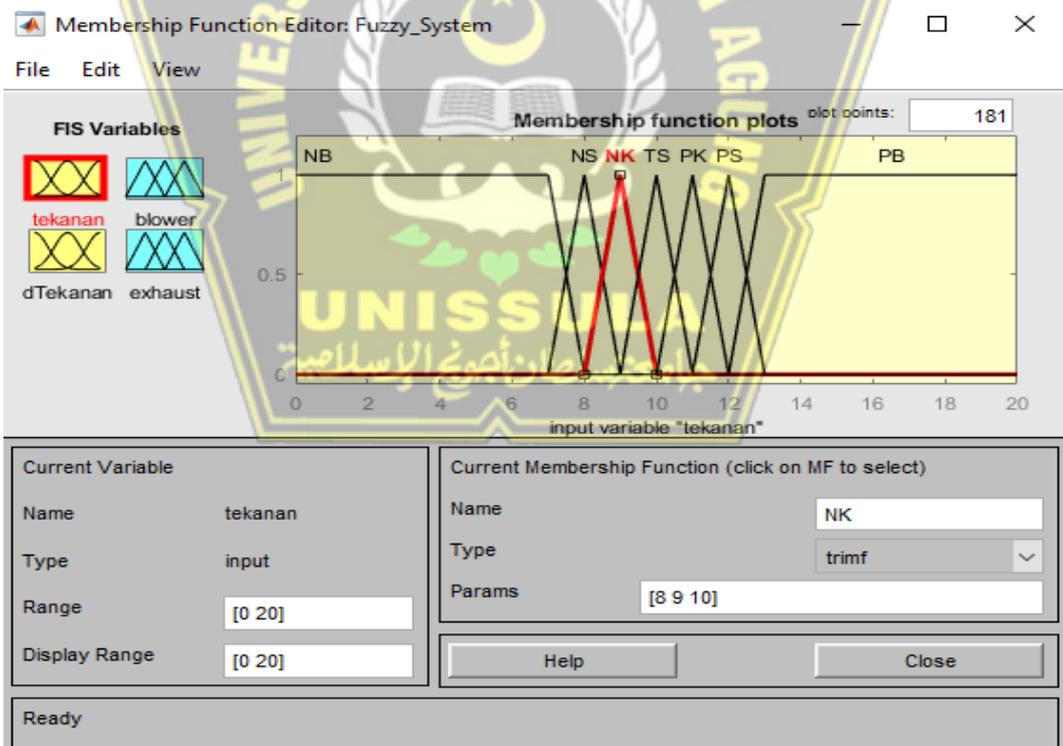


Gambar 3.14 Derajat keanggotaan simpangan negative besar (NB) dari sensor tekanan dengan parameter [0;0;7;8]

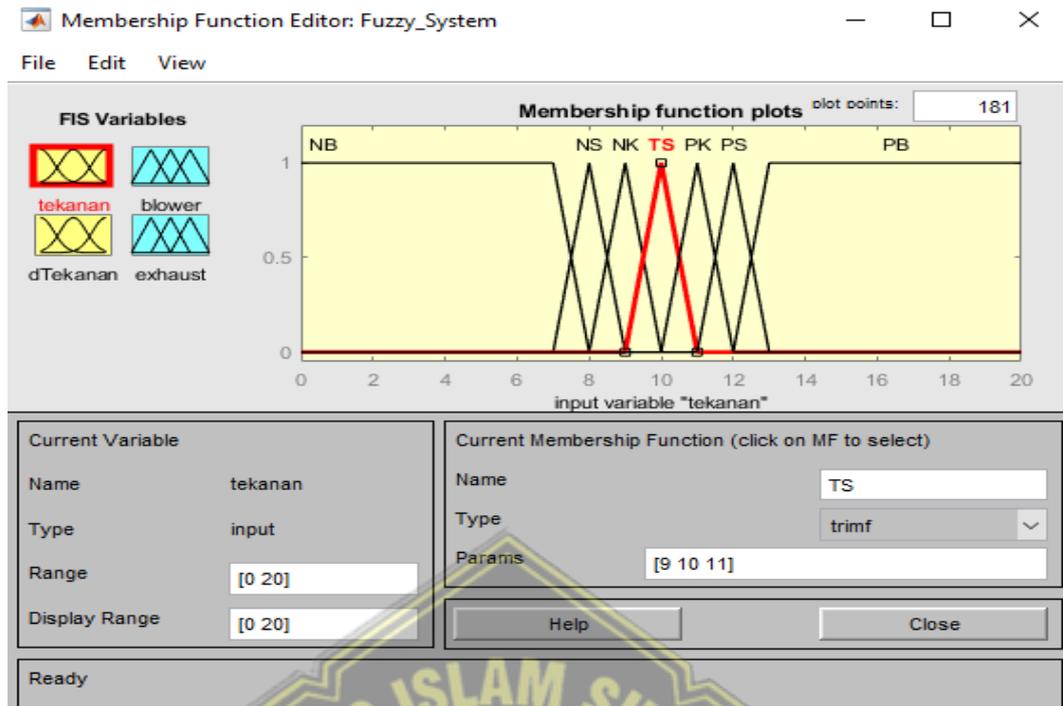
Pada pemodelan input sensor yang dibuat terdapat dua buah tipe model yaitu model trapezium dan model segitiga. Model trapezium memiliki 4 titik seperti yang terlihat pada model NB atau input sensor dengan koefisien negative besar atau bisa dikatakan memiliki nilai lebih kecil atau sama dengan 7 pascal. Titik parameter pada input NB adalah 0;0;7;8.



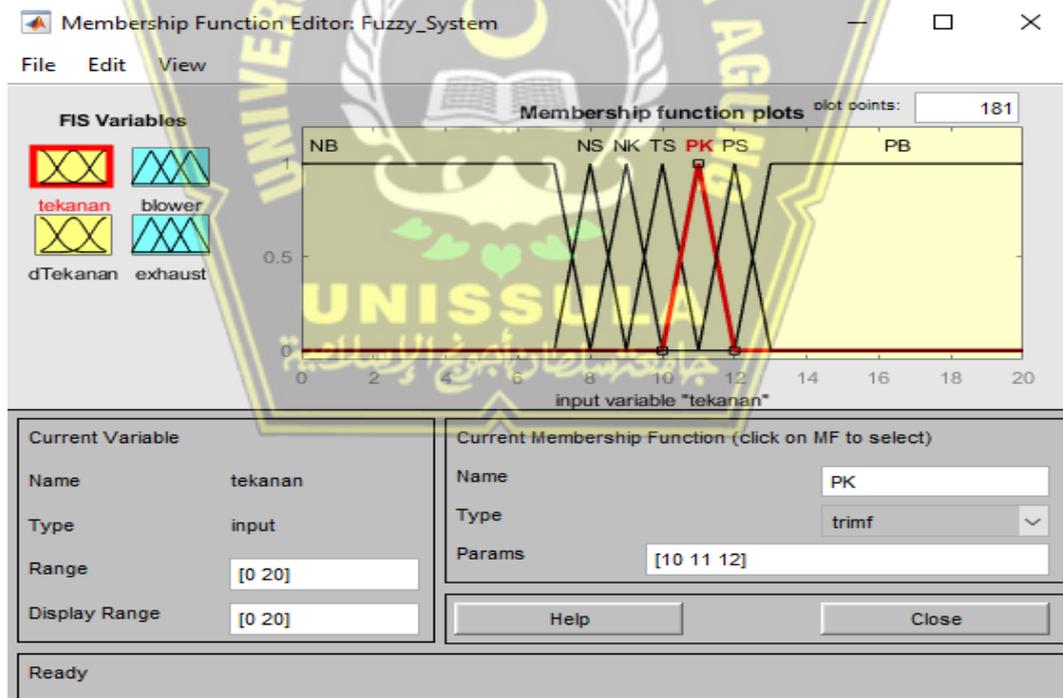
Gambar 3.15 Derajat keanggotaan simpangan negative sedang(NS) dari sensor tekanan dengan parameter [7;8;9]



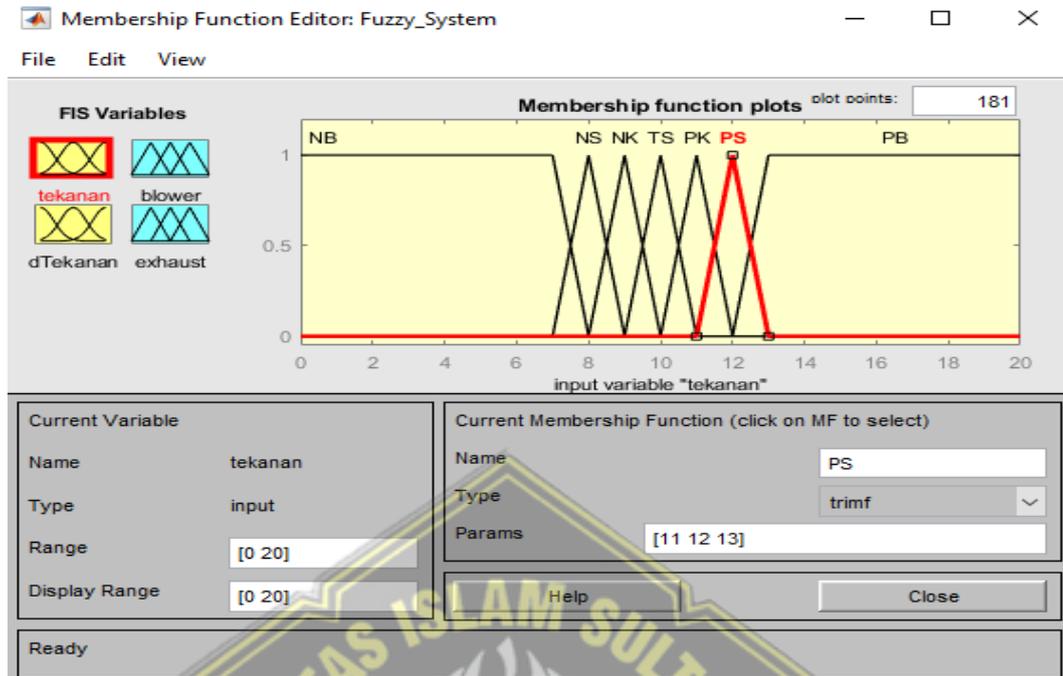
Gambar 3.16 Derajat keanggotaan simpangan negative kecil (kecil) dari sensor tekanan dengan parameter [8;9;10]



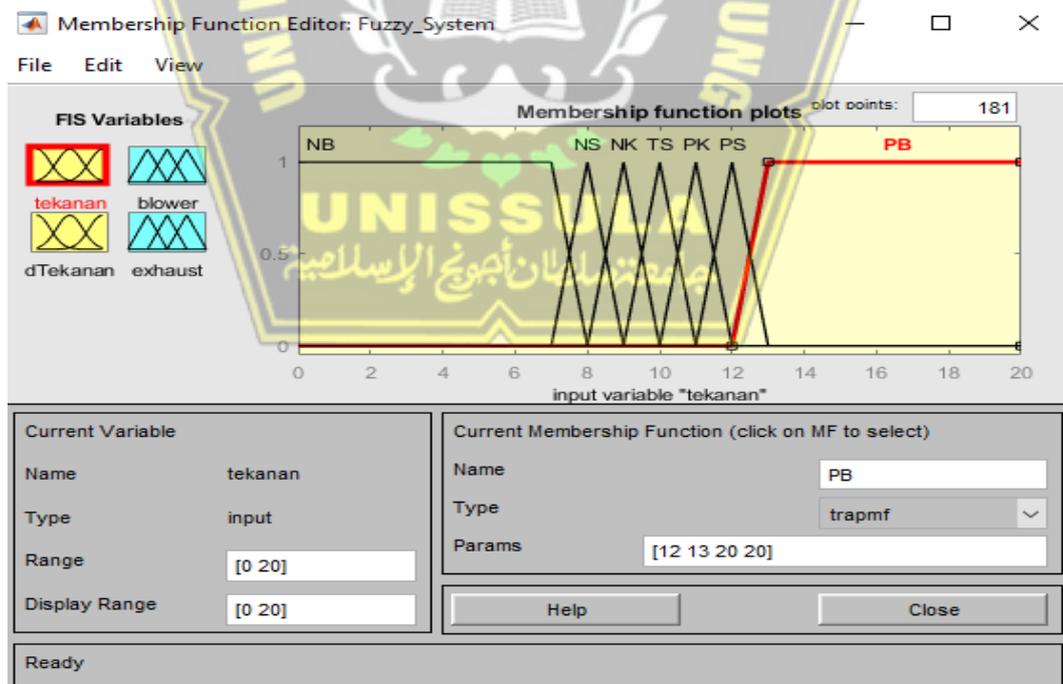
Gambar 3.17 Derajat keanggotaan kondisi sesuai dari sensor tekanan dengan parameter [9;10;11]



Gambar 3.18 Derajat keanggotaan kondisi simpangan positive kecil dari sensor tekanan dengan parameter [10;11;12]



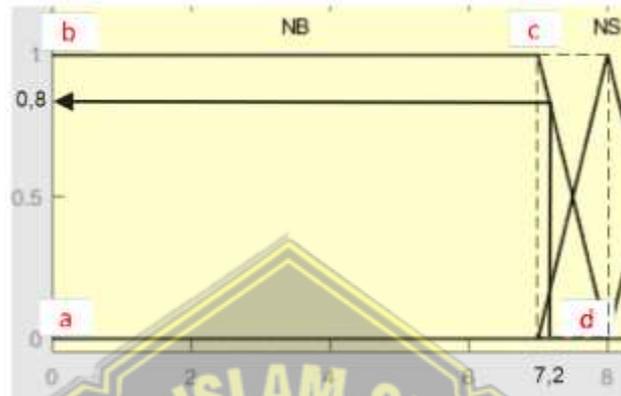
Gambar 3.19 Derajat keanggotaan kondisi simpangan positive sedang dari sensor tekanan dengan parameter [11;12;13]



Gambar 3.20 Derajat keanggotaan kondisi simpangan positive besar dari sensor tekanan dengan input [12;13;20;20]

3.3.2.1 Fuzzyfikasi sensor tekanan dengan fungsi keanggotaan “error simpangan tekanan negative besar”

Dalam perancangan ini error dari fungsi keanggotaan “error simpangan tekanan negative besar” menggunakan aturan trapezium dengan posisi kemiringan di kanan.



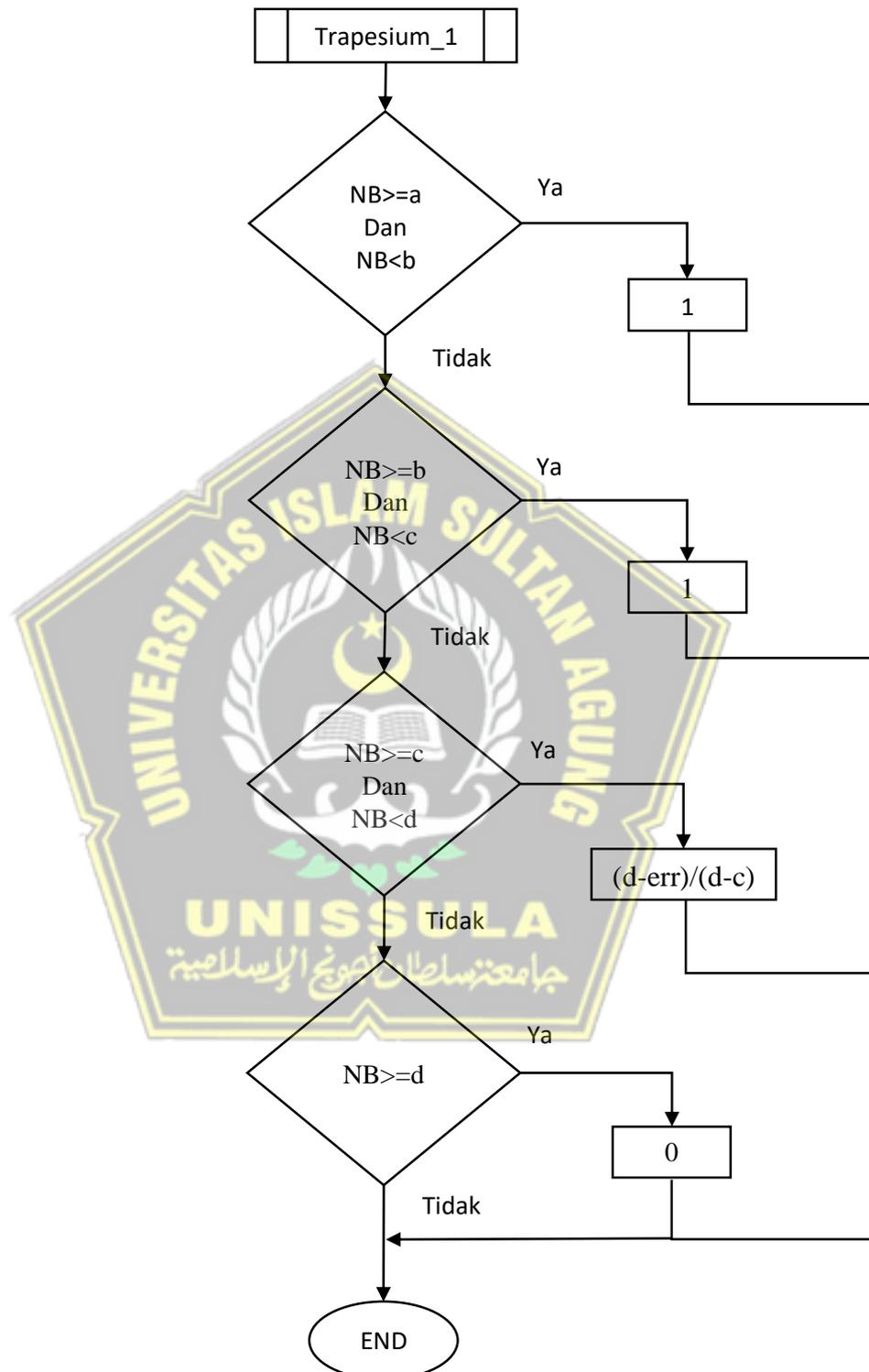
Gambar 3.21 Ruas trapezium untuk input dengan simpangan error negative besar

Dari gambar 3.21 diatas untuk menemukan nilai derajat keanggotaan dari input tekanan dapat dihitung dengan persamaan:

$$\text{Trapezium}(x: a, b, c, d) = \begin{cases} 0, & x \leq a \\ 0, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c \leq x \leq d \\ 0, & x \geq d \end{cases} \quad (3.1)$$

Pada rumus diatas ditunjukkan bahwa model input dari sensor tekanan dengan simpangan negative besar adalah trapesium. Dengan titik a, b, c, d merupakan batasan-batasan dari nilai simpangan negative besar. Pada gambar 3.21 bentuk trapesium yang digambarkan tidak terlihat sempurna karena untuk titik a berada pada titik 0,0 dan titik b berada pada titik 0,1.

Maka dari gambar 3.21 dan persamaan diatas, keanggotaan dapat digambarkan dengan flowchart seperti berikut:



Gambar 3.22 Flowchart Fungsi Keanggotaan simpangan error negative besar

Pada Flowchart Gambar 3.22 ditunjukkan bahwa ketika error yang terjadi adalah antara a dan b dimana a dan b adalah 0 maka nilai keanggotaannya adalah 1, ketika error yang terjadi antara b dan c dimana $b=0$ dan $c=7$ maka nilai keanggotaannya adalah 1, ketika error yang terjadi antara c dan d dimana nilai $c=7$ dan $d=8$ maka nilai keanggotaannya dapat dihitung dengan menggunakan rumus $\mu[x]=(d-x)/(d-c)$ atau $\mu[x]=(8-x)/(8-7)$, sedangkan apabila error melebihi atau sama dengan d maka nilai keanggotaannya adalah 0.

Pada saat kondisi simpangan tekanan negative besar:

Jika tekanan \leq dari 7, maka $\mu[x]$ adalah 1

Jika tekanan ≥ 7 dan ≤ 8 , maka $\mu[x]$ adalah $(8-x)/1$

Jika tekanan \geq dari 8, maka $\mu[x]$ adalah 0

Dari persamaan diatas dapat dicontohkan sebagai berikut

Misal tekanan terukur 7,2 maka:

$$\begin{aligned}\mu_{\text{(negative besar)}}[x] &= (8 - x)/1 \\ &= (8 - 7,2)/1 \\ &= 0,8/1 = 0,8\end{aligned}$$

Misal tekanan terukur 7,4 maka:

$$\begin{aligned}\mu_{\text{(negative besar)}}[x] &= (8 - x)/1 \\ &= (8 - 7,4)/1 \\ &= 0,6\end{aligned}$$

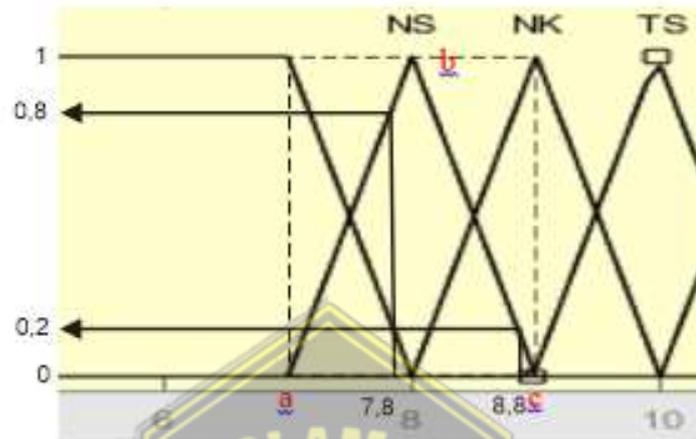
Misal tekanan terukur 7,8 maka:

$$\begin{aligned}\mu_{\text{(negative besar)}}[x] &= (8 - x)/1 \\ &= (8 - 7,8)/1 \\ &= 0,2\end{aligned}$$

Dari contoh yang diperlihatkan nilai koefisien negative besar pada sisi diagonal memiliki nilai semakin kecil jika posisi perpotongan dengan model inputan lain semakin besar.

3.3.2.2 Fuzzyfikasi sensor tekanan dengan fungsi keanggotaan “error simpangan tekanan negative sedang”

Dalam perancangan ini error dari fungsi keanggotaan “error simpangan tekanan negative sedang” menggunakan aturan segitiga.



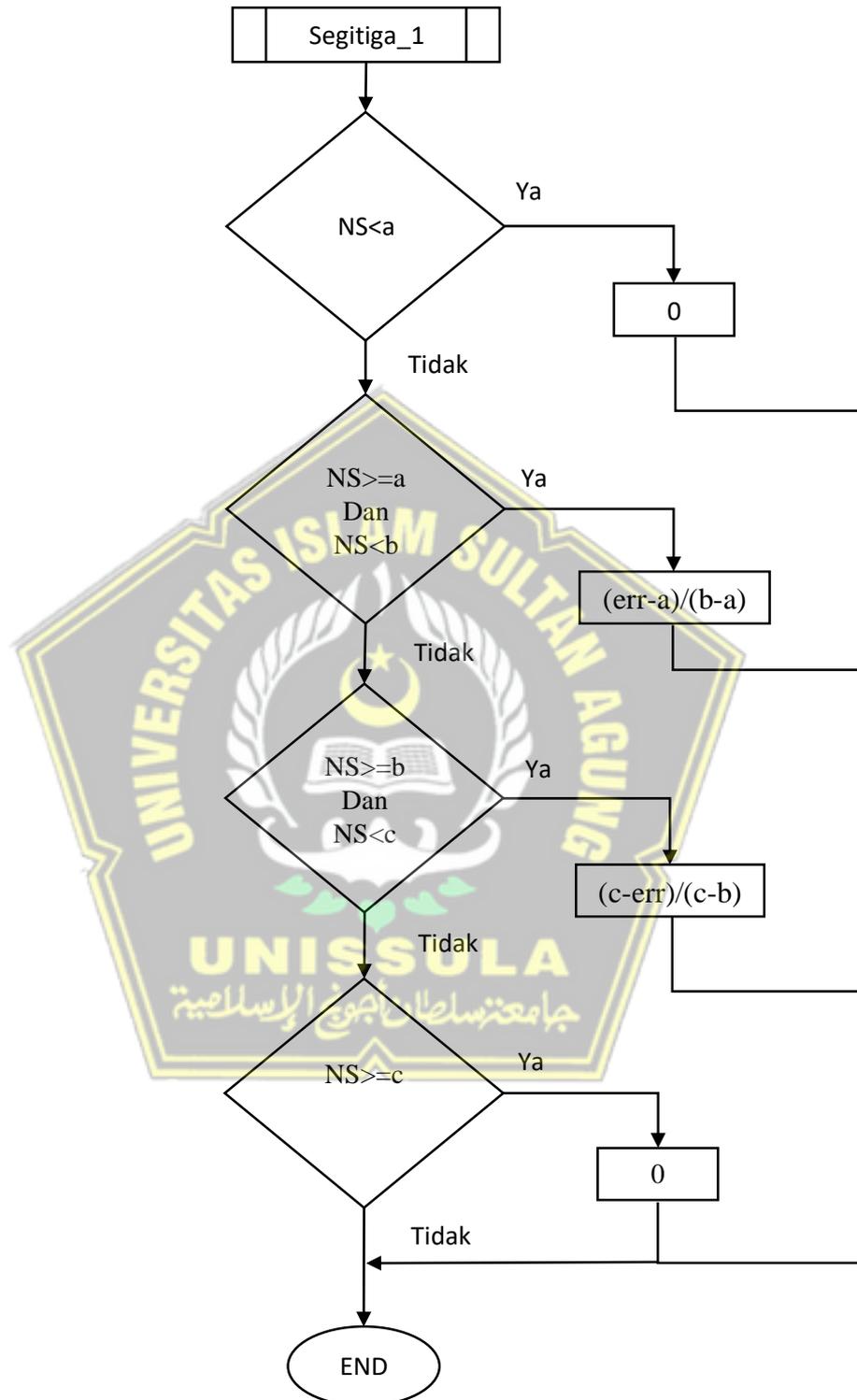
Gambar 3.23 Ruas segitiga untuk input dengan simpangan error negative sedang.

Dari gambar 3.23 diatas untuk menemukan nilai derajat keanggotaan dari input tekanan dapat dihitung dengan persamaan:

$$\text{Segitiga}(x: a, b, c) = \begin{cases} 0 & x < a \text{ atau } x > c \\ \frac{(x - a)}{(b - a)} & a \leq x \leq b \\ \frac{(c - x)}{(c - b)} & b \leq x \leq c \end{cases} \quad (3.2)$$

Dalam menentukan rumus untuk mencari derajat keanggotaan dari inputan sensor perlu memperhatikan bentuk tipe model inputan. Nilai derajat keanggotaan yang tidak bersifat diskrit adalah derajat keanggotaan yang terletak pada sisi miring pada model input. Pada gambar 3.23 ditunjukkan bahwa model input negative sedang adalah segitiga yang berpotongan dengan model input negative besar yang berbentuk trapezium dan model inputan negative kecil yang berbentuk segitiga.

Maka dari gambar 3.23 dan persamaan diatas, keanggotaan dapat digambarkan dengan flowchart seperti berikut:



Gambar 3.24 Flowchart Fungsi Keanggotaan simpangan error negative sedang.

Pada Flowchart 3.24 ditunjukkan bahwa ketika error yang terjadi memiliki nilai kurang dari a maka nilai keanggotaannya adalah 0, ketika error yang terjadi antara a dan b dimana $a=7$ dan $b=8$ maka nilai keanggotaannya adalah $(err-a)/(b-a)$ atau $\mu[x]=(error-7)/(8-7)$, ketika error yang terjadi antara b dan c dimana nilai $b=8$ dan $c=9$ maka nilai keanggotaannya dapat dihitung dengan menggunakan rumus $\mu[x]=(c-error)/(c-b)$ atau $\mu[x]=(9-error)/(9-8)$, sedangkan apabila error melebihi c maka nilai keanggotaannya adalah 0.

