



TUGAS AKHIR - TF 181801

**PERANCANGAN *SMART ROOM SYSTEM*
DENGAN INDIKATOR KENYAMANAN SUHU DAN
PENCAHAYAAN BERBASIS KONTROL LOGIKA
*FUZZY-PID (KLF-PID)***

**ARDYAS NUR AUFA
NRP. 02311540000106**

**Dosen Pembimbing:
Dr. Imam Abadi, S.T., M.T.**

**Departemen Teknik Fisika
Fakultas Teknologi Industri dan Rekayasa Sistem
Institut Teknologi Sepuluh Nopember
Surabaya
2020**

“Halaman ini sengaja dikosongkan”



FINAL PROJECT - TF 181801

***SMART ROOM SYSTEM DESIGN WITH
TEMPERATURE AND LIGHTING COMFORT
INDICATOR BASED ON FUZZY LOGIC
CONTROL-PID (FLC-PID)***

***ARDYAS NUR AUFA
NRP. 02311540000106***

***Supervisor:
Dr. Imam Abadi, S.T., M.T.***

***ENGINEERING PHYSICS DEPARTMENT
Faculty of Industrial Technology and System Engineering
Sepuluh Nopember Institute of Technology
Surabaya
2020***

“Halaman ini sengaja dikosongkan”

PERNYATAAN BEBAS PLAGIARISME

Saya yang bertanda tangan di bawah ini:

Nama : Ardyas Nur Aufa
NRP : 02311540000106
Departemen : Teknik Fisika FTIRS-ITS

Dengan ini menyatakan bahwa tugas akhir saya yang berjudul "PERANCANGAN *SMART ROOM SYSTEM* DENGAN INDIKATOR KENYAMANAN SUHU DAN PENCAHAYAAN BERBASIS KONTROL LOGIKA *FUZZY-PID* (KLF-PID)" adalah bebas plagiarisme. Apabila pernyataan ini terbukti tidak benar, maka saya bersedia menerima sanksi sesuai ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat dengan sebenar-benarnya.

Surabaya, 24 Januari 2020
Yang membuat pernyataan,



Ardyas Nur Aufa
NRP. 02311540000106

“Halaman ini sengaja dikosongkan”

LEMBAR PENGESAHAN

**PERANCANGAN *SMART ROOM SYSTEM* DENGAN
INDIKATOR KENYAMANAN SUHU DAN
PENCAHAYAAN BERBASIS KONTROL LOGIKA
FUZZY-PID (KLF-PID)**

TUGAS AKHIR

OLEH :

**Ardyas Nur Aufa
NRP. 02311540000106**

**Surabaya, 24 Januari 2020
Mengetahui/Menyetujui**

Pembimbing,



**Dr. Imam Abadi , S.T., M.T.
NIP. 19761006 199903 1 002**



“Halaman ini sengaja dikosongkan”

**PERANCANGAN *SMART ROOM SYSTEM* DENGAN
INDIKATOR KENYAMANAN SUHU DAN
PENCAHAYAAN BERBASIS KONTROL LOGIKA
*FUZZY-PID***

TUGAS AKHIR


Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Teknik
pada
Bidang Minat Rekayasa Instrumentasi
Program Studi S-1 Teknik Fisika
Fakultas Teknologi Industri dan Rekayasa Sistem
Institut Teknologi Sepuluh Nopember

Oleh :

Ardyas Nur Aufa
NRP. 02311540000106

Disetujui oleh Tim Penguji Tugas Akhir

1. Dr. Imam Abadi, S.T., M.T.

 (Pembimbing)


2. Dr. Ir. Ali Musyafa', M.Sc.

 (Ketua Penguji)

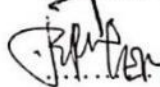
3. Dr. Ir. Ronny Dwi Noriyati, M. Kes.

 (Penguji)

4. Dr. Suyanto, S.T., M.T.

 (Penguji)

5. Erna Septyaningrum, S.T., M.T.

 (Penguji)

**SURABAYA
JANUARI, 2020**

“Halaman ini sengaja dikosongkan”

**PERANCANGAN *SMART ROOM SYSTEM* DENGAN
INDIKATOR KENYAMANAN SUHU DAN
PENCAHAYAAN BERBASIS KONTROL LOGIKA
*FUZZY-PID (KLF-PID)***

Nama Mahasiswa : Ardyas Nur Aufa
NRP : 02311540000106
Program Studi : S1 Teknik Fisika
Dosen Pembimbing : Dr. Imam Abadi, S.T., M.T.

ABSTRAK

Sering kita temui kasus pemakaian lampu dan AC secara tidak efektif. Ketika pencahayaan alami telah mencapai tingkatan yang sesuai SNI 03-6575-2001, yaitu 120-250 lux, tetapi menggunakan semua lampu di dalam ruangan. Ketika cuaca sedang panas, AC diatur dengan suhu paling rendah secara terus menerus, padahal suhu AC dapat disesuaikan dengan perubahan suhu ruangan, agar mencapai suhu yang sesuai SNI 03-6572-2001 yaitu $25 \pm 1^\circ\text{C}$. Oleh karena itu dilakukan perancangan *Smart Room System* dengan indikator kenyamanan suhu dan pencahayaan berbasis Kontrol Logika *Fuzzy-PID (KLF-PID)*. Pengambilan data suhu menggunakan sensor DHT22. Komunikasi data menggunakan *wireless-IR*. Pendeteksi gerakan manusia menggunakan sensor PIR. Pengaturan jeda ON-OFF lampu menggunakan modul *relay*. Sistem kontrol suhu disimulasikan pada Matlab dengan respons sistem memiliki *rise time* 3,078 detik, *peak time* 3,103 detik, *undershoot* $24,87^\circ\text{C}$ (0,52%), dan *settling time* 3,056 detik. Dengan nilai akhir *gain* KLF-PID: $K_p=2,379$; $K_i=0,9961$; dan $K_d=1,939$. Sedangkan pengujian pada *real pant* menghasilkan respons sistem yang memiliki *rise time* 95 detik, *peak time* 110 detik, *undershoot* $24,6^\circ\text{C}$ (1,6%) dan *settling time* 82 detik. Dengan nilai akhir *gain* KLF-PID: $K_p=2,35$; $K_i=1,5$; dan $K_d=1,76$.

Kata kunci: *Smart Room System, Pencahayaan, Suhu, Kontrol Logika Fuzzy, PID, KLF-PID.*

“Halaman ini sengaja dikosongkan”

**SMART ROOM SYSTEM DESIGN WITH
THERMAL AND LIGHTING COMFORT INDICATOR
BASED ON FUZZY LOGIC CONTROL-PID (FLC-PID)**

Name : Ardyas Nur Aufa
NRP : 02311540000106
Study Program : S1 Teknik Fisika
Supervisor : Dr. Imam Abadi, S.T., M.T.

ABSTRACT

Often, we encounter cases of ineffective use of lamps and air conditioners. When natural lighting has reached the appropriate level of SNI 03-6575-2001, which is 120-250 lux, but we use all the lights in the room. When the weather is hot, the air conditioner is set to the lowest temperature continuously, even though the temperature of the air conditioner can be adjusted to changes in room temperature, in order to reach the appropriate temperature SNI 03-6572-2001 which is $25 \pm 1^{\circ}\text{C}$. Therefore, a Smart Room System design with a comfort indicator of temperature and lighting based on Fuzzy-PID Logic Control (KLF-PID) was carried out. Temperature data acquisition uses the DHT22 sensor. Data communication uses wireless-IR. Human motion detection uses a PIR sensor. The lamp ON-OFF pause setting uses a relay module. The temperature control system is simulated on Matlab with the system response having a rise time of 3.078 seconds, peak time of 3.103 seconds, overshoot of 24.87°C (0.52%), and settling time of 3.056 seconds. With the final value of the KLF-PID gain: $K_p = 2,379$; $K_i = 0.9961$; and $K_d = 1,939$. While testing on a real plant produces a system response that has a rise time of 95 seconds, a peak time of 110 seconds, an undershoot of 24.6°C (1.6%) and a settling time of 82 seconds. With the final value of KLF-PID gain: $K_p = 2.35$; $K_i = 1.5$; and $K_d = 1.76$.

Keywords: *Smart Room System, Lighting, Temperature, Fuzzy Logic Control, PID, KLF-PID.*

“Halaman ini sengaja dikosongkan”

KATA PENGANTAR

Puji syukur kepada Allah SWT yang senantiasa melimpahkan rahmat dan hidayah-Nya, serta selawat dan salam kepada Nabi Muhammad SAW sehingga dapat terselesaikannya tugas akhir beserta laporan tugas akhir yang berjudul **“Perancangan *Smart Room System* Dengan Indikator Kenyamanan Suhu Dan Pencahayaan Berbasis Kontrol Logika *Fuzzy-PID* (KLF-PID)”**.

Penulis telah banyak memperoleh bantuan dari berbagai pihak dalam penyelesaian tugas akhir dan laporan tugas akhir ini. Penulis mengucapkan terima kasih kepada:

1. Bapak Dr. Suyanto, S.T., M.T. selaku Kepala Departemen Teknik Fisika ITS yang telah memberikan ilmu, bimbingan serta sarana dan prasarana selama menempuh pendidikan di Teknik Fisika ITS.
2. Bapak Dr. Imam Abadi, S.T., M.T. selaku dosen pembimbing yang telah dengan sabar memberikan petunjuk, ilmu, serta bimbingan dalam pengerjaan tugas akhir ini.
3. Bapak Dr. Ir. Ali Musyafa', M. Sc., Ibu Dr. Ir. Ronny Dwi N, M.Kes., dan Ibu Erna Septyaningrum, S.T., M.T. selaku dosen penguji yang telah memberikan saran kepada penulis.
4. Ibu Dr. Ing. Doty Dewi Risanti, S.T., M.T. selaku dosen wali yang telah membimbing penulis selama perkuliahan dan memberikan ilmu serta nasihat yang sangat bermanfaat.
5. Seluruh Dosen Departemen Teknik Fisika ITS atas segala ilmu, bimbingan, petunjuk, dan nasihat yang sangat bermanfaat.
6. Seluruh Karyawan Departemen Teknik Fisika ITS yang telah membantu proses administrasi selama masa perkuliahan.
7. Kedua orang tua dan keluarga yang telah mendoakan penulis, mendukung, memberikan semangat dan kasih sayang, serta memberikan dukungan moril dan materiil.
8. Faruq, Gugus, Rifki, Rokhmat, dan Irul atas segala semangat dan bantuannya selama pengerjaan tugas akhir ini.

9. Teman-teman *Transcendent Frontier*, Teknik Fisika angkatan 2015 terima kasih atas pelajarannya, atas kekeluargaannya, terima kasih untuk kisahnya selama ini.
10. Seluruh mahasiswa Teknik Fisika ITS yang telah memberikan banyak pelajaran, membimbing, dan memberikan saran, serta doa kepada penulis.
11. Dan segenap orang-orang berjasa dalam hidup saya yang tidak bisa saya sebutkan satu persatu yang telah memberikan dukungan, semangat, dan doa untuk saya.

Penulis sadar bahwa penulisan laporan tugas akhir ini masih terdapat kekurangan. Namun, semoga laporan ini dapat memberikan kontribusi yang berarti dan menambah wawasan yang bermanfaat bagi pembaca, keluarga besar Teknik Fisika ITS khususnya, dan sivitas akademika ITS pada umumnya.

Surabaya, 24 Januari 2020

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
COVER PAGE	iii
PERNYATAAN BEBAS PLAGIARISME	v
LEMBAR PENGESAHAN.....	vii
ABSTRAK	xi
<i>ABSTRACT</i>	xiii
KATA PENGANTAR	xv
DAFTAR ISI.....	xvii
DAFTAR GAMBAR.....	xix
DAFTAR TABEL	xxi
DAFTAR NOTASI.....	xxiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan.....	3
1.4 Batasan Masalah	3
1.5 Sistematika Laporan.....	4
BAB II TINJAUAN PUSTAKA	7
2.1 <i>Smart Room System</i>	7
2.2 Model Sistem Kontrol Suhu Ruangan ber-AC	7
2.3 Mekanisme Sensor <i>Passive Infrared</i>	9
2.4 Kontrol PID	10
2.5 Kontrol Logika <i>Fuzzy</i>	12
2.6 Kontrol Logika <i>Fuzzy</i> -PID	16
BAB III METODOLOGI PENELITIAN.....	19
3.1 Studi Literatur	20
3.2 Pengambilan Data Sekunder.....	21
3.2.1 Parameter Termal Ruangan.....	21
3.2.2 Aktuator	21
3.2.3 Sensor	22
3.2.4 Mikrokontroler	23
3.3 Pengambilan Data Primer	23
3.4 Pemodelan Sistem Kontrol Suhu Ruangan.....	24
3.5 Perancangan <i>Hardware</i>	25

3.6 Pengujian Komponen	28
3.7 Perancangan Sistem Kontrol Logika <i>Fuzzy</i> -PID.....	29
3.8 Pemrograman Arduino	34
BAB IV ANALISIS DAN PEMBAHASAN	39
4.1 Hasil Pengujian Sensor DHT22	39
4.2 Hasil Pengujian Sensor PIR.....	40
4.3 Hasil Simulasi Sistem Kontrol Logika <i>Fuzzy</i> -PID.....	41
4.3.1 Uji <i>Open Loop</i>	42
4.3.2 Uji Respons Masukan <i>Step</i>	42
4.2.3 Uji <i>Tracking Setpoint</i> Naik	44
4.3.4 Uji <i>Tracking Setpoint</i> Turun	46
4.4 Hasil Pengujian <i>Smart Room System</i>	48
4.5 Komparasi Hasil Simulasi dengan Pengujian <i>Real Plant</i> ..	51
BAB V PENUTUP	53
5.1 Kesimpulan.....	53
5.2 Saran.....	54
DAFTAR PUSTAKA.....	55
LAMPIRAN 1 Pengambilan Data Primer.....	59
LAMPIRAN 2 <i>Source Code</i> Arduino.....	61
BIODATA PENULIS	87

DAFTAR GAMBAR

Gambar 2. 1 Rangkaian dan Model Sensor PIR [17]	10
Gambar 2. 2 Skema Kontrol PID [19].....	11
Gambar 2. 3 Skema Kontrol Logika Fuzzy [22].....	13
Gambar 2. 4 Fungsi Keanggotaan Triangular [22].....	14
Gambar 2. 5 Fungsi Keanggotaan Trapezoid [22]	14
Gambar 2. 6 Fungsi Keanggotaan Gaussian [22].....	15
Gambar 2. 7 Skema Kontrol Logika Fuzzy-PID [24].....	16
Gambar 2. 8 Diagram Alir KLF-PID	17
Gambar 3. 1 Diagram Alir Metode Penelitian	19
Gambar 3. 2 Diagram Alir Metode Penelitian (lanjutan)	20
Gambar 3. 3 Rancangan Hardware dengan Software Fritzing..	25
Gambar 3. 4 Sub-sistem Kontrol Suhu.....	27
Gambar 3. 5 Sub-sistem Kontrol Pencahayaan.....	27
Gambar 3. 6 Integrasi Sistem.....	28
Gambar 3. 7 Realisasi <i>Hardware</i>	28
Gambar 3. 8 Fungsi Keanggotaan Error	29
Gambar 3. 9 Fungsi Keanggotaan Delta Error.....	29
Gambar 3. 10 Fungsi Keanggotaan $K'p$	31
Gambar 3. 11 Fungsi Keanggotaan $K'd$	31
Gambar 3. 12 Fungsi Keanggotaan α	31
Gambar 3. 13 Blok Sistem Kontrol Logika Fuzzy-PID.....	33
Gambar 3. 14 Blok Sistem Kontrol PID	33
Gambar 3. 15 Blok Sistem Kontrol Logika Fuzzy-PID.....	34
Gambar 3. 16 Blok Sistem Perbandingan Sistem Kontrol.....	34
Gambar 3. 17 Penambahan <i>Library</i> Sistem dan Penentuan <i>Pin</i> DHT22, Sensor PIR, <i>Relay</i>	35
Gambar 3. 18 Penentuan <i>RAW Data</i> AC dan Variabel untuk Kontrol Suhu dengan <i>IR</i>	35
Gambar 3. 19 Penentuan Variabel untuk Kontrol PID dan FIS	36
Gambar 3. 20 <i>Setup</i> Program Utama.....	36