Pada saat kondisi simpangan tekanan negative sedang:

Jika tekanan \leq dari 7, maka $\mu[x]$ adalah 0

Jika tekanan ≥ 7 dan ≤ 8 , maka $\mu[x]$ adalah $(x-7)/(8-7)$

Jika tekanan ≥ 8 dan ≤ 9 , maka $\mu[x]$ adalah $(9-x)/(9-8)$

Jika tekanan \geq dari 9, maka $\mu[x]$ adalah 0

Dari persamaan diatas dapat dicontohkan sebagai berikut

Misal tekanan terukur 7,8 maka:

$$\begin{aligned}\mu_{\text{(negative sedang)}}[x] &= (x-7)/1 \\ &= (7,8-7)/1 \\ &= 0,8/1 = 0,8\end{aligned}$$

Misal tekanan terukur 7,4 maka:

$$\begin{aligned}\mu_{\text{(negative sedang)}}[x] &= (x-7)/1 \\ &= (7,4-7)/1 \\ &= 0,4/1 = 0,4\end{aligned}$$

Misal tekanan terukur 8,8 maka:

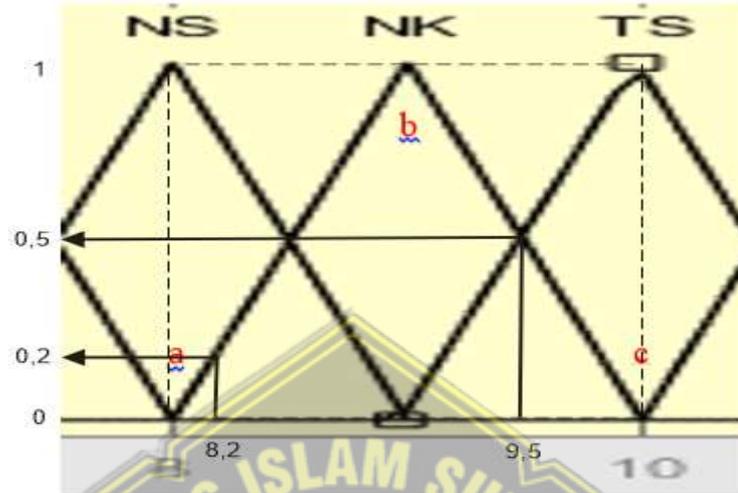
$$\begin{aligned}\mu_{\text{(negative sedang)}}[x] &= (9 - x)/1 \\ &= (9 - 8,8)/1 \\ &= 0,2/1 = 0,2\end{aligned}$$

Misal tekanan terukur 8,3 maka:

$$\begin{aligned}\mu_{\text{(negative sedang)}}[x] &= (9 - x)/1 \\ &= (9 - 8,3)/1 \\ &= 0,3/1 = 0,3\end{aligned}$$

3.3.2.3 Fuzzyfikasi sensor tekanan dengan fungsi keanggotaan “error simpangan tekanan negative kecil”

Dalam perancangan ini error dari fungsi keanggotaan “error simpangan tekanan negative kecil” menggunakan aturan segitiga.



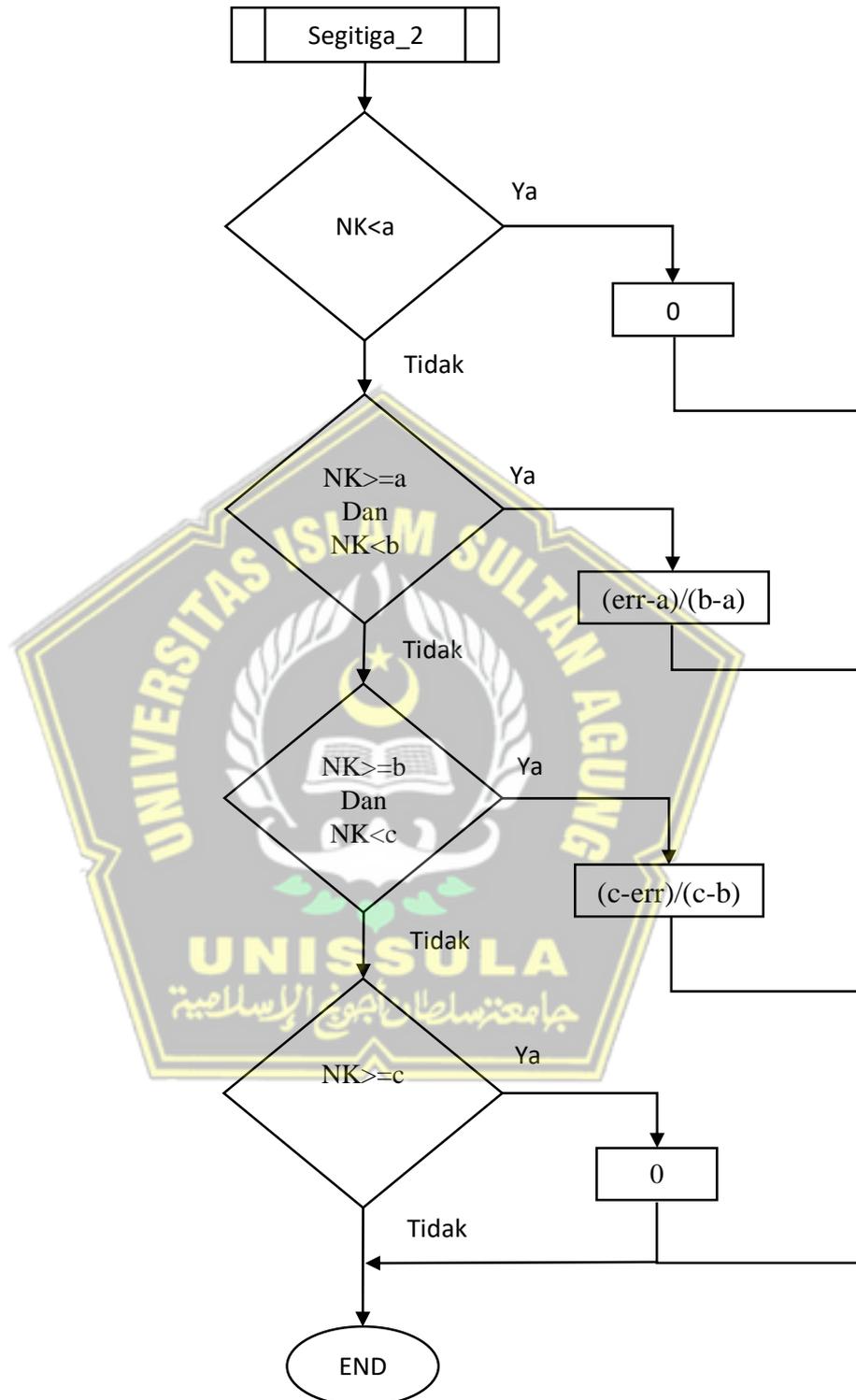
Gambar 3.25 Ruas segitiga untuk input dengan simpangan error negative kecil.

Dari gambar 3.25 diatas untuk menemukan nilai derajat keanggotaan dari input tekanan dapat dihitung dengan persamaan:

$$Segitiga(x: a, b, c) = \begin{cases} 0 & x < a \text{ atau } x > c \\ \frac{(x - a)}{(b - a)} & a \leq x \leq b \\ \frac{(c - x)}{(c - b)} & b \leq x \leq c \end{cases} \quad (3.3)$$

Pada rumus diatas ditunjukkan bahwa model input dari sensor tekanan dengan simpangan negative kecil, berbeda dengan inputan sensor negatif besar model pada inputan ini berbentuk segitiga. Dengan titik a, b, c merupakan batasan-batasan dari nilai simpangan negative kecil. Pada gambar 3.25 bentuk segitiga yang digambarkan saling berpotongan dengan bentuk model input dari klasifikasi inputan yang lain.

Maka dari gambar 3.25 dan persamaan diatas, keanggotaan dapat digambarkan dengan flowchart seperti berikut:



Gambar 3.26 Flowchart Fungsi Keanggotaan simpangan error negative kecil.

Pada Flowchart 3.26 ditunjukkan bahwa ketika error yang terjadi memiliki nilai kurang dari a maka nilai keanggotaannya adalah 0, ketika error yang terjadi antara a dan b dimana $a=8$ dan $b=9$ maka nilai keanggotaannya adalah $(err-a)/(b-a)$ atau $\mu[x]=(error-8)/(9-8)$, ketika error yang terjadi antara b dan c dimana nilai $b=9$ dan $c=10$ maka nilai keanggotaannya dapat dihitung dengan menggunakan rumus $\mu[x]=(c-error)/(c-b)$ atau $\mu[x]=(10-error)/(10-9)$, sedangkan apabila error melebihi c maka nilai keanggotaannya adalah 0.

Pada saat kondisi simpangan tekanan negative kecil:

Jika tekanan \leq dari 8, maka $\mu[x]$ adalah 0

Jika tekanan ≥ 8 dan ≤ 9 , maka $\mu[x]$ adalah $(x-8)/(9-8)$

Jika tekanan ≥ 9 dan ≤ 10 , maka $\mu[x]$ adalah $(10-x)/(10-9)$

Jika tekanan \geq dari 10, maka $\mu[x]$ adalah 0

Dari persamaan diatas dapat dicontohkan sebagai berikut

Misal tekanan terukur 8,2 maka:

$$\begin{aligned}\mu_{\text{(negative kecil)}}[x] &= (x-8)/1 \\ &= (8,2-8)/1 \\ &= 0,2/1 = 0,2\end{aligned}$$

Misal tekanan terukur 8,6 maka:

$$\begin{aligned}\mu_{\text{(negative kecil)}}[x] &= (x-8)/1 \\ &= (8,6-8)/1 \\ &= 0,6/1 = 0,6\end{aligned}$$

Misal tekanan terukur 9,5 maka:

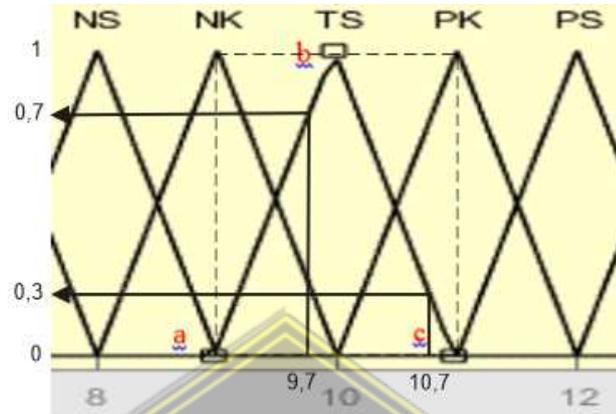
$$\begin{aligned}\mu_{\text{(negative kecil)}}[x] &= (10 - x)/1 \\ &= (10 - 9,5)/1 \\ &= 0,5/1 = 0,5\end{aligned}$$

Misal tekanan terukur 9,8 maka:

$$\begin{aligned}\mu_{\text{(negative kecil)}}[x] &= (10 - x)/1 \\ &= (10 - 9,8)/1 \\ &= 0,8/1 = 0,8\end{aligned}$$

3.3.2.4 Fuzzyfikasi sensor tekanan dengan fungsi keanggotaan “error simpangan tekanan sesuai”

Dalam perancangan ini error dari fungsi keanggotaan “error simpangan tekanan sesuai” menggunakan aturan segitiga.



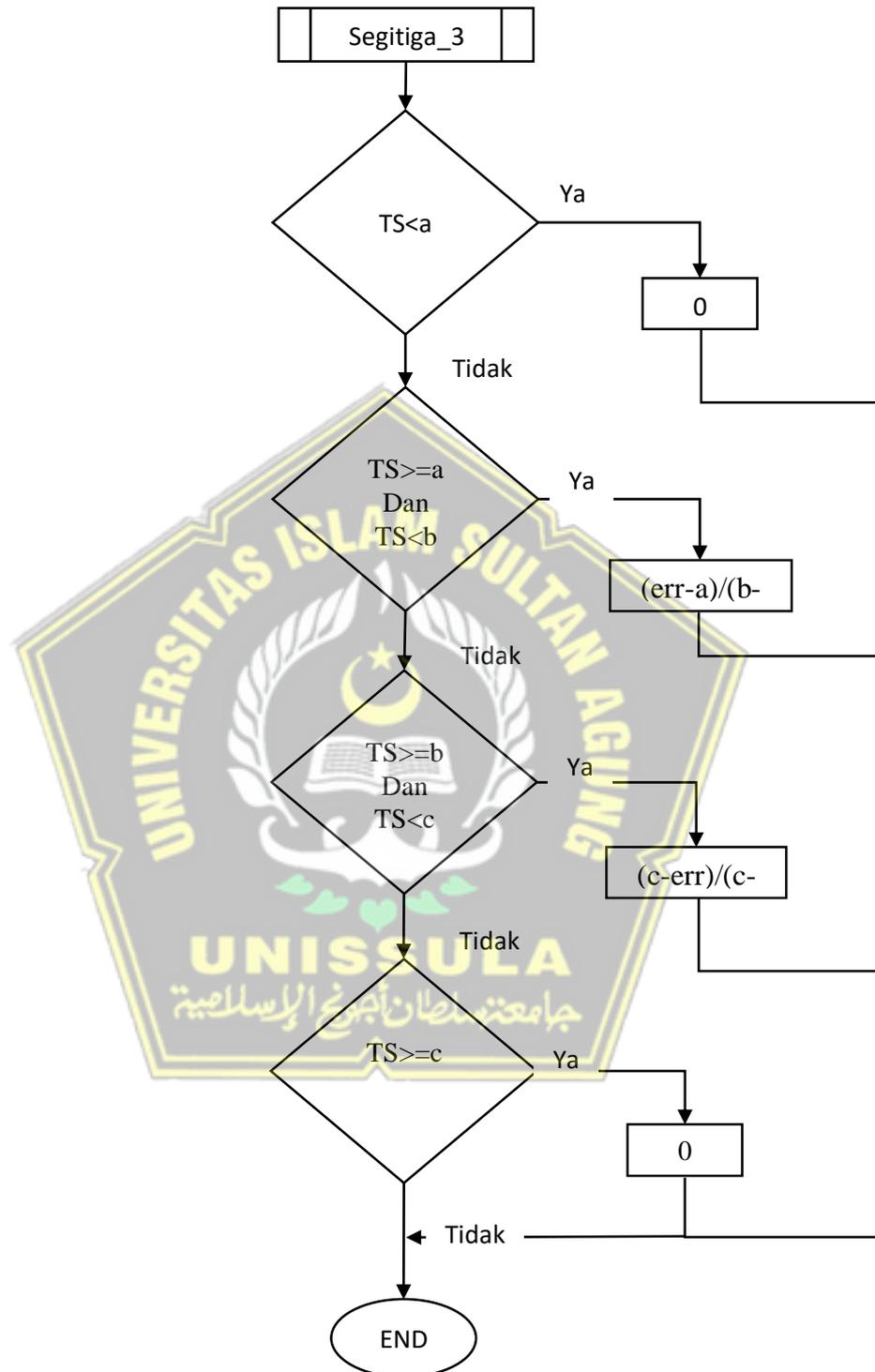
Gambar 3.27 Ruas segitiga untuk input dengan simpangan error tekanan sesuai.

Dari gambar 3.27 diatas untuk menemukan nilai derajat keanggotaan dari input tekanan dapat dihitung dengan persamaan:

$$Segitiga(x: a, b, c) = \begin{cases} 0 & x < a \text{ atau } x > c \\ \frac{(x - a)}{(b - a)} & a \leq x \leq b \\ \frac{(c - x)}{(c - b)} & b \leq x \leq c \end{cases} \quad (3.4)$$

Pada rumus diatas ditunjukkan bahwa model input dari sensor tekanan dengan simpangan tekanan sesuai segitiga. Dengan titik a, b, c merupakan batasan-batasan dari nilai simpangan negative kecil. Pada gambar 3.27 bentuk segitiga yang digambarkan saling berpotongan dengan bentuk model input dari klasifikasi inputan yang lain yang juga berbentuk segitiga.

Maka dari gambar 3.27 dan persamaan diatas, keanggotaan dapat digambarkan dengan flowchart seperti berikut:



Gambar 3.28 Flowchart Fungsi Keanggotaan simpangan error tekanan sesuai.

Pada Flowchart 3.28 ditunjukkan bahwa ketika error yang terjadi memiliki nilai kurang dari a maka nilai keanggotaannya adalah 0, ketika error

yang terjadi antara a dan b dimana $a=9$ dan $b=10$ maka nilai keanggotaannya adalah $(err-a)/(b-a)$ atau $\mu[x]=(error-9)/(10-9)$, ketika error yang terjadi antara b dan c dimana nilai $b=10$ dan $c=11$ maka nilai keanggotaannya dapat dihitung dengan menggunakan rumus $\mu[x]=(c-error)/(c-b)$ atau $\mu[x]=(11-error)/(11-10)$, sedangkan apabila error melebihi c maka nilai keanggotaannya adalah 0.

Pada saat kondisi simpangan tekanan sesuai:

Jika tekanan \leq dari 9, maka $\mu[x]$ adalah 0

Jika tekanan \geq 9 dan \leq 10, maka $\mu[x]$ adalah $(x-9)/(10-9)$

Jika tekanan \geq 10 dan \leq 11, maka $\mu[x]$ adalah $(11-x)/(11-10)$

Jika tekanan \geq dari 11, maka $\mu[x]$ adalah 0

Dari persamaan diatas dapat dicontohkan sebagai berikut

Misal tekanan terukur 9,7 maka:

$$\begin{aligned}\mu_{\text{(tekanan sesuai)}}[x] &= (x-9)/1 \\ &= (9,7-9)/1 \\ &= 0,7/1 = 0,7\end{aligned}$$

Misal tekanan terukur 9,3 maka:

$$\begin{aligned}\mu_{\text{(tekanan sesuai)}}[x] &= (x-9)/1 \\ &= (9,3-9)/1 \\ &= 0,3/1 = 0,3\end{aligned}$$

Misal tekanan terukur 10,7 maka:

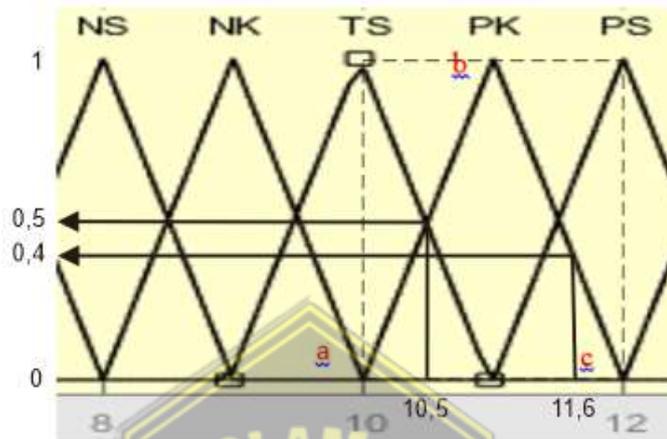
$$\begin{aligned}\mu_{\text{(tekanan sesuai)}}[x] &= (11 - x)/1 \\ &= (11 - 10,7)/1 \\ &= 0,3/1 = 0,3\end{aligned}$$

Misal tekanan terukur 10,3 maka:

$$\begin{aligned}\mu_{\text{(tekanan sesuai)}}[x] &= (11 - x)/1 \\ &= (11 - 10,3)/1 \\ &= 0,7/1 = 0,7\end{aligned}$$

3.3.2.5 Fuzzyfikasi sensor tekanan dengan fungsi keanggotaan “error simpangan positive kecil”

Dalam perancangan ini error dari fungsi keanggotaan “error simpangan tekanan sesuai” menggunakan aturan segitiga.



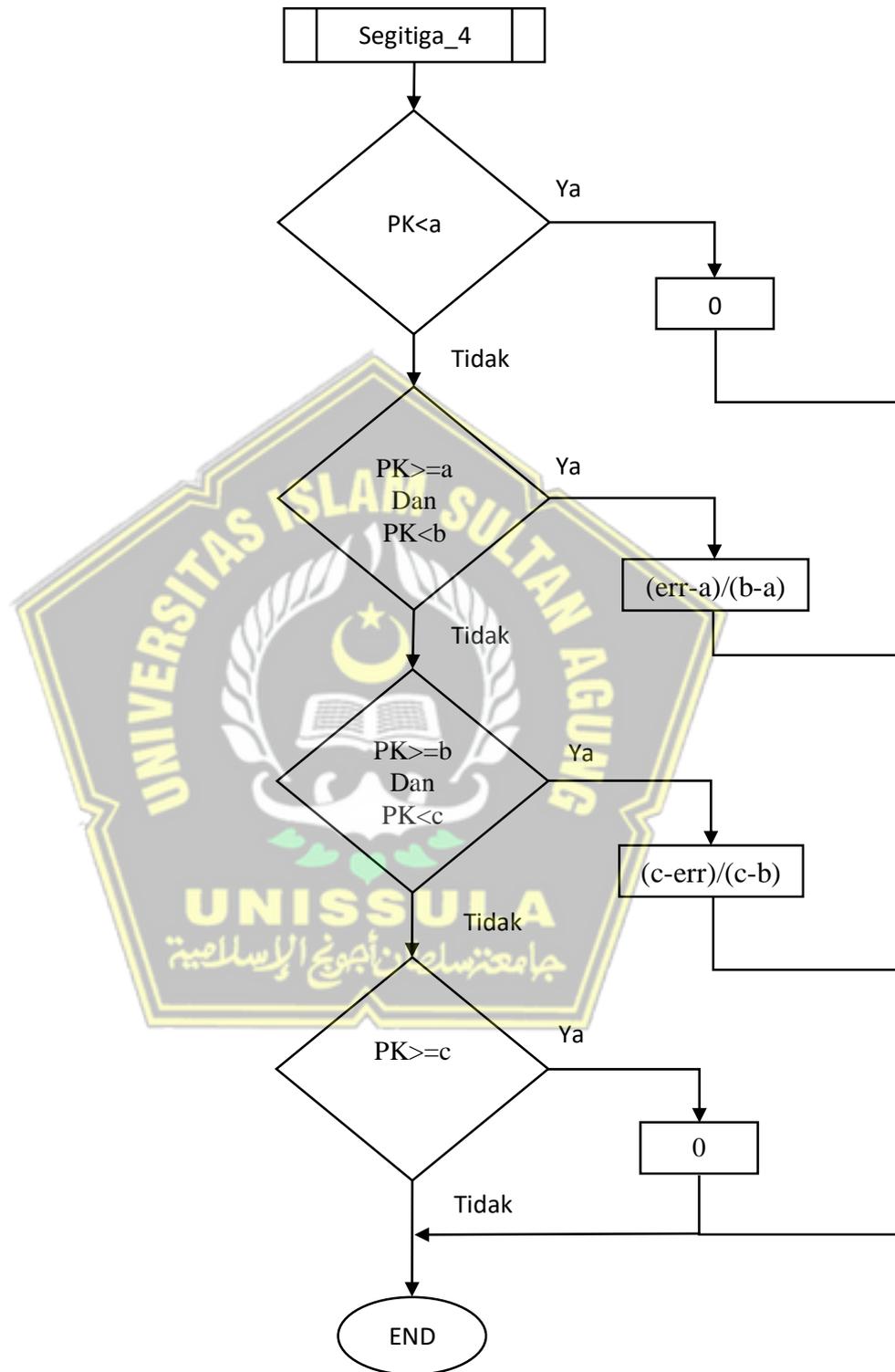
Gambar 3.29 Ruas segitiga untuk input dengan simpangan error tekanan positive kecil.

Dari gambar 3.29 diatas untuk menemukan nilai derajat keanggotaan dari input tekanan dapat dihitung dengan persamaan:

$$\text{Segitiga}(x: a, b, c) = \begin{cases} 0 & x < a \text{ atau } x > c \\ \frac{(x - a)}{(b - a)} & a \leq x \leq b \\ \frac{(c - x)}{(c - b)} & b \leq x \leq c \end{cases} \quad (3.5)$$

Pada rumus diatas ditunjukkan bahwa model input dari sensor tekanan dengan simpangan tekanan positive kecil berbentuk segitiga. Dengan titik a, b, c merupakan batasan-batasan dari nilai simpangan positive kecil. Pada gambar 3.29 bentuk segitiga yang digambarkan saling berpotongan dengan bentuk model input dari klasifikasi inputan yang lain yang juga berbentuk segitiga yaitu model input simpangan tekanan sesuai dan simpangan positif sedang.

Maka dari gambar 3.29 dan persamaan diatas, keanggotaan dapat digambarkan dengan flowchart seperti berikut:



Gambar 3.30 Flowchart Fungsi Keanggotaan simpangan error positif kecil

Pada Flowchart 3.30 ditunjukkan bahwa ketika error yang terjadi memiliki nilai kurang dari a maka nilai keanggotaannya adalah 0, ketika error yang terjadi antara a dan b dimana $a=10$ dan $b=11$ maka nilai keanggotaannya adalah $(err-a)/(b-a)$ atau $y=(error-10)/(11-10)$, ketika error yang terjadi antara b dan c dimana nilai $b=11$ dan $c=12$ maka nilai keanggotaannya dapat dihitung dengan menggunakan rumus $y=(c-error)/(c-b)$ atau $y=(12-error)/(12-11)$, sedangkan apabila error melebihi c maka nilai keanggotaannya adalah 0.

Pada saat kondisi simpangan tekanan positive kecil:

Jika tekanan \leq dari 10, maka $\mu[x]$ adalah 0

Jika tekanan ≥ 10 dan ≤ 11 , maka $\mu[x]$ adalah $(x-10)/(11-10)$

Jika tekanan ≥ 11 dan ≤ 12 , maka $\mu[x]$ adalah $(12-x)/(12-11)$

Jika tekanan \geq dari 12, maka $\mu[x]$ adalah 0

Dari persamaan diatas dapat dicontohkan sebagai berikut

Misal tekanan terukur 10,5 maka:

$$\begin{aligned}\mu_{\text{(positive kecil}[x])} &= (x-10)/1 \\ &= (10,5-10)/1 \\ &= 0,5/1 = 0,5\end{aligned}$$

Misal tekanan terukur 10,8 maka:

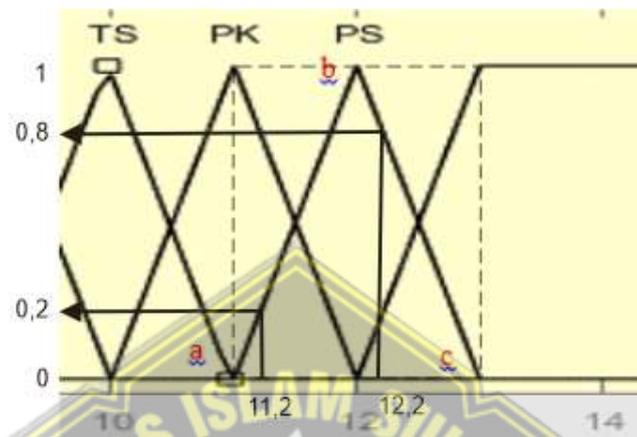
$$\begin{aligned}\mu_{\text{(positive kecil}[x])} &= (x-10)/1 \\ &= (10,8-10)/1 \\ &= 0,8/1 = 0,8\end{aligned}$$

Misal tekanan terukur 11,6 maka:

$$\begin{aligned}\mu_{\text{(positive kecil [x])}} &= (12 - x)/1 \\ &= (12 - 11,6)/1 \\ &= 0,4/1 = 0,4\end{aligned}$$

3.3.2.6 Fuzzyfikasi sensor tekanan dengan fungsi keanggotaan “error simpangan positive sedang”

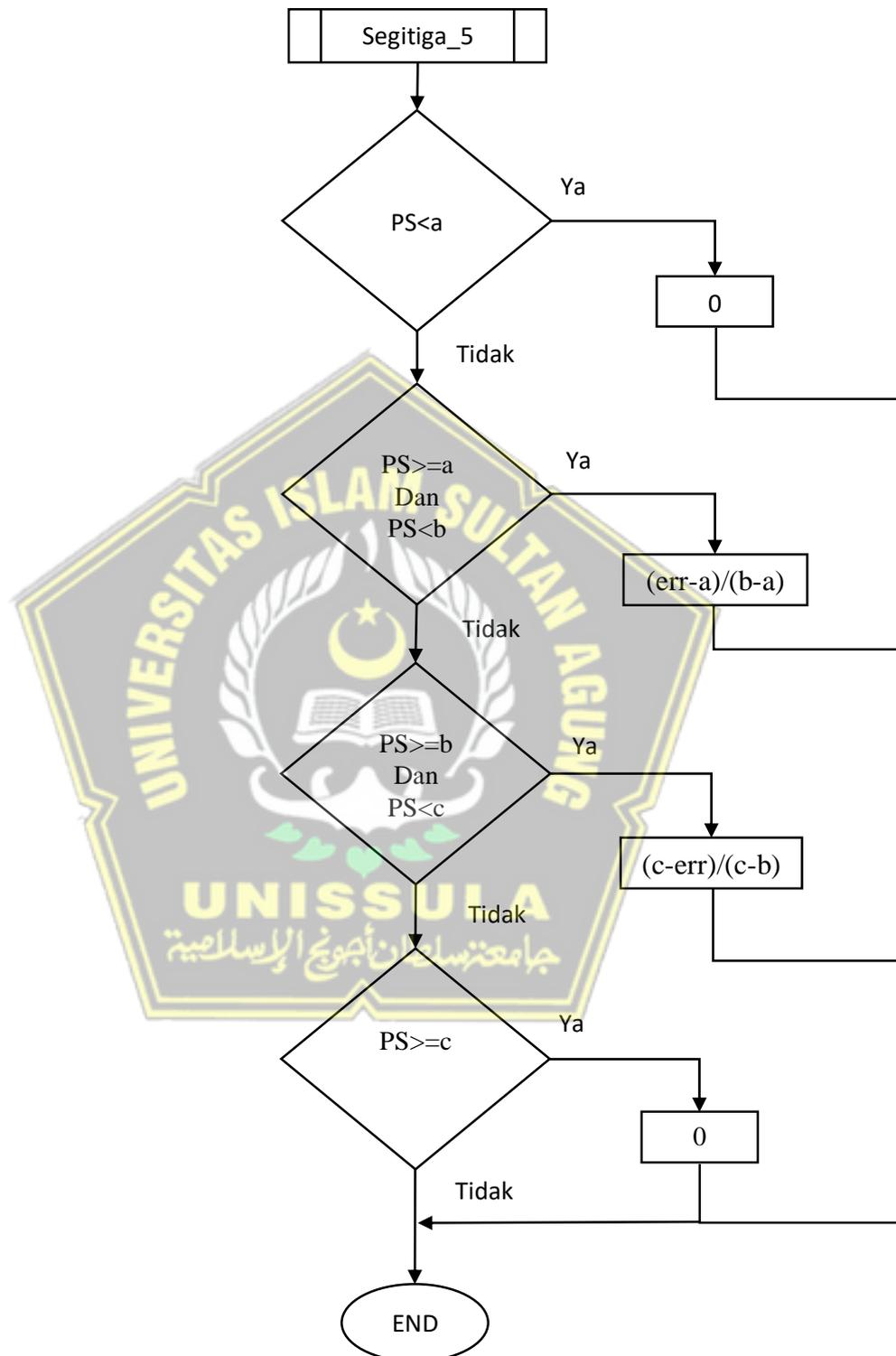
Dalam perancangan ini error dari fungsi keanggotaan “error simpangan tekanan sedang” menggunakan aturan segitiga.



Gambar 3.31 Ruas segitiga untuk input dengan simpangan error tekanan positive sedang.

Dari gambar 3.31 diatas untuk menemukan nilai derajat keanggotaan dari input tekanan dapat dihitung dengan persamaan segitiga seperti diatas. Persamaan segitiga dirumuskan untuk bisa mendapatkan nilai dari sisi diagonal. Dengan melihat pada gambar 3.31 untuk mendapat sebuah nilai respon dapat dilakukan dengan menarik garis secara vertikal dari nilai input sensor sampai pada batasan model segitiga, dilanjutkan dengan menarik garis secara horizontal menuju pada garis nilai respon. Nilai respon yang didapatkan selanjutnya akan dijadikan salah satu faktor untuk menentukan besarnya sinyal manipulasi untuk menggerakkan aktuator berupa blower dan exhaust.

Maka dari gambar 3.24 dan persamaan diatas, keanggotaan dapat digambarkan dengan flowchart seperti berikut:



Gambar 3.32 Flowchart Fungsi Keanggotaan simpangan error positive sedang.

Pada Flowchart 3.32 ditunjukkan bahwa ketika error yang terjadi memiliki nilai kurang dari a maka nilai keanggotaannya adalah 0, ketika error yang terjadi antara a dan b dimana $a=11$ dan $b=12$ maka nilai keanggotaannya adalah $(err-a)/(b-a)$ atau $\mu[x]=(error-11)/(12-11)$, ketika error yang terjadi antara b dan c dimana nilai $b=12$ dan $c=13$ maka nilai keanggotaannya dapat dihitung dengan menggunakan rumus $\mu[x]=(c-error)/(c-b)$ atau $\mu[x]=(13-error)/(13-12)$, sedangkan apabila error melebihi c maka nilai keanggotaannya adalah 0.

Pada saat kondisi simpangan tekanan positive sedang:

Jika tekanan \leq dari 10, maka $\mu[x]$ adalah 0

Jika tekanan ≥ 11 dan ≤ 12 , maka $\mu[x]$ adalah $(x-11)/(12-11)$

Jika tekanan ≥ 12 dan ≤ 13 , maka $\mu[x]$ adalah $(13-x)/(13-12)$

Jika tekanan \geq dari 13, maka $\mu[x]$ adalah 0

Dari persamaan diatas dapat dicontohkan sebagai berikut,

Misal tekanan terukur 11,2 maka:

$$\begin{aligned}\mu_{\text{(positive sedang)}}[x] &= (x-11)/1 \\ &= (11,2-11)/1 \\ &= 0,2/1 = 0,2\end{aligned}$$

Misal tekanan terukur 11,7 maka:

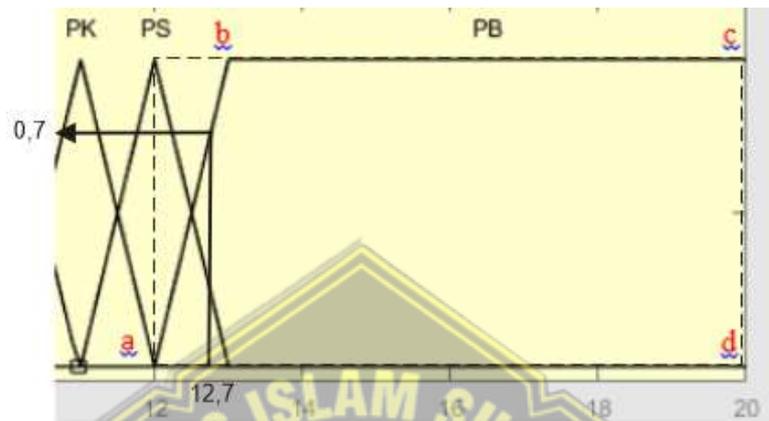
$$\begin{aligned}\mu_{\text{(positive sedang)}}[x] &= (x-11)/1 \\ &= (11,7-11)/1 \\ &= 0,7/1 = 0,7\end{aligned}$$

Misal tekanan terukur 12,2 maka:

$$\begin{aligned}\mu_{\text{(positive sedang)}}[x] &= (13 - x)/1 \\ &= (13 - 12,2)/1 \\ &= 0,8/1 = 0,8\end{aligned}$$

3.3.2.7 Fuzzyfikasi sensor tekanan dengan fungsi keanggotaan “error simpangan tekanan positive besar”

Dalam perancangan ini error dari fungsi keanggotaan “error simpangan tekanan positive besar” menggunakan aturan trapezium dengan posisi kemiringan di kanan.



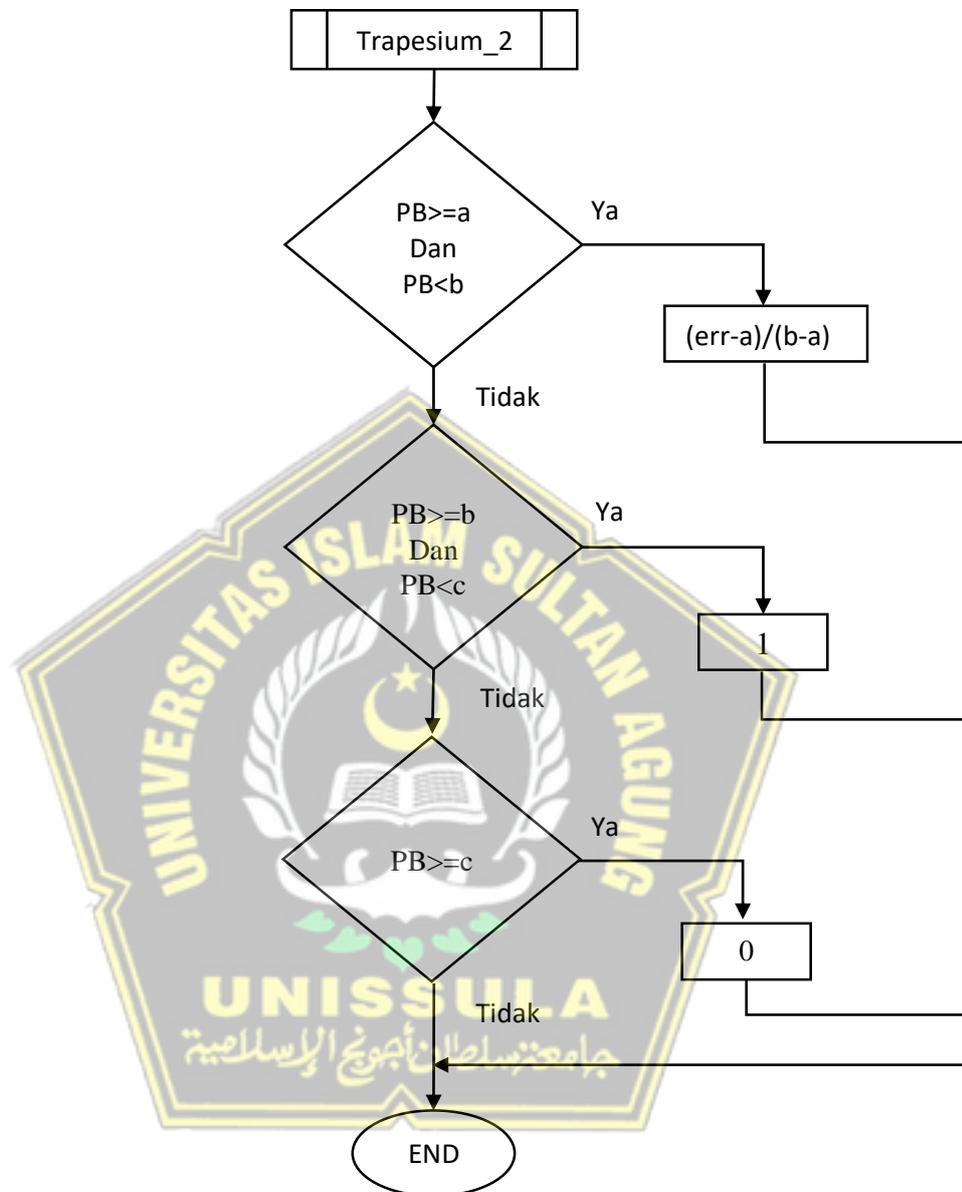
Gambar 3.33 Ruas trapesium untuk input dengan simpangan error positive besar

Dari gambar 3.33 diatas untuk menemukan nilai derajat keanggotaan dari input tekanan dapat dihitung dengan persamaan:

$$\text{Trapezium}(x: a, b, c, d) = \begin{cases} 0, & x \leq a \\ \frac{x - a}{b - a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d - x}{d - c}, & c \leq x \leq d \\ 0, & d \leq x \end{cases} \quad (3.6)$$

Input tekanan yang dihitung dengan persamaan dan menjadi derajat keanggotaan akan menghasilkan nilai bervariasi untuk mendukung detail respon yang lebih baik dengan inputan sensor yang bernilai float atau memiliki nilai koma.

Maka dari gambar 3.26 dan persamaan diatas, keanggotaan dapat digambarkan dengan flowchart seperti berikut:



Gambar 3.34 Flowchart Fungsi Keanggotaan simpangan error positive besar

Pada Flowchart 3.34 ditunjukkan bahwa ketika error yang terjadi adalah antara a dan b maka nilai keanggotaannya adalah $\mu[x]=\frac{err-a}{b-a}$ atau $\mu[x]=\frac{err-12}{13-12}$, ketika error yang terjadi antara b dan c maka nilai keanggotaannya adalah 1, ketika error yang terjadi antara c dan d dimana nilai

maka nilai keanggotaannya adalah 0, sedangkan apabila error melebihi atau sama dengan d maka nilai keanggotaannya adalah juga 0.

Pada saat kondisi simpangan tekanan positive besar:

Jika tekanan \leq dari 12, maka $\mu[x]$ adalah 0

Jika tekanan \geq 12 dan \leq 13, maka $\mu[x]$ adalah $(x-12)/1$

Jika tekanan \geq dari 13, maka $\mu[x]$ adalah 1

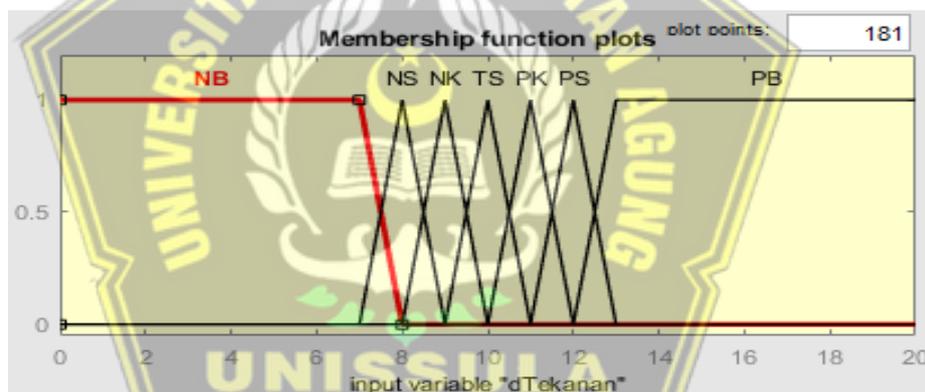
Dari persamaan diatas dapat dicontohkan sebagai berikut

Misal tekanan terukur 12,7 maka:

$$\begin{aligned} \mu_{\text{(positive besar}[x])} &= (x- 12)/1 \\ &= (12,7 - 12)/1 \\ &= 0,7/1 = 0,7 \end{aligned}$$

3.3.3 Turunan Input Sensor Tekanan

Input kedua yang membantu dalam proses fuzzifikasi adalah input yang berasal dari nilai turunan dari sensor tekanan dengan member seperti berikut:



Gambar 3.35 Anggota himpunan dari turunan sensor tekanan

Karena merupakan turunan dari pembacaan sesnsor tekanan makan model fungsi yang digunakan sama dengan model fungsi sensor tekanan. Sehingga rumusnya pun juga sama. Adapun bentuk penalaran inputnya adalah sebagai berikut:

Tabel 3.2 Penalaran sistem input

Tekanan \ dTekanan	NB	NS	NK	TS	PK	PS	PB
NB	NB	NB	NB	NS	TS	PK	PS
NS	NB	NB	NB	NS	TS	PK	PS
NK	NB	NB	NS	NK	PK	PS	PB
TS	NB	NS	NK	TS	PK	PS	PB
PK	NB	NS	NK	PK	PS	PB	PB
PS	NS	NS	NK	PK	PB	PB	PB
PB	NS	NK	TS	PS	PB	PB	PB

3.3.4 Proses Penalaran Mesin (Inference Engine)

Proses penalaran yang digunakan untuk penelitian ini adalah MAX – MIN dimana kita ingin mencari nilai maksimum dari perbandingan nilai-nilai minimum pada error. Berikut adalah penalaran yang dirancang untuk output blower:

Tabel 3.3 Penalaran sistem output blower

Tekanan \ Selisih	NB	NS	NK	TS	PK	PS	PB
-4	cepat						
-3	cepat						
-2		sedang					
-1			pelan				
0				stop			
1					Stop		
2						Stop	
3							Stop
4							Stop

Tabel 3.4 Penalaran sistem output exhaust

Tekanan \ Selisih	NB	NS	NK	TS	PK	PS	PB
-4	Stop						
-3	Stop						
-2		Stop					
-1			Stop				
0				stop			
1					Pelan		
2						Sedang	
3							Cepat
4							Cepat

Untuk mendapatkan himpunan sinyal control yang akan dipakai sebagai masukan defuzzyfikasi diperlukan mesin penalaran sebagai berikut $U = \max(U, \min(ETekanan))$.

3.3.4.1 Rules pemodelan Fuzzy

1. If (tekanan is NB) and (dTekanan is NB) then (blower is cepat)(exhaust is stop) (1)
2. If (tekanan is NB) and (dTekanan is NS) then (blower is cepat)(exhaust is stop) (1)

3. If (tekanan is NB) and (dTekanan is NK) then (blower is cepat)(exhaust is stop) (1)

4. If (tekanan is NB) and (dTekanan is TS) then (blower is cepat)(exhaust is stop) (1)

5. If (tekanan is NB) and (dTekanan is PK) then (blower is cepat)(exhaust is stop) (1)

6. If (tekanan is NB) and (dTekanan is PS) then (blower is sedang)(exhaust is stop) (1)

7. If (tekanan is NB) and (dTekanan is PB) then (blower is sedang)(exhaust is stop) (1)

8. If (tekanan is NS) and (dTekanan is NB) then (blower is cepat)(exhaust is stop) (1)

9. If (tekanan is NS) and (dTekanan is NS) then (blower is cepat)(exhaust is stop) (1)

10. If (tekanan is NS) and (dTekanan is NK) then (blower is cepat)(exhaust is stop) (1)

11. If (tekanan is NS) and (dTekanan is TS) then (blower is sedang)(exhaust is stop) (1)

12. If (tekanan is NS) and (dTekanan is PK) then (blower is sedang)(exhaust is stop) (1)

13. If (tekanan is NS) and (dTekanan is PS) then (blower is sedang)(exhaust is stop) (1)

14. If (tekanan is NS) and (dTekanan is PB) then (blower is pelan)(exhaust is stop) (1)

15. If (tekanan is NK) and (dTekanan is NB) then (blower is cepat)(exhaust is stop) (1)

16. If (tekanan is NK) and (dTekanan is NS) then (blower is cepat)(exhaust is stop) (1)

17. If (tekanan is NK) and (dTekanan is NK) then (blower is sedang)(exhaust is stop) (1)

18. If (tekanan is NK) and (dTekanan is TS) then (blower is pelan)(exhaust is stop) (1)

19. If (tekanan is NK) and (dTekanan is PK) then (blower is pelan)(exhaust is stop) (1)

20. If (tekanan is NK) and (dTekanan is PS) then (blower is pelan)(exhaust is stop) (1)

21. If (tekanan is NK) and (dTekanan is PB) then (blower is stop)(exhaust is stop) (1)

22. If (tekanan is TS) and (dTekanan is NB) then (blower is sedang)(exhaust is stop) (1)

23. If (tekanan is TS) and (dTekanan is NS) then (blower is sedang)(exhaust is stop) (1)

24. If (tekanan is TS) and (dTekanan is NK) then (blower is pelan)(exhaust is stop) (1)

25. If (tekanan is TS) and (dTekanan is TS) then (blower is stop)(exhaust is stop) (1)

26. If (tekanan is TS) and (dTekanan is PK) then (blower is stop)(exhaust is pelan) (1)

27. If (tekanan is TS) and (dTekanan is PS) then (blower is stop)(exhaust is pelan) (1)
28. If (tekanan is TS) and (dTekanan is PB) then (blower is stop)(exhaust is sedang) (1)
29. If (tekanan is PK) and (dTekanan is NB) then (blower is stop)(exhaust is stop) (1)
30. If (tekanan is PK) and (dTekanan is NS) then (blower is stop)(exhaust is stop) (1)
31. If (tekanan is PK) and (dTekanan is NK) then (blower is stop)(exhaust is pelan) (1)
32. If (tekanan is PK) and (dTekanan is TS) then (blower is stop)(exhaust is pelan) (1)
33. If (tekanan is PK) and (dTekanan is PK) then (blower is stop)(exhaust is sedang) (1)
34. If (tekanan is PK) and (dTekanan is PS) then (blower is stop)(exhaust is cepat) (1)
35. If (tekanan is PK) and (dTekanan is PB) then (blower is stop)(exhaust is cepat) (1)
36. If (tekanan is PS) and (dTekanan is NB) then (blower is stop)(exhaust is pelan) (1)
37. If (tekanan is PS) and (dTekanan is NS) then (blower is stop)(exhaust is pelan) (1)
38. If (tekanan is PS) and (dTekanan is NK) then (blower is stop)(exhaust is sedang) (1)

39. If (tekanan is PS) and (dTekanan is TS) then (blower is stop)(exhaust is sedang) (1)

40. If (tekanan is PS) and (dTekanan is PK) then (blower is stop)(exhaust is cepat) (1)

41. If (tekanan is PS) and (dTekanan is PS) then (blower is stop)(exhaust is cepat) (1)

42. If (tekanan is PS) and (dTekanan is PB) then (blower is stop)(exhaust is cepat) (1)

43. If (tekanan is PB) and (dTekanan is NB) then (blower is stop)(exhaust is sedang) (1)

44. If (tekanan is PB) and (dTekanan is NS) then (blower is stop)(exhaust is sedang) (1)

45. If (tekanan is PB) and (dTekanan is NK) then (blower is stop)(exhaust is cepat) (1)

46. If (tekanan is PB) and (dTekanan is TS) then (blower is stop)(exhaust is cepat) (1)

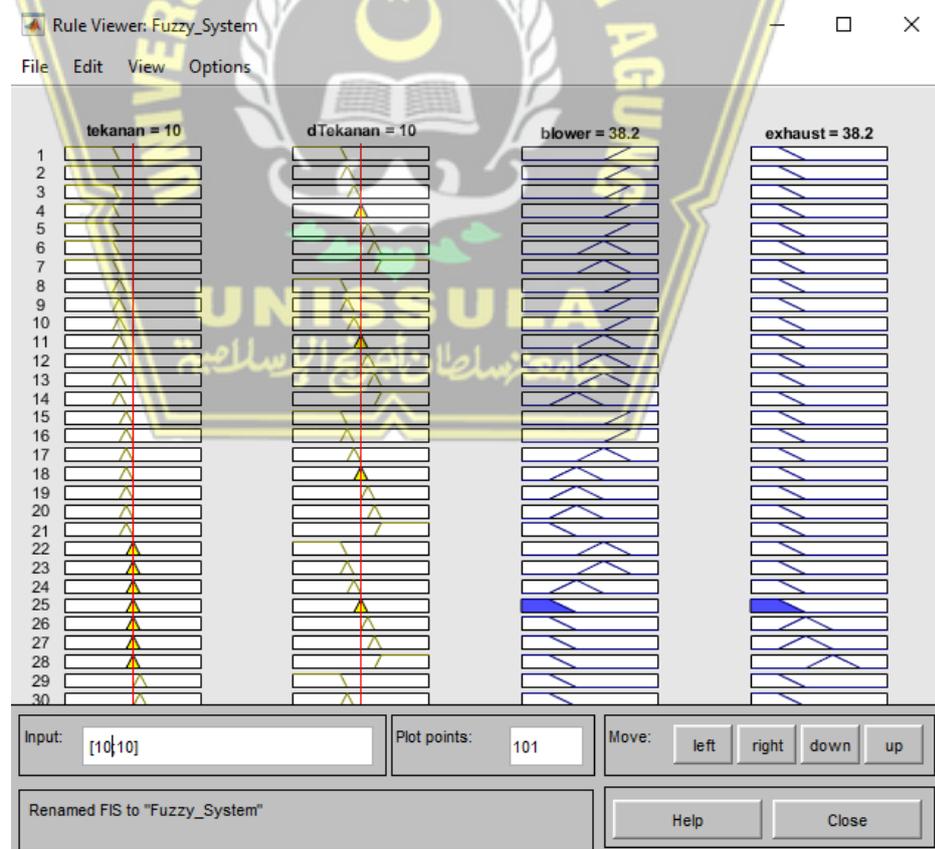
47. If (tekanan is PB) and (dTekanan is PK) then (blower is stop)(exhaust is cepat) (1)

48. If (tekanan is PB) and (dTekanan is PS) then (blower is stop)(exhaust is cepat) (1)

49. If (tekanan is PB) and (dTekanan is PB) then (blower is stop)(exhaust is cepat) (1)

Pada pemodelan Fuzzy dalam penelitian ini ada 49 rules yang ditentukan seperti penalaran yang sudah dijelaskan di atas dimana:

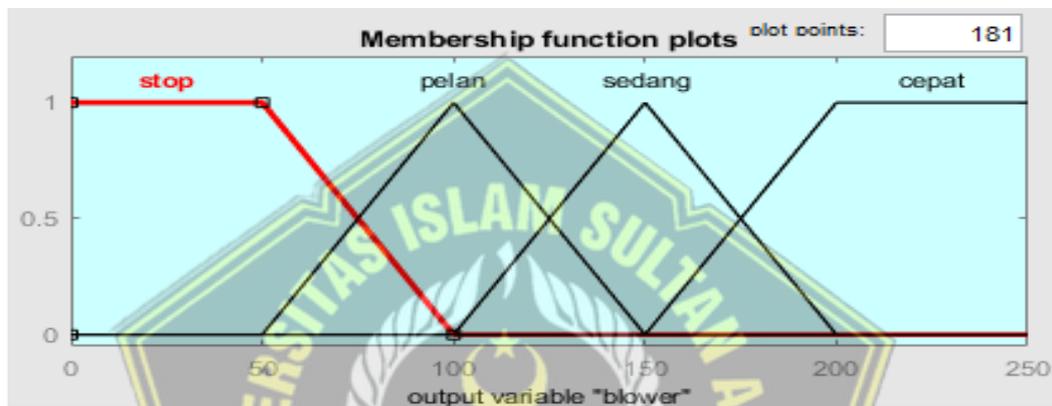
- a. Jika nilai sensor sesuai dengan set poin (TS) maka blower berputar pelan dan exhaust berhenti berputar.
- b. Jika nilai sensor selisih dari set poin dengan nilai penalaran negative kecil (NK) maka blower berputar pelan dan exhaust berhenti berputar.
- c. Jika nilai sensor selisih dari set poin dengan nilai penalaran negative sedang (NS) maka blower berputar sedang dan exhaust berhenti berputar.
- d. Jika nilai sensor selisih dari set poin dengan nilai penalaran negative besar (NB) maka blower berputar cepat dan exhaust berhenti berputar.
- e. Jika nilai sensor selisih dari set poin dengan nilai penalaran positive kecil (PK) maka blower berputar pelan dan exhaust berputar pelan.
- f. Jika nilai sensor selisih dari set poin dengan nilai penalaran positive sedang (PS) maka blower berputar pelan dan exhaust berputar sedang.
- g. Jika nilai sensor selisih dari set poin dengan nilai penalaran positive besar (PB) maka blower berhenti berputar dan exhaust berputar cepat.



Gambar 3.36 Pemodelan Rules Fuzzy Logic

3.3.5 Proses Defuzzyfikasi

Proses terakhir dari algoritma fuzzy adalah proses defuzzyfikasi yang akan merubah himpunan sinyal control dalam fuzzy menjadi sebuah sinyal control yang digunakan untuk menggerakkan plant. Plant berupa sebuah motor blower dan motor exhaust yang akan mensuplay udara dan membuang udara dari dalam miniature ruang operasi. Kondisi pengaturan nilai PWM untuk pengkondisian putaran motor blower dan exhaust akan menghasilkan kondisi putaran blower dan exhaust : Stop, Pelan, Sedang, dan Cepat.