Gambar 3. 21	Fungsi Kontrol Pencahayaan	37
Gambar 3. 22	Fungsi Kontrol PID	37
Gambar 3. 23	Fungsi FIS.....	38
Gambar 4. 1	Pengujian Sensor DHT22	39
Gambar 4. 2	Hasil Pengujian Sensor DHT22	40
Gambar 4. 3	Grafik Respons Sistem Open Loop.....	42
Gambar 4. 4	Grafik Respons Masukan Step.....	43
Gambar 4. 5	Nilai Gain KLF-PID Respons Masukan Step	44
Gambar 4. 6	Grafik Tracking Setpoint Naik.....	45
Gambar 4. 7	Nilai Gain KLF-PID Tracking Setpoint Naik	46
Gambar 4. 8	Grafik Tracking Setpoint Turun.....	47
Gambar 4. 9	Nilai Gain KLF-PID Tracking Setpoint Turun	48
Gambar 4. 10	Hasil Pengujian Smart Room System.....	49
Gambar 4. 11	Nilai Gain KLF-PID pada Smart Room System	49
Gambar 4. 12	Peletakan Lampu dan Sensor PIR	50
Gambar 4. 13	Pengukuran Tingkat Pencahayaan.....	50
Gambar 4. 14	Hasil Uji Beban Smart Room System	51

DAFTAR TABEL

Tabel 3. 1 Parameter Termal Ruangan [26].....	21
Tabel 3. 2 Spesifikasi AC (Mitsubishi SRK10CEV-4).....	22
Tabel 3. 3 Spesifikasi Lampu LED (Philips LED Bulb).....	22
Tabel 3. 4 Spesifikasi Sensor DHT22.....	22
Tabel 3. 5 Spesifikasi Sensor PIR (HC-SR501).....	23
Tabel 3. 6 Spesifikasi Mikrokontroler (Arduino Mega 2560) ...	23
Tabel 3. 7 Sambungan Pin Komponen dan Arduino.....	26
Tabel 3. 8 Rule Base Keluaran $K'p$	32
Tabel 3. 9 Rule Base Keluaran $K'd$	32
Tabel 3. 10 Rule Base Keluaran α	33
Tabel 4. 1 Hasil Pengujian Sensor PIR.....	40
Tabel 4. 2 Hasil Pengujian Sensor PIR (lanjutan).....	41
Tabel 4. 3 Indikator Performansi Respons Masukan Step.....	43
Tabel 4. 4 Indikator Performansi Tracking Setpoint Naik.....	45
Tabel 4. 5 Indikator Performansi Tracking Setpoint Turun.....	47
Tabel 4. 6 Komparasi Karakteristik Respons Sistem.....	51
Tabel 4. 7 Komparasi Indikator Performansi Sistem.....	52

“Halaman ini sengaja dikosongkan”

DAFTAR NOTASI

C_1	= koefisien kapasitas kalor ruangan, $\text{kJ}/^\circ\text{C}$
θ_0	= suhu udara dalam ruangan, $^\circ\text{C}$
Q_c	= panas yang diserap oleh <i>fan coil unit</i> , W
G_s	= volume suplai udara di ruangan ber-AC, Kg/s
c_1	= kalor jenis udara, $\text{KJ}/(\text{Kg}^\circ\text{C})$
θ'_0	= suhu udara dalam ruangan sebelum diolah, $^\circ\text{C}$
Q_2	= disipasi panas (alat elektronik, manusia), W
Q_4	= perpindahan panas dari eksterior ke interior, W
θ_4	= suhu struktur eksternal, $^\circ\text{C}$
R	= resistansi termal permukaan internal, $\text{m}^2\text{C}/\text{W}$
K	= faktor pembesaran <i>regulated object</i>
T	= konstanta waktu <i>regulating object</i>
θ_c	= variasi suhu suplai udara <i>regulating quantity</i>
θ_f	= variasi suhu suplai udara <i>interference quantity</i>
τ	= <i>transport delay</i> ,
K_p	= <i>proportional gain</i>
K_i	= <i>integral gain</i>
K_d	= <i>derivative gain</i>

“Halaman ini sengaja dikosongkan”

BAB I

PENDAHULUAN

1.1 Latar Belakang

Energi listrik merupakan salah satu sumber energi yang sangat penting bagi manusia. Energi listrik digunakan sebagai sumber utama dalam kebutuhan rumah tangga maupun digunakan oleh berbagai perusahaan dan industri yang berperan sebagai penggerak ekonomi negara. Dengan berbagai kemudahan bagi masyarakat Indonesia dalam menggunakan energi listrik, sering terjadi kasus pemborosan energi listrik yang tidak disadari dan berdampak pada meningkatnya tarif pemakaian listrik. Oleh karena itu berbagai metode penelitian telah dilakukan untuk menghemat pemakaian daya listrik.

Dalam aktivitas harian, sering ditemukan fenomena-fenomena pemborosan listrik yang terjadi. Misalnya ketika kita telah selesai menggunakan ruangan untuk bekerja maupun belajar, lampu dan pendingin ruangan (AC) tetap dibiarkan dalam kondisi menyala. Hal ini menyebabkan pemakaian listrik akan terus berlangsung dan berdampak pada pemborosan energi, padahal di Indonesia konsumsi listrik terus mengalami peningkatan setiap tahunnya di berbagai sektor pemakaian listrik. Konsumsi listrik nasional pada tahun 2017 mencapai 1.012 kWh/kapita, naik 5,9% dari tahun sebelumnya [1]. Sedangkan, tarif listrik rata-rata untuk pelanggan rumah tangga adalah Rp 1.006 per kWh atau naik 18,91 persen dari tarif akhir 2016. Kenaikan ini merupakan yang tertinggi, jika dibandingkan golongan lainnya [2]. Sehingga penghematan pemakaian daya listrik memegang peranan penting dalam keberlanjutan energi.

Lampu dan AC merupakan contoh peralatan yang sering digunakan untuk memenuhi kenyamanan seseorang dalam bekerja dan beraktivitas lainnya. Dalam kasus normal, sering ditemukan adanya pemakaian lampu dan AC secara tidak efektif. Misal ketika pencahayaan alami dari cahaya matahari telah mencapai tingkatan yang sesuai dengan SNI, pada ruang kerja adalah 120-250 lux, tetapi tetap menggunakan semua lampu yang ada dalam ruangan.

Pada pemakaian AC, penggunaan yang tidak efektif sering terjadi ketika cuaca sedang panas, AC diatur dengan suhu paling rendah secara terus menerus, padahal semestinya pengaturan suhu AC dapat disesuaikan dengan perubahan suhu ruangan, agar dapat mencapai suhu ruangan sesuai SNI yaitu $25\pm 1^{\circ}\text{C}$. Hal ini diperlukan untuk mencegah pemborosan energi listrik karena setiap penambahan 1°C pada AC akan menghemat 3-5% dari daya keseluruhan unit AC [3].

Untuk mengatasi kelalaian manusia ketika menggunakan energi listrik, dibutuhkan suatu sistem cerdas yang dapat mencegah pemborosan energi listrik. Salah satu upaya yang dapat dilakukan untuk mewujudkan hal tersebut yaitu otomatisasi perangkat listrik pada ruangan. Sebelumnya telah dilakukan penelitian tentang perancangan sistem penghematan energi otomatis. Pada sistem ini dirancang sebuah sistem cerdas pada ruangan yang meliputi lampu, kipas, dan detektor asap yang dikontrol secara otomatis oleh sebuah mikrokontroler [4]. Penelitian lain juga telah dilakukan untuk merancang sistem pengendali lampu dan kipas otomatis menggunakan sensor inframerah [5] [6] [7] [8]. Sistem ini mampu mengatasi masalah pada penelitian sebelumnya terkait kerancuan arah pada sensor inframerah untuk menghitung jumlah orang yang memasuki ruangan.

Pada umumnya di Indonesia, perangkat listrik yang digunakan pada sebuah ruangan tidak hanya terdiri dari lampu saja, namun juga terdapat AC yang menyumbang pemakaian daya yang cukup besar. Sehingga perlu dirancang sebuah sistem kontrol temperatur AC agar menyesuaikan dengan kondisi ruangan. Sebelumnya telah dilakukan penelitian yang membandingkan respons sistem kontrol antara metode PID dan *Fuzzy* pada AC, hasilnya Kontrol Logika *Fuzzy* memiliki respons yang lebih cepat dan hasil keluaran yang lebih akurat [9] [10] [11]. Pada penelitian lain dikembangkan sistem kontrol HVAC dengan membandingkan metode PID, *Fuzzy*, *Fuzzy-PI*, dan *Fuzzy-PID*. Pada penelitian tersebut didapatkan bahwa Kontrol Logika *Fuzzy-PID* memiliki respons yang lebih cepat, nilai *overshoot* yang lebih rendah, dan hasil keluaran yang lebih akurat [12] [13] [14].

Pada akhirnya, sistem ruangan cerdas yang mampu mengatur penggunaan lampu dan AC secara otomatis sangat dibutuhkan agar mengurangi pemborosan pemakaian daya listrik sehari-hari. Di samping itu, perlu juga untuk dipilih metode kontrol yang sesuai agar sistem ruangan cerdas yang dibuat dapat memberikan respons yang baik. Oleh karena itu, pada tugas akhir ini akan dibuat *Smart Room System* dengan judul “Perancangan *Smart Room System* dengan Indikator Kenyamanan Suhu dan Pencahayaan Berbasis Kontrol Logika *Fuzzy*-PID (KLF-PID)”.

1.2 Rumusan Masalah

Dari latar belakang yang telah diuraikan di atas, maka didapatkan permasalahan dalam tugas akhir ini, yaitu sebagai berikut:

- a. Bagaimana merancang *Smart Room System* dengan indikator kenyamanan suhu dan pencahayaan?
- b. Bagaimana menentukan parameter kontrol PID menggunakan algoritma *Fuzzy* untuk mendapatkan sistem kontrol yang stabil?

1.3 Tujuan

Tujuan yang akan dicapai dari penelitian tugas akhir ini adalah sebagai berikut:

- a. Mampu merancang *Smart Room System* dengan indikator kenyamanan suhu dan pencahayaan.
- b. Mampu menentukan parameter kontrol PID menggunakan algoritma *Fuzzy* untuk mendapatkan sistem kontrol yang stabil.

1.4 Batasan Masalah

Adapun batasan masalah dalam penelitian ini antara lain:

- a. Metode kontrol suhu yang digunakan yaitu Kontrol Logika *Fuzzy*-PID (KLF-PID).
- b. Metode kontrol pencahayaan yang digunakan yaitu *switch* ON-OFF dengan masukan berupa daerah yang digunakan manusia untuk beraktivitas dari sensor gerakan PIR.

- c. Mikrokontroler yang digunakan dalam penelitian ini yaitu *Arduino Mega 2560 Rev 3*.
- d. Jenis sensor yang digunakan dalam penelitian ini yaitu sensor termistor (DHT22) untuk sensor suhu dan TSOP34438 untuk sensor inframerah.
- e. Komunikasi data yang digunakan untuk menghubungkan mikrokontroler dan AC adalah *Wireless-IR*.
- f. Ruang yang dijadikan objek dalam penelitian ini adalah ruangan P-212 Departemen Teknik Fisika ITS.
- g. Simulasi yang dilakukan dalam penelitian ini menggunakan *software* Matlab 2018b.
- h. Perancangan *hardware* dilakukan menggunakan *software* Fritzing 0.9.2b.
- i. Pengambilan data primer tingkat pencahayaan dilakukan pada tanggal 3 Mei 2019 dan parameter sistem suhu ruangan pada tanggal 11 September 2019.
- j. Pengambilan data untuk pengujian *smart room system* dilakukan pada tanggal 6 Januari 2020.

1.5 Sistematika Laporan

Sistematika penulisan laporan tugas akhir adalah sebagai berikut:

- a. **BAB I PENDAHULUAN**
Pada bab I ini terdiri dari latar belakang, perumusan masalah, tujuan, batasan masalah, dan sistematika laporan.
- b. **BAB II TINJAUAN PUSTAKA**
Pada bab II ini dibahas mengenai teori-teori penunjang yang berkaitan dengan penelitian yang akan dilakukan, seperti pemodelan sistem kontrol suhu ruangan ber-AC, mekanisme sensor *Passive Infrared*, kontrol PID, Kontrol Logika *Fuzzy*, dan Kontrol Logika *Fuzzy*-PID.
- c. **BAB III METODOLOGI PENELITIAN**
Pada bab III ini berisi mengenai penjelasan secara rinci mengenai metode yang harus dilalui untuk mencapai tujuan dan simpulan akhir dari penelitian. Produk akhir dari tahap

ini adalah perancangan *hardware/prototype* yang siap untuk diuji dan dianalisis.

d. **BAB IV ANALISIS DATA DAN PEMBAHASAN**

Pada bab IV ini berisi tentang analisis data dan pembahasan hasil simulasi menggunakan *software* Matlab dan hasil pengambilan data dari *hardware/prototype*.

e. **BAB V KESIMPULAN DAN SARAN**

Pada bab V ini berisi kesimpulan tentang tugas akhir yang telah dilakukan berdasarkan analisis data dan pembahasan serta saran sebagai penunjang maupun rekomendasi pengembangan untuk tugas akhir yang selanjutnya.

“Halaman ini sengaja dikosongkan”

BAB II

TINJAUAN PUSTAKA

Pada bagian ini berisi tentang teori yang berkaitan dengan Tugas Akhir, di antaranya adalah model sistem kontrol suhu ruangan ber-AC, mekanisme sensor *Passive Infrared*, kontrol PID, Kontrol Logika *Fuzzy*, dan Kontrol Logika *Fuzzy*-PID.

2.1 Smart Room System

Smart Room System adalah suatu kumpulan dari beberapa komponen dalam sebuah ruangan yang saling berkaitan dan mampu menyelesaikan suatu masalah atau tujuan tertentu secara otomatis. Sebelumnya telah dilakukan penelitian tentang *Smart Meeting Room System*, pada penelitian tersebut dirancang sistem yang mampu mendukung kegiatan pertemuan, yaitu mengatur telepon selular agar mati, sistem pencahayaan, sistem pendingin ruangan, penyeduh kopi, dll [15]. Pada penelitian lain dirancang *Smart Room System* yang mampu memantau kondisi kesehatan pasien di rumah sakit, yaitu suhu pasien, tingkat salinitas, dan status keberadaan pasien di tempat tidur [16].

Pada penelitian ini dirancang *Smart Room System* yang mampu memenuhi kenyamanan pada ruang kerja. Parameter yang digunakan untuk memenuhi kondisi nyaman pada ruang kerja, yaitu suhu dan pencahayaan. Standar kenyamanan suhu untuk perancangan umum di Indonesia menggunakan SNI 03-6572-2001 yaitu $25 \pm 1^\circ\text{C}$. Sedangkan standar kenyamanan pencahayaan pada ruang kerja di Indonesia menggunakan SNI 03-6575-2001, yaitu 120-250 lux.

2.2 Model Sistem Kontrol Suhu Ruangan ber-AC

Sistem kontrol suhu ruangan ber-AC merupakan sistem kontrol yang kompleks. Gangguan internal ruangan ber-AC adalah hasil dari panas dan kelembaban yang dihasilkan oleh peralatan listrik, lampu, tubuh manusia, dll. Gangguan eksternal ruang

berasal dari panas radiasi matahari dan panas yang dihasilkan oleh perbedaan suhu *indoor* dan *outdoor*. Beberapa faktor ini stabil sementara beberapa tidak stabil dan berubah dengan musim. Karena itu, penting bagi sistem pendingin udara untuk menetapkan model objek tertentu terlebih dahulu.

Dalam penelitian ini, digunakan satu ruang pendingin udara sebagai objek penelitian di bagian ini. Seluruh sistem pendingin udara terdiri dari aktuator yang berupa *fan coil unit* dan *plant* yang berupa ruangan ber-AC. Menurut hukum konservasi energi, jumlah panas yang memasuki ruangan per unit waktu dikurangi jumlah panas yang mengalir keluar ruangan per unit waktu sama dengan laju perubahan jumlah panas yang disimpan di ruangan ber-AC. Hubungan berikut dapat diperoleh dalam persamaan 2.1 berikut:

$$C_1 \frac{d\theta_0}{dt} = Q_c + G_s c_1 \theta'_0 Q_4 + Q_2 - G_s c_1 \theta'_0 + \frac{\theta_4 \theta_0}{R} + Q_2 - G_s c_1 \theta'_0 \quad (2.1)$$

Di mana C_1 adalah koefisien kapasitas kalor ruangan, $\text{kJ}/^\circ\text{C}$; θ_0 adalah suhu udara dalam ruangan, $^\circ\text{C}$; Q_c adalah panas yang diserap oleh *fan coil unit*, W ; G_s adalah volume suplai udara di ruangan ber-AC, Kg/s ; c_1 adalah kalor jenis udara, $\text{KJ}/(\text{Kg}^\circ\text{C})$; θ'_0 adalah suhu udara dalam ruangan sebelum diolah, $^\circ\text{C}$; Q_2 adalah pembuangan panas peralatan dalam ruangan, pencahayaan dan tubuh manusia, W ; Q_4 adalah perpindahan panas dari struktur eksterior ke interior, W ; θ_4 adalah suhu struktur eksternal, $^\circ\text{C}$; R adalah resistansi termal permukaan ruangan internal, $\text{m}^2^\circ\text{C}/\text{W}$ [9].

Berdasarkan persamaan 2.1, dapat disederhanakan menjadi persamaan 2.2 berikut ini:

$$T \frac{d\Delta\theta_0}{dt} + \Delta\theta_0 = K(\theta_c + \theta_f) \quad (2.2)$$

Di mana, K adalah faktor pembesaran untuk *regulated object*, dan T adalah konstanta waktu untuk *regulating object*. θ_c adalah variasi suhu suplai udara yang dikonversi dari

regulating quantity. θ_f adalah variasi suhu suplai udara yang dikonversi dari *interference quantity* [9].

Dari persamaan 2.2 didapatkan persamaan diferensial *interference channel* seperti pada persamaan 2.3 berikut ini:

$$T \frac{d\Delta\theta_0}{dt} + \Delta\theta_0 = K\theta_f \quad (2.3)$$

Sedangkan persamaan diferensial *regulating channel* didapatkan seperti pada persamaan 2.4 berikut ini:

$$T \frac{d\Delta\theta_0}{dt} + \Delta\theta_0 = K\theta_c \quad (2.4)$$

Dengan mempertimbangkan *transport delay*, τ , dari objek sistem, yaitu ruangan aktual, fungsi alih dapat dihitung dengan persamaan 2.5 berikut ini:

$$W(s) = Ke^{-\tau s}Ts + 1 \quad (2.5)$$

Konstanta waktu, T , merepresentasikan karakteristik dinamis objek, dan faktor pembesaran, K , merepresentasikan karakteristik statis objek dan menentukan pengaruh sinyal masukan terhadap nilai kestabilan.

Nilai inersia dari sensor suhu dan aktuator pada sistem dapat diabaikan untuk penyederhanaan sistem, serta variasi gangguan internal dan eksternal dikonversi menjadi variasi suhu suplai udara.

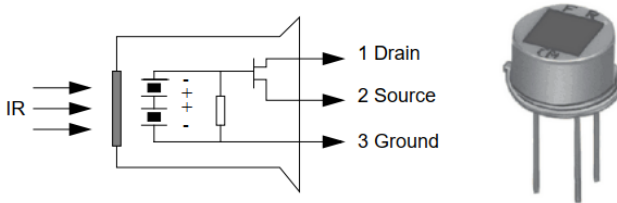
2.3 Mekanisme Sensor *Passive Infrared*

Sensor yang paling banyak digunakan dalam rangka penghematan energi dan biaya adalah sensor berbasis PIR (*Passive Infrared*). Sensor ini mendeteksi gerakan manusia, hewan, bahkan mobil, lalu menyalakan semua perangkat listrik / elektronik yang terhubung dengannya. Komponen kunci dari modul sensor ini adalah elemen *pyro-electric*. Biasanya terdapat minimum 2 elemen yang memungkinkan pembatalan bising dan peningkatan deteksi.

Sensor PIR, seperti namanya, mendeteksi perubahan dalam IR (radiasi inframerah) yang dipicu oleh benda yang bergerak

panas seperti manusia atau gerakan hewan berdarah panas lainnya, gerakan mobil atau bahkan angin sepoi-sepoi yang hangat. Semua gerakan ini mewakili perubahan IR yang bervariasi waktu yang memicu fluktuasi sinyal kecil pada elemen *pyro-electric*. Fluktuasi sinyal kecil ini diperkuat dan sinyal dikondisikan dalam sirkuit pengkondisian sinyal [17].

Salah satu aspek penting dari sensor PIR adalah elemen *pyro-electric*, yang tidak memancarkan energi secara aktif untuk mendeteksi gerakan. Oleh karena itu, selama operasi *standby* yang berkepanjangan ketika ada sedikit atau tidak ada gerakan yang terdeteksi, sensor PIR akan lebih hemat energi. Keunggulan yang lain adalah saat sensor ini digunakan untuk konservasi energi listrik melalui aplikasi seperti sakelar lampu otomatis, karena konsumsi energi mandiri sakelar kontrol PIR sangat minim.



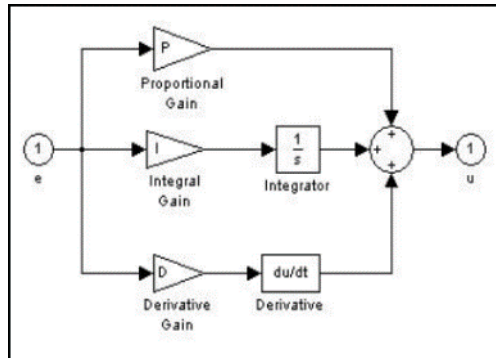
Gambar 2. 1 Rangkaian dan Model Sensor PIR [17]

Sebagaimana pada Gambar 2.1, Sensor PIR ditempatkan dalam kaleng logam tertutup rapat untuk meningkatkan imunitas terhadap kebisingan, suhu, maupun kelembaban. Ada jendela yang terbuat dari bahan *IR-transmissive* (biasanya dilapisi silikon karena sangat mudah didapat) yang melindungi elemen penginderaan. Di belakang jendela ada dua elemen sensor yang dipasang seimbang [17].

2.4 Kontrol PID

Kontrol PID (*Proportional Integral Derivative*) adalah jenis mode pengendali yang sering digunakan untuk keperluan otomatisasi di industri. Hal ini dikarenakan fleksibilitas dari pengendali PID, sehingga banyak digunakan dalam berbagai

situasi.[18] Di dalam sistem kontrol ini, tersusun dari beberapa parameter kontrol, yaitu *proportional* (P), *integral* (I), dan *derivative* (D), sebagaimana pada Gambar 2.2 berikut ini:



Gambar 2. 2 Skema Kontrol PID [19]

Fungsi alih yang didapatkan dari kontrol PID sederhana adalah seperti pada persamaan 2.6 dan 2.7:

$$C(S) = K_p + \frac{K_i}{s} + K_D s \quad (2.6)$$

$$C(S) = K_p s + K_i + K_D s^2 \quad (2.7)$$

Di mana K_p adalah *proportional gain*, K_i adalah *integral gain*, dan K_d adalah *derivative gain*.

Setiap parameter kontrol tersebut memiliki fungsi tersendiri dalam mencapai respons seperti yang diharapkan. K_p digunakan untuk mengurangi *rise time*. K_i digunakan untuk mengurangi *overshoot* dan *settling time*. K_d digunakan untuk mengeliminasi *steady-state error* [19].

Sinyal kontrol, u , dari pengontrol ke instalasi sama dengan *Proportional gain* dikalikan besarnya *error* ditambah *integral gain* dikalikan *integral error* ditambah *derivative gain* dikalikan turunan dari *error*. Hal tersebut seperti pada persamaan 2.8:

$$u = K_p \cdot e + K_i \int e \cdot dt + K_d \frac{de}{dt} \quad (2.8)$$

Pada respons *step* sistem *loop* tertutup, didapatkan beberapa karakteristik sebagai berikut [19]:

- a. *Rise time*, adalah waktu yang diperlukan oleh respons untuk naik melampaui 90% dari tingkat yang diinginkan untuk pertama kalinya.
- b. *Overshoot*, adalah nilai puncak suatu respons, dibandingkan dengan nilai pada kondisi *steady-state*.
- c. *Settling time*, adalah waktu yang diperlukan oleh respons untuk mencapai konvergensi ke kondisi *steady-state*
- d. *Steady-state error*, adalah selisih antara keluaran *steady-state* dan keluaran akhir yang diharapkan.

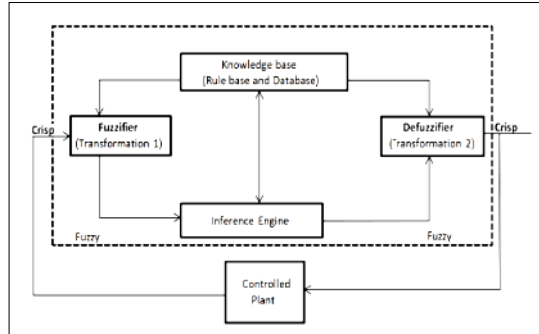
Hal yang perlu diperhatikan dalam penalaan parameter pengendali PID adalah: Gain K_p tidak boleh terlalu besar, karena semakin besar K_p sistem akan semakin sensitif dan cenderung tidak stabil, sedangkan jika K_p kecil akan mengakibatkan *offset* yang besar; K_i yang kecil akan bermanfaat untuk menghilangkan *offset*, tetapi cenderung membawa sistem menjadi lebih sensitif dan mudah berosilasi. Sedangkan K_i besar belum tentu efektif menghilangkan *offset* dan juga membuat respons menjadi lebih lambat; dan K_d yang besar akan membuat unsur *derivative* menjadi lebih menonjol sehingga respons cenderung cepat, sedangkan K_d yang kecil kurang membantu pada saat pencapaian *setpoint* [20].

2.5 Kontrol Logika Fuzzy

Logika *Fuzzy* merupakan sebuah logika yang prinsipnya didasari oleh metode dasar dalam berpikir yang dimiliki oleh manusia. Berbeda dengan logika *boolean* di mana fungsi keanggotaannya hanya memungkinkan nilai 0 dan 1 saja, logika *Fuzzy* memungkinkan adanya daerah fungsi keanggotaan ketiga yang terletak di antara nilai 0 dan 1. Dengan basis aturan “JIKA-MAKA (*IF-THEN*)”, logika *Fuzzy* dapat digunakan untuk memodelkan fungsi nonlinear dari sebuah sistem yang kompleks, yaitu memiliki beberapa masukan dan keluaran [21].

Fuzzy Inference System (FIS) dikembangkan dari logika *Fuzzy* dengan melakukan pemetaan secara nonlinear dari vektor

data masukan ke keluaran dalam bentuk saklar dengan menggunakan aturan-aturan *Fuzzy* yang telah disepakati. Secara umum dasar struktur logika *Fuzzy* dapat digambarkan sebagaimana yang ada pada Gambar 2.3 berikut ini:



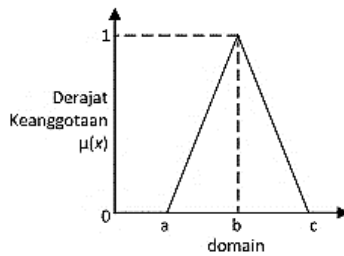
Gambar 2. 3 Skema Kontrol Logika *Fuzzy* [22]

Dari skema pada Gambar 2.3, didapatkan beberapa blok yang memiliki fungsi sebagai berikut ini:

- Fuzzifier*, adalah blok yang berfungsi untuk memetakan data masukan dalam bentuk *crisp* (variabel fisis) atau bilangan *real* menjadi bilangan *Fuzzy* dengan menentukan derajat keanggotaan *Fuzzy* untuk masing-masing nilai masukan.
- Knowledge base*, adalah kumpulan aturan-aturan dasar yang mendefinisikan himpunan *Fuzzy* atas daerah-daerah masukan dan keluaran serta sebagai acuan sistem. Aturan dasar ini menggunakan persamaan *IF – THEN*.
- Inference engine*, adalah proses penalaran bilangan *Fuzzy* yang didasari oleh aturan-aturan pada *knowledge base*, serta karakteristik fungsi keanggotaan yang telah ditentukan sebelumnya.
- Defuzzifier*, adalah blok yang berfungsi untuk memetakan keluaran himpunan *Fuzzy* ke dalam data *crisp*, sehingga mengembalikan setiap keluaran himpunan *Fuzzy* menjadi bilangan *crisp*.

Dalam implementasinya pada suatu sistem kontrol, karakteristik himpunan *Fuzzy* dapat ditentukan melalui fungsi keanggotaannya. Fungsi keanggotaan pada satu himpunan *Fuzzy* dapat direpresentasikan ke dalam beberapa bentuk berikut ini [22]:

- a. Fungsi keanggotaan *triangular*, adalah fungsi keanggotaan yang memiliki 3 parameter dalam implementasinya. Ketiga parameter tersebut adalah $\{a,b,c\}$ yang dideskripsikan sebagaimana pada Gambar 2.4.



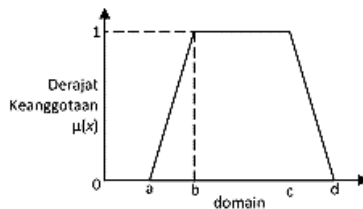
Gambar 2. 4 Fungsi Keanggotaan *Triangular* [22]

Persamaan matematis dari fungsi *triangular* didapatkan melalui persamaan 2.9 dan 2.10.

$$\mu[x] = 0; x \leq a \mid x \geq c \mu[x] = \frac{(x - a)}{(b - a)}; a < x \leq b \quad (2.9a)$$

$$\mu[x] = \frac{(c - x)}{(c - b)}; b < x \leq c \quad (2.9b)$$

$$\mu[x] = \max\left(\min\left(\frac{(x - a)}{(b - a)}, \frac{(c - x)}{(c - b)}\right), 0\right) \quad (2.10)$$



Gambar 2. 5 Fungsi Keanggotaan *Trapezoid* [22]

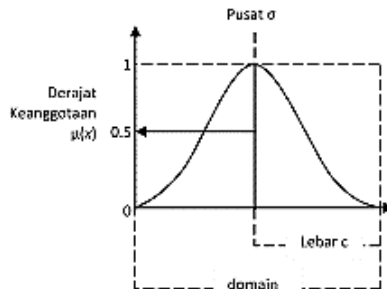
- b. Fungsi keanggotaan *trapezoid*, adalah fungsi keanggotaan yang memiliki 4 parameter dalam pembentukannya. Keempat parameter tersebut adalah $\{a,b,c,d\}$ yang dapat dideskripsikan sebagaimana pada Gambar 2.5.