Gambar 3.37 Membership Output Blower

Keanggotaan Blower Stop:

$$\mu_{\text{output - Stop}}[x] = \begin{cases} 1 & x \leq 50 \\ \frac{100 - x}{50} & 50 \leq x \leq 100 \\ 0 & x \geq 100 \end{cases} \quad (3.7)$$

Pada saat output blower Stop :

Jika output \leq dari 50, maka $\mu[x]$ adalah 1

Jika output \geq 50 dan \leq 100, maka $\mu[x]$ adalah $100 - x/50$

Jika output \geq 100, maka $\mu[x]$ adalah 0

Dari persamaan diatas dapat dicontohkan sebagai berikut

Misal output blower 60 maka :

$$\begin{aligned} \mu_{\text{(output - Stop)}}[x] &= (100 - x) / 50 \\ &= (100 - 60) / 50 \\ &= 0,8 \end{aligned}$$

Keanggotaan Blower Pelan:

$$\mu_{\text{output - Pelan}}[x] \begin{cases} 0 & x \leq 50 \\ \frac{x-50}{50}, & 50 \leq x \leq 100 \\ \frac{150-x}{50}, & 100 \leq x \leq 150 \\ 0 & x \geq 150 \end{cases} \quad (3.8)$$

Pada saat output blower pelan:

Jika output \leq dari 50, maka $\mu[x]$ adalah 0

Jika output \geq 50 dan \leq 100, maka $\mu[x]$ adalah $(x-50)/(50)$

Jika output \geq 100 dan \leq 150, maka $\mu[x]$ adalah $(150-x)/(50)$

Jika output \geq dari 150, maka $\mu[x]$ adalah 0

Dari persamaan diatas dapat dicontohkan sebagai berikut

Misal output blower 70 maka:

$$\begin{aligned} \mu_{\text{(output Pelan)}}[x] &= (x-50)/50 \\ &= (70-50)/50 \\ &= 20/50 = 0,4 \end{aligned}$$

Misal output blower 120 maka:

$$\begin{aligned} \mu_{\text{(output Pelan)}}[x] &= (150 - x)/50 \\ &= (150 - 120)/50 \\ &= 30/50 = 0,6 \end{aligned}$$

Keanggotaan Blower Sedang:

$$\mu_{\text{output - Sedang}}[x] \begin{cases} 0 & x \leq 100 \\ \frac{x-100}{50}, & 100 \leq x \leq 150 \\ \frac{200-x}{50}, & 150 \leq x \leq 200 \\ 0 & x \geq 200 \end{cases} \quad (3.9)$$

Pada saat output blower sedang:

Jika output \leq dari 100, maka $\mu[x]$ adalah 0

Jika output \geq 100 dan \leq 150, maka $\mu[x]$ adalah $(x-100)/50$

Jika output \geq 150 dan \leq 200, maka $\mu[x]$ adalah $(200-x)/50$

Jika output \geq dari 200, maka $\mu[x]$ adalah 0

Dari persamaan diatas dapat dicontohkan sebagai berikut

Misal output blower 130 maka:

$$\begin{aligned}\mu_{\text{(output Pelan}[x]} &= (x-100)/50 \\ &= (130-100)/50 \\ &= 30/50 = 0,6\end{aligned}$$

Misal output blower 180 maka:

$$\begin{aligned}\mu_{\text{(output Pelan}[x]} &= (200 - x)/50 \\ &= (200 - 180)/50 \\ &= 20/50 = 0,4\end{aligned}$$

Keanggotaan Blower Cepat:

$$\mu_{\text{output - Cepat}[x]} = \begin{cases} 0 & x \leq 150 \\ \frac{250 - x}{50}, & 150 \leq x \leq 200 \\ 1 & x \geq 200 \end{cases} \quad (3.10)$$

Pada saat output blower Cepat :

Jika output \leq dari 150, maka $\mu[x]$ adalah 0

Jika output \geq 150 dan \leq 200 , maka $\mu[x]$ adalah $200 - x/50$

Jika output \geq 200, maka $\mu[x]$ adalah 1

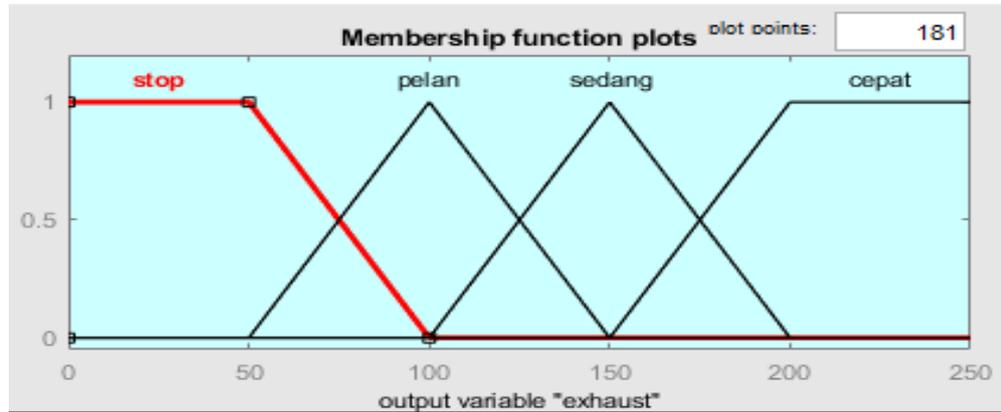
Dari persamaan diatas dapat dicontohkan sebagai berikut

Misal output blower 190 maka :

$$\begin{aligned}\mu_{\text{(output -Stop}[x]} &= (200 - x) / 50 \\ &= (200 - 190) / 50 \\ &= 10/50 = 0,2\end{aligned}$$

Misal output blower 175 maka :

$$\begin{aligned}\mu_{\text{(output -Stop}[x]} &= (200 - x) / 50 \\ &= (200 - 175) / 50 \\ &= 25/50 = 0,5\end{aligned}$$



Gambar 3.38 Membership Output Exhaust

Keanggotaan Exhaust Stop:

$$\mu_{\text{output - Stop}}[x] = \begin{cases} 1 & x \leq 50 \\ \frac{100 - x}{50} & 50 \leq x \leq 100 \\ 0 & x \geq 100 \end{cases} \quad (3.11)$$

Pada saat output Exhaust Stop :

Jika output \leq dari 50, maka $\mu[x]$ adalah 1

Jika output \geq 50 dan \leq 100, maka $\mu[x]$ adalah $(100 - x)/50$

Jika output \geq 100, maka $\mu[x]$ adalah 0

Dari persamaan diatas dapat dicontohkan sebagai berikut

Misal output Exhaust 80 maka :

$$\begin{aligned} \mu_{\text{(output - Stop)}}[x] &= (100 - x)/50 \\ &= (100 - 80)/50 \\ &= 0,4 \end{aligned}$$

Keanggotaan Exhaust Pelan:

$$\mu_{\text{output - Pelan}}[x] = \begin{cases} 0 & x \leq 50 \\ \frac{x - 50}{50} & 50 \leq x \leq 100 \\ \frac{150 - x}{50} & 100 \leq x \leq 150 \\ 0 & x \geq 150 \end{cases} \quad (3.12)$$

Pada saat output Exhaust pelan:

Jika output \leq dari 50, maka $\mu[x]$ adalah 0

Jika output \geq 50 dan \leq 100, maka $\mu[x]$ adalah $(x - 50)/(50)$

Jika output ≥ 100 dan ≤ 150 , maka $\mu[x]$ adalah $(150-x)/(50)$

Jika output \geq dari 150, maka $\mu[x]$ adalah 0

Dari persamaan diatas dapat dicontohkan sebagai berikut

Misal output Exhaust 65 maka:

$$\begin{aligned}\mu_{\text{output Pelan}}[x] &= (x-50)/50 \\ &= (65-50)/50 \\ &= 15/50 = 0,3\end{aligned}$$

Misal output Exhaust 130 maka:

$$\begin{aligned}\mu_{\text{output Pelan}}[x] &= (150 - x)/50 \\ &= (150 - 130)/50 \\ &= 20/50 = 0,4\end{aligned}$$

Keanggotaan Exhaust Sedang:

$$\mu_{\text{output - Sedang}}[x] = \begin{cases} 0 & x \leq 100 \\ \frac{x-100}{50}, & 100 \leq x \leq 150 \\ \frac{200-x}{50}, & 150 \leq x \leq 200 \\ 0 & x \geq 200 \end{cases} \quad (3.13)$$

Pada saat output Exhaust sedang:

Jika output \leq dari 100, maka $\mu[x]$ adalah 0

Jika output ≥ 100 dan ≤ 150 , maka $\mu[x]$ adalah $(x-100)/50$

Jika output ≥ 150 dan ≤ 200 , maka $\mu[x]$ adalah $(200-x)/50$

Jika output \geq dari 200, maka $\mu[x]$ adalah 0

Dari persamaan diatas dapat dicontohkan sebagai berikut

Misal output Exhaust 130 maka:

$$\begin{aligned}\mu_{\text{output Pelan}}[x] &= (x-100)/50 \\ &= (130-100)/50 \\ &= 30/50 = 0,6\end{aligned}$$

Misal output Exhaust 180 maka:

$$\begin{aligned}\mu_{\text{output Pelan}}[x] &= (200 - x)/50 \\ &= (200 - 180)/50 \\ &= 20/50 = 0,4\end{aligned}$$

Keanggotaan Exhaust Cepat:

$$\mu_{\text{output - Cepat}}[x] \begin{cases} 0 & x \leq 150 \\ \frac{250 - x}{50} & 150 \leq x \leq 200 \\ 1 & x \geq 200 \end{cases} \quad (3.14)$$

Pada saat output Exhaust Cepat :

Jika output \leq dari 150, maka $\mu[x]$ adalah 0

Jika output \geq 150 dan \leq 200 , maka $\mu[x]$ adalah $200 - x/50$

Jika output \geq 200, maka $\mu[x]$ adalah 1

Dari persamaan diatas dapat dicontohkan sebagai berikut

Misal output blower 185 maka :

$$\begin{aligned} \mu_{\text{(output - Stop)}}[x] &= (200 - x) / 50 \\ &= (200 - 185) / 50 \\ &= 10 / 50 = 0,3 \end{aligned}$$

Rumus Defusifikasi untuk Output Blower:

$$def = \frac{\sum x \mu R(x)}{\sum \mu R(x)} \quad (3.15)$$

$$\begin{aligned} def = & ((x1 * rule1) + (x2 * rule2) + (x3 * rule3) + (x4 * rule4) + (x5 * \\ & rule5) + (x6 * rule6) + (x7 * rule7) + (x8 * rule8) + (x9 * rule9) + (x10 * \\ & rule10) + (x11 * rule11a) + (x11 * rule11b) + (x12 * rule12a) + (x12 * \\ & rule12b) + (x13 * rule13a) + (x13 * rule13b) + (x14 * rule14a) + (x14 * \\ & rule14b) + (x15 * rule15a) + (x15 * rule15b) + (x16 * rule16a) + (x16 * \\ & rule16b) + (x17 * rule17a) + (x17 * rule17b) + (x18 * rule18a) + (x18 * \\ & rule18b) + (x19 * rule19a) + (x19 * rule19b) + (x20 * rule20a) + (x20 * \\ & rule20b) + (x21 * rule21a) + (x21 * rule21b) + (x22 * rule22a) + (x22 * \\ & rule22b) + (x23 * rule23a) + (x24 * rule24) + (x25 * rule25) + (x26 * rule26) + \\ & (x27 * rule27) + (x28 * rule28) + (x29 * rule29) + (x30 * rule30) + (x31 * \\ & rule31) + (x32 * rule32) + (x33 * rule33) + (x34 * rule34) + (x35 * rule35) \\ & + (x36 * rule36) + (x37 * rule37) + (x38 * rule38) + (x39 * rule39) + (x40 * \\ & rule40) + (x41 * rule41) + (x42 * rule42) + (x43 * rule43) + (x44 * rule44) + \end{aligned}$$

$(x_{45} * \text{rule}_{45}) + (x_{46} * \text{rule}_{46}) + (x_{47} * \text{rule}_{47}) + (x_{48} * \text{rule}_{48}) + (x_{49} * \text{rule}_{49}) / (x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11} + x_{11} + x_{12} + x_{12} + x_{13} + x_{13} + x_{14} + x_{14} + x_{15} + x_{15} + x_{16} + x_{16} + x_{17} + x_{17} + x_{18} + x_{18} + x_{19} + x_{19} + x_{20} + x_{20} + x_{21} + x_{21} + x_{22} + x_{22} + x_{23} + x_{23} + x_{24} + x_{25} + x_{26} + x_{27} + x_{28} + x_{29} + x_{30} + x_{31} + x_{32} + x_{33} + x_{34} + x_{35} + x_{36} + x_{37} + x_{38} + x_{39} + x_{40} + x_{41} + x_{42} + x_{43} + x_{44} + x_{45} + x_{46} + x_{47} + x_{48} + x_{49});$

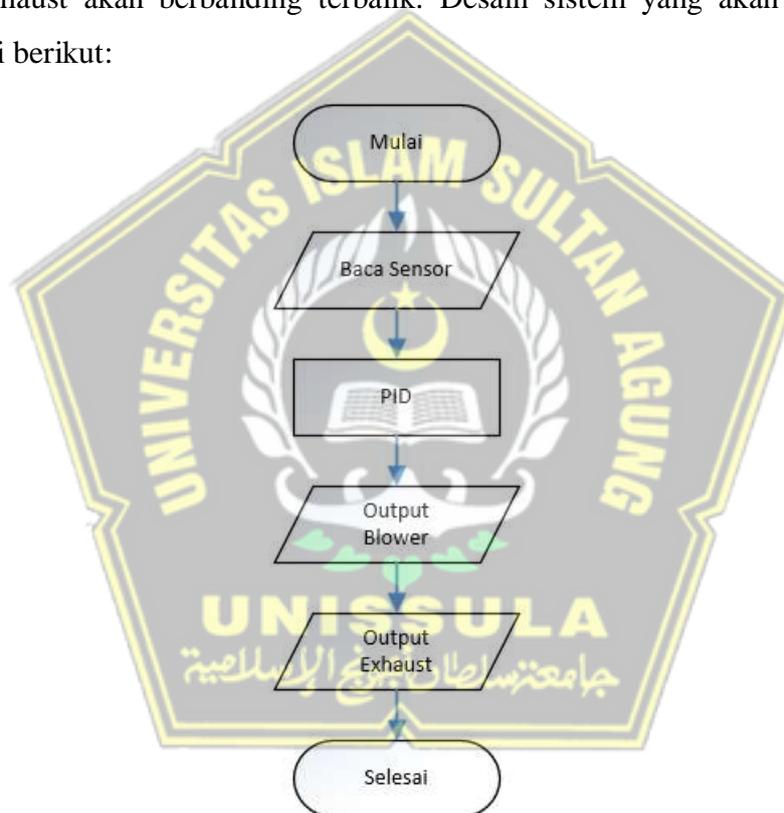
Rumus Defusifikasi untuk Output Exhaust:

$$\text{def}x = \frac{\sum x\mu R(x)}{\sum \mu R(x)} \quad (3.16)$$

$\text{def}x = ((x_1 * \text{rule}_{1x}) + (x_2 * \text{rule}_{2x}) + (x_3 * \text{rule}_{3x}) + (x_4 * \text{rule}_{4x}) + (x_5 * \text{rule}_{5x}) + (x_6 * \text{rule}_{6x}) + (x_7 * \text{rule}_{7x}) + (x_8 * \text{rule}_{8x}) + (x_9 * \text{rule}_{9x}) + (x_{10} * \text{rule}_{10x}) + (x_{11} * \text{rule}_{11x}) + (x_{12} * \text{rule}_{12x}) + (x_{13} * \text{rule}_{13x}) + (x_{14} * \text{rule}_{14x}) + (x_{15} * \text{rule}_{15x}) + (x_{16} * \text{rule}_{16x}) + (x_{17} * \text{rule}_{17x}) + (x_{18} * \text{rule}_{18x}) + (x_{19} * \text{rule}_{19x}) + (x_{20} * \text{rule}_{20x}) + (x_{21} * \text{rule}_{21x}) + (x_{22} * \text{rule}_{22x}) + (x_{23} * \text{rule}_{23x}) + (x_{24} * \text{rule}_{24x}) + (x_{25} * \text{rule}_{25x}) + (x_{26} * \text{rule}_{26x}) + (x_{27} * \text{rule}_{27x}) + (x_{28} * \text{rule}_{28ax}) + (x_{28} * \text{rule}_{28bx}) + (x_{29} * \text{rule}_{29ax}) + (x_{29} * \text{rule}_{29bx}) + (x_{30} * \text{rule}_{30ax}) + (x_{30} * \text{rule}_{30bx}) + (x_{31} * \text{rule}_{31ax}) + (x_{31} * \text{rule}_{31bx}) + (x_{32} * \text{rule}_{32ax}) + (x_{32} * \text{rule}_{32bx}) + (x_{33} * \text{rule}_{33ax}) + (x_{33} * \text{rule}_{33bx}) + (x_{34} * \text{rule}_{34ax}) + (x_{34} * \text{rule}_{34bx}) + (x_{35} * \text{rule}_{35ax}) + (x_{35} * \text{rule}_{35bx}) + (x_{36} * \text{rule}_{36ax}) + (x_{36} * \text{rule}_{36bx}) + (x_{37} * \text{rule}_{37ax}) + (x_{37} * \text{rule}_{37bx}) + (x_{38} * \text{rule}_{38ax}) + (x_{38} * \text{rule}_{38bx}) + (x_{39} * \text{rule}_{39ax}) + (x_{39} * \text{rule}_{39bx}) + (x_{40} * \text{rule}_{40x}) + (x_{41} * \text{rule}_{41x}) + (x_{42} * \text{rule}_{42x}) + (x_{43} * \text{rule}_{43x}) + (x_{44} * \text{rule}_{44x}) + (x_{45} * \text{rule}_{45x}) + (x_{46} * \text{rule}_{46x}) + (x_{47} * \text{rule}_{47x}) + (x_{48} * \text{rule}_{48x}) + (x_{49} * \text{rule}_{49x}) / (x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + x_{16} + x_{17} + x_{18} + x_{19} + x_{20} + x_{21} + x_{22} + x_{23} + x_{24} + x_{25} + x_{26} + x_{27} + x_{28} + x_{28} + x_{29} + x_{29} + x_{30} + x_{30} + x_{31} + x_{31} + x_{32} + x_{32} + x_{33} + x_{33} + x_{34} + x_{34} + x_{35} + x_{35} + x_{36} + x_{36} + x_{37} + x_{37} + x_{38} + x_{38} + x_{39} + x_{39} + x_{40} + x_{41} + x_{42} + x_{43} + x_{44} + x_{45} + x_{46} + x_{47} + x_{48} + x_{49});$

3.4 PID

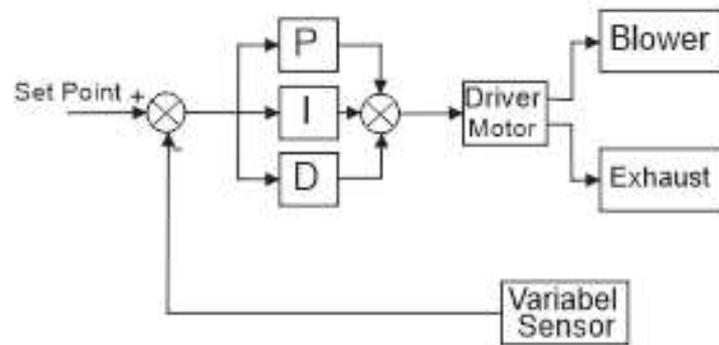
Setelah pembuatan sistem Fuzzi selesai, maka berikutnya sistem fuzzi akan digantikan dengan sistem PID. Sistem PID ini akan digunakan untuk mengendalikan blower dan exhaust. Adapun hardware yang digunakan adalah hardware yang sama seperti digunakan dalam sistem Fuzzy. Dalam pembuatan sistem PID ini, akan menggunakan inputan yang berasal dari sensor tekanan, dimana variable yang berasal dari sensor tekanan akan diproses dengan algoritma PID sehingga menghasilkan nilai output yang digunakan untuk mengendalikan motor blower dan exhaust, dimana nantinya direncanakan nilai manipulasi blower dan exhaust akan berbanding terbalik. Desain sistem yang akan dibuat adalah sebagai berikut:



Gambar 3.39 Flowchart Sederhana Sistem PID

Pada Flowchart diatas ditunjukkan bahwa sistem dimulai dengan pembacaan sensor tekanan dimana dari pembacaan inilah akan didapatkan nilai variable yang digunakan sebagai inputan sistem PID. Inputan dari sensor inilah yang nantinya akan dibandingkan dengan nilai set pada sistem PID dimana selisih antara nilai terukur dengan nilai set menjadi sebuah triger untuk sistem PID mengeluarkan output. Output yang dikeluarkan adalah manipulated value yang digunakan untuk

menggerakkan motor blower dan motor exhaust. Untuk bentuk bagan proses PID seperti yang dijelaskan di atas ditunjukkan dengan bagan berikut:



Gambar 3.40 Algoritma Sederhana Sistem PID

Variabel sensor yang masuk ke dalam sistem akan dijabarkan menjadi beberapa proses variable dimana jumlah proses variable yang digunakan sama dengan ragam variable yang digunakan dalam proses Fuzzy yaitu total ada 7 proses variable, yaitu:

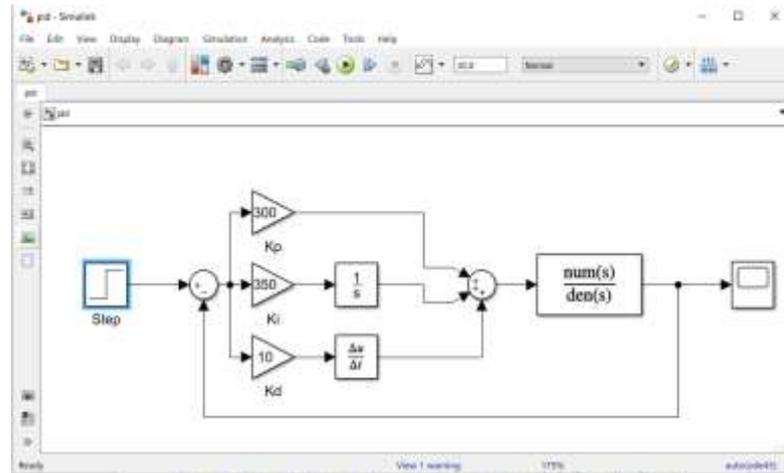
Tabel 3.5 Mapping Nilai Proses pada Sistem PID

Nilai Simpangan Terukur	Nilai Proses Blower	Nilai Proses Exhaust
0 (Sesui set value)	0	0
-1 Pascal	-1	1
-2 Pascal	-4	4
<-2 Pascal	-8	8
+1 Pascal	1	-1
+2 Pascal	4	-4
>+2 Pascal	8	-8

Adapun formula perhitungan PWM motor blower dan exhaust sebagai berikut:

$$\text{motorBlower} = \text{BasePWM} + \text{PID1}; \quad (\text{Motor Blower}) \quad (3.17)$$

$$\text{motorExhaust} = \text{BasePWM} - \text{PID2}; \quad (\text{Motor Exhaust})$$



Gambar 3.41 Pemodelan PID pada kendali tekanan positive ruang operasi.

Metode tuning PID yang digunakan adalah metode eksperimen, dengan melakukan tuning pada koefisien Propotional terlebih dahulu baru diikuti oleh Koefisien integral dan derivative. Penentuan nilai koefisien PID akan diambil setelah menemukan hasil percobaan yang paling ideal.

3.5 Pelaksanaan

Pelaksanaan sistem dalam penelitian ini dilakukan setelah Analisa kebutuhan dan perancangan seluruh sistem ditentukan. Dari segi Hardware, pelaksanaan mengacu pada perakitan komponen elektronik serta mekanik dengan tujuan dapat mendukung kinerja sistem secara baik tanpa ada kendala teknis terkait peralatan yang dapat mengganggu kinerja sistem. Selain itu, dari segi software, sistem dikembangkan berdasarkan algoritma dan struktur program yang telah ditentukan sesuai dengan prosedur standar ruang operasi dengan baik, sehingga software yang dikembangkan dapat mendukung kinerja hardware dengan sempurna.

3.6 Pengujian dan Analisa

Pengujian dan Analisa merupakan suatu langkah yang paling menentukan dari suatu penelitian, karena pengujian dan analisa berfungsi untuk menyimpulkan hasil penelitian. Pengujian dan analisis dapat dilakukan melalui tahap berikut ini :

1. Pengujian
 - a. Pengujian Sensor Tekanan
 - b. Pengujian Sistem input Variable
 - c. Pengujian system penampil data
 - d. Pengujian Sistem Blower dan Exhaust
 - e. Pengujian Respon system

2. Analisis

- a. Pengujian/kalibrasi sensor
- b. Penentuan data base Fuzzy
- c. Tuning parameter PID
- d. Analisis respon system terhadap kondisi pembacaan sensor tekanan



BAB IV

HASIL DAN PENGUJIAN

Setelah melakukan perancangan baik hardware dan software maka proses berikutnya akan dilakukan proses pengujian dari beberapa bagian yang dirancang. Dimulai dari pengujian sensor, input tombol, penampil LCD serta pengujian system fuzzy dan PID yang tertanam dalam mikrokontroler.

Penelitian ini dimaksudkan untuk merancang sistem pengendalian tekanan udara positif pada ruang operasi. Terdapat 2 (dua) pendekatan dalam penelitian ini yaitu pengendalian tekanan udara positif berbasis Fuzzy Logic Controller dan pengendalian tekanan udara berbasis PID Controller. Berdasarkan metode yang dijelaskan pada Bab 3, penelitian ini berjalan dengan beberapa tahapan. Adapun tahapan pelaksanaan pada penelitian ini meliputi Studi Pustaka, Analisis Kebutuhan, Perancangan, Pelaksanaan, serta Pengujian dan Analisa. Untuk memperjelas hasil dari tahapan Pengujian dan Analisa, maka tahapan tersebut akan dibahas pada Bab 4 ini.

4.1. Pengujian Alat

4.1.1. Sensor Tekanan

Sensor tekanan yang digunakan pada penelitian ini adalah sensor mps20n0040d-d. Dari beberapa jenis sensor tekanan yang ada di pasaran, sensor ini dipilih untuk digunakan dalam penelitian ini dikarenakan memiliki range pengukuran antara 0-40kpa dimana dapat mendukung akuisisi data dalam penelitian ini. Hasil pengujian sensor tekanan mps20n0040d-d dapat dilihat pada Tabel 4.1.