Persamaan matematis dari fungsi *trapezoid* didapatkan melalui persamaan 2.11 dan 2.12.

$$\begin{aligned}\mu[x] &= 0; x \leq a \text{ atau } x \geq c \\ \mu[x] &= \frac{(x - a)}{(b - a)}; a < x \leq b \\ \mu[x] &= 1; b \leq x \leq c \\ \mu[x] &= \frac{(d - x)}{(d - b)}; c < x \leq d\end{aligned}\quad (2.11)$$

$$\mu[x] = \max\left(\min\left(\frac{(x - a)}{(b - a)}, 1, \frac{(d - x)}{(d - b)}\right), 0\right) \quad (2.12)$$

- c. Fungsi keanggotaan *gaussian*, adalah fungsi keanggotaan yang memiliki 2 parameter $\{c,\sigma\}$ dalam pembentukannya. Fungsi parameter c adalah untuk mempresentasikan pusat fungsi keanggotaan dan σ mempresentasikan lebar fungsi keanggotaan. Fungsi keanggotaan *gaussian* dideskripsikan sebagaimana pada Gambar 2.6 berikut ini:



Gambar 2. 6 Fungsi Keanggotaan *Gaussian* [22]

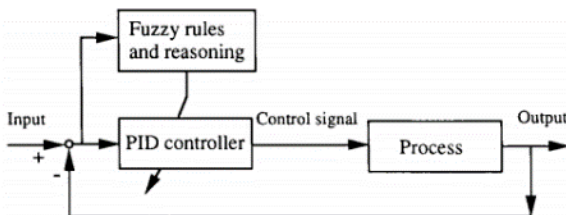
Persamaan matematis dari fungsi *gaussian* didapatkan melalui persamaan 2.13 berikut ini:

$$\mu[x] = e^{-\frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2} \quad (2.13)$$

2.6 Kontrol Logika *Fuzzy*-PID

Suatu proses dalam suatu sistem kontrol dapat berubah-ubah sejalan dengan kondisi operasi proses. Salah satu sumber perubahan bisa diakibatkan oleh nonlinearitas suatu sistem. Keadaan tersebut dapat diminimalkan melalui modifikasi nilai parameter / *gain* dari pengontrol yang didasarkan pada pengukuran kondisi operasi dari proses untuk mengimbangi variasi-variasi perubahan parameter atau nonlinearitas dari proses. Modifikasi nilai *gain* ini lebih dikenal dengan istilah *gain scheduling* [23].

Masalah utama dalam merancang sistem dengan *gain scheduling* adalah untuk mendapatkan variabel *gain scheduling* yang sesuai. Ketika variabel *gain scheduling* didapatkan, maka penalaan parameter pengendali dihitung berdasarkan jumlah kondisi operasi dengan menggunakan metode perancangan yang sesuai. Pengendali kemudian ditala untuk setiap kondisi operasi. *Gain scheduling* mempunyai keuntungan lebih yaitu bahwa parameter pengendali dapat berubah dengan cepat dalam menanggapi perubahan proses [20].

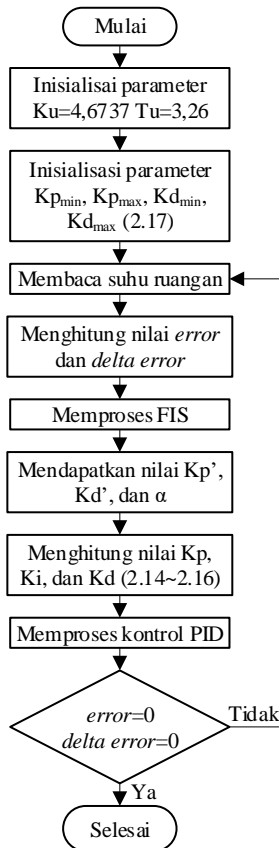


Gambar 2. 7 Skema Kontrol Logika *Fuzzy*-PID [24]

Kontrol Logika *Fuzzy*-PID merupakan salah satu aplikasi dari teori himpunan *Fuzzy*. Logika *Fuzzy* dibuat agar dapat

menjadwal parameter dalam pengontrol PID, hingga *error* dapat dihilangkan sampai pada batas yang diinginkan. Skema Kontrol Logika *Fuzzy*-PID dijelaskan pada Gambar 2.7.

Pada Gambar 2.8 adalah diagram alir yang dapat menjelaskan tahapan dalam proses yang berlangsung pada Kontrol Logika *Fuzzy*-PID.



Gambar 2. 8 Diagram Alir KLF-PID

Masukan sistem kontrol berupa *error*, yaitu selisih nilai pengukuran terhadap nilai *setpoint* dan *delta error*, yaitu laju

perubahan *error*. Sedangkan keluaran sistem kontrol berupa beberapa konstanta yang nantinya akan digunakan untuk menghitung nilai *gain* kontrol PID, yaitu nilai K'_p , K'_d , dan α .

Setelah didapatkan keluaran dari Kontrol Logika *Fuzzy*, maka *gain* pengontrol PID dapat dihitung melalui persamaan 2.14, 2.15, dan 2.16 berikut ini:

$$K_p = (K_{p,max} - K_{p,min})K'_p + K_{p,min} \quad (2.14)$$

$$K_d = (K_{d,max} - K_{d,min})K'_d + K_{d,min} \quad (2.15)$$

$$K_i = K_p^2 + (\alpha K_d) \quad (2.16)$$

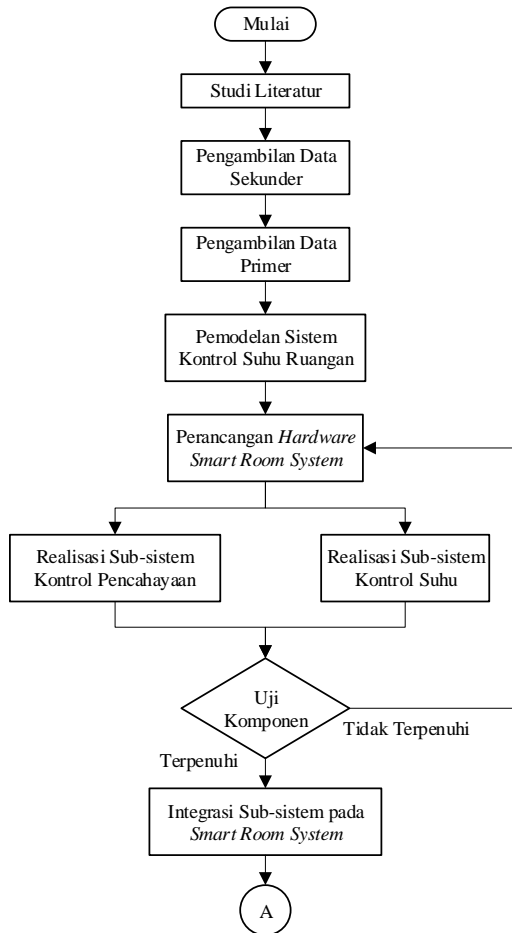
Dari hasil studi simulasi secara ekstensif pada proses yang beraneka ragam, aturan untuk menentukan *range* dari K_p dan K_d didapatkan melalui persamaan 2.17 berikut ini:

$$\begin{aligned} K_{p,min} &= 0,32K_u, & K_{p,max} &= 0,6K_u \\ K_{d,min} &= 0,08K_u T_u, & K_{d,max} &= 0,15K_u T_u \end{aligned} \quad (2.17)$$

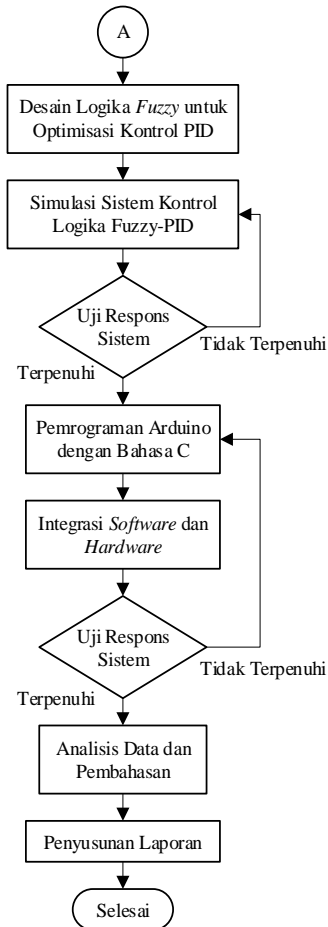
Di mana nilai K_u dan T_u , secara berurutan, merupakan *gain* dan periode osilasi respons pada batas stabilitas *P-Control* [25].

BAB III METODOLOGI PENELITIAN

Pada penelitian tugas akhir ini, dilakukan beberapa tahapan yang digambarkan dalam suatu diagram alir. Berikut pada Gambar 3.1 dan 3.2 merupakan diagram alir pengerjaan tugas akhir:



Gambar 3. 1 Diagram Alir Metode Penelitian



Gambar 3. 2 Diagram Alir Metode Penelitian (lanjutan)

Berikut merupakan penjelasan dari diagram alir penelitian:

3.1 Studi Literatur

Studi literatur dibutuhkan untuk menunjang pengetahuan penulis terhadap topik tugas akhir yang dikerjakan baik dari *handbook*, jurnal dan laman internet. Beberapa pengetahuan

mengenai model sistem kontrol suhu ruangan ber-AC, mekanisme sensor *passive infrared*, kontrol PID, Kontrol Logika *Fuzzy*, serta Kontrol Logika *Fuzzy*-PID.

3.2 Pengambilan Data Sekunder

Untuk mempermudah perancangan dari *smart room system* maka dibutuhkan data sekunder. Adapun data sekunder yang dibutuhkan adalah parameter termal ruangan, spesifikasi mikrokontroler, aktuator sistem berupa AC dan lampu LED, serta data spesifikasi sensor sistem, yaitu berupa sensor DHT22 dan sensor PIR.

3.2.1 Parameter Termal Ruangan

Parameter termal ruangan dibutuhkan untuk membuat model ruangan ber-AC yang akan digunakan pada simulasi sistem kontrol menggunakan *software* MATLAB. Parameter termal ruangan tertera pada Tabel 3.1 berikut ini:

Tabel 3. 1 Parameter Termal Ruangan [26]

Jenis Objek	Nilai Disipasi
6 orang (duduk, mengetik)	648W
1 TLD	36 W
2 laptop	106 W
6 telepon seluler	215 W
Jenis Material	Nilai Resistansi
Papan asbes	0,037 m ² C/W
Ubin keramik	0,44 m ² C/W
Papan gipsum	0,08 m ² C/W

3.2.2 Aktuator

Aktuator yang digunakan dalam *smart room system* berupa *Air Conditioner* dan lampu bohlam LED. *Air Conditioner* digunakan untuk mengontrol suhu ruangan, sedangkan lampu bohlam LED digunakan untuk mengontrol pencahayaan ruangan. Data spesifikasi aktuator tertera pada Tabel 3.2 dan 3.3 berikut ini:

Tabel 3. 2 Spesifikasi AC (*Mitsubishi SRK10CEV-4*)

Karakteristik	Nilai
Tegangan Suplai	220~240 V
<i>Cooling Capacity</i>	2,638 kW
<i>Cooling Input</i>	0,803 kW
Arus Kerja	4,2 A
COP	3,29
Dimensi	250 x 815 x 249 mm
<i>Air Flow Supply</i>	8,5 CCM
Berat	10,5 kg

Tabel 3. 3 Spesifikasi Lampu LED (*Philips LED Bulb*)

Karakteristik	Nilai
Tegangan Suplai	220~240 V
Daya Kerja	8 W
Intensitas Cahaya	770 lumen
Suhu Warna	6500 K
Sudut Cahaya	200°

3.2.3 Sensor

Sensor yang digunakan dalam *smart room system* berupa termistor DHT22 dan sensor PIR. Termistor DHT22 digunakan untuk mengukur suhu ruangan, sedangkan sensor PIR digunakan untuk mengindra gerakan manusia. Data spesifikasi sensor tertera pada Tabel 3.4 dan 3.5 berikut ini:

Tabel 3. 4 Spesifikasi Sensor DHT22

Karakteristik	Nilai
Tegangan Suplai	3,3~6 VDC
<i>Sensing Element</i>	DS18B20
<i>Range</i>	-40~125°C
Akurasi	0,2°C
Resolusi	0,1°C
<i>Sensing Period</i>	2 detik

Tabel 3. 5 Spesifikasi Sensor PIR (HC-SR501)

Karakteristik	Nilai
Tegangan Suplai	4,5~20 VDC
<i>Delay Time</i>	5~300 detik
<i>Sensing Range</i>	$\leq 110^\circ$; 3~7 m
Suhu Operasi	-15~70°C
Konsumsi Daya	65 mA

3.2.4 Mikrokontroler

Mikrokontroler yang digunakan dalam *smart room system* adalah *Arduino Mega 2560* dengan spesifikasi yang tertera pada Tabel 3.6 berikut ini:

Tabel 3. 6 Spesifikasi Mikrokontroler (Arduino Mega 2560)

Karakteristik	Nilai
<i>Microchip</i>	ATmega 2560
Tegangan Operasi	5 V
Tegangan Input	6~20 V
Pin I/O Digital	54 buah
Pin I/O Analog	16 buah
Arus DC pin I/O	20 mA
Arus DC pin 3.3V	50 mA
<i>Flash Memory</i>	256 KB
SRAM/EEPROM	8 KB/4 KB
<i>Clock Speed</i>	16 MHz

3.3 Pengambilan Data Primer

Selain dibutuhkankannya data sekunder berupa parameter termal ruangan, spesifikasi mikrokontroler, aktuator dan sensor, diperlukan juga pengambilan data primer. Data primer yang diambil berupa dimensi ruangan, suhu internal dan eksternal ruangan untuk membuat model suhu ruangan ber-AC, serta tingkat pencahayaan alami untuk menentukan jenis lampu yang akan digunakan.

Pengambilan data primer dilakukan di Ruang P-212, Departemen Teknik Fisika, Fakultas Teknologi Industri, ITS. Pengambilan data tingkat pencahayaan alami dilakukan pada tanggal 3 Mei 2019 pukul 11.00 WIB. Sedangkan pengambilan data dimensi ruangan, suhu internal dan eksternal ruangan dilakukan pada tanggal 11 September 2019 pukul 11.00 WIB. Data lampiran dapat dilihat pada Lampiran 1.

Untuk mengetahui protokol data yang digunakan dalam komunikasi data *wireless-IR*, maka dilakukan pengambilan *RAW data* sinyal inframerah dari *remote control* AC. Pengambilan data dilakukan dengan menggunakan *IR receiver* berupa TSOP34438. Hasil pengambilan data tersusun di dalam *source code* Arduino yang dapat dilihat pada Lampiran 2.