Tabel 4.1 Hasil pengujian sensor

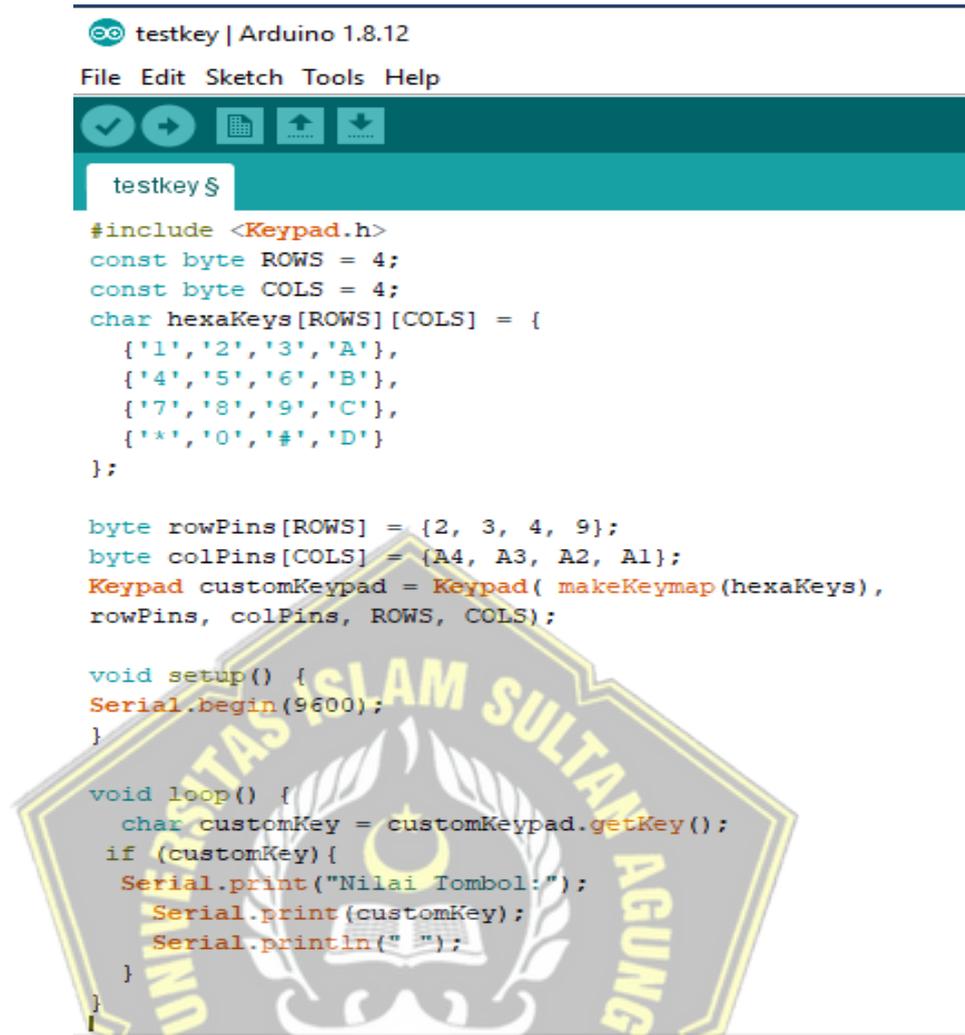
Urutan Pengukuran	Hasil Terukur S1 Dalam Ruang (Nilai Data)	Hasil Terukur S2 Luar Ruang (Nilai Data)
1	101372	101827
2	101401	101913

3	101371	101824
4	101387	101839
5	101392	101845
6	101385	101836
7	101402	101914

Berdasarkan Tabel diketahui bahwa sensor mps20n0040d-d berfungsi dengan baik dengan menghasilkan data pengukuran ketika blower dan exhaust belum aktif. Tetapi didapatkan perbedaan pengukuran antara sensor 1 dalam ruang dan sensor 2 di luar ruang, sehingga perlu dilakukan kalibrasi untuk mendapatkan pengukuran yang hamper seimbang antara sensor 1 dan sensor 2 saat sistem belum aktif. Sistem kalibrasi ini dilakukan dengan cara menghitung selisih antara 2 sensor tersebut dan menambahkan nilai sesuai selisih sensor pada nilai sensor yang lebih rendah.

4.1.2. Sistem Input

Sistem input yang diimplementasikan dalam penelitian ini berupa matrik keypad 4x4. Keypad matrik 4x4 dipilih dalam penelitian ini karena dapat mengakomodasi 16 input button dengan hanya memerlukan 8 pin I/O pada mikrokontroler. Pengujian keypad dilakukan dengan cara memasukkan program yang tertampil pada gambar 4.x pada mikrokontroler dan melihat hasil outputnya dengan menggunakan serial monitor.



```

testkey | Arduino 1.8.12
File Edit Sketch Tools Help
testkey $
#include <Keypad.h>
const byte ROWS = 4;
const byte COLS = 4;
char hexaKeys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};

byte rowPins[ROWS] = {2, 3, 4, 9};
byte colPins[COLS] = {A4, A3, A2, A1};
Keypad customKeypad = Keypad( makeKeymap(hexaKeys),
rowPins, colPins, ROWS, COLS);

void setup() {
  Serial.begin(9600);
}

void loop() {
  char customKey = customKeypad.getKey();
  if (customKey){
    Serial.print("Nilai Tombol:");
    Serial.print(customKey);
    Serial.println(" ");
  }
}

```

Gambar 4.1 Program pengetesan input keypad

Hasil pengujian sistem input keypad matrik 4x4 dapat dilihat pada Tabel 4.2.

Tabel 4.2 Hasil pengujian keypad

Tombol Ditekan	Nilai tertampil pada serial monitor
Angka 0	Nilai Tombol:0
Angka 1	Nilai Tombol:1

Angka 2	Nilai Tombol:2
Angka 3	Nilai Tombol:3
Angka 4	Nilai Tombol:4
Angka 5	Nilai Tombol:5
Angka 6	Nilai Tombol:6
Angka 7	Nilai Tombol:7
Angka 8	Nilai Tombol:8
Angka 9	Nilai Tombol:9
Huruf A	Nilai Tombol:A
Huruf B	Nilai Tombol:B
Huruf C	Nilai Tombol:C
Huruf D	Nilai Tombol:D
Tanda *	Nilai Tombol:*
Tanda #	Nilai Tombol:#

Berdasarkan Tabel diketahui bahwa sistem input keypad matrik 4x4 pada penelitian ini berfungsi dengan baik dengan kinerja keenambelas button berfungsi dengan baik.

4.1.3. Sistem Penampil Data

Sistem penampil data pada penelitian ini berupa Liquid Crystal Display (LCD) yang didukung dengan 20x4 karakter. LCD 20x4 karakter ini dipilih untuk digunakan dalam penelitian ini dikarenakan jumlah karakter yang cukup untuk mengakomodasi sistem monitoring data. Hasil pengujian sistem penampil data berupa LCD 20x4 dapat dilihat pada Tabel.

Tabel 4.3 Hasil pengujian penampil data

Baris pada LCD	Jumlah karakter yang tertampil	Hasil Tampilan
Baris 1	20 Karakter	
Baris 2	20 Karakter	
Baris 3	20 Karakter	
Baris 4	20 Karakter	

Berdasarkan Tabel diketahui bahwa sistem penampil data berjalan dengan baik dengan keseluruhan pixel dapat diakses untuk menampilkan data keseluruhan sistem.

4.1.4. Sistem Blower dan Exhaust

Sistem blower dan exhaust pada penelitian ini didukung dengan penerapan Motor DC 12v. Motor DC dipilih untuk digunakan dalam penelitian ini dikarenakan hasil dari sistem pengendalian pada penelitian ini adalah besaran nilai Pulse Width Modulation (PWM) dimana dengan mudah dapat diterapkan pada pengendalian Motor DC.

Hasil pengujian motor Blower dan Exhaust dapat dilihat pada Tabel 4.4

Tabel 4.4 Hasil pengujian penampil data

Nilai PWM	Blower	Exhaust	Terukur
0	Berhenti	Berhenti	0.15 Volt
75	Putar Pelan	Putar Pelan	3.4 Volt
150	Putar Sedang	Putar Sedang	7.3 Volt
255	Putar Maksimal	Putar Maksimal	11.8 Volt

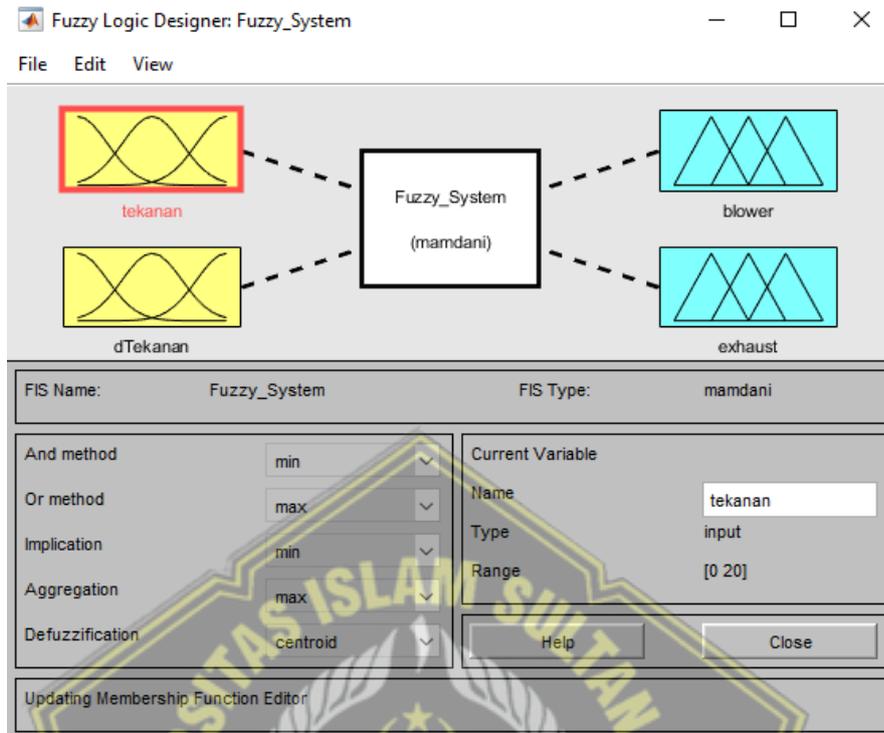
Berdasarkan Tabel diketahui bahwa sistem Motor DC 12v untuk mendukung sistem Blower dan Exhaust pada penelitian ini berjalan dengan baik dengan penerapan berbagai nilai PWM.

4.2. Analisis Metode

4.2.1. Fuzzy Logic Controller

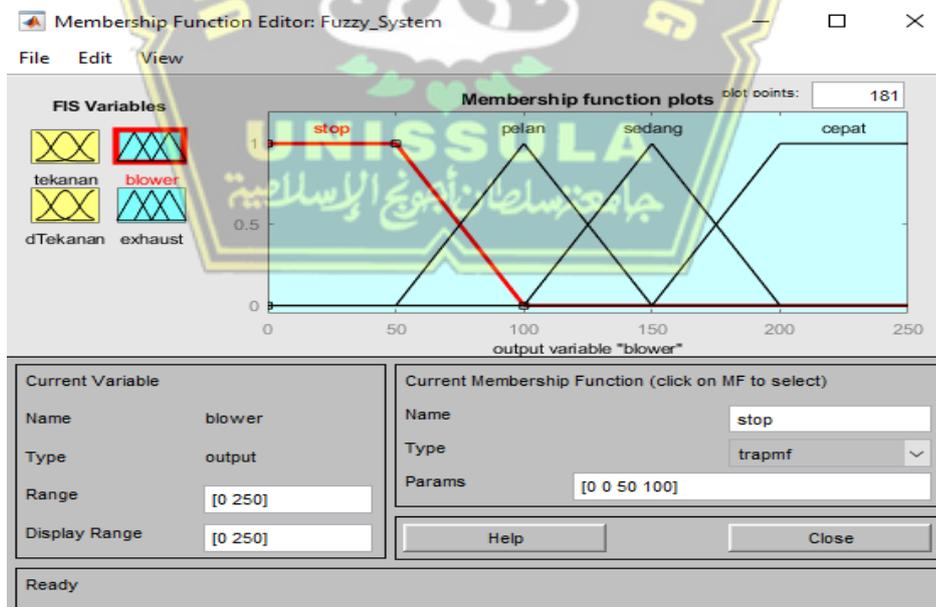
Pengujian sistem menggunakan FLC dilaksanakan guna mengetahui performa dari sistem FLC yang diimplementasikan dalam penelitian ini.

Pengujian sistem Fuzzy diawali dengan analisis output dimana desain output ditampilkan pada rancangan sistem Fuzzy berikut:



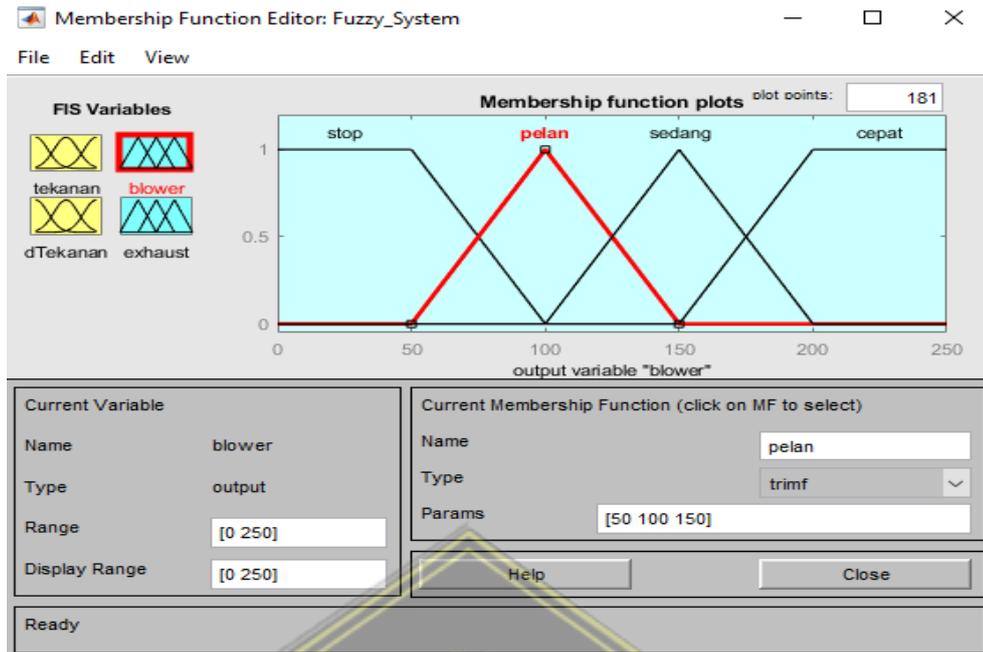
Gambar 4.2 Rancangan sistem Fuzzy pada kendali tekanan positif ruang operasi

Pada gambar 4.2 ditunjukkan ada dua buah output yaitu blower dan exhaust.



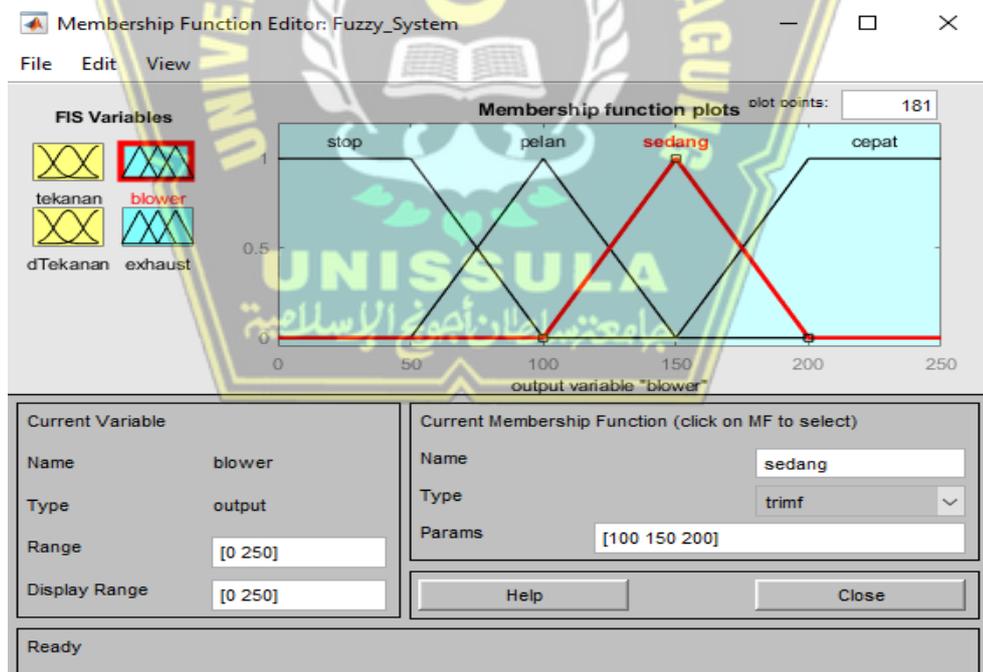
Gambar 4.3 Output Fuzzy blower berhenti berputar

Pada gambar 4.3 ditunjukkan nilai-nilai dari output blower berhenti berputar dengan parameter 0; 0; 50; dan 100.



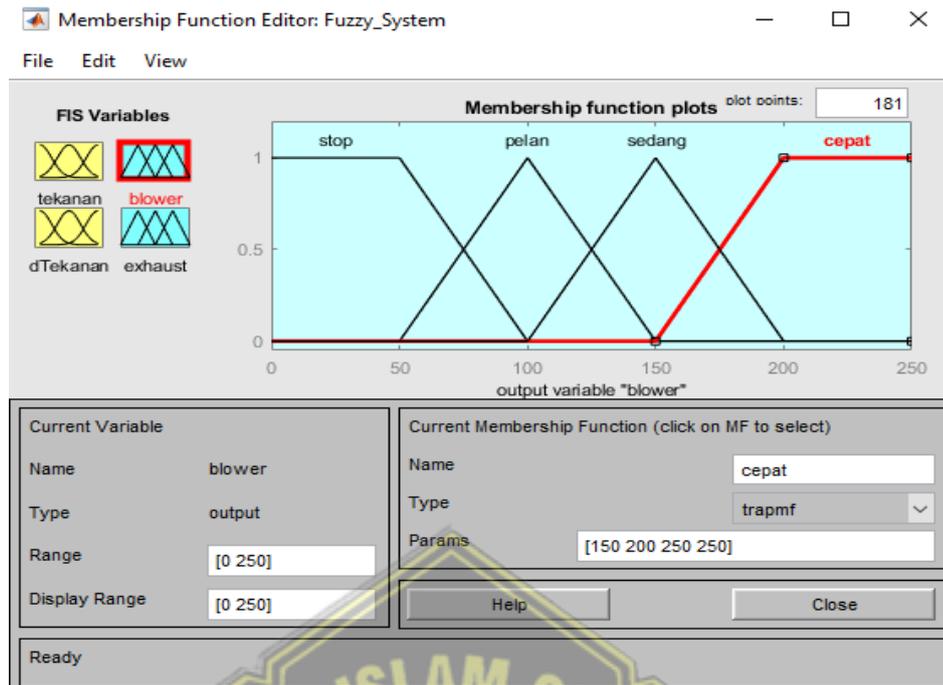
Gambar 4.4 Output Fuzzy blower putaran pelan.

Pada gambar 4.4 ditunjukkan nilai-nilai dari derajat keanggotaan output blower berputar pelan dengan parameter 50; 100; dan 150.



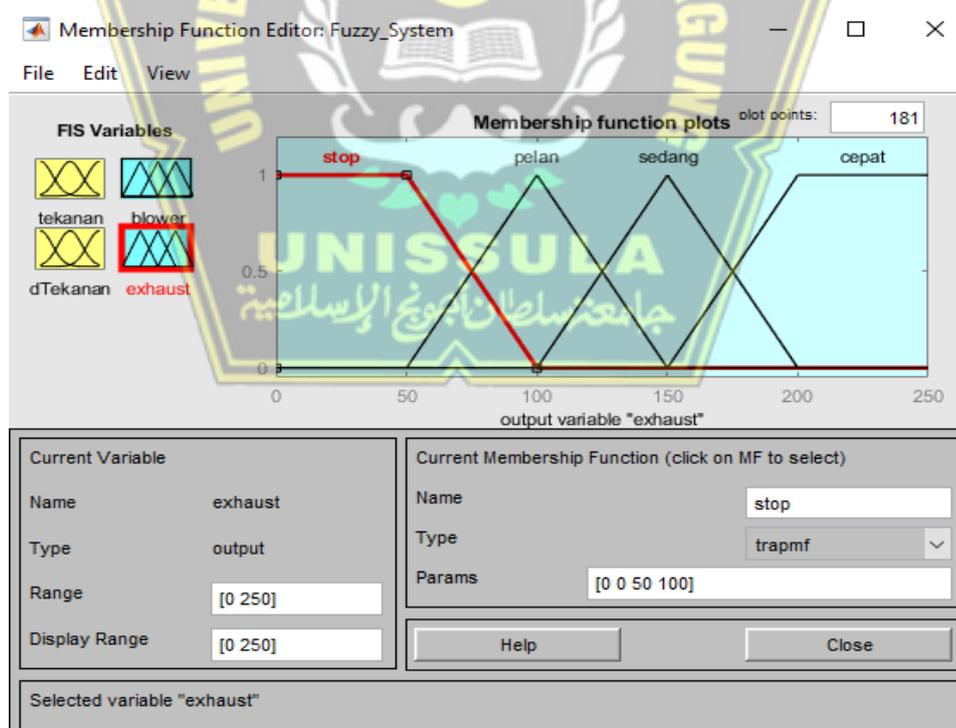
Gambar 4.5 Output Fuzzy blower putaran sedang.

Pada gambar 4.5 ditunjukkan nilai-nilai dari derajat keanggotaan output blower berputar sedang dengan parameter 102;153; dan 204.



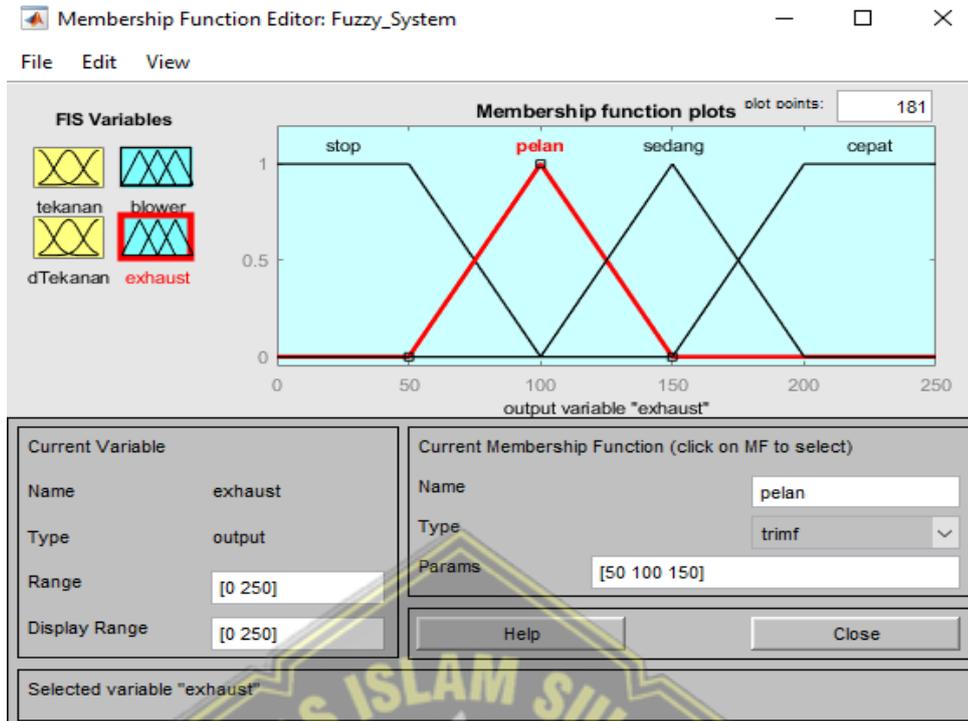
Gambar 4.6 Output Fuzzy blower putaran cepat.

Pada gambar 4.6 ditunjukkan nilai-nilai dari derajat keanggotaan output blower berputar cepat dengan parameter 150;200;250; dan 250.



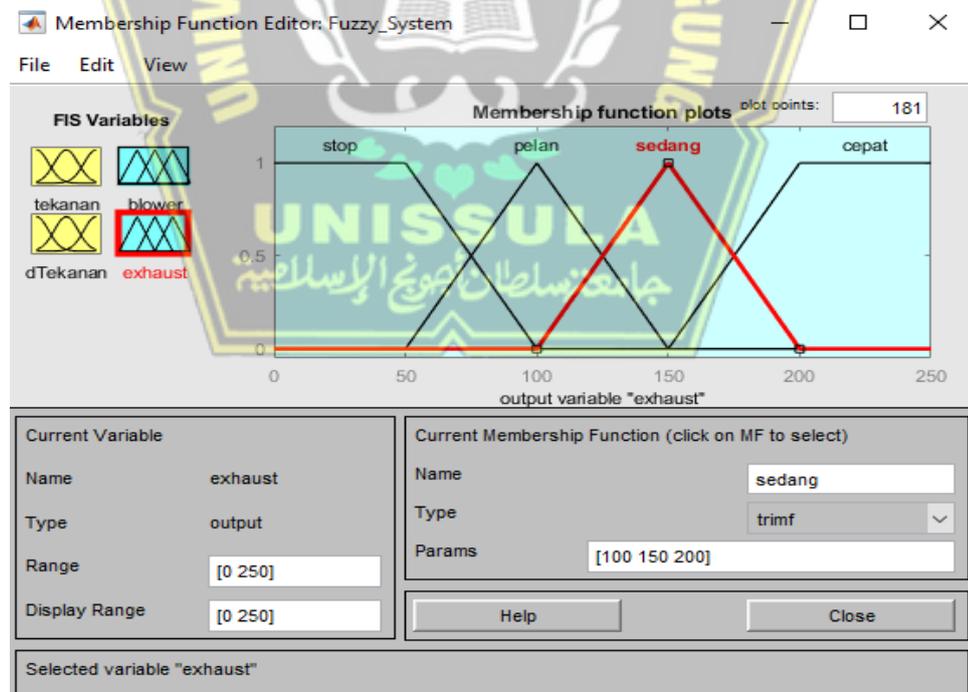
Gambar 4.7 Output Fuzzy exhaust berhenti berputar.

Pada gambar 4.7 ditunjukkan nilai-nilai dari derajat keanggotaan output exhaust berhenti berputar dengan parameter 0;0;50; dan 100.



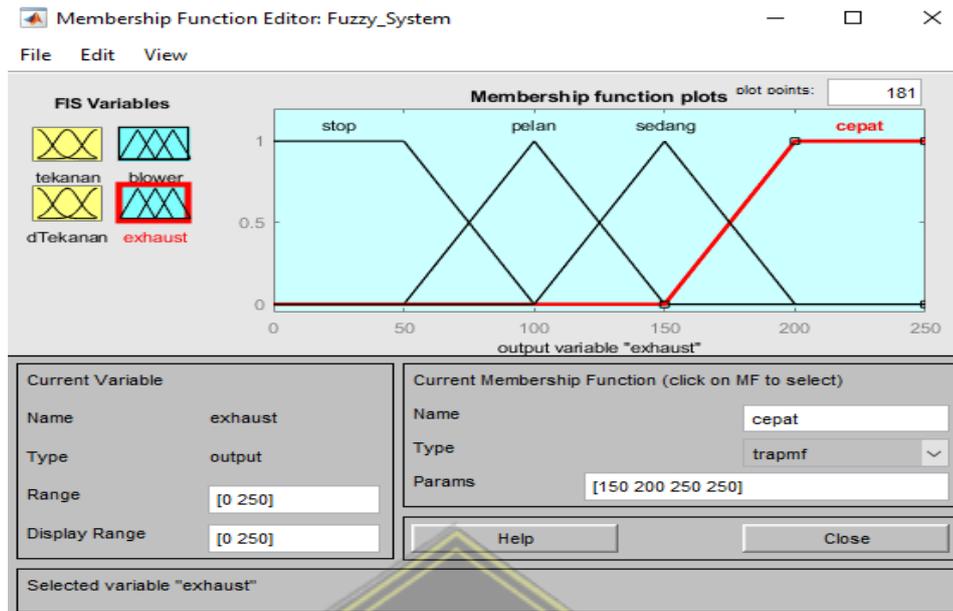
Gambar 4.8 Output Fuzzy exhaust berputar pelan.

Pada gambar 4.8 ditunjukkan nilai-nilai dari derajat keanggotaan output exhaust berputar pelan dengan parameter 50;100; dan 150.



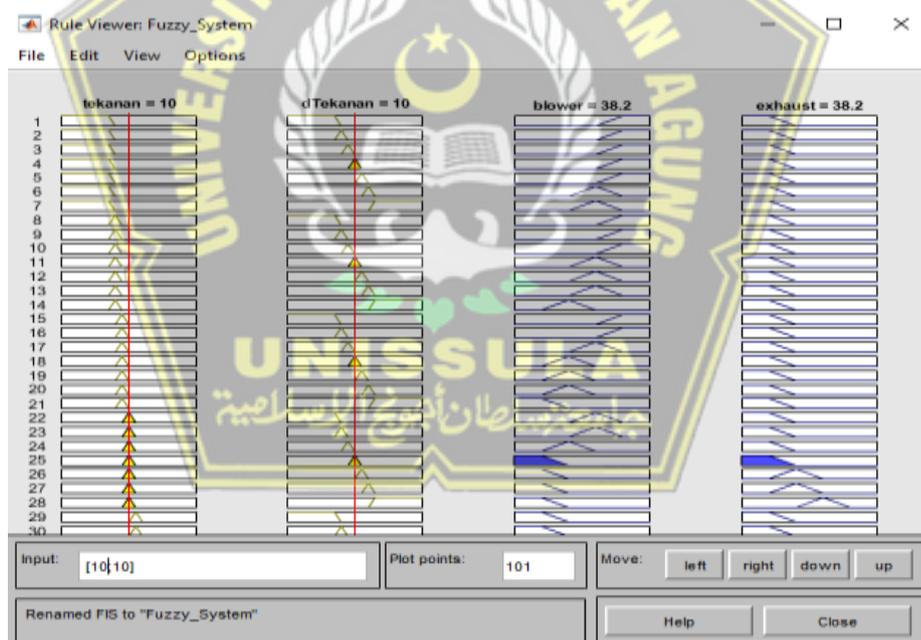
Gambar 4.9 Output Fuzzy exhaust berputar sedang.

Pada gambar 4.9 ditunjukkan nilai-nilai dari derajat keanggotaan output exhaust berputar sedang dengan parameter 100;150; dan 200.



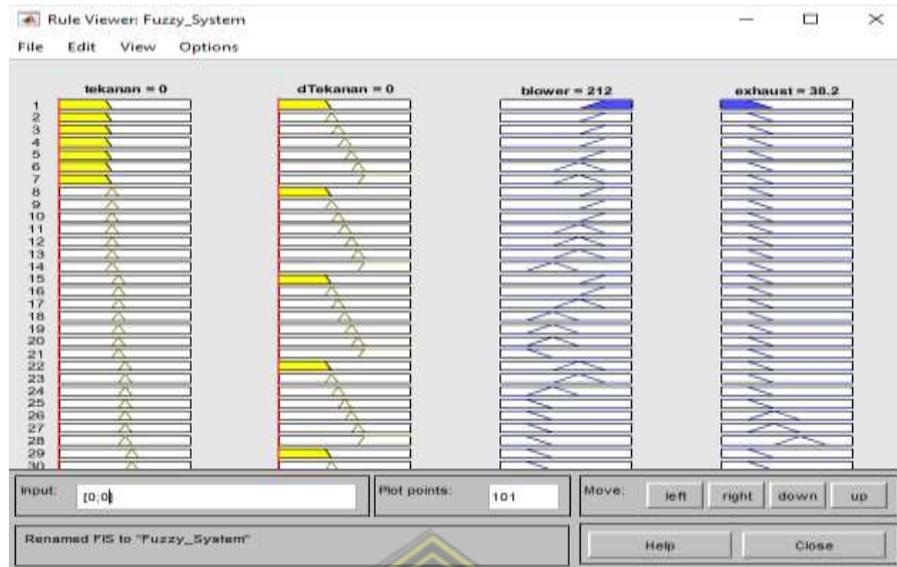
Gambar 4.10 Output Fuzzy exhaust berputar cepat.

Pada gambar 4.10 ditunjukkan nilai-nilai dari derajat keanggotaan output exhaust berputar cepat dengan parameter 150;200;250; dan 250.



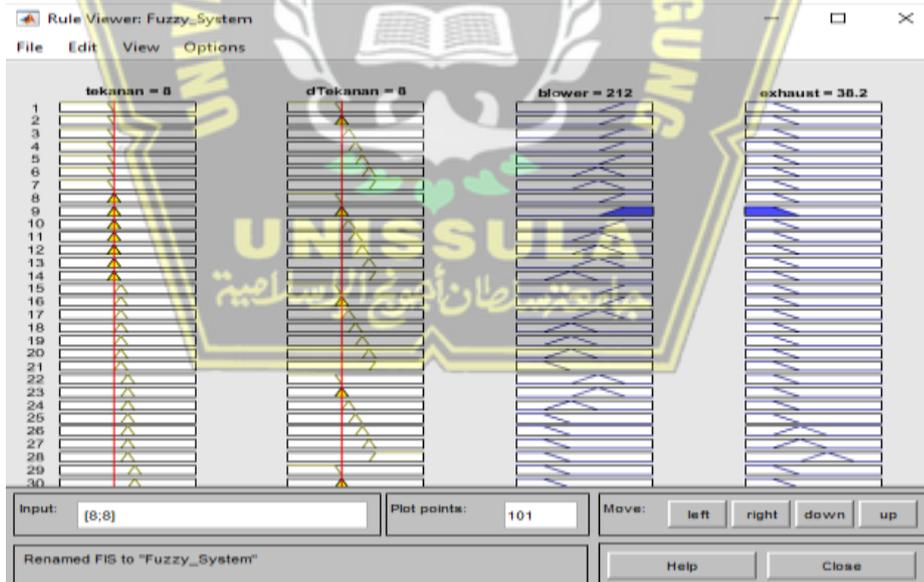
Gambar 4.11 Pemodelan ketika input bernilai sama dengan set poin atau simpangan tekanan sesuai.

Ditunjukkan pada gambar 4.11 dimana input bernilai 10 dan nilai 10 inilah nilai yang diinginkan, dengan kondisi seperti ini output PWM blower adalah 38,2 dan output PWM exhaust adalah 38,2.



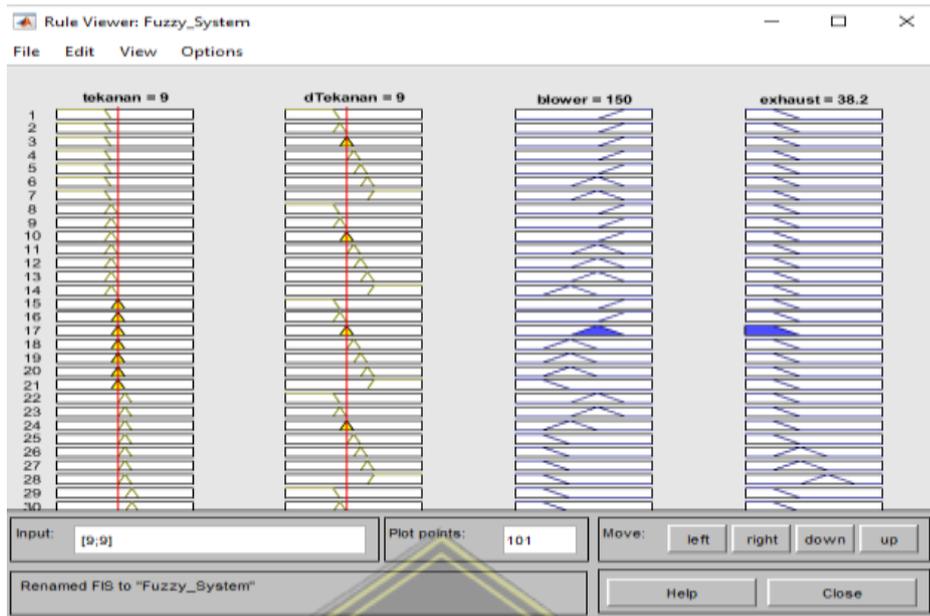
Gambar 4.12 Pemodelan ketika input bernilai 0 atau simpangan tekanan negative besar.

Ditunjukkan pada gambar 4.12 dimana input bernilai 0 dan nilai 0 ini diklasifikasikan dalam kategori simpangan tekanan negative besar, dengan kondisi seperti ini output PWM blower adalah 212 dan output PWM exhaust adalah 38,2.

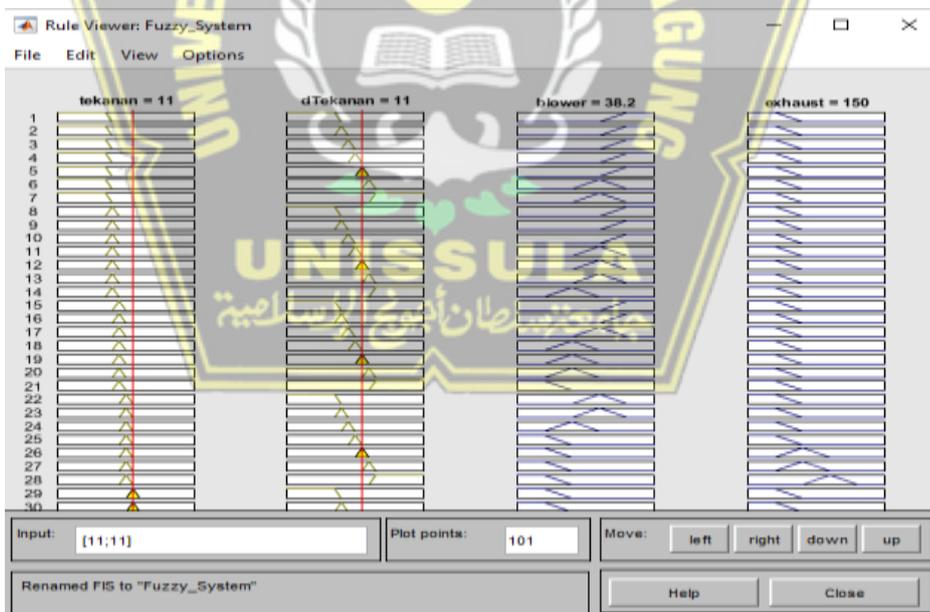


Gambar 4.13 Pemodelan ketika input bernilai 8 atau simpangan tekanan negative sedang.

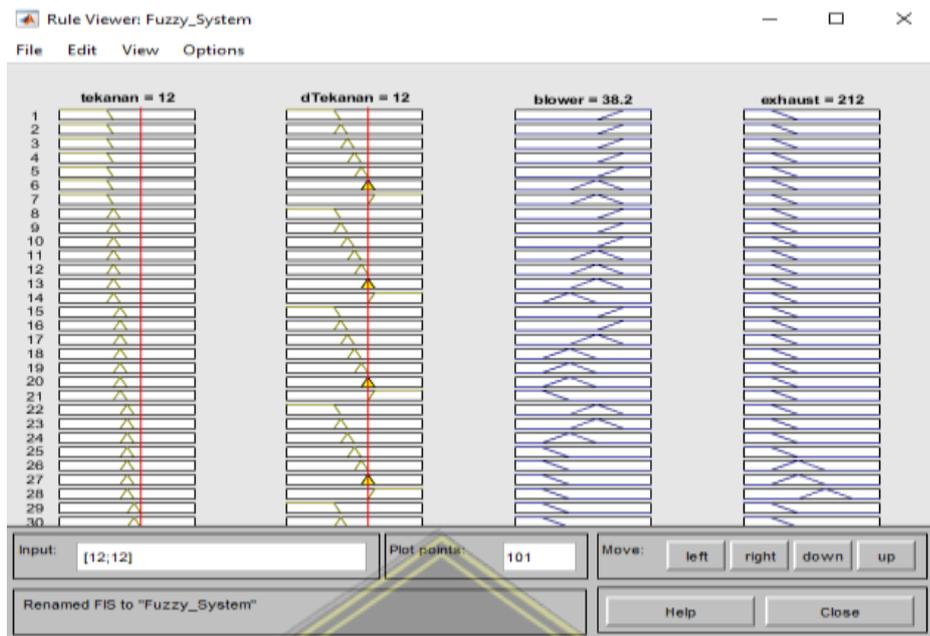
Ditunjukkan pada gambar 4.13 dimana input bernilai 8 dan nilai 8 ini diklasifikasikan dalam kategori simpangan tekanan negative sedang, dengan kondisi seperti ini output PWM blower adalah 212 dan output PWM exhaust adalah 38,2.



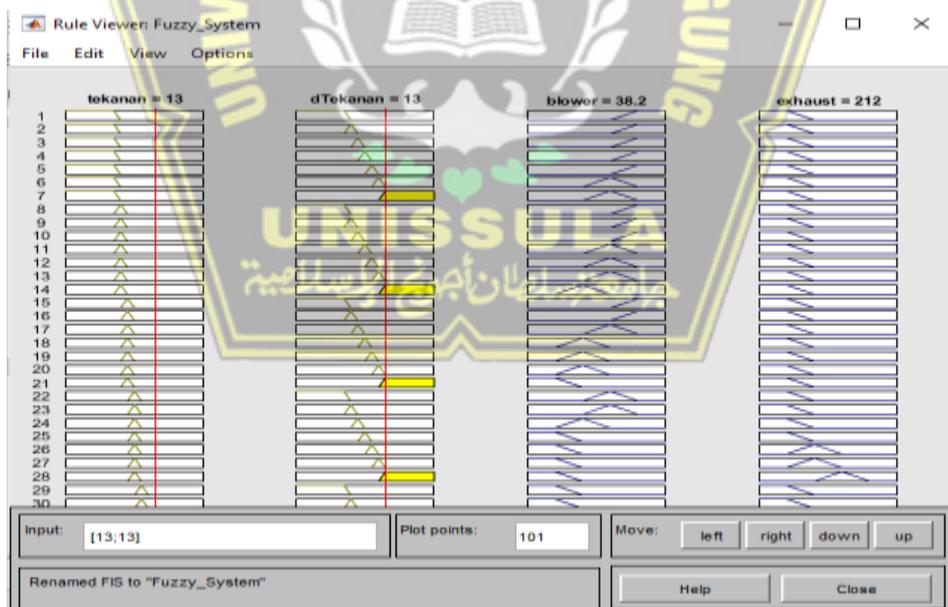
Gambar 4.14 Pemodelan ketika input bernilai 9 atau simpangan negative kecil. Ditunjukkan pada gambar 4.14 dimana input bernilai 9 dan nilai 9 ini diklasifikasikan dalam kategori simpangan tekanan negative kecil, dengan kondisi seperti ini output PWM blower adalah 150 dan output PWM exhaust adalah 38,2.



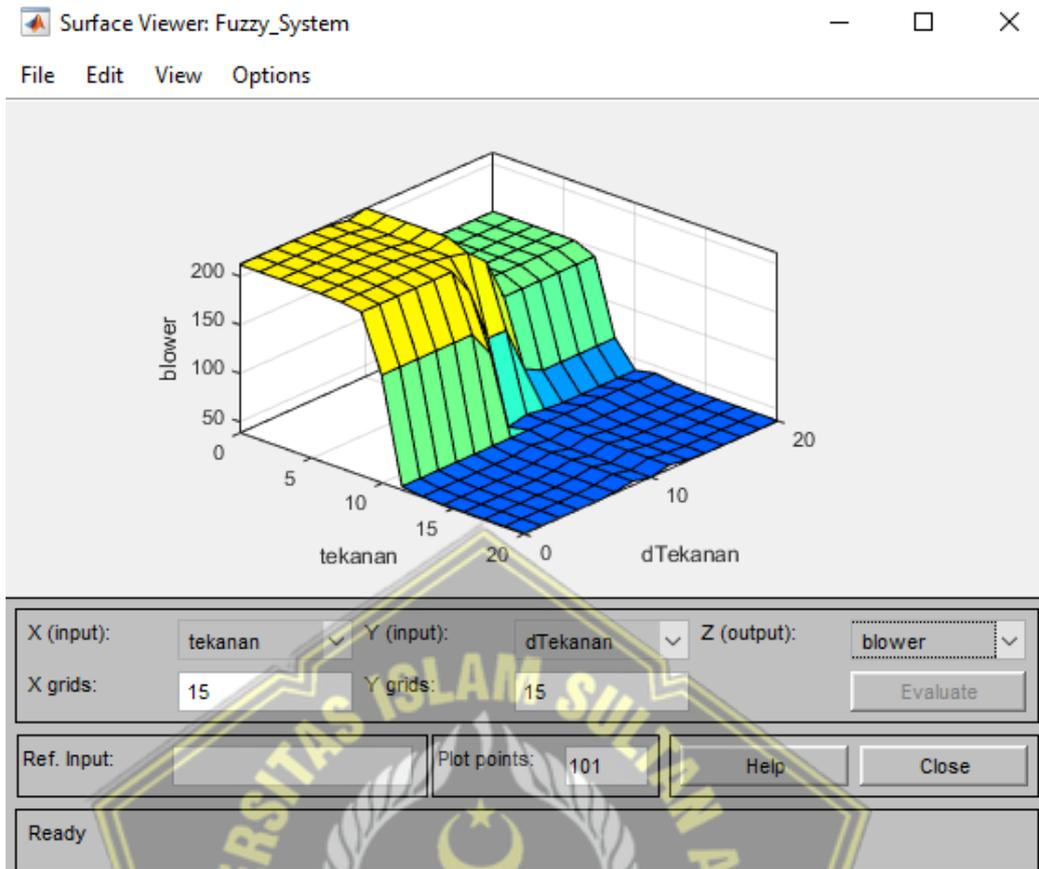
Gambar 4.15 Pemodelan ketika input bernilai 11 atau simpangan positive kecil. Ditunjukkan pada gambar 4.15 dimana input bernilai 11 dan nilai 11 ini diklasifikasikan dalam kategori simpangan tekanan positive kecil, dengan kondisi seperti ini output PWM blower adalah 38,2 dan output PWM exhaust adalah 150.



Gambar 4.16 Pemodelan ketika input bernilai 12 atau simpangan positive sedang. Ditunjukkan pada gambar 4.16 dimana input bernilai 12 dan nilai 12 ini diklasifikasikan dalam kategori simpangan tekanan positive sedang, dengan kondisi seperti ini output PWM blower adalah 38,2 dan output PWM exhaust adalah 212.

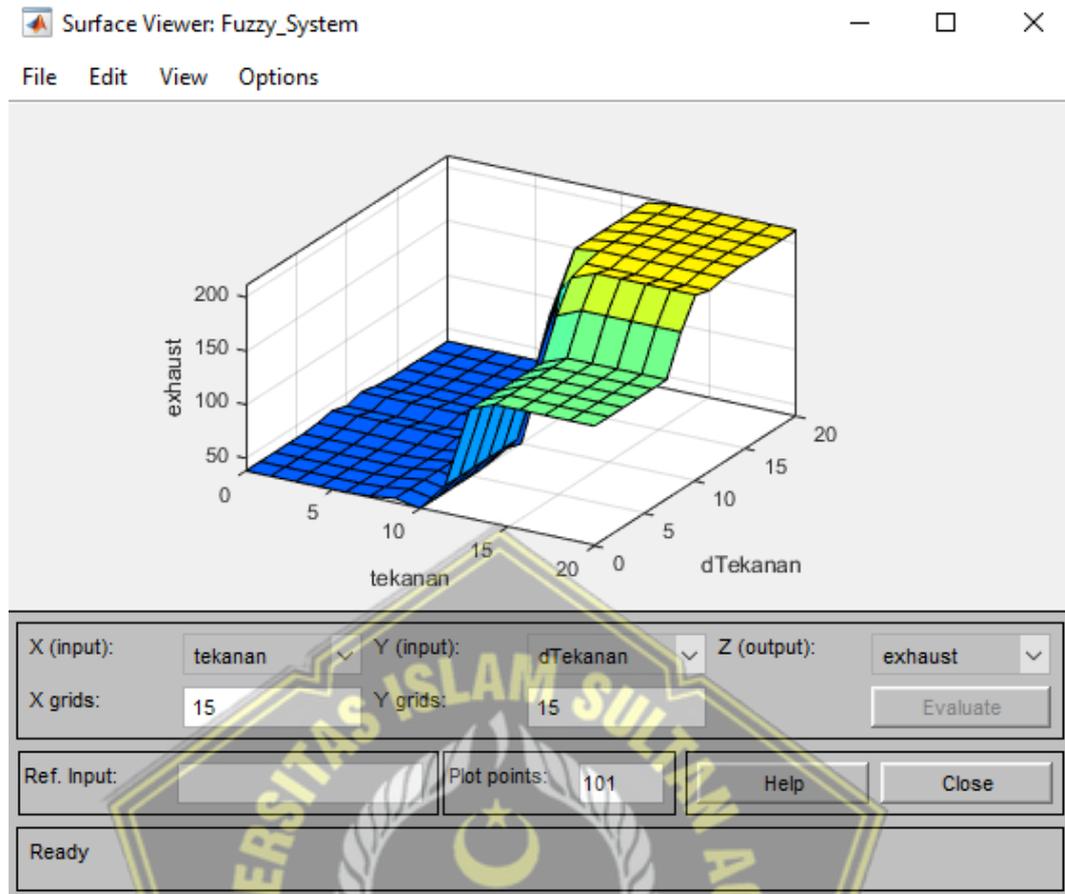


Gambar 4.17 Pemodelan ketika input bernilai 13 atau simpangan positive besar. Ditunjukkan pada gambar 4.17 dimana input bernilai 13 dan nilai 13 ini diklasifikasikan dalam kategori simpangan tekanan positive besar, dengan kondisi seperti ini output PWM blower adalah 39 dan output PWM exhaust adalah 216.



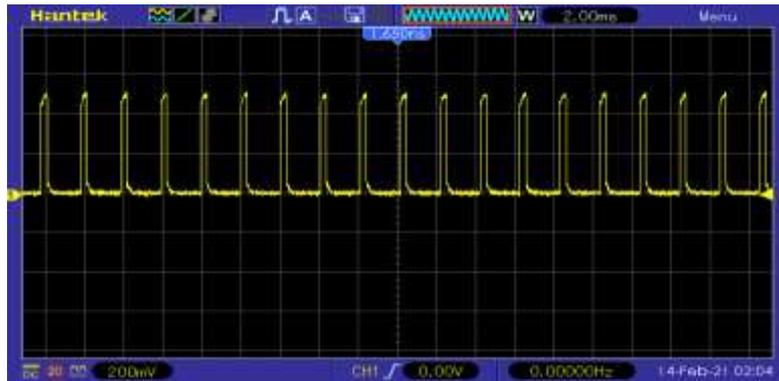
Gambar 4.18 Grafik nilai PWM Blower terhadap nilai input dari sensor tekanan.

Pada gambar di atas diperlihatkan bahwa respon putaran blower terhadap nilai input dari sensor berbanding terbalik, dimana semakin kecil nilai input sensor maka semakin besar nilai PWM dan semakin besar nilai input sensor maka semakin kecil nilai PWM. Hal ini jelas terlihat pada range input sensor antara 5 sampai 15. Sedangkan untuk nilai dibawah 5 nilai PWM cenderung stabil mengikuti nilai terbesar PWM sebelumnya. Begitu juga dengan nilai sensor diatas 15 yang memiliki output nilai PWM yang cenderung sama dengan nilai PWM terkecil sebelumnya

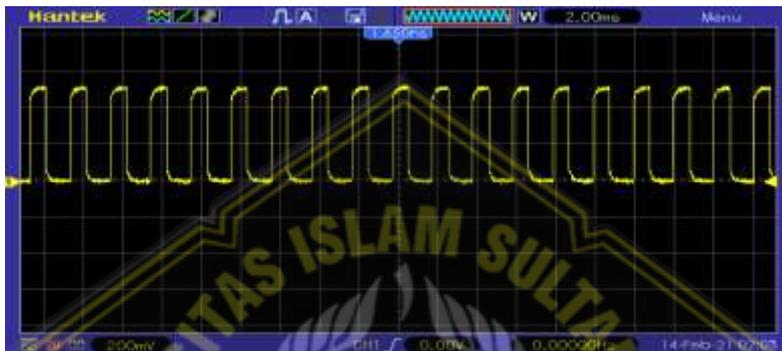


Gambar 4.19 Grafik nilai PWM Exhaust terhadap nilai input dari sensor tekanan.

Pada gambar di atas diperlihatkan bahwa respon putaran Exhaust terhadap nilai input dari sensor berbanding lurus, dimana semakin kecil nilai input sensor maka semakin kecil juga nilai PWM dan semakin besar nilai input sensor maka semakin besar juga nilai PWM. Hal ini jelas terlihat pada range input sensor antara 10 sampai 15. Sedangkan untuk nilai dibawah 10 nilai PWM cenderung stabil mengikuti nilai terkecil PWM sebelumnya. Begitu juga dengan nilai sensor diatas 15 yang memiliki output nilai PWM yang cenderung sama dengan nilai PWM terbesar sebelumnya.



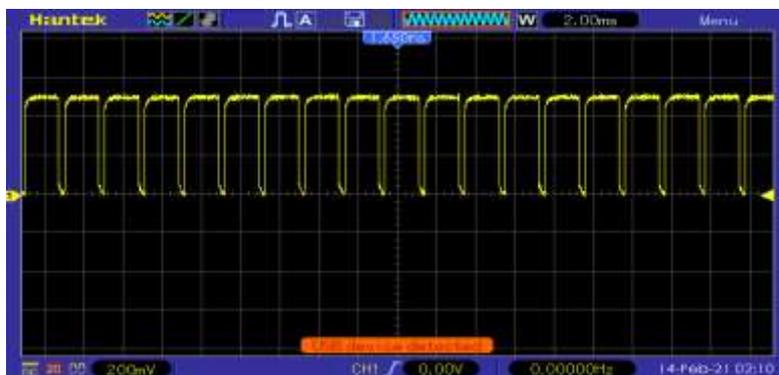
Gambar 4.20 Sinyal PWM dengan nilai PWM 38,2



Gambar 4.21 Sinyal PWM dengan nilai PWM 100



Gambar 4.22 Sinyal PWM dengan nilai PWM 150



Gambar 4.23 Sinyal PWM dengan nilai PWM 212

Hasil dari pengujian FLC pada penelitian ini dapat dilihat pada Tabel 4.5

Tabel 4.5 Hasil pengujian matlab kendali Fuzzy

Nilai sensor	Klasifikasi Input	Nilai PWM Blower	Nilai PWM Exhaust
0	NB	212	38,2
1	NB	212	38,2
2	NB	212	38,2
3	NB	212	38,2
4	NB	212	38,2
5	NB	212	38,2
6	NB	212	38,2
7	NB	212	38,2
8	NS	150	38,2
9	NK	100	38,2
10	TS	38,2	38,2
11	PK	38,2	100
12	PS	38,2	150
13	PB	38,2	212
14	PB	38,2	212
15	PB	38,2	212
16	PB	38,2	212
17	PB	38,2	212
18	PB	38,2	212

19	PB	38,2	212
20	PB	38,2	212

Berdasarkan Tabel diketahui bahwa pengujian FLC pada penelitian ini berjalan baik untuk mengendalikan motor DC berdasarkan nilai PWM yang dihasilkan sistem FLC. Terjadi perubahan respon terhadap perubahan klasifikasi input sensor tekanan.