3.4 Pemodelan Sistem Kontrol Suhu Ruangan

Sebelum dilakukan simulasi, terlebih dahulu ditentukan model suhu ruangan ber-AC dan *fan coil unit*. Model suhu ruangan ber-AC didapatkan melalui persamaan 2.1. Dikarenakan nilai Q_c yang selalu berubah menyesuaikan suhu AC yang diatur, serta nilai Q_2 yang selalu berubah menyesuaikan aktivitas manusia dan peralatan elektronik yang ada pada ruangan, maka persamaan 2.1 dapat diubah menjadi persamaan 3.1 berikut:

$$C_1 \frac{d\Delta\theta_0}{dt} = G_s c_1 \Delta\theta'_0 (Q_4 - 2) + \frac{\theta_4 \Delta\theta_0}{R} \quad (3.1)$$

Kemudian didapatkan persamaan fungsi alih model suhu ruangan ber-AC melalui persamaan 3.2~3.8 berikut ini:

$$C_1 \frac{d\theta\Delta_0}{dt} - \frac{\theta_4 \Delta\theta_0}{R} = G_s c_1 \Delta\theta'_0 (Q_4 - 2) \quad (3.2)$$

$$-\frac{R}{\theta_4} \left(C_1 \frac{d\Delta\theta_0}{dt} \right) + \Delta\theta_0 = -\frac{R}{\theta_4} (G_s c_1 \Delta\theta'_0 (Q_4 - 2)) \quad (3.3)$$

$$\begin{aligned} & -\frac{0,557}{28,16} \left(136.735 \frac{d\Delta\theta_0}{dt} \right) + \Delta\theta_0 = \\ & -\frac{0,557}{28,16} (0,174.1,005. \Delta\theta'_0 (192,637 - 2)) \end{aligned} \quad (3.4)$$

$$-2,705 \left(\frac{d\Delta\theta_0}{dt} \right) + \Delta\theta_0 = (-0,658)\Delta\theta'_0 \quad (3.5)$$

$$(2,705)s \cdot \Delta\theta_0(s) + \Delta\theta_0(s) = (0,658)\Delta\theta'_0(s) \quad (3.6)$$

$$(2,705s + 1)\Delta\theta_0(s) = (0,658)\Delta\theta'_0(s) \quad (3.7)$$

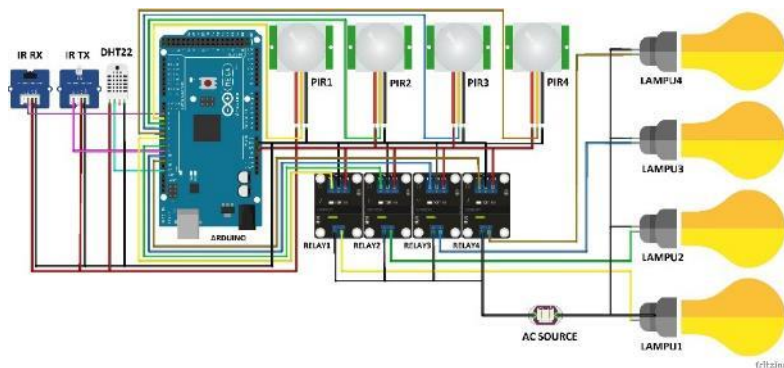
$$\frac{\Delta\theta_0(s)}{\Delta\theta'_0(s)} = \frac{0,658}{2,705s + 1} \quad (3.8)$$

Sedangkan persamaan fungsi alih dari *fan coil unit* didapatkan dari melalui persamaan 2.4 dan 2.5 sehingga menjadi persamaan 3.9 berikut ini [9]:

$$\frac{\Delta\theta_c(s)}{W(s)} = \frac{26}{2,5s + 1} \quad (3.9)$$

3.5 Perancangan *Hardware*

Setelah didapatkan data primer maupun sekunder, dilakukan perancangan *hardware* menggunakan *software* Fritzing, dengan hasil perancangan seperti pada Gambar 3.3 berikut ini:



Gambar 3. 3 Rancangan *Hardware* dengan *Software* Fritzing

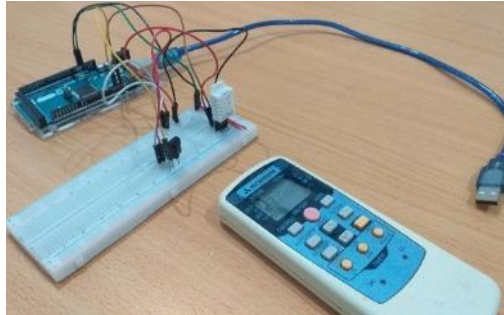
Dalam perancangan *hardware* digunakan beberapa komponen utama sebagai berikut: 1 buah IR *receiver*, 1 buah IR *emitter*, 1 buah sensor termistor DHT22, 4 buah sensor PIR, 1 buah *relay* SSR 4 kanal, 4 buah lampu bohlam LED, dan 1 buah mikrokontroler Arduino Mega2560 Rev3.

Sambungan *pin* antara komponen dan Arduino tertera pada Tabel 3.7 berikut ini:

Tabel 3. 7 Sambungan *Pin* Komponen dan Arduino

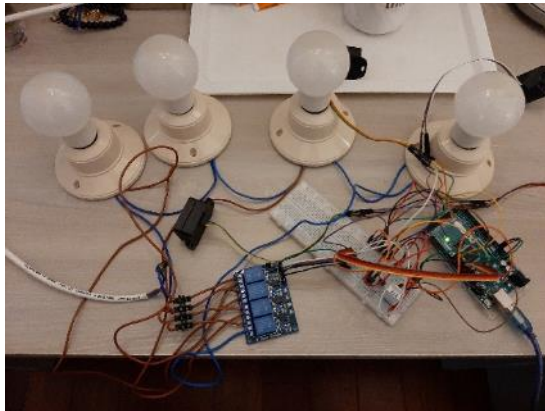
<i>Pin</i> Komponen	<i>Pin</i> Arduino
DHT22 – Vcc	5V
DHT22 – Data	13
DHT22 – Gnd	Gnd
IR Receiver – Vcc	5V
IR Receiver – Data	2
IR Receiver – Gnd	Gnd
IR LED – Anoda	9
IR LED – Katoda	Gnd
Relay – Vcc	5V
Relay – 1	6
Relay – 2	7
Relay – 3	8
Relay – 4	10
Relay – Gnd	Gnd
PIR – Vcc	5V
PIR – 1	5
PIR – 2	4
PIR – 3	3
PIR – 4	11
PIR – Gnd	Gnd

Pada perancangan *hardware*, terdapat 2 sub-sistem, yaitu sub-sistem kontrol suhu dan sub-sistem kontrol pencahayaan. Pada sub-sistem kontrol suhu terdiri dilakukan terlebih dahulu proses *decoding* protokol komunikasi inframerah dari *remote control* AC, menggunakan IR *Receiver*. Proses *decoding* dilakukan dengan menggunakan metode *trial and error* sehingga mendapatkan *raw data* yang sesuai dengan informasi yang tertera pada *remote control* AC. Setelah itu, dirangkai sub-sistem kontrol suhu seperti pada Gambar 3.4 berikut ini:



Gambar 3. 4 Sub-sistem Kontrol Suhu

Setelah dirancang sub-sistem kontrol suhu, selanjutnya ditambahkan rancangan sub-sistem kontrol pencahayaan. Kontrol pencahayaan pada sistem ini menggunakan mekanisme deteksi gerakan manusia dari sensor PIR dan menggunakan *relay* untuk mengatur ON-OFF lampu yang digunakan, sebagaimana pada Gambar 3.5 berikut ini:



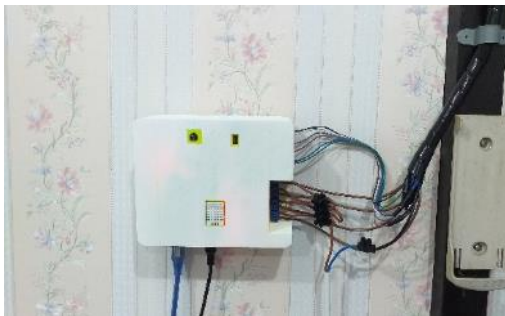
Gambar 3. 5 Sub-sistem Kontrol Pencahayaan

Proses integrasi sistem dilakukan melalui pemrograman Arduino. Bila salah satu sensor PIR mendeteksi gerakan manusia maka AC akan menyala, sedangkan pada saat tidak ada sensor PIR yang mendeteksi gerakan manusia maka AC akan mati, sebagaimana pada Gambar 3.6 berikut ini:



Gambar 3. 6 Integrasi Sistem

Setelah sistem berjalan dengan baik, maka dilakukan realisasi *hardware* dengan mengaplikasikan komponen pada papan PCB dan pemberian kotak panel sebagai komponen pelindung sistem. Berikut pada Gambar 3.7 adalah realisasi *hardware*:



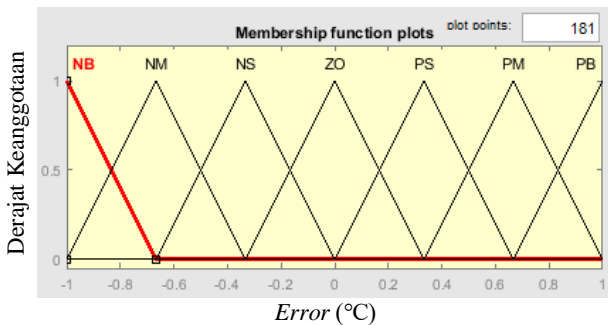
Gambar 3. 7 Realisasi *Hardware*

3.6 Pengujian Komponen

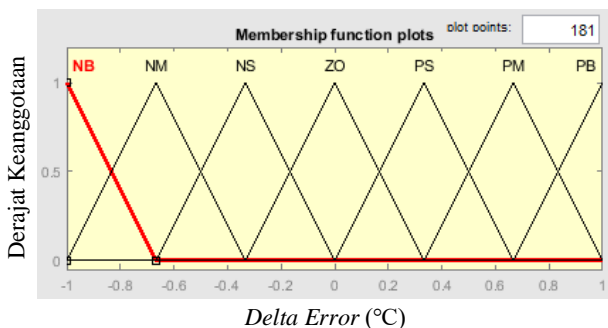
Pengujian komponen dilakukan dengan menggunakan sensor DHT22 untuk pembacaan suhu dan sensor PIR untuk mendeteksi gerakan. Pengujian ini bertujuan untuk memastikan komponen berfungsi dengan baik dan mengetahui karakteristik performansi serta pembacaan *hardware*. Hasil pengujian beberapa komponen tersebut akan dijabarkan pada bab selanjutnya.

3.7 Perancangan Sistem Kontrol Logika *Fuzzy*-PID

Sebelum mendesain sistem kontrol utama, dirancang terlebih dahulu FIS yang akan digunakan pada sistem kontrol utama. Nilai masukan FIS ada dua yaitu *error* dan *delta error* yang akan menghasilkan tiga keluaran yaitu nilai K'_p , K'_d , dan α . yang nantinya akan dihitung untuk didapatkan parameter PID, yaitu nilai K_p , K_i , dan K_d . Parameter tersebut diproses oleh pengontrol PID untuk menentukan selisih suhu yang akan ditambahkan pada ruangan, serta memilih nilai suhu yang akan dikirimkan ke AC.



Gambar 3. 8 Fungsi Keanggotaan *Error*



Gambar 3. 9 Fungsi Keanggotaan *Delta Error*

Kemudian dilakukan penentuan fungsi dan jumlah keanggotaan masukan *Fuzzy*. Jumlah keanggotaan *Fuzzy* yang digunakan yaitu 7 yang terdiri dari bentuk fungsi keanggotaan *triangular*. Hal ini bertujuan untuk mendapatkan respons sistem

dengan nilai *overshoot* dan *error* keseluruhan sistem yang rendah. Nilai *range* yang digunakan adalah -1~1. Hal ini bertujuan untuk mendapatkan waktu respons sistem yang cepat. Bentuk interval fungsi keanggotaan dari masukan sistem tertera pada Gambar 3.8 dan 3.9.

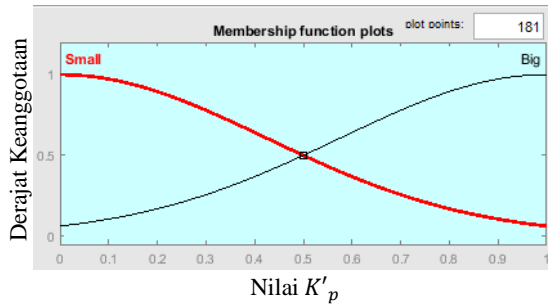
Nilai parameter variabel masukan *error*:

NB	: [-1 -1 -0.6667]
NM	: [-1 -0.6667 -0.3333]
NS	: [-0.6667 -0.3333 0]
ZO	: [-0.3333 0 0.3333]
PS	: [0 0.3333 0.6667]
PM	: [0.3333 0.6667 1]
PB	: [0.6667 1 1]

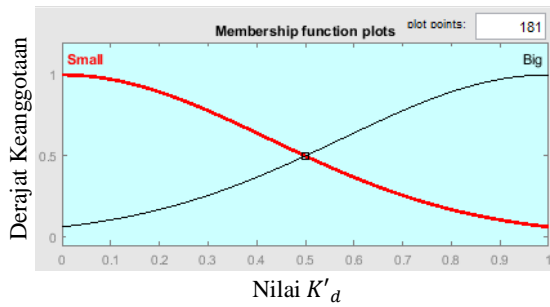
Nilai parameter variabel masukan *delta error*:

NB	: [-1 -1 -0.6667]
NM	: [-1 -0.6667 -0.3333]
NS	: [-0.6667 -0.3333 0]
ZO	: [-0.3333 0 0.3333]
PS	: [0 0.3333 0.6667]
PM	: [0.3333 0.6667 1]
PB	: [0.6667 1 1]

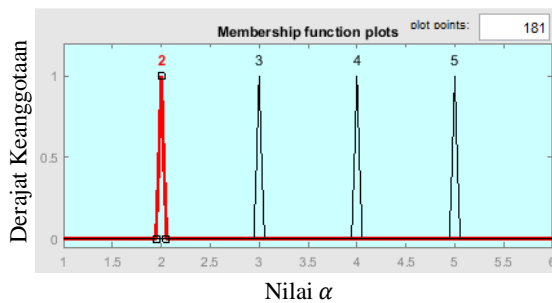
Kemudian dilakukan penentuan fungsi dan jumlah keanggotaan keluaran *Fuzzy*. Jumlah keanggotaan *Fuzzy* yang digunakan pada keluaran K'_p dan K'_d , yaitu 2 yang terdiri dari bentuk fungsi keanggotaan *gaussian* dengan nilai *range* adalah 0~1. Sedangkan, jumlah keanggotaan *Fuzzy* yang digunakan pada keluaran α , yaitu 4 yang terdiri dari bentuk fungsi keanggotaan *triangular*. Nilai *range* yang digunakan adalah 1~6. Fungsi keanggotaan dan *range* yang tersebut dirancang untuk memenuhi kebutuhan kontrol PID. Pada nilai *error* yang tinggi maka dibutuhkan sinyal kontrol yang besar. K_p harus bernilai besar, sedangkan K_i dan K_d harus bernilai kecil, begitu pula sebaliknya. Bentuk interval fungsi keanggotaan dari keluaran sistem tertera pada Gambar 3.10, 3.11 dan 3.12 sebagai berikut ini:



Gambar 3. 10 Fungsi Keanggotaan K'_p



Gambar 3. 11 Fungsi Keanggotaan K'_d



Gambar 3. 12 Fungsi Keanggotaan α

Nilai parameter variabel keluaran K'_p dan K'_d :

Small : [0.4247 0]

Big : [0.4247 1]

Nilai parameter variabel keluaran α :

- 2 : [1.95 2 2.05]
 3 : [2.95 3 3.05]
 4 : [3.95 4 4.05]
 5 : [4.95 5 5.05]

Pada sistem ini menggunakan *rule base* yang berjumlah 49 karena memiliki dua masukan yaitu *error* dan *delta error* di mana masing-masing masukan berisi 7 fungsi keanggotaan. Dalam menentukan *rule base*, penentuan derajat keanggotaan keluaran terhadap fungsi keanggotaan masukan disesuaikan dengan karakteristik kontrol PID. Sehingga untuk mendapatkan masing-masing nilai keluaran dapat ditentukan menggunakan Tabel 3.8, 3.9, dan 3.10.