Tabel 4.6 Hasil pengujian alat dengan sistem Fuzzy

Tampilan Display	Tekanan	Blower	Exhaust
	9,19	160	58
	9,45	145	53
	8,7	166	53
	9,65	74	48

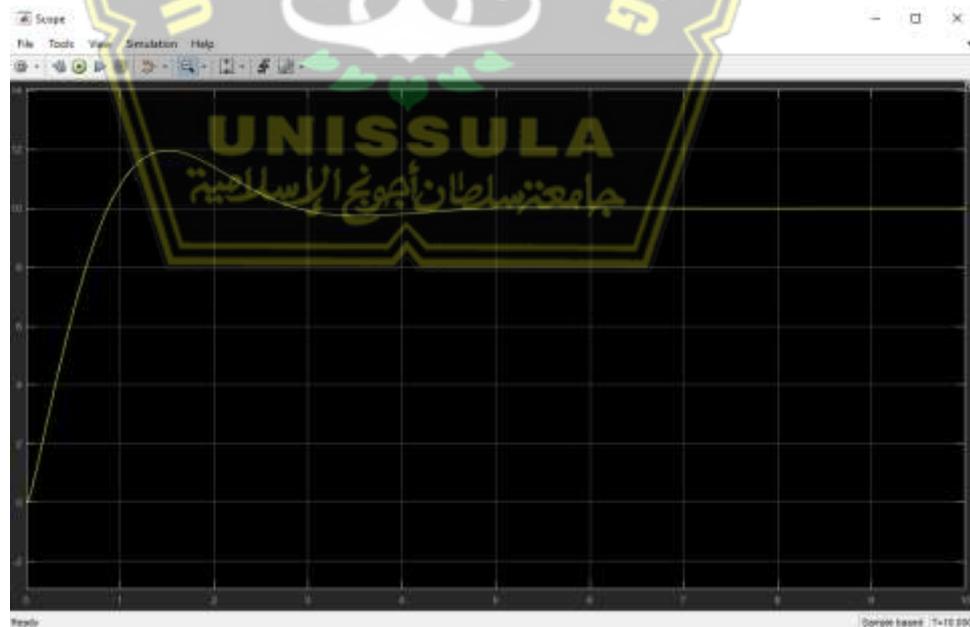
	10,9	105	60
	13,1	50	200

4.2.2. PID Controller

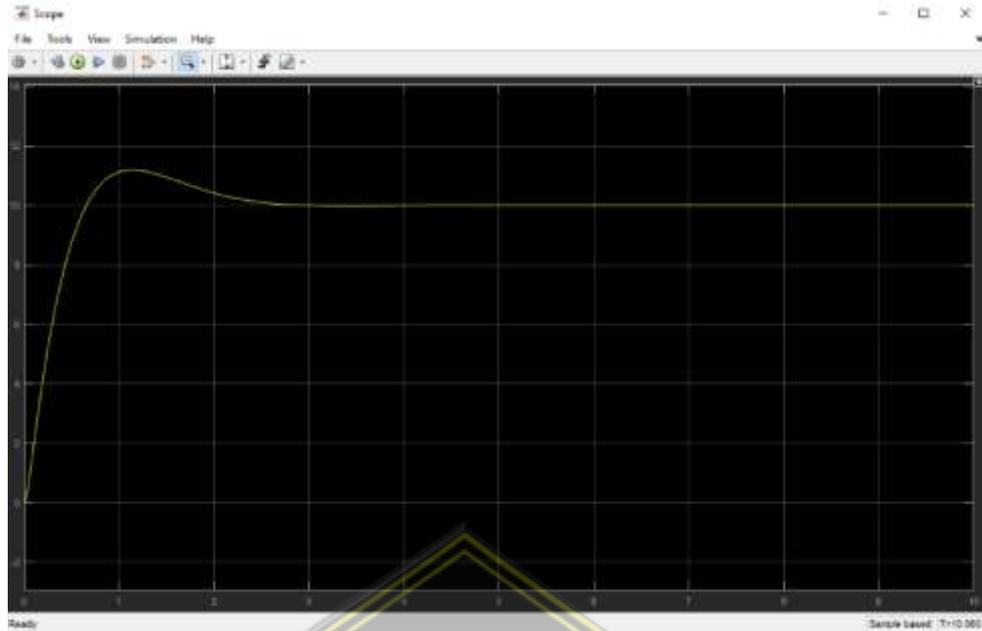
Seperti pada penerapan FLC, Pengujian sistem PID dilaksanakan guna mengetahui performa kendali dengan pendekatan PID yang diimplementasikan dalam penelitian ini.

4.2.2.1 Tuning sistem PID

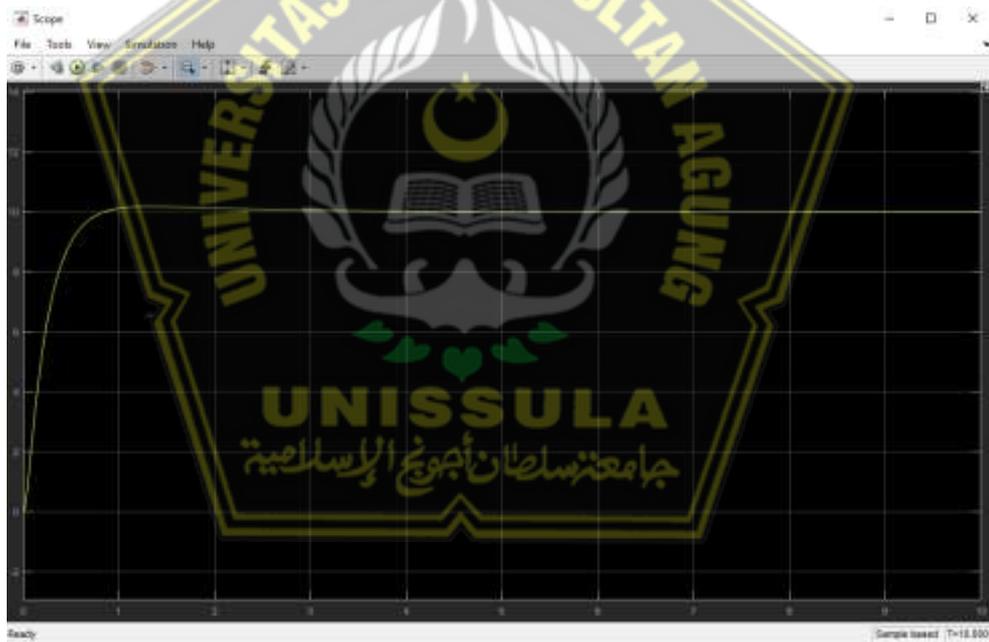
Tuning koefisien PID dilakukan dengan metode percobaan dimana data dari beberapa percobaan yang dilakukan ditampilkan di bawah ini.



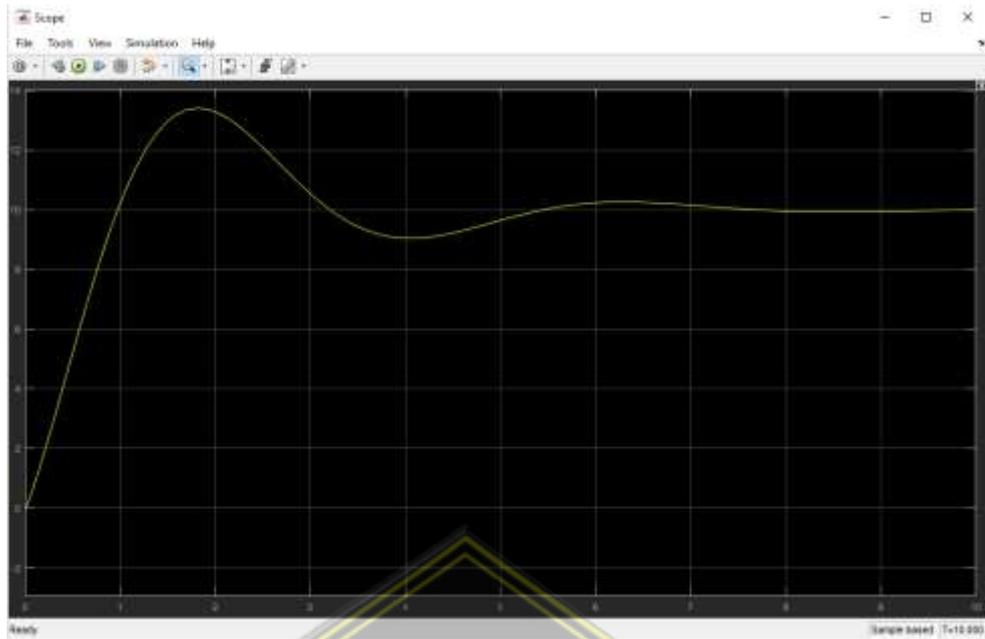
Gambar 4.24 Kurva respon dengan nilai P:2; I:5; dan D:1



Gambar 4.25 Kurva respon dengan nilai P:4; I:7; dan D:1



Gambar 4.26 Kurva respon dengan nilai P:6; I:5; dan D:1



Gambar 4.27 Kurva respon dengan nilai P:3; I:8; dan D:3



Gambar 4.28 Kurva respon dengan nilai P:2; I:10; dan D:1

4.2.2.2 Hasil analisis respon sistem PID sampai pada PWM tertinggi terhadap waktu.

Tabel 4.6 Tabel respon sistem PID terhadap waktu

Set Point	Nilai Sensor	P	I	D	Waktu
10	0	2	5	1	3,29 detik
10	0	4	7	1	1,84 detik
10	0	6	5	1	2,56 detik
10	0	3	8	3	1,58 detik
10	0	2	10	1	1,05 detik

Dari hasil data analisis diatas maka diambil nilai PID dengan koefisien P:2 ; I:5 ; D:1 dengan waktu mulai sampai mencapai PWM maksimal adalah 3,29 detik. Terlihat bahwa respon berdasarkan catatan waktu adalah yang paling lambat tetapi sistem PID dengan koefisien ini cukup stabil dan berhati-hati sehingga menghasilkan respon kestabilan pada set point yang bagus.

Hasil keluaran PWM yang dihasilkan dari sistem PID guna minimalisasi Error serta responsivitas terhadap waktu digunakan sebagai tolak ukur pengujian pada sistem PID dalam penelitian ini. Hasil dari pengujian PID pada penelitian ini dapat dilihat pada Tabel 4.7.

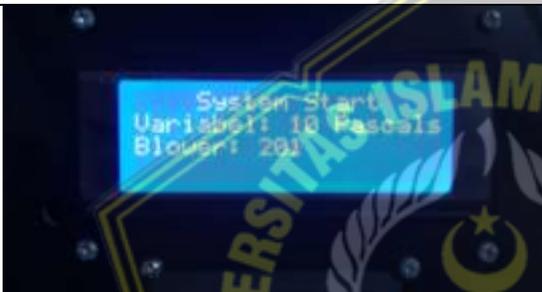
Tabel 4.7 Hasil pengujian output kendali PID

Input Sensor	Kondisi Blower	Kondisi Exhaust
0	ON	OFF
1	ON	OFF
2	ON	OFF
3	ON	OFF

4	ON	OFF
5	ON	OFF
6	ON	OFF
7	ON	OFF
8	ON	OFF
9	ON	OFF
10	OFF	OFF
11	OFF	ON
12	OFF	ON
13	OFF	ON
14	OFF	ON
15	OFF	ON
16	OFF	ON

Tabel 4.8 Hasil pengujian kendali PID

Tampilan Display	Tekanan	Blower
	3	204

	9	220
	10	220
	10	201
	14	220

BAB 5

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Sistem pengendalian tekanan pada ruang operasi berbasis Fuzzy Logic Controller dan PID Controller berhasil dikembangkan untuk dapat menjaga stabilitas tekanan udara positif pada ruang operasi. Sistem dikembangkan dengan 2 pendekatan. Pendekatan pertama dirancang menggunakan sistem kendali FLC dan pendekatan kedua dirancang menggunakan sistem kendali PID. Pengembangan sistem didukung dengan sensor tekanan mps20n0040d-d, mikrokontroler, LCD 20x4 karakter, serta motor DC sebagai penggerak blower dan exhaust. Pada perbandingan sistem kendali, sistem Fuzzy Logic dan PID yang dibangun memiliki kemampuan untuk menjaga tekanan stabil pada 10 Pascal. Berdasarkan pada pengamatan dari 2 jenis kendali yang diterapkan, memiliki karakter yang berbeda dimana pada fuzzy logic respon sistem yang lebih kasar dibandingkan kendali PID. Dari segi ketepatan PID memerlukan proses osilasi sebelum sampai pada titik stabilnya, sedangkan kendali Fuzzy lebih tepat dalam menuju titik stabil pada 10 Pascal.

5.2. Saran

Berdasar pada keseluruhan proses penelitian, masih terdapat keterbatasan dan kekurangan yang dihasilkan. Penambahan sensor perlu dipertimbangkan untuk mendukung beberapa faktor terkait standarisasi lingkungan ruang operasi agar tetap terjaga secara otomatis. Sensor tekanan perlu dikembangkan untuk mendapatkan nilai pengukuran yang stabil dan penempatan sensor deteksi pada bukaan pintu ruang operasi perlu ditambahkan. Peningkatan algoritma untuk membuat sistem semakin responsif juga diperlukan mengingat stabilitas keadaan lingkungan ruang operasi merupakan hal yang sangat penting pada bagian rumah sakit. Dengan adanya beberapa peningkatan tersebut pada penelitian selanjutnya, diharapkan dapat membantu untuk semakin meningkatkan kualitas pelayanan rumah sakit terutama pada bagian fasilitas ruang operasi.

DAFTAR PUSTAKA

- [1] BSN, “SNI - 03 - 6572 - 2001, Tata Cara Perancangan Sistem Ventilasi dan Pengkondisian Udara pada Bangunan Gedung,” pp. 1–55, 2003, [Online]. Available:
<http://staffnew.uny.ac.id/upload/132100514/pendidikan/perencanaan-pendingin.pdf>.
- [2] puput dani Prasetyoadi, “MENGUNAKAN MIKROKONTROLLER ARDUINO DAN METODE LOGIKA FUZZY Studi Kasus : Rumah Sakit Umum Nabire Papua,” *Fateska*, vol. 1, no. 1, pp. 11–20, 2016.
- [3] Menkes RI, “Pedoman teknis bangunan rumah sakit ruang gawat darurat ruang operasi,” *Direktorat Bina Pelayanan Penunjang Med. dan Sarana Kesehat.*, 2012.
- [4] R. Nugraheni, S. Winarni, R. Setjonegoro, and K. Wonosobo, “Infeksi Nosokomial di RSUD Setjonegoro Kabupaten Wonosobo,” *Media Kesehat. Masy. Indones.*, vol. 11, no. 1, pp. 94–100, 2012, doi: 10.14710/mkmi.11.1.94-100.
- [5] F. Wahab, A. Sumardiono, A. R. Al Tahtawi, and A. F. A. Mulayari, “Desain dan Purwarupa Fuzzy Logic Control untuk Pengendalian Suhu Ruangan,” *J. Teknol. Rekayasa*, vol. 2, no. 1, p. 1, 2017, doi: 10.31544/jtera.v2.i1.2017.1-8.
- [6] Y. D. Aryandhi and M. W. Talakua, “Penerapan Inferensi Fuzzy Untuk Pengendali Suhu Ruangan Secara Otomatis Pada Air Conditioner (AC),” *Pros. FMIPA Univ. Pattimura 2013*, pp. 177–185, 2013.
- [7] S. K. Putra, E. K. Simanjuntak, W. Hendradjit, and S. Hadisupadmo, “Perancangan dan Kontrol Mode Operasi Tata Udara Ruang Bedah,” pp. 121–128, 2016, doi: 10.5614/sniko.2015.19.
- [8] A. Shodiq, “Kamar operasi 1,” pp. 1–10, 2009.
- [9] Menkes RI, “Pedoman teknis bangunan rumah sakit ruang gawat darurat,” *Direktorat Bina Pelayanan Penunjang Med. dan Sarana Kesehat.*, 2012.
- [10] H. Nasution, “Implementasi Logika Fuzzy pada Sistem Kecerdasan Buatan,” *ELKHA J. Tek. Elektro*, vol. 4, no. 2, pp. 4–8, 2012.

- [11] I. K. Putri, "Aplikasi Metode Fuzzy Min-Max (Mamdani) Dalam Menentukan Jumlah Produksi Perusahaan," *UIN Alauddin Makassar*, 2017.
- [12] E. Yazid, "Penerapan Kendali Cerdas Pada Sistem Tangki Air Menggunakan," *Himpun. Fis. Indones.*, vol. 2009, no. 2, pp. 11–23, 2009, [Online]. Available: Lembaga Ilmu Pengetahuan Indonesia.
- [13] "dokumen." [Online]. Available: <https://dokumen.tips/download/link/konsep-dasar-logika-fuzzy>.
- [14] A. Note, "Dt-51," *Univ. Stuttgart*, pp. 1–18, 1960.
- [15] H. Thendean and M. Sugiarto, "Penerapan Fuzzy If-Then Rules Untuk Peningkatan Kontras Pada Citra Hasil Mammografi," *J. Inform.*, vol. 9, no. 1, pp. 1–7, 2009, doi: 10.9744/informatika.9.1.1-7.
- [16] S. Kuswandi, *Kendali Cerdas*. Yogyakarta: CV. Andi Offset, 2007.
- [17] E. Dodu, "Sistem Kendali Umpan-Balik (Feedback Control) Menggunakan Teknik Pengendalian PID (PRoporsional-Integral-Derivatif)," *Jimt*, vol. 6, p. 1, 2009.
- [18] D. Irawan and P. Perdana SS, "Kontrol Motor Brushless DC (BLDC) Berbasis Algoritma AI - PID," *J. Tek. Elektro dan Komputasi*, vol. 2, no. 1, pp. 41–48, 2020, doi: 10.32528/elkom.v2i1.3146.
- [19] K. Ogata, *Teknik Kontrol Automatik*, 2nd ed. Erlangga, 1977.
- [20] M. Ali, "Pembelajaran Perancangan Sistem Kontrol Pid Dengan Software Matlab," *J. Edukasi Elektro*, vol. 1, no. 1, p. 2, 2004.
- [21] E. Indonesia, "Pengenalan Metode Ziegler-Nichols pada Perancangan Kontroler PID," 1998. .
- [22] A. Braun and A. Braun, "Auto-Tuning," *Optim. und Adapt. Regelung Tech. Syst.*, pp. 213–219, 2020, doi: 10.1007/978-3-658-30916-9_12.
- [23] A. E. Putra, *Belajar Mikrokontroler*. Gava Media, 2006.
- [24] R. Hendrawan and M. A. R. Darjat, "Perancangan Arithmetic Logic Unit (Alu) Pada Prosesor Risc 16-Bit Dengan Teknologi 600Nm Menggunakan Electric," 2015.
- [25] N. Ferdiansyah Kusna, S. Rizqika Akbar, and D. Syauqy, "Rancang Bangun Pengenalan Modul Sensor Dengan Konfigurasi Otomatis Berbasis Komunikasi I2C," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no.

- 10, pp. 3200–3209, 2018, [Online]. Available: <http://j-ptiik.ub.ac.id>.
- [26] “Pengertian Arduino Uno.” .
- [27] Noviardi, “Aplikasi Kominikasi Serial Arduino Uno R3 Pada Pengontrolan Dengan Menggunakan Visual Studio 2012 Dan Sql Server 2008,” *Jte-Itp*, vol. 5, no. 1, pp. 57–64, 2016.
- [28] P. K. Selam and H. M. I, “CARA KERJA DAN PENGGUNAAN MOTOR DIRECT CURRENT (DC),” vol. 23, no. Dc, pp. 509–523, 2011.
- [29] C. Cholish, R. Rimbawati, and A. A. Hutasuhut, “Analisa Perbandingan Switch Mode Power Supply (SMPS) dan Transformator Linear Pada Audio Amplifier,” *CIRCUIT J. Ilm. Pendidik. Tek. Elektro*, vol. 1, no. 2, pp. 90–102, 2017, doi: 10.22373/crc.v1i2.2079.
- [30] M. Amin, R. Ananda, and J. Eska, “Analisis Penggunaan Driver Mini Victor L298N Terhadap Mobil Robot Dengan Dua Perintah Android Dan Arduino Nano,” *J. Teknol. dan Sist. Inf.*, vol. VI, no. 1, pp. 51–58, 2019, doi: <https://doi.org/10.33330/jurteks.v6i1.396>.
- [31] R. Bangun, A. Pembersih, D. A. N. Penyortir, T. Asin, and B. Arduino, “Jurnal Skripsi- "RANCANG BANGUN ALAT PEMBERSIH DAN PENYORTIR UKURAN TELUR ASIN BERBASIS ARDUINO MEGA 2560" Muhammad Sa'ad Rosyidi, M. Ibrahim Ashari, ST, MT., Dr. Eng. I Komang Somawirata, ST., MT. Institut Teknologi Nasional, Malang, Indonesia.,” *Inst. Teknol. Nas. Malang*, pp. 1–17, 2015.

LAMPIRAN

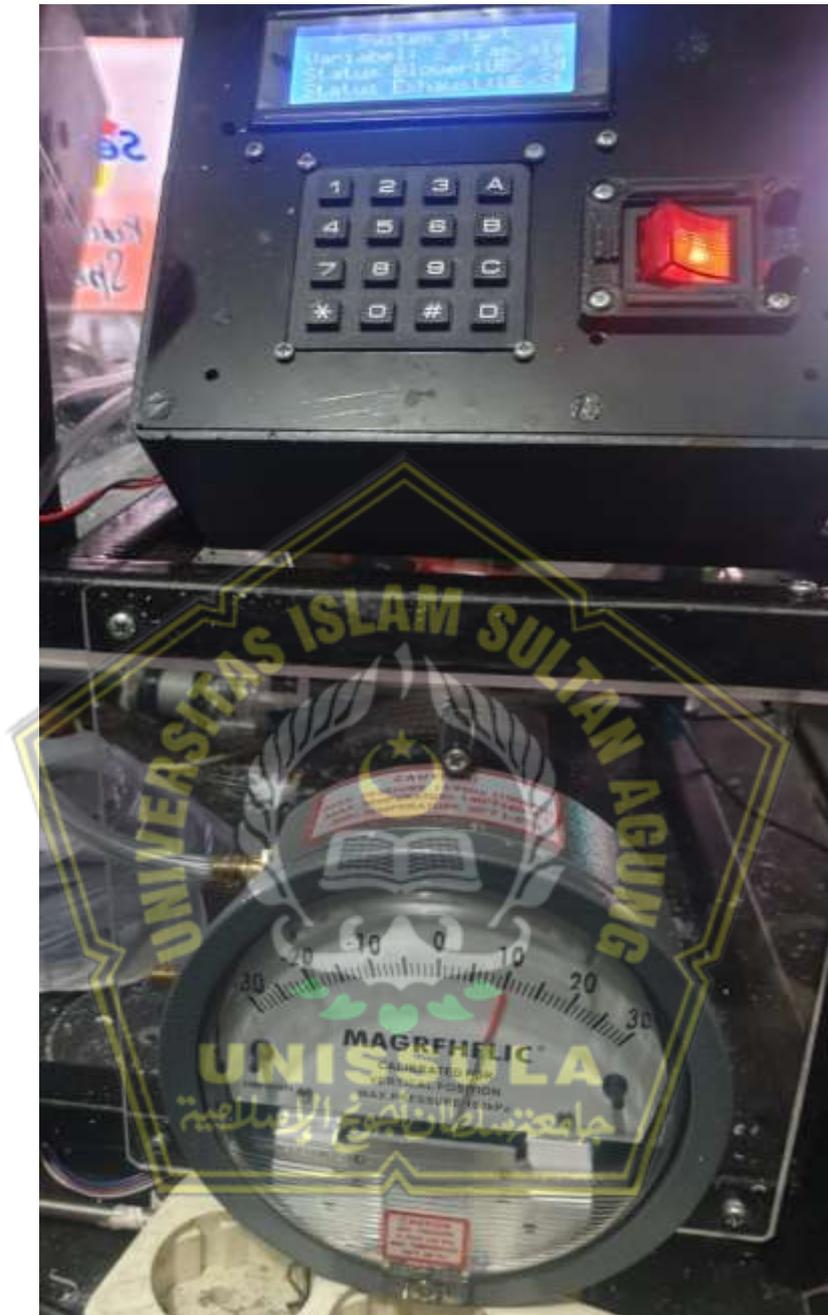
Gambar ketika posisi pengukuran sesuai dengan target atau set point.



Gambar alat ukur tekanan terkalibrasi



Gambar tampilan pengukuran sistem dan pengukuran alat ukur terkalibrasi



Program

Program Fuzzy:

```
#include <LiquidCrystal.h>
#include <Q2HX711.h>
float error_1,error,dError;
float anggotaErrorNB,anggotadErrorNB,anggotapwmNB;
float anggotaErrorNS,anggotadErrorNS,anggotapwmNS;
float anggotaErrorNK,anggotadErrorNK,anggotapwmNK;
float anggotaErrorTS,anggotadErrorTS,anggotapwmTS;
float anggotaErrorPK,anggotadErrorPK,anggotapwmPK;
float anggotaErrorPS,anggotadErrorPS,anggotapwmPS;
float anggotaErrorPB,anggotadErrorPB,anggotapwmPB;
float rule1,rule2,rule3,rule4,rule5,rule6,rule7,rule8,rule9,rule10;
float rule11a,rule11b,rule12a,rule12b,rule13a,rule13b,rule14a,rule14b,
rule15a,rule15b;
float rule16a,rule16b,rule17a,rule17b,rule18a,rule18b,rule19a,rule19b,
rule20a,rule20b;
float rule21a,rule21b,rule22a,rule22b,rule23a,rule23b;
float rule24,rule25,rule26,rule27,rule28,rule29,rule30,rule31,rule32,rule33,
rule34,rule35;
float rule36,rule37,rule38,rule39,rule40,rule41,rule42,rule43,rule44,rule45,
rule46,rule47,rule48,rule49;

float rule1x,rule2x,rule3x,rule4x,rule5x,rule6x,rule7x,rule8x,rule9x,rule10x;
float rule11x,rule12x,rule13x,rule14x,rule15x,rule16x,rule17x,rule18x,
rule19x,rule20x;
float rule21x,rule22x,rule23x;
float rule24x,rule25x,rule26x,rule27x;
float rule28ax,rule28bx,rule29ax,rule29bx,rule30ax,rule30bx,rule31ax,rule31bx;
float rule32ax,rule32bx,rule33ax,rule33bx,rule34ax,rule34bx,rule35ax,rule35bx,
rule36ax,rule36bx,rule37ax,rule37bx,rule38ax,rule38bx,rule39ax,rule39bx;
```

```

float rule40x,rule41x,rule42x,rule43x,rule44x,rule45x,rule46x,rule47x,rule48x,
rule49x;
int def;
int defx;
float tekanan=0;
const int hx711_data_pin = A5;
const int hx711_clock_pin = A4;
const int hx711_data_pin1 = 7;
const int hx711_clock_pin1 = 4 ;
Q2HX711 hx711(hx711_data_pin, hx711_clock_pin); //s1
Q2HX711 hx7112(hx711_data_pin1, hx711_clock_pin1); //s2
const int rs = 13, en = 8, d4 = 9, d5 = 10, d6 = 11, d7 = 12;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
int exhaust=5;
int blower=6;
float vs;
float p1;
float p2;
float calp1;
float calp2;
float cal;

void error_tekanan_NB(){
if(error<=7){anggotaErrorNB = 1;} else if(error > 7 && error <8){anggotaErrorNB
= (8 - error)/1;} else if (error >= 0) {anggotaErrorNB = 0;}
}

void error_tekanan_NS(){
if (error <=7){ anggotaErrorNS = 0;} else if (error > 7 && error <8 ){
anggotaErrorNS = (error-7)/1;}
else if (error >= 8 && error < 9){ anggotaErrorNS = (9 - error)/1;} else if (error >=
9) {anggotaErrorNS = 0;}
}

```

```

void error_tekanan_NK(){
if (error <=8){ anggotaErrorNK = 0;} else if (error > 8 && error <9 ){
anggotaErrorNK = (error-8)/1;}
else if (error >= 9 && error < 10){ anggotaErrorNK = (10 - error)/1;} else if (error
>= 10) {anggotaErrorNK = 0;}
}

```

```

void error_tekanan_TS(){
if (error <=9){ anggotaErrorTS = 0;} else if (error > 9 && error <10 ){
anggotaErrorTS = (error-9)/1;}
else if (error >= 10 && error < 11){ anggotaErrorTS = (11 - error)/1;} else if (error
>= 11) {anggotaErrorTS = 0;}
}

```

```

void error_tekanan_PK(){
if (error <=10){ anggotaErrorPK = 0;} else if (error > 10 && error <11 ){
anggotaErrorPK = (error-10)/1;}
else if (error >= 11 && error < 12){ anggotaErrorPK = (12 - error)/1;} else if (error
>= 12) {anggotaErrorPK = 0;}
}

```

```

void error_tekanan_PS(){
if (error <=10){ anggotaErrorPS = 0;} else if (error > 11 && error <12 ){
anggotaErrorPS = (error-11)/1;}
else if (error >= 12 && error < 13){ anggotaErrorPS = (13 - error)/1;} else if (error
>= 13) {anggotaErrorPS = 0;}
}

```

```

void error_tekanan_PB(){
if (error >= 13){ anggotaErrorPB = 1;}
else if (error < 13 && error > 12){ anggotaErrorPB = (error - 12)/1;}
else if (error < 12 ) {anggotaErrorPB = 0;}
}

```

```

//=====
void dError_tekanan_NB(){
if (error <=7){ anggotadErrorNB = 1;} else if (dError > 7 && dError < 8){
anggotadErrorNB = (8 - dError)/1;} else if (dError >= 0) {anggotadErrorNB = 0;}
}
void dError_tekanan_NS(){
if (dError <=7){ anggotadErrorNS = 0;} else if (dError > 7 && dError <8 ){
anggotadErrorNS = (dError-7)/1;}
else if (dError >= 8 && dError < 9){ anggotadErrorNS = (9 - dError)/1;} else if
(dError >= 9) {anggotadErrorNS = 0;}
}
void dError_tekanan_NK(){
if (dError <=8){ anggotadErrorNK = 0;} else if (dError > 8 && dError <9 ){
anggotadErrorNK = (dError-8)/1;}
else if (dError >= 9 && dError < 10){ anggotadErrorNK = (10 - dError)/1;} else if
(dError >= 10) {anggotadErrorNK = 0;}
}
void dError_tekanan_TS(){
if (dError <=9){ anggotadErrorTS = 0;} else if (dError > 9 && dError <10 ){
anggotadErrorTS = (dError-9)/1;}
else if (dError >= 10 && dError < 11){ anggotadErrorTS = (11 - dError)/1;} else if
(dError >= 11) {anggotadErrorTS = 0;}
}

void dError_tekanan_PK(){
if (dError <=10){ anggotadErrorPK = 0;} else if (dError > 10 && dError <11 ){
anggotadErrorPK = (dError-10)/1;}
else if (dError >= 11 && dError < 12){ anggotadErrorPK = (12 - dError)/1;} else
if (dError >= 12) {anggotadErrorPK = 0;}
}
void dError_tekanan_PS(){

```

```

if (dError <=10){ anggotadErrorPS = 0;} else if (dError > 11 && dError <12 ){
anggotadErrorPS = (dError-11)/1;}
else if (dError >= 12 && dError < 13){ anggotadErrorPS = (13 - dError)/1;} else if
(dError >= 13) {anggotadErrorPS = 0;}
}
void dError_tekanan_PB(){
if (dError >= 13){ anggotadErrorPB = 1;}
else if (dError < 13 && dError > 12){ anggotadErrorPB = (dError - 12)/1;}
else if (dError < 12 ){anggotadErrorPB = 0;}
}

```

```

void fuzifikasi() {
error_tekanan_NB();
error_tekanan_NS();
error_tekanan_NK();
error_tekanan_TS();
error_tekanan_PK();
error_tekanan_PS();
error_tekanan_PB();

dError_tekanan_NB();
dError_tekanan_NS();
dError_tekanan_NK();
dError_tekanan_TS();
dError_tekanan_PK();
dError_tekanan_PS();
dError_tekanan_PB();
}

```

```

float Min (float a, float b) {
if (a < b) {
return a;
}
}

```

```

}
else if (b < a) {
return b;
}
else return a;
}

```

```

void rule() {
fuzifikasi();
//NB Cepat
float x1 = Min (anggotaErrorNB,anggotadErrorNB);
rule1 = 150 + (100 * x1);
rule1x=150-(100*x1);
float x2 = Min (anggotaErrorNB,anggotadErrorNS);
rule2 = 150 + (100 * x2);
rule2x=150-(100*x2);
float x3 = Min (anggotaErrorNB,anggotadErrorNK);
rule3 = 150 + (100 * x3);
rule3x=150-(100*x3);
float x4 = Min (anggotaErrorNB,anggotadErrorTS);
rule4 = 150 + (100 * x4);
rule4x=150-(100*x4);
float x5 = Min (anggotaErrorNB,anggotadErrorPK);
rule5 = 150 + (100 * x5);
rule5x=150-(100*x5);
float x6 = Min (anggotaErrorNS,anggotadErrorNB);
rule6 = 150 + (100 * x6);
rule6x=150-(100*x6);
float x7 = Min (anggotaErrorNS,anggotadErrorNS);
rule7 = 150 + (100 * x7);
rule7x=150-(100*x7);
float x8 = Min (anggotaErrorNS,anggotadErrorNK);

```

```

rule8 = 150 + (100 * x8);
rule8x=150-(100*x8);
float x9 = Min (anggotaErrorNK,anggotadErrorNB);
rule9 = 150 + (100 * x9);
rule9x=150-(100*x9);
float x10 = Min (anggotaErrorNK,anggotadErrorNS);
rule10 = 150 + (100 * x10);
rule10x=150-(100*x10);

```

```
//NS Sedang
```

```

float x11=Min(anggotaErrorNB,anggotadErrorPS);
rule11a =100+(50*x11);
rule11b =200-(50*x11);
rule11x=150-(100*x11);
float x12=Min(anggotaErrorNB,anggotadErrorPB);
rule12a =100+(50*x12);
rule12b =200-(50*x12);
rule12x=150-(100*x12);
float x13=Min(anggotaErrorNS,anggotadErrorTS);
rule13a =100+(50*x13);
rule13b =200-(50*x13);
rule13x=150-(100*x13);
float x14=Min(anggotaErrorNS,anggotadErrorPK);
rule14a =100+(50*x14);
rule14b =200-(50*x14);
rule14x=150-(100*x14);
float x15=Min(anggotaErrorNS,anggotadErrorPS);
rule15a =100+(50*x15);
rule15b =200-(50*x15);
rule15x=150-(100*x15);
float x16=Min(anggotaErrorNK,anggotadErrorNK);

```

```

rule16a =100+(50*x16);
rule16b =200-(50*x16);
rule16x=150-(100*x16);
float x17=Min(anggotaErrorTS,anggotadErrorNB);
rule17a =100+(50*x17);
rule17b =200-(50*x17);
rule17x=150-(100*x17);
float x18=Min(anggotaErrorTS,anggotadErrorNS);
rule18a =100+(50*x18);
rule18b =200-(50*x18);
rule18x=150-(100*x18);

```

```
//NK Pelan
```

```

float x19 = Min (anggotaErrorNS,anggotadErrorPB);
rule19a = 50 + (50 * x19) ;
rule19b = 150 - (50 * x19) ;
rule19x=150-(100*x19);
float x20 = Min (anggotaErrorNK,anggotadErrorTS);
rule20a = 50 + (50 * x20) ;
rule20b = 150 - (50 * x20) ;
rule20x=150-(100*x20);
float x21 = Min (anggotaErrorNK,anggotadErrorPK);
rule21a = 50 + (50 * x21) ;
rule21b = 150 - (50 * x21) ;
rule21x=150-(100*x21);
float x22 = Min (anggotaErrorNK,anggotadErrorPS);
rule22a = 50 + (50 * x22) ;
rule22b = 150 - (50 * x22) ;
rule22x=150-(100*x22);
float x23 = Min (anggotaErrorTS,anggotadErrorNK);
rule23a = 50 + (50 * x23) ;

```

```
rule23b = 150 - (50 * x23) ;  
rule23x=150-(100*x23);
```

```
//TS STOP
```

```
float x24 = Min (anggotaErrorNK,anggotadErrorPB);  
rule24 = 150 - (100 * x24);  
rule24x=150-(100*x24);  
float x25 = Min (anggotaErrorTS,anggotadErrorTS);  
rule25 = 150 - (100 * x25);  
rule25x=150-(100*x25);  
float x26 = Min (anggotaErrorPK,anggotadErrorNB);  
rule26 = 150 - (100 * x26);  
rule26x=150-(100*x26);  
float x27 = Min (anggotaErrorPK,anggotadErrorNS);  
rule27 = 150 - (100 * x27);  
rule27x=150-(100*x27);
```

```
//PK
```

```
float x28 = Min (anggotaErrorTS,anggotadErrorPK);  
rule28 = 150 - (100 * x28);  
rule28ax = 50 + (50 * x28) ;  
rule28bx = 150 - (50 * x28) ;  
float x29 = Min (anggotaErrorTS,anggotadErrorPS);  
rule29 = 150 - (100 * x29);  
rule29ax = 50 + (50 * x29) ;  
rule29bx = 150 - (50 * x29) ;  
float x30 = Min (anggotaErrorPK,anggotadErrorNK);  
rule30 = 150 - (100 * x30);  
rule30ax = 50 + (50 * x30) ;  
rule30bx = 150 - (50 * x30) ;
```

```

float x31 = Min (anggotaErrorPK,anggotadErrorTS);
rule31 = 150 - (100 * x31);
rule31ax = 50 + (50 * x31) ;
rule31bx = 150 - (50 * x31) ;
float x32 = Min (anggotaErrorPS,anggotadErrorNB);
rule32 = 150 - (100 * x32);
rule32ax = 50 + (50 * x32) ;
rule32bx = 150 - (50 * x32) ;
float x33 = Min (anggotaErrorPS,anggotadErrorNS);
rule33 = 150 - (100 * x33);
rule33ax = 50 + (50 * x33) ;
rule33bx = 150 - (50 * x33) ;

//PS
float x34 = Min(anggotaErrorTS,anggotadErrorPB);
rule34 = 150 - (100 * x34);
rule34ax =100+(50*x34);
rule34bx =200-(50*x34);
float x35 = Min(anggotaErrorPK,anggotadErrorPK);
rule35 = 150 - (100 * x35);
rule35ax =100+(50*x35);
rule35bx =200-(50*x35);
float x36 = Min(anggotaErrorPS,anggotadErrorNK);
rule36 = 150 - (100 * x36);
rule36ax =100+(50*x36);
rule36bx =200-(50*x36);
float x37 = Min(anggotaErrorPS,anggotadErrorTS);
rule37 = 150 - (100 * x37);
rule37ax =100+(50*x37);
rule37bx =200-(50*x37);
float x38 = Min(anggotaErrorPB,anggotadErrorNB);

```

```

rule38 = 150 - (100 * x38);
rule38ax = 100 + (50 * x38);
rule38bx = 200 - (50 * x38);
float x39 = Min(anggotaErrorPB, anggotadErrorNS);
rule39 = 150 - (100 * x39);
rule39ax = 100 + (50 * x39);
rule39bx = 200 - (50 * x39);

```

```
//PB
```

```

float x40 = Min(anggotaErrorPK, anggotadErrorPS);
rule40 = 150 - (100 * x40);
rule40x = 150 + (100 * x40);
float x41 = Min(anggotaErrorPK, anggotadErrorPB);
rule41 = 150 - (100 * x41);
rule41x = 150 + (100 * x41);
float x42 = Min(anggotaErrorPS, anggotadErrorPK);
rule42 = 150 - (100 * x42);
rule42x = 150 + (100 * x42);
float x43 = Min(anggotaErrorPS, anggotadErrorPS);
rule43 = 150 - (100 * x43);
rule43x = 150 + (100 * x43);
float x44 = Min(anggotaErrorPS, anggotadErrorPB);
rule44 = 150 - (100 * x44);
rule44x = 150 + (100 * x44);
float x45 = Min(anggotaErrorPB, anggotadErrorNK);
rule45 = 150 - (100 * x45);
rule45x = 150 + (100 * x45);
float x46 = Min(anggotaErrorPB, anggotadErrorTS);
rule46 = 150 - (100 * x46);
rule46x = 150 + (100 * x46);
float x47 = Min(anggotaErrorPB, anggotadErrorPK);

```

```

rule47 = 150 - (100 * x47);
rule47x = 150 + (100 * x47);
float x48 = Min(anggotaErrorPB,anggotadErrorPS);
rule48 = 150 - (100 * x48);
rule48x = 150 + (100 * x8);
float x49 = Min(anggotaErrorPB,anggotadErrorPB);
rule49 = 150 - (100 * x49);
rule49x = 150 + (100 * x49);

```

```

def = ((x1 * rule1) + (x2 * rule2) +(x3 * rule3)+( x4 * rule4) +(x5 * rule5)+(x6 *
rule6) + (x7 * rule7) +(x8 * rule8)+( x9 * rule9) +(x10 * rule10) + (x11 * rule11a)
+(x11 * rule11b)+ (x12 * rule12a) +(x12 * rule12b)+ (x13 * rule13a)+ (x13 *
rule13b)+ (x14 * rule14a)+ (x14 * rule14b)+ (x15 * rule15a)+ (x15 * rule15b)+
(x16 * rule16a)+ (x16 * rule16b)+ (x17 * rule17a)+ (x17 * rule17b)+ (x18 *
rule18a)+ (x18 * rule18b)+ (x19 * rule19a)+ (x19 * rule19b)+ (x20 * rule20a)+
(x20 * rule20b)+ (x21 * rule21a)+ (x21 * rule21b)+ (x22 * rule22a)+ (x22 *
rule22b)+ (x23 * rule23a)+(x24 * rule24) + (x25 * rule25) +(x26 * rule26)+ (x27
* rule27) +(x28 * rule28)+(x29 * rule29) + (x30 * rule30) +(x31 * rule31)+ (x32 *
rule32) +(x33 * rule33)+(x34 * rule34) + (x35 * rule35) +(x36 * rule36)+ (x37 *
rule37) +(x38 * rule38)+(x39 * rule39) + (x40 * rule40) +(x41 * rule41)+ (x42 *
rule42) +(x43 * rule43)+(x44 * rule44) + (x45 * rule45) +(x46 * rule46)+ (x47 *
rule47) +(x48 * rule48)+(x49 * rule49)) / (x1 + x2+x3 + x4+x5 + x6 + x7 + x8+x9
+ x10 +x11+ x11+x12 + x12+x13 + x13+x14 + x14 + x15 + x15+x16 + x16 +x17+
x17+x18 + x18+x19 + x19+x20 + x20 + x21 + x21+x22 + x22 +x23+ x23+x24 +
x25+x26 + x27+x28 + x29 + x30 + x31+x32 + x33 +x34+ x35+x36 + x37 +x38+
x39+ x40+x41 + x42 +x43+ x44+x45 + x46 +x47+ x48+x49);

```

```

defx = ((x1 * rule1x) + (x2 * rule2x) +(x3 * rule3x)+( x4 * rule4x) +(x5 *
rule5x)+(x6 * rule6x) + (x7 * rule7x) +(x8 * rule8x)+ (x9 * rule9x) +(x10 *
rule10x) + (x11 * rule11x) +(x12 * rule12x)+ (x13 * rule13x) +(x14 * rule14x)+
(x15 * rule15x)+ (x16 * rule16x)+ (x17 * rule17x)+ (x18 * rule18x)+ (x19 *
rule19x)+ (x20 * rule20x)+ (x21 * rule21x)+ (x22 * rule22x)+ (x23 * rule23x)+

```

```

(x24 * rule24x)+ (x25 * rule25x)+ (x26 * rule26x)+ (x27 * rule27x)+ (x28 *
rule28ax)+ (x28 * rule28bx)+ (x29 * rule29ax)+ (x29 * rule29bx)+ (x30 *
rule30ax)+ (x30 * rule30bx)+ (x31 * rule31ax)+(x31 * rule31bx) + (x32 *
rule32ax) +(x32 * rule32bx)+ (x33 * rule33ax) +(x33 * rule33bx)+(x34 * rule34ax)
+ (x34 * rule34bx) +(x35 * rule35ax)+ (x35 * rule35bx)+(x36 * rule36ax) + (x36
* rule36bx) +(x37 * rule37ax)+ (x37 * rule37bx) +(x38 * rule38ax)+(x38 *
rule38bx) + (x39 * rule39ax) +(x39 * rule39bx)+ (x40 * rule40x) +(x41 * rule41x)+
(x42 * rule42x) + (x43 * rule43x) +(x44 * rule44x)+ (x45 * rule45x) +(x46 *
rule46x)+(x47 * rule47x)+(x48 * rule48x)+(x49 * rule49x)) / (x1 + x2+x3 + x4+x5
+ x6 + x7 + x8+x9 + x10 +x11+ x12+x13 + x14+x15 + x16+x17 + x18 + x19 +
x20+x21 + x22 +x23+ x24+x25 + x26+x27 + x28+x28 + x29 + x29 + x30+x30 +
x31 +x31+ x32+x32 + x33+x33 + x34+x34 + x35 + x35 + x36+x36 + x37 +x37+
x38+x38 + x39 +x39+ x40+ x41+x42 + x43 +x44+ x45+x46 + x47 +x48+ x49)-
50;
}

```

```

void runFuzzy() {
error_1 = error;
bacaTekanan();
error = tekanan;
dError = error_1;
rule();
analogWrite(blower, def);
analogWrite(exhaust, defx);
delay(100);
}

```

```

void setup()
{
pinMode(exhaust,OUTPUT);
pinMode(blower,OUTPUT);
lcd.begin(20, 4);
}

```

```

//Serial.begin(9600);
calib();
}
void loop()
{
  main:
  digitalWrite(blower,LOW);
  digitalWrite(exhaust,LOW);
  start1();
}

```

```

void calib(){
for (int x=0; x<100; x++){
calp1=hx711.read()/100.0;
calp2=hx7112.read()/100.0;
cal=calp2-calp1;
lcd.setCursor(0,1);
lcd.print(" Sensor Calibration ");
lcd.setCursor(0,3);
lcd.print("   WAIT!!   ");
delay(10);
}
lcd.clear();
}

```

```

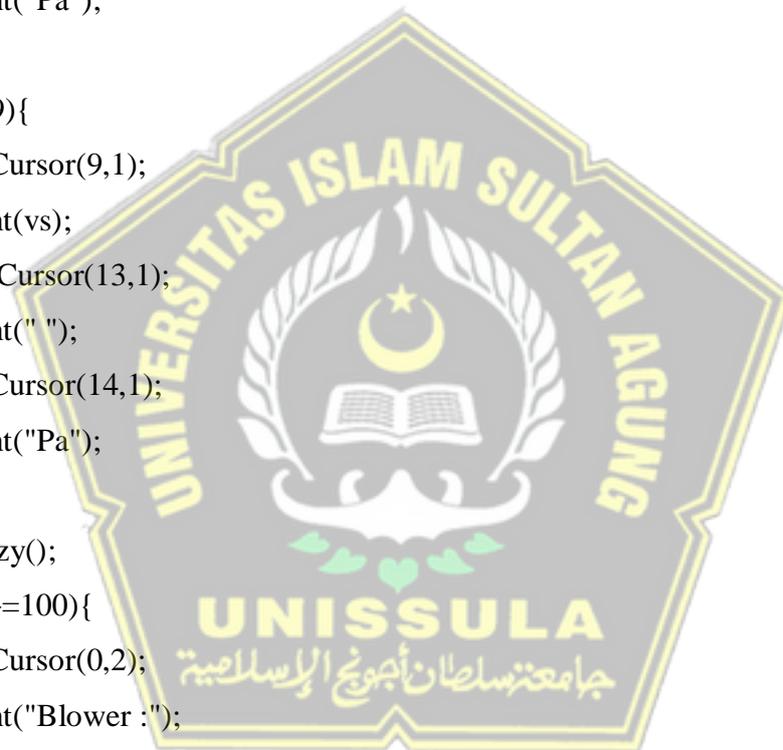
void start1(){
xx:
lcd.setCursor(0,0);
lcd.print(" Fuzzy Start ");
lcd.setCursor(0,1);
lcd.print("Tekanan:");
p1=hx711.read()/100.0+cal;

```

```

p2=hex7112.read()/100.0;
vs=p1-p2;
vs=vs/10+2;
if (vs<10){
lcd.setCursor(9,1);
lcd.print(vs);
lcd.setCursor(12,1);
lcd.print(" ");
lcd.setCursor(14,1);
lcd.print("Pa");
}
if (vs>9){
lcd.setCursor(9,1);
lcd.print(vs);
lcd.setCursor(13,1);
lcd.print(" ");
lcd.setCursor(14,1);
lcd.print("Pa");
}
runFuzzy();
if (def>=100){
lcd.setCursor(0,2);
lcd.print("Blower :");
lcd.setCursor(9,2);
lcd.print(def);
}
if (def<100){
lcd.setCursor(0,2);
lcd.print("Blower :");
lcd.setCursor(9,2);
lcd.print(def);
lcd.setCursor(11,2);

```



```

lcd.print(" ");
}
if (defx>=100){
lcd.setCursor(0,2);
lcd.print("Exhaust:");
lcd.setCursor(9,2);
lcd.print(defx);
}
if (defx<100){
lcd.setCursor(0,3);
lcd.print("Exhaust:");
lcd.setCursor(9,3);
lcd.print(defx);
lcd.setCursor(11,3);
lcd.print(" ");
}
goto xx;
}

void bacaTekanan() {
//float f=analogRead(A0); //Nilai sensor
tekanan =vs;//map(f,0,1023,0*100,20*100)/100.0;
delay(100);
}

```

Program PID:

```

#include <LiquidCrystal.h>
#include <Keypad.h>
#include <Q2HX711.h>

const byte hx711_data_pin = A5;
const byte hx711_clock_pin = A4;

```

```

const byte hx711_data_pin1 = 7;
const byte hx711_clock_pin1 = 6 ;

Q2HX711 hx711(hx711_data_pin, hx711_clock_pin); //s1
Q2HX711 hx7112(hx711_data_pin1, hx711_clock_pin1); //s2

const int rs = 13, en = 8, d4 = 9, d5 = 10, d6 = 11, d7 = 12;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
const byte ROWS = 4;
const byte COLS = 4;
char hexaKeys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};

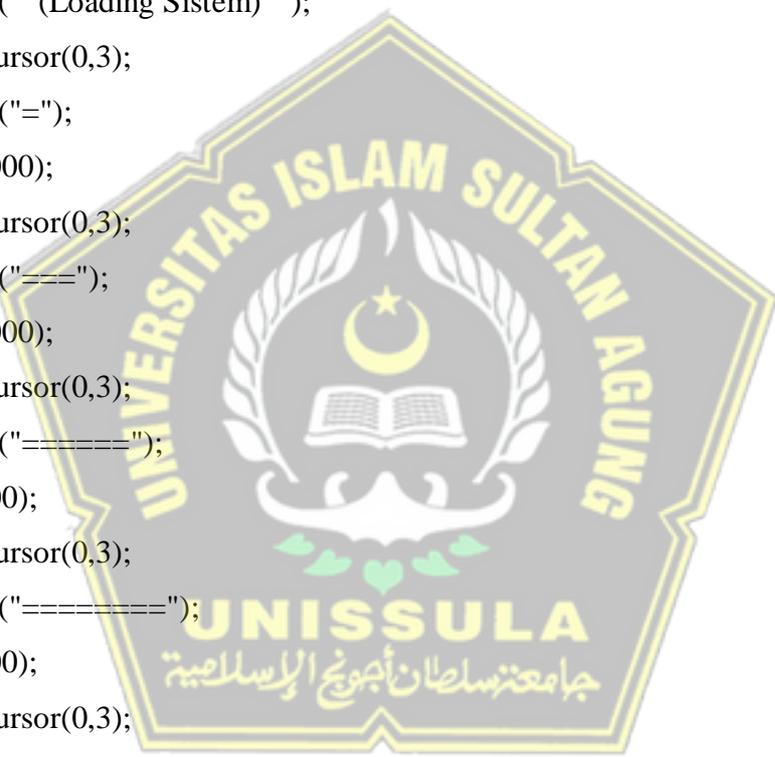
byte rowPins[ROWS] = {A3, A2, A1, A0};
byte colPins[COLS] = {2, 3, 4, 5};
Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins, colPins,
ROWS, COLS);
long p1;
long p2;
long u;
long vs;
void setup() {
  lcd.begin(20, 4);
  lcd.setCursor(0,0);
  lcd.print(" Sistem Kendali ");
  lcd.setCursor(0,1);
  lcd.print(" Tekanan Posistif ");

```

```

lcd.setCursor(0,2);
lcd.print(" Ruang Operasi ");
lcd.setCursor(0,3);
lcd.print("-----");
delay(3000);
lcd.clear();
lcd.setCursor(0,0);
lcd.print(" Susilo Ari W ");
lcd.setCursor(0,2);
lcd.print(" (Loading Sistem) ");
lcd.setCursor(0,3);
lcd.print("=");
delay(1000);
lcd.setCursor(0,3);
lcd.print("====");
delay(1000);
lcd.setCursor(0,3);
lcd.print("=====");
delay(500);
lcd.setCursor(0,3);
lcd.print("=====");
delay(500);
lcd.setCursor(0,3);
lcd.print("=====");
delay(300);
lcd.setCursor(0,3);
lcd.print("=====");
delay(300);
lcd.setCursor(0,3);
lcd.print("=====");
delay(100);
lcd.setCursor(0,3);

```



```

lcd.print("=====");
delay(100);
lcd.clear();
}

```

```

void loop() {
  main:
  monitorx();
  goto main;
}

```

```

void monitorx(){
  char customKey = customKeypad.getKey();
  if (customKey=='A'){
    lcd.clear();
    start1();
  }
  p1=hx711.read()/100.0+1500;
  p2=hx7112.read()/100.0;
  lcd.setCursor(0,0);
  lcd.print("  monitor Sistem  ");
  lcd.setCursor(0,1);
  lcd.print("Kendali: PID");
  lcd.setCursor(0,2);
  lcd.print("S1:");
  lcd.setCursor(3,2);
  lcd.print(p1);
  lcd.setCursor(10,2);
  lcd.print("S2:");
  lcd.setCursor(14,2);
  lcd.print(p2);
  lcd.setCursor(0,3);

```

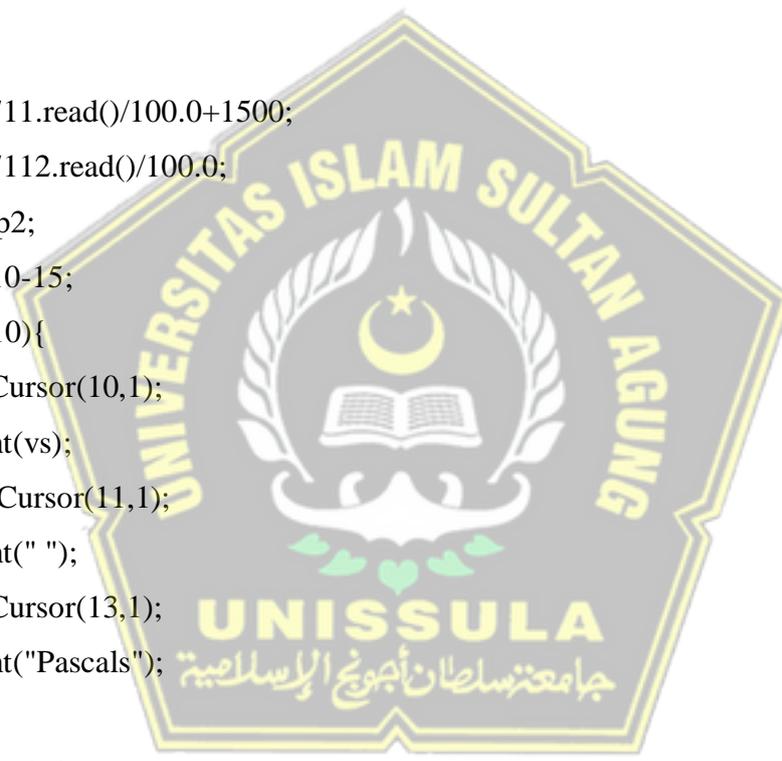


```

lcd.print("D Start, A Menu");
delay(200);
}

void start1(){
  xx:
  char customKey = customKeypad.getKey();
  if (customKey=='D'){
    lcd.clear();
    loop();
  }
  p1=hx711.read()/100.0+1500;
  p2=hx7112.read()/100.0;
  vs=p1-p2;
  vs=vs/10-15;
  if (vs<10){
    lcd.setCursor(10,1);
    lcd.print(vs);
    lcd.setCursor(11,1);
    lcd.print(" ");
    lcd.setCursor(13,1);
    lcd.print("Pascals");
  }
  if (vs>=10){
    lcd.setCursor(10,1);
    lcd.print(vs);
    lcd.setCursor(12,1);
    lcd.print(" ");
    lcd.setCursor(13,1);
    lcd.print("Pascals");
  }
  if(vs>10){

```



```

lcd.setCursor(0,2);
lcd.print("Status Blower: OFF");
}
if(vs<=10){
lcd.setCursor(0,2);
lcd.print("Status Blower: ON ");
}
if(vs>11){
lcd.setCursor(0,3);
lcd.print("Status Exhaust: ON ");
}
if(vs<=10){
lcd.setCursor(0,3);
lcd.print("Status Exhaust: OFF ");
}
lcd.setCursor(0,0);
lcd.print("  System Start  ");
lcd.setCursor(0,1);
lcd.print("Variabel:");
lcd.setCursor(0,2);
lcd.print("Status Blower:");
lcd.setCursor(0,3);
lcd.print("Status Exhaust:");
delay(200);
goto xx;
}

```