Tabel 3. 8 *Rule Base* Keluaran K'_p

		<i>delta error</i>						
		NB	NM	NS	ZO	PS	PM	PB
<i>error</i>	NB	B	B	B	B	B	B	B
	NM	S	B	B	B	B	B	S
	NS	S	S	B	B	B	S	S
	ZO	S	S	S	B	S	S	S
	PS	S	S	B	B	B	S	S
	PM	S	B	B	B	B	B	S
	PB	B	B	B	B	B	B	B

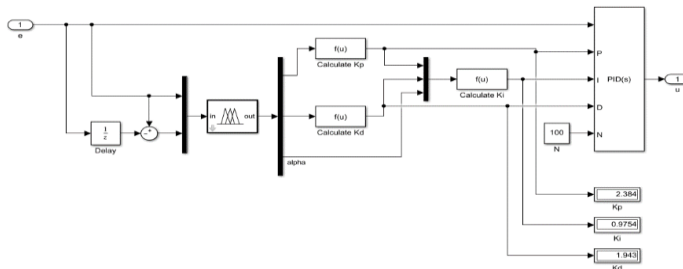
Tabel 3. 9 *Rule Base* Keluaran K'_d

		<i>delta error</i>						
		NB	NM	NS	ZO	PS	PM	PB
<i>error</i>	NB	S	S	S	S	S	S	S
	NM	B	B	S	S	S	B	B
	NS	B	B	B	S	B	B	B
	ZO	B	B	B	B	B	B	B
	PS	B	B	B	S	B	B	B
	PM	B	B	S	S	S	B	B
	PB	S	S	S	S	S	S	S

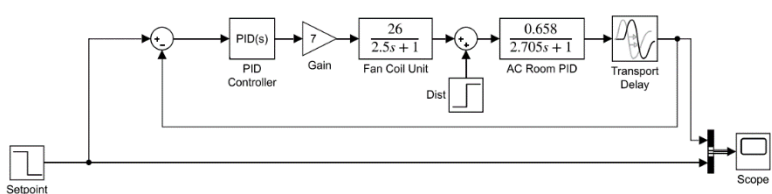
Tabel 3. 10 Rule Base Keluaran α

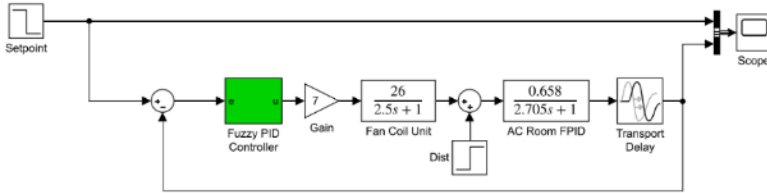
		<i>delta error</i>						
		NB	NM	NS	ZO	PS	PM	PB
<i>error</i>	NB	2	2	2	2	2	2	2
	NM	3	3	2	2	2	3	3
	NS	4	3	3	2	3	3	4
	ZO	5	4	3	3	3	4	5
	PS	4	3	3	2	3	3	4
	PM	3	3	2	2	2	3	3
	PB	2	2	2	2	2	2	2

Skema Kontrol Logika *Fuzzy*-PID dirancang pada Simulink Matlab dengan mengimpor *file* FIS pada *workspace* Matlab terlebih dahulu. Skema Kontrol Logika *Fuzzy*-PID pada Simulink Matlab dapat dilihat pada Gambar 3.13 berikut ini:

**Gambar 3. 13** Blok Sistem Kontrol Logika *Fuzzy*-PID

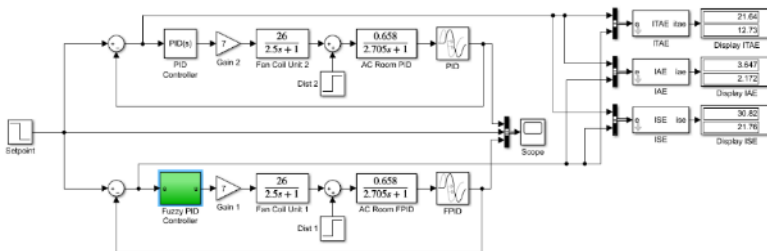
Kemudian dilakukan perancangan sistem dengan *software* Matlab menggunakan fitur Simulink. Berdasarkan persamaan 3.8 dan 3.9 maka dapat dirancang blok sistem kontrol PID dan Kontrol Logika *Fuzzy*-PID seperti pada Gambar 3.14 dan 3.15.

**Gambar 3. 14** Blok Sistem Kontrol PID



Gambar 3. 15 Blok Sistem Kontrol Logika *Fuzzy*-PID

Untuk mempermudah analisis perbandingan karakteristik dan performansi dari sistem kontrol PID dan sistem Kontrol Logika *Fuzzy*-PID, maka dirancang skema perbandingan sistem kontrol PID dan *Fuzzy*-PID seperti pada Gambar 3.16 berikut ini:



Gambar 3. 16 Blok Sistem Perbandingan Sistem Kontrol

3.8 Pemrograman Arduino

Setelah rancangan sistem kontrol disimulasikan, maka dilakukan pemrograman melalui *software* Arduino IDE dengan menggunakan bahasa C. Perincian *source code* dapat dilihat pada Lampiran 2. Pada pemrograman Arduino, digunakan *library* DHT untuk melakukan pemrograman sensor DHT22 dan *library* IR Remote untuk melakukan pemrograman komunikasi data menggunakan *wireless-IR*. Selain itu, juga dilakukan konversi FIS terlebih dahulu dari *file format* Matlab (.fis) menjadi *file format* Arduino (.ino) untuk mempermudah proses pemrograman. Proses konversi dilakukan menggunakan *toolbox* Arduino FIST dari MakeProto. Proses *data logging* ketika melakukan pengujian *smart room system* dilakukan dengan menggunakan fitur ekstensi dari

Microsoft Excel yaitu PLX DAQ v.2.11 yang telah terintegrasi dengan *serial* COM Arduino.

Tahapan pada pemrograman Aduino dilakukan sebagaimana tertera pada Gambar 3.17~3.23 berikut ini:

```
//Menambahkan Lib & Header ke Program
#include <IRremote.h> //Lib untuk Wireless-IR
#include <DHT.h> // Lib untuk Sensor DHT22
#include "fis_header.h" // Header untuk FIS

//Inisialisasi Sensor DHT22
#define DHTPIN 13
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);

// Menentukan Pin Input: Sensor PIR
#define PIR_1 5
#define PIR_2 4
#define PIR_3 3
#define PIR_4 11

// Menentukan Pin Output: Relay
#define RELAY_1 6
#define RELAY_2 7
#define RELAY_3 8
#define RELAY_4 10
```

Gambar 3. 17 Penambahan *Library* Sistem dan Penentuan *Pin* DHT22, Sensor PIR, *Relay*

Pada Gambar 3.17 tertera *source code* yang digunakan untuk menambahkan *library* untuk *IR Remote*, sensor DHT22 dan *header* FIS. Pada tahapan ini juga ditentukan *pin* yang digunakan pada Arduino Mega untuk sensor DHT22, sensor PIR, dan *relay*.

```
// Menentukan RAW Data setiap Fungsi AC
unsigned int AC18[] = {3208, 1576, 408, 388, 404, 1180, 408,
unsigned int AC19[] = {3208, 1576, 404, 388, 404, 1184, 404,
unsigned int AC20[] = {3208, 1576, 404, 392, 400, 1184, 404,
unsigned int AC21[] = {3208, 1576, 404, 388, 404, 1180, 408,
unsigned int AC22[] = {3208, 1576, 408, 388, 404, 1180, 404,
unsigned int AC23[] = {3208, 1576, 408, 388, 404, 1180, 404,
unsigned int AC24[] = {3208, 1572, 408, 388, 404, 1180, 408,
unsigned int AC25[] = {3208, 1576, 408, 388, 404, 1180, 404,
unsigned int AC26[] = {3208, 1576, 408, 388, 404, 1180, 404,
unsigned int AC27[] = {3208, 1576, 408, 388, 404, 1180, 404,
unsigned int AC28[] = {3208, 1576, 404, 388, 404, 1184, 404,
unsigned int AC29[] = {3208, 1576, 404, 392, 404, 1180, 404,
unsigned int AC30[] = {3208, 1576, 408, 388, 404, 1180, 408,

//Menentukan Variabel untuk Kontrol Suhu
IRsend irsend;
int khz = 38; // 38kHz frekuensi karier untuk protokol NEC
int AC_Temp, suhuACint;
```

Gambar 3. 18 Penentuan *RAW Data* AC dan Variabel untuk Kontrol Suhu dengan *IR*

Pada Gambar 3.18 tertera *source code* yang digunakan untuk menentukan *RAW data* setiap suhu AC, dari 18°C sampai 30°C, dan variabel yang dibutuhkan untuk komunikasi antara Arduino Mega dengan AC menggunakan *IR Remote*.

```
//Menentukan Variabel untuk Kontrol PID
double Kp, Ki, Kd, Kpp, Kdp, alpha;
double Ku=4.6737, Tu=3.26;
double Kpmin=0.32*Ku, Kpmax=0.6*Ku, Kdmin=0.08*Ku*Tu, Kdmax=0.15*Ku*Tu;
unsigned long currentTime, previousTime;
double elapsedTime;
double error, delta_error;
double lastError;
double input, output, setpoint;
double cumError, rateError;

// Jumlah Input FIS
const int fis_gcI = 2;
// Jumlah Output FIS
const int fis_gcO = 3;
// Jumlah Rules FIS
const int fis_gcR = 49;

FIS_TYPE g_fisInput[fis_gcI];
FIS_TYPE g_fisOutput[fis_gcO];
```

Gambar 3. 19 Penentuan Variabel untuk Kontrol PID dan FIS

Pada Gambar 3.19 tertera *source code* yang digunakan untuk menentukan variabel yang dibutuhkan untuk menjalankan fungsi kontrol PID pada program utama dan inisialisasi variabel yang dibutuhkan dalam menjalankan FIS.

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

  // inisialisasi input&output sub-sistem pencahayaan
  Init_cahaya();

  // memulai pengambilan data DHT22
  dht.begin();

  // menentukan setpoint suhu ruangan
  setpoint = 25;

  // inisialisasi PLX-DAQ
  Init_PlxDaq();
}
```

Gambar 3. 20 *Setup* Program Utama

Pada Gambar 3.20 tertera *source code* yang digunakan untuk melakukan *setup* pada program utama yang meliputi inisialisasi

sensor PIR, *relay*, DHT22, memulai *serial monitor* dengan *baud rate* 9600, pengaturan *setpoint* suhu, dan inisialisasi *data logging*.

```
//Lampu 1
void lampuON1()
{
  digitalWrite(RELAY_1,HIGH);
}
void lampuOFF1()
{
  digitalWrite(RELAY_1,LOW);
}
boolean deteksil()
{
  int sensorValue1 = digitalRead(PIR_1);
  if(sensorValue1 == LOW)//if the sensor value is HIGH?
  {
    return true;//yes,return ture
  }
  else
  {
    return false;//no,return false
  }
}
```

Gambar 3. 21 Fungsi Kontrol Pencahayaan

Pada Gambar 3.21 tertera *source code* yang digunakan untuk menyalakan lampu, mematikan lampu, serta logika *boolean* untuk menentukan kondisi sensor PIR dalam mendeteksi objek.

```
// fungsi untuk kontrol PID
double computePID(double input){
  currentTime = millis();
  elapsedTime = (double)(currentTime - previousTime);

  error = setpoint - input;
  cumError += error * elapsedTime;
  rateError = (error - lastError)/elapsedTime;

  double output = Kp*error + Ki*cumError + Kd*rateError;

  lastError = error;
  previousTime = currentTime;

  return output;
}
```

Gambar 3. 22 Fungsi Kontrol PID

Pada Gambar 3.22 tertera *source code* yang digunakan untuk menentukan struktur fungsi kontrol PID. Pada tahapan ini dihitung

keluaran kontrol PID berdasarkan nilai Kp, Ki, dan Kd yang didapatkan dari keluaran FIS.

```
void fis_evaluate()
{
    FIS_TYPE fuzzyInput0[] = { 0, 0, 0, 0, 0, 0, 0 };
    FIS_TYPE fuzzyInput1[] = { 0, 0, 0, 0, 0, 0, 0 };
    FIS_TYPE* fuzzyInput[fis_gcI] = { fuzzyInput0, fuzzyInput1, };
    FIS_TYPE fuzzyOutput0[] = { 0, 0 };
    FIS_TYPE fuzzyOutput1[] = { 0, 0 };
    FIS_TYPE fuzzyOutput2[] = { 0, 0, 0, 0 };
    FIS_TYPE* fuzzyOutput[fis_gcO] = { fuzzyOutput0, fuzzyOutput1,
    FIS_TYPE fuzzyRules[fis_gcR] = { 0 };
    FIS_TYPE fuzzyFires[fis_gcR] = { 0 };
    FIS_TYPE* fuzzyRuleSet[] = { fuzzyRules, fuzzyFires };
    FIS_TYPE sW = 0;
```

Gambar 3. 23 Fungsi FIS

Pada Gambar 3.23 tertera cuplikan *source code* yang digunakan untuk menentukan fungsi FIS. Pada tahapan ini terjadi proses fuzifikasi, penalaran *Fuzzy* berdasarkan *rule base*, serta defuzifikasi. Keluaran dari fungsi FIS adalah Kp, Ki, dan Kd yang akan diproses oleh fungsi kontrol PID.

BAB IV

ANALISIS DAN PEMBAHASAN

Analisis dan pembahasan berisi tentang hasil pengujian sensor DHT22, serta perbandingan respons sistem kontrol hasil simulasi dan pengujian menggunakan *hardware*.

4.1 Hasil Pengujian Sensor DHT22

Pengujian sensor DHT22 dilakukan dengan tujuan mengetahui nilai dari pembacaan sensor yang digunakan. Pengujian menggunakan sensor DHT22 dan termokopel tipe-K yang telah terkalibrasi sebelumnya. Berikut pada Gambar 4.1 adalah proses pengujian sensor DHT22.

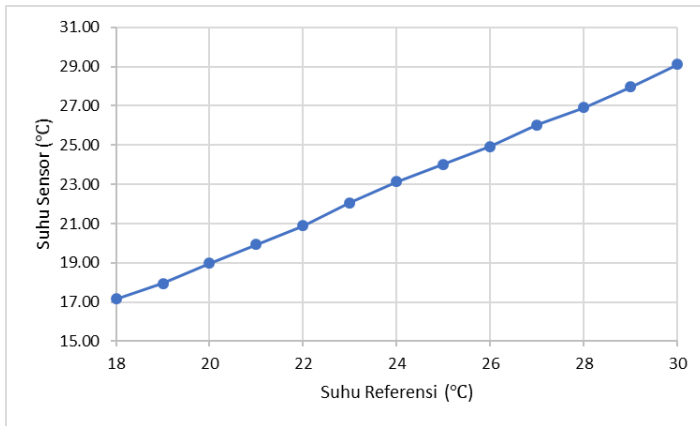


Gambar 4. 1 Pengujian Sensor DHT22

Hasil pembacaan sensor dibandingkan dengan termokopel tipe K sebagai pembacaan referensi untuk diketahui nilai ketidakpastian pembacaan suhu dari sensor DHT22 dari pengukuran berulang (tipe A). Adapun pengertian ketidakpastian pengukuran adalah sebuah rentang nilai di mana nilai sebenarnya dari sebuah pengukuran tersebut diperkirakan berada. Hasil pembacaan tertera pada Gambar 4.2.

Berdasarkan hasil pengambilan data, sebagaimana yang terdapat pada lampiran, nilai ketidakpastian terbesar berada pada pembacaan suhu 18°C dengan nilai sebesar 0,04°C. Pada titik

tersebut diperoleh rerata pembacaan suhu $17,16^{\circ}\text{C}$ dengan standar deviasi sebesar $0,089^{\circ}\text{C}$. Sehingga nilai pembacaan sesungguhnya sensor DHT22 pada suhu 18°C adalah $18 \pm 0,089^{\circ}\text{C}$. Sedangkan rerata koreksi pembacaan suhu sensor DHT22 adalah $0,99^{\circ}\text{C}$.



Gambar 4. 2 Hasil Pengujian Sensor DHT22

4.2 Hasil Pengujian Sensor PIR

Pengujian sensor PIR dilakukan dengan tujuan mengetahui batas pembacaan sensor dalam mendeteksi gerakan manusia. Variasi jarak dan sudut diterapkan untuk mengetahui batas maksimum jarak dan sudut antara sensor dan objek. Hasil pengujian terdapat pada Tabel 4.1 berikut ini:

Tabel 4. 1 Hasil Pengujian Sensor PIR

Sudut ($^{\circ}$)	Jarak (m)	PIR			
		1	2	3	4
0	1	V	V	V	V
	2	V	V	V	V
	3	V	V	V	V
	4	V	V	V	V

Tabel 4. 2 Hasil Pengujian Sensor PIR (lanjutan)

Sudut (^o)	Jarak (m)	PIR			
		1	2	3	4
10	1	V	V	V	V
	2	V	V	V	V
	3	V	V	V	V
	4	V	V	V	V
20	1	V	V	V	V
	2	V	V	V	V
	3	V	V	V	V
	4	V	V	V	V
30	1	V	V	V	V
	2	V	V	V	V
	3	V	V	V	V
	4	V	V	V	V
40	1	V	V	V	V
	2	V	V	V	V
	3	V	V	V	V
	4	X	X	X	X
*Keterangan: V = Terdeteksi X = Tidak Terdeteksi					

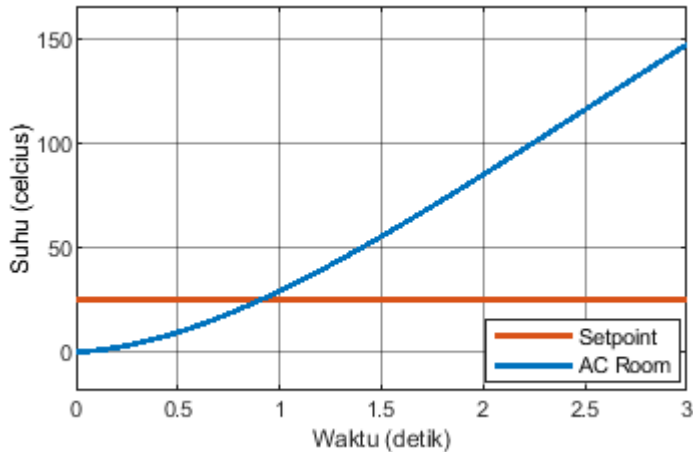
Dari Tabel 4.1-4.2, didapatkan bahwa sensor PIR dapat mendeteksi gerakan dengan baik pada jarak 1~4 meter. Serta jangkauan sudut optimum untuk mendeteksi gerakan adalah 0~40°. Jika selisih jarak dan sudut melebihi nilai tersebut, maka gerakan tidak terdeteksi. Hasil pengujian sesuai dengan kebutuhan sistem dengan jangkauan objek 2~3 meter dan sudut maksimum 40°.

4.3 Hasil Simulasi Sistem Kontrol Logika *Fuzzy*-PID

Simulasi sistem Kontrol Logika *Fuzzy*-PID dilakukan pada *software* Matlab 2018b. Untuk menguji karakteristik sistem, maka dilakukan beberapa pengujian seperti uji *open loop*, uji respons masukan *step*, serta uji *tracking setpoint* naik dan uji *tracking setpoint* turun. Berikut ini adalah hasil simulasi sistem Kontrol Logika *Fuzzy*-PID:

4.3.1 Uji *Open Loop*

Pengujian *open loop* dilakukan untuk mengetahui sejauh mana performansi sistem tanpa adanya pengendali. Masukan *setpoint* berupa sinyal *step*. Berikut pada Gambar 4.3 adalah hasil pengujian sistem *open loop* pada *software* Matlab:

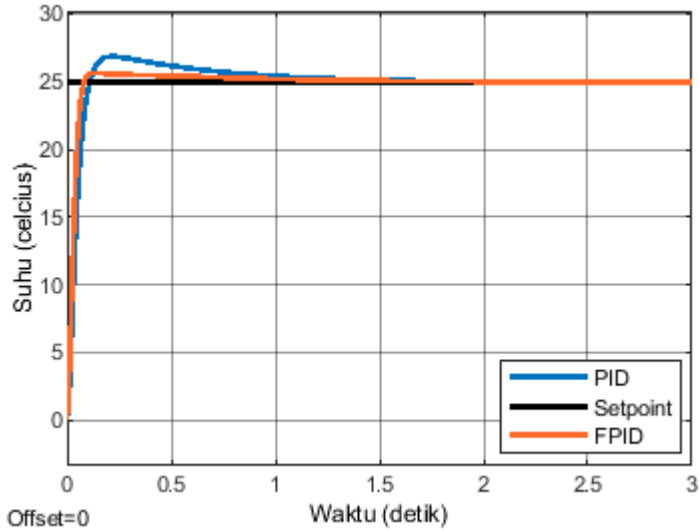


Gambar 4. 3 Grafik Respons Sistem *Open Loop*

Dari Gambar 4.3 dapat dilihat bahwa dengan pengujian sistem *open loop* tanpa adanya pengendali, suhu tidak bisa mencapai nilai *setpoint*, karena performansi sistem tersebut, jauh dari nilai yang diinginkan, yaitu sebesar 25°C. Oleh karena itu penggunaan pengendali pada sistem ini sangat penting untuk mengendalikan suhu agar sistem dapat mencapai nilai yang diinginkan dan tidak menimbulkan kegagalan pada sistem.

4.3.2 Uji Respons Masukan *Step*

Pada pengujian ini diberikan *setpoint*, berupa masukan sinyal *step* sebesar 25°C, nilai ini disesuaikan dengan standar kenyamanan suhu ruangan pada SNI 03-6572-2001. Berikut pada Gambar 4.4 adalah hasil pengujian respons masukan *step* pada *software* Matlab:



Gambar 4. 4 Grafik Respons Masukan *Step*

Pada simulasi ini, respons keluaran pada suhu ruangan menunjukkan nilai yang berbeda untuk kontrol PID dan Kontrol Logika *Fuzzy*-PID. Pada kontrol PID, respons sistem memiliki *rise time* sebesar 0,11 detik, *peak time* sebesar 0,206 detik, *overshoot* sebesar 26,85°C (7,4%) dan *settling time* sebesar 0,367 detik. Sedangkan pada Kontrol Logika *Fuzzy*-PID, respons sistem memiliki *rise time* sebesar 0,082 detik, *peak time* sebesar 0,125 detik, *overshoot* sebesar 25,62°C (2,48%), dan *settling time* sebesar 0,319 detik.

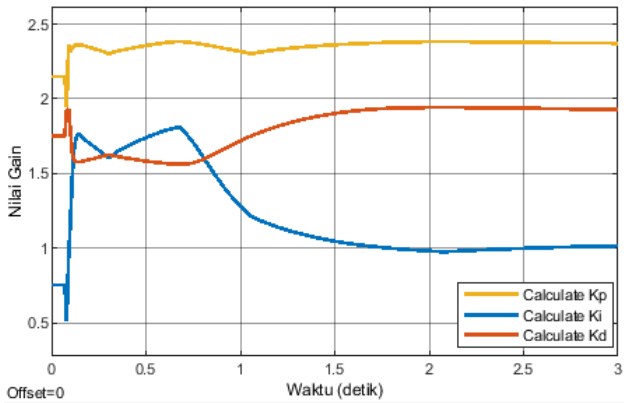
Perbandingan nilai indikator performansi respons masukan *step* dari kontrol PID dan Kontrol Logika *Fuzzy*-PID dijelaskan pada Tabel 4.3 berikut ini:

Tabel 4. 3 Indikator Performansi Respons Masukan *Step*

Jenis Indikator	PID	KLF-PID
ISE	18,44	13
IAE	2,172	1,237
ITAE	0,7092	0,3512

Berdasarkan pada Tabel 4.2 terlihat bahwa sistem KLF-PID memiliki nilai *error* yang lebih rendah daripada sistem kontrol PID baik pada nilai ISE, IAE, dan ITAE. Sehingga didapatkan bahwa performansi sistem kontrol KLF-PID lebih baik daripada sistem kontrol PID.

Berikut pada Gambar 4.5 adalah dinamika nilai *gain* kontrol *Fuzzy*-PID pada pengujian respons masukan *step*:



Gambar 4. 5 Nilai *Gain* KLF-PID Respons Masukan *Step*

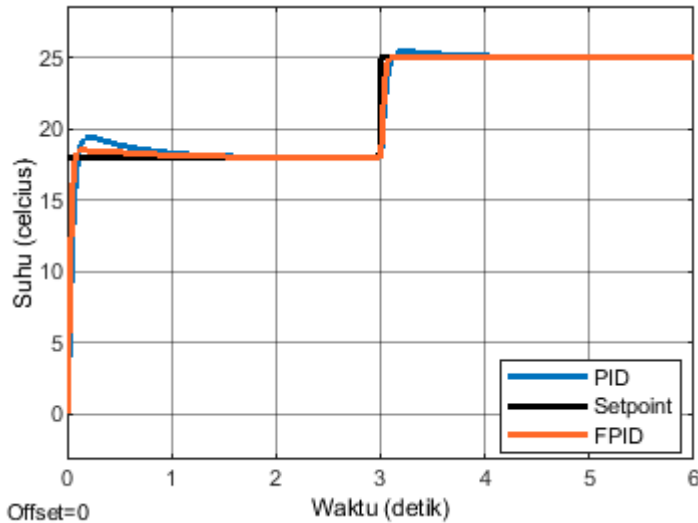
Dari Gambar 4.5 didapatkan bahwa nilai *gain* KLF-PID mengalami dinamika agar sistem dapat mencapai nilai *setpoint* yang ditentukan. Didapatkan pula nilai *gain* akhir pada respons masukan *step* adalah $K_p=2,372$; $K_i=1,016$; dan $K_d=1,927$.

4.2.3 Uji *Tracking Setpoint Naik*

Pada pengujian ini diberikan *setpoint* awal berupa masukan sinyal *step* sebesar 18°C , lalu diberikan nilai *setpoint* kedua sebesar 25°C pada saat sistem telah berjalan selama 3 detik. Pada Gambar 4.6 adalah hasil pengujian *tracking setpoint* naik pada *software* Matlab

Pada simulasi ini respons keluaran pada suhu ruangan menunjukkan nilai yang berbeda untuk kontrol PID dan Kontrol Logika *Fuzzy*-PID. Pada kontrol PID, respons sistem memiliki

rise time sebesar 3,115 detik, *peak time* sebesar 3,198 detik, *overshoot* sebesar 25,42°C (1,68%) dan *settling time* sebesar 3,09 detik. Sedangkan pada Kontrol Logika Fuzzy-PID, respons sistem memiliki *rise time* sebesar 3,089 detik, *peak time* sebesar 3,107 detik, *overshoot* sebesar 25,07°C (0,28%), dan *settling time* sebesar 3,063 detik.



Gambar 4. 6 Grafik *Tracking Setpoint Naik*

Perbandingan nilai indikator performansi *tracking setpoint* naik dari kontrol PID dan Kontrol Logika Fuzzy-PID dijelaskan pada Tabel 4.4 berikut ini:

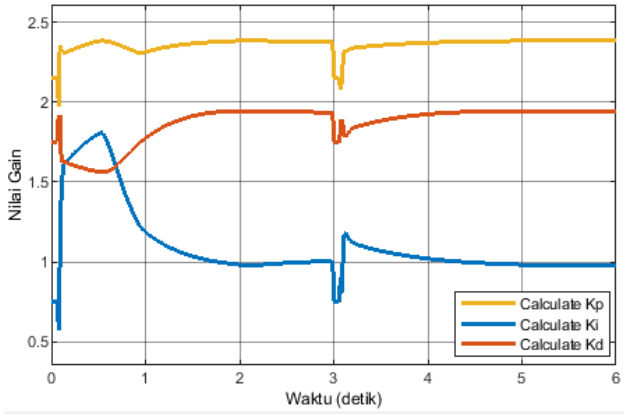
Tabel 4. 4 Indikator Performansi *Tracking Setpoint Naik*

Jenis Indikator	PID	KLF-PID
ISE	11,01	7,768
IAE	2,152	1,196
ITAE	2,378	1,115

Berdasarkan pada Tabel 4.4 terlihat bahwa sistem KLF-PID memiliki nilai *error* yang lebih rendah daripada sistem kontrol PID baik pada nilai ISE, IAE, dan ITAE. Sehingga

didapatkan bahwa performansi sistem KLF-PID lebih baik daripada sistem kontrol PID.

Berikut pada Gambar 4.6 adalah dinamika nilai *gain* kontrol *Fuzzy*-PID pada pengujian *tracking setpoint* naik:



Gambar 4. 7 Nilai *Gain* KLF-PID *Tracking Setpoint* Naik

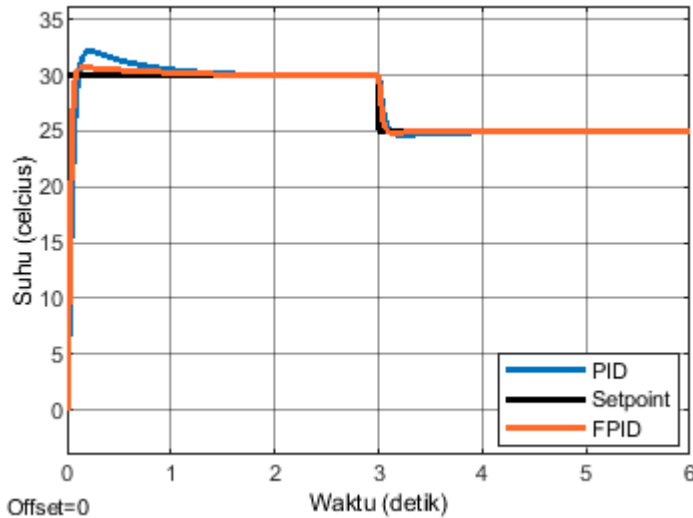
Dari Gambar 4.7 didapatkan bahwa nilai *gain* KLF-PID mengalami dinamika agar sistem dapat mencapai nilai *setpoint* yang ditentukan. Didapatkan pula nilai *gain* akhir pada respons masukan *step* adalah $K_p=2,383$; $K_i=0,9832$; dan $K_d=1,942$.

4.3.4 Uji *Tracking Setpoint* Turun

Pada pengujian ini diberikan *setpoint* awal berupa masukan sinyal *step* sebesar 30°C , lalu diberikan nilai *setpoint* kedua sebesar 25°C pada saat sistem telah berjalan selama 3 detik. Pada Gambar 4.8 adalah hasil pengujian *tracking setpoint* turun pada *software* Matlab:

Pada simulasi ini respons keluaran pada suhu ruangan menunjukkan nilai yang berbeda untuk kontrol PID dan Kontrol Logika *Fuzzy*-PID. Pada kontrol PID, respons sistem memiliki *rise time* sebesar 3,11 detik, *peak time* sebesar 3,2 detik, *overshoot* sebesar $24,65^{\circ}\text{C}$ (1,4%) dan *settling time* sebesar

3,081 detik. Sedangkan pada Kontrol Logika *Fuzzy*-PID, respons sistem memiliki *rise time* sebesar 3,078 detik, *peak time* sebesar 3,103 detik, *undershoot* sebesar 24,87°C (0,52%), dan *settling time* sebesar 3,056 detik.



Gambar 4. 8 Grafik *Tracking Setpoint Turun*

Perbandingan nilai indikator performansi *tracking setpoint* turun dari kontrol PID dan Kontrol Logika *Fuzzy*-PID dijelaskan pada Tabel 4.5 berikut ini:

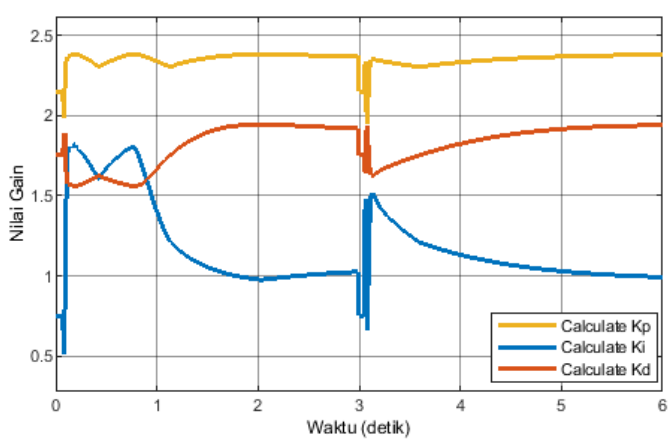
Tabel 4. 5 Indikator Performansi *Tracking Setpoint Turun*

Jenis Indikator	PID	KLF-PID
ISE	27,29	19,25
IAE	3,038	1,797
ITAE	2,357	1,662

Berdasarkan pada Tabel 4.5 terlihat bahwa sistem KLF-PID memiliki nilai *error* yang lebih rendah daripada sistem kontrol PID baik pada nilai ISE, IAE, dan ITAE. Sehingga

didapatkan bahwa performansi sistem KLF-PID lebih baik daripada sistem kontrol PID.

Berikut pada Gambar 4.9 adalah dinamika nilai *gain* kontrol *Fuzzy*-PID pada pengujian *tracking setpoint* turun:

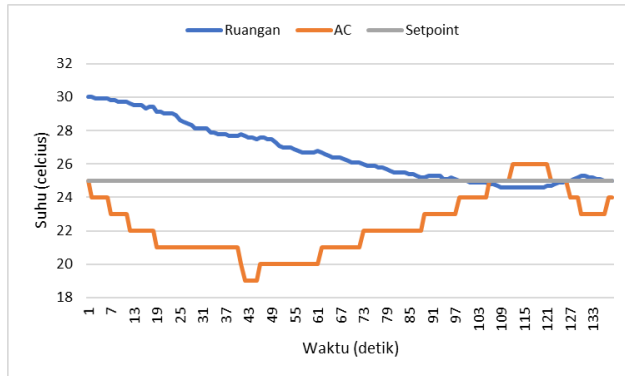


Gambar 4.9 Nilai *Gain* KLF-PID *Tracking Setpoint* Turun

Dari Gambar 4.9 didapatkan bahwa nilai *gain* KLF-PID mengalami dinamika agar sistem dapat mencapai nilai *setpoint* yang ditentukan. Didapatkan pula nilai *gain* akhir pada respons masukan *step* adalah $K_p=2,379$; $K_i=0,9961$; dan $K_d=1,939$.

4.4 Hasil Pengujian *Smart Room System*

Pengujian dilakukan untuk mengetahui karakteristik respons sistem suhu ruangan ber-AC pada Ruangan P-212, Departemen Teknik Fisika ITS, menggunakan *smart room system* berbasis KLF-PID. Pengujian dilakukan pada suhu ruangan awal 30°C. Nilai *setpoint* yang diberikan pada sistem adalah 25°C. Nilai ini disesuaikan berdasarkan pada SNI 03-6572-2001. Pengujian sistem KLF-PID pada dilakukan pada hari Kamis 5 Desember 2019 pukul 09.30 WIB. Berikut pada Gambar 4.10 adalah hasil pengujian respons *smart room system*:



Gambar 4. 10 Hasil Pengujian *Smart Room System*

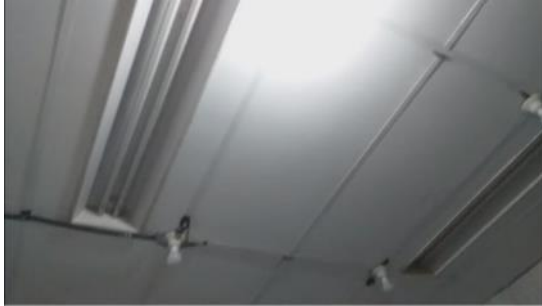
Dari Gambar 4.10, didapatkan bahwa respons sistem memiliki *rise time* sebesar 95 detik, *peak time* sebesar 110 detik, *undershoot* sebesar 24,6°C (1,6%) dan *settling time* sebesar 82 detik. Berikut pada Gambar 4.11 adalah dinamika nilai *gain* kontrol *Fuzzy*-PID pada pengujian *smart room system*:



Gambar 4. 11 Nilai *Gain* KLF-PID pada *Smart Room System*

Dari Gambar 4.11 didapatkan bahwa nilai *gain* KLF-PID mengalami dinamika agar sistem dapat mencapai nilai *setpoint* yang ditentukan. Didapatkan pula nilai *gain* akhir pada respons masukan *step* adalah $K_p=2,35$; $K_i=1,5$; dan $K_d=1,76$.

Pada *smart room system* juga dilakukan pengujian tingkat pencahayaan buatan untuk memastikan agar sistem mampu mencapai standar SNI 03-6575-2001, yaitu 120-250 lux. Berikut pada Gambar 4.12 adalah peletakan lampu dan sensor PIR pada ruangan dan Gambar 4.13 adalah proses pengukuran tingkat pencahayaan buatan:



Gambar 4. 12 Peletakan Lampu dan Sensor PIR

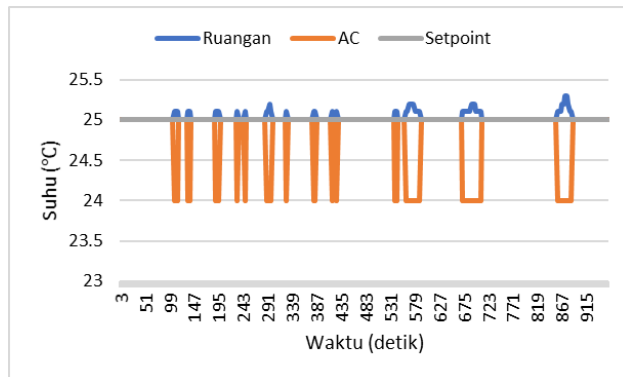


Gambar 4. 13 Pengukuran Tingkat Pencahayaan

Berdasarkan hasil pengukuran yang terdapat pada lampiran, rerata tingkat pencahayaan hasil perancangan *smart room system* adalah sebesar 161,783 lux.

Selain uji masukan *setpoint*, dilakukan pula uji beban pada *real plant*. Uji beban dilakukan pada 16 Januari 2020 pukul 11.40 WIB. Pada uji beban, diberikan beban berupa jumlah orang yang ditambahkan secara bertahap. Dimulai dengan 1 manusia pada

detik ke-0, 2 manusia pada detik ke-90, 3 manusia pada detik ke-270, 4 manusia pada detik ke-450, 5 manusia pada detik ke-630, dan 6 manusia pada detik ke-810. Berikut pada Gambar 4.14 adalah hasil uji beban *Smart Room System*:



Gambar 4. 14 Hasil Uji Beban *Smart Room System*

Dari hasil uji beban didapatkan bahwa, pada setiap penambahan beban pada ruangan, sistem kontrol mampu mengembalikan suhu ruangan menuju suhu *setpoint*, yaitu 25°C.

4.5 Komparasi Hasil Simulasi dengan Pengujian *Real Plant*

Setelah dilakukan pengujian *smart room system*, dilakukan komparasi hasil simulasi pada *software* Matlab dengan hasil pengujian *real plant*, sebagaimana pada Tabel 4.6, untuk diketahui apakah perancangan sistem KLF-PID dapat diterapkan untuk mengendalikan suhu pada ruangan ber-AC.

Tabel 4. 6 Komparasi Karakteristik Respons Sistem

Karakteristik Sistem	Hasil Simulasi	Pengujian <i>Real Plant</i>
<i>Rise Time</i>	3,078 detik	95 detik
<i>Peak Time</i>	3,103 detik	110 detik
<i>Undershoot</i>	24,87°C (0,52%)	24,6°C (1,6%)
<i>Settling Time</i>	3,056 detik	82 detik

Berdasarkan Tabel 4.6, didapatkan bahwa hasil pengujian *smart room system* berbasis KLF-PID memiliki karakteristik sistem yang berbeda dari hasil simulasi pada Simulink Matlab. Pada pengujian *smart room system*, nilai *rise time*, *peak time*, dan *settling time* jauh lebih lama. Nilai *undershoot* hasil simulasi memiliki nilai yang lebih baik daripada hasil pengujian *smart room system*. Hal ini dimungkinkan karena beberapa faktor yang terjadi saat dilakukan pengujian, antara lain:

- a. Pada mekanisme pendinginan AC terdapat *delay time* beberapa detik, terutama saat baru dinyalakan.
- b. Terdapat *delay time* rata-rata 3 detik pada kinerja *smart room system*, yang berasal dari pembacaan sensor DHT22 dan proses penghitungan Kontrol Logika *Fuzzy-PID*.

Perbandingan nilai indikator performansi dilakukan antara hasil simulasi dengan hasil pengujian *real plant* menggunakan *smart room system*, sebagaimana pada Tabel 4.7 berikut ini:

Tabel 4. 7 Komparasi Indikator Performansi Sistem

Indikator Performansi	Hasil Simulasi	Pengujian Real Plant
MSE	3,958	5,634
RMSE	1,989	2,374

Berdasarkan pada Tabel 4.7 terlihat bahwa hasil *pengujian real plant* memiliki nilai *error* yang lebih tinggi daripada hasil simulasi baik pada nilai MSE maupun RMSE. Hal ini dimungkinkan karena pada penelitian ini pemodelan aktuator sistem, berupa *fan coil unit*, tidak disesuaikan dengan aktuator yang sebenarnya pada pengujian *real plant*. Perbedaan kapasitas pendinginan antara simulasi dan *real plant* mengakibatkan kedua sistem tersebut memiliki waktu respons yang jauh berbeda dan nilai *error* yang jauh berbeda pula.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil analisis data yang telah dilakukan maka didapatkan kesimpulan dari tugas akhir mengenai “Perancangan *Smart Room System* dengan Indikator Kenyamanan Suhu dan Pencahayaan Berbasis Kontrol Logika *Fuzzy-PID* (KLF-PID)” adalah sebagai berikut:

1. *Smart Room System* merupakan integrasi dari 2 sub-sistem. Pertama adalah sistem kontrol suhu ruangan ber-AC. Di mana nilai *setpoint* adalah 25°C, sesuai dengan SNI 03-6572-2001, *feedback* sistem berupa suhu ruangan didapatkan dengan menggunakan sensor DHT22, komunikasi data dengan AC menggunakan *wireless-IR*, serta kontrol menggunakan KLF-PID. Kedua adalah sistem kontrol pencahayaan. Di mana target tingkat pencahayaan adalah 120-250 lux, sesuai dengan SNI 03-6575-2001, *feedback* sistem berupa daerah yang digunakan manusia untuk beraktivitas yang dideteksi menggunakan sensor PIR, dan *relay* digunakan untuk mengatur ON-OFF pada lampu.
2. Berdasarkan hasil simulasi sistem Kontrol Logika *Fuzzy-PID* pada *software* Matlab, didapatkan bahwa respons *tracking setpoint* turun Kontrol Logika *Fuzzy-PID* memiliki *rise time* 3,078 detik, *peak time* 3,103 detik, *undershoot* 24,87°C (0,52%), dan *settling time* 3,056 detik. Dengan nilai akhir *gain* Kontrol Logika *Fuzzy-PID*: $K_p=2,379$; $K_i=0,9961$; dan $K_d=1,939$. Secara umum, pada uji respons masukan *step* maupun uji *tracking setpoint* naik dan turun, menunjukkan bahwa Kontrol Logika *Fuzzy-PID* mampu memberikan nilai indikator performansi kontrol yang lebih baik dari kontrol PID. Sedangkan dari hasil pengujian Kontrol Logika *Fuzzy-PID* pada *smart room system*, didapatkan bahwa respons sistem memiliki *rise time* 95

detik, *peak time* 110 detik, *undershoot* 24,6°C (1,6%) dan *settling time* 82 detik. Dengan nilai akhir *gain* Kontrol Logika Fuzzy-PID: $K_p=2,35$; $K_i=1,5$; dan $K_d=1,76$.

5.2 Saran

Saran yang diberikan untuk pengembangan tugas akhir pada penelitian ini adalah:

1. Menggunakan mikrokontroler dengan kapasitas EEPROM yang lebih tinggi dari Arduino Mega 2560, untuk menghindari *delay time* dan konflik kontrol antar sub-sistem.
2. Komunikasi data dari sistem ke AC maupun koneksi antara lampu dan sistem dapat menggunakan modul IoT agar sistem menjadi lebih fleksibel.
3. Menggunakan model sistem yang lebih akurat, baik aktuator maupun *plant* agar simulasi dapat dilakukan mendekati kondisi *real plant*.
4. Menambahkan parameter kelembaban dan jumlah orang dalam ruangan sebagai masukan sistem, agar sistem dapat mencapai standar kenyamanan termal yang lebih baik.

DAFTAR PUSTAKA

- [1] Kementerian Energi dan Sumber Daya Mineral, “Statistik Konsumsi Listrik Nasional Tahun 2014-2017,” Kementerian Energi dan Sumber Daya Mineral, Jakarta, 2017.
- [2] PT (Persero) Perusahaan Listrik Nasional, “Statistik Tarif Listrik Nasional Semua Golongan Tahun 2014-2017,” PT (Persero) Perusahaan Listrik Nasional, Jakarta, 2017.
- [3] *American Council of Energy Efficiency Economy (ACEEE), “Study on Energy Efficiency of Air Conditioner Usage in Household System,” American Council of Energy Efficiency Economy (ACEEE), 2014*
- [4] Majumdar, M K Hassan. Biswas, Himel. Shaim, MH Ali. Ahmmed, K Tanvir., “Automated Energy Saving and Safety System,” *International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT), 2014.*
- [5] R.R., Yuganandhine. Kiruthika, J. Manju, S. Mythili, K. Manikandan, R.K., “Automatic Room Light and Fan Controller with Bidirectional Visitor Counter,” *International Journal of Recent Trends in Engineering & Research (IJRTER) Conference on Electronics, Information and Communication Systems (CELICS’17), 2017.*
- [6] Waradkar, G., Ramina, H., Maitry, V., Ansurkar, T., Rawat, A., Das, P., ... & Sion, P. V. P. P. C. O. E. (2016). *Automated room light controller with visitor counter. Imperial Journal of Interdisciplinary Research (IJIR), 2(4).*
- [7] Farooq, M. U., Shakoor, A., & Siddique, A. B. (2016). *ARM based Bidirectional Visitor Counter and Automatic Room Light Controller using PIR sensors. Advances in Science, Technology and Engineering Systems Journal, 1(5), 10-15.*

- [8] Kumar, M. R., Goswami, M. S., Singh, M. D., & Bisht, M. U. (2018). *Automatic Room Light and Other Appliances Controller with Bidirectional Visitor Counter. International Journal of Creative Research Thoughts (IJCRT)*, 6(2), 194-208.
- [9] Guanzhou, Du. Yue, Shang. Jia, Qin. Xiaoming, Ma., "Research on Temperature Control System of Air Conditioning Room Based on Fuzzy Control," *IEEE 3rd International Conference on Control Science and Systems Engineering*, 2018.
- [10] Ying-hong, D. U. A. N. (2008). *Research of Predictive Fuzzy-PID Controller in Temperature of Central Air-conditioning System [J]*. *Journal of System Simulation*, 3, 620-622.
- [11] Xing, L., Yang, S., & Lu, Q. (2010, June). *Self-tuning fuzzy PID controller for temperature control in Variable Air Volume air conditioning systems. In The 2010 IEEE International Conference on Information and Automation* (pp. 2117-2120). IEEE.
- [12] Khooban, M Hasan., Soltanpourb, M Reza., Abadi, D N Maryam., Esfahani, Zahra., "Optimal Intelligent Control for HVAC Systems," *Journal of Power Technologies*, Vol. No.92, No.3 (2012), pp 192–200.
- [13] Ying-Guo, P., Hua-Guang, Z., & Bien, Z. (1998, May). *A simple fuzzy adaptive control method and application in HVAC. In 1998 IEEE International Conference on Fuzzy Systems Proceedings. IEEE World Congress on Computational Intelligence* (Cat. No. 98CH36228) (Vol. 1, pp. 528-532). IEEE.
- [14] Wang, J., An, D., & Lou, C. (2006, June). *Application of fuzzy-PID controller in heating ventilating and air-conditioning system. In 2006 International Conference on Mechatronics and Automation* (pp. 2217-2222). IEEE.
- [15] Waibel, A., Schultz, T., Bett, M., Denecke, M., Malkin, R., Rogina, I., ... & Yang, J. (2003, April). SMART: The smart meeting room task at ISL. In *2003 IEEE*

- International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. ICASSP'03.* (Vol. 4, pp. IV-752). IEEE.
- [16] Arnil, J., Punsawad, Y., & Wongsawat, Y. (2011, December). Wireless sensor network-based smart room system for healthcare monitoring. In *2011 IEEE International Conference on Robotics and Biomimetics* (pp. 2073-2076). IEEE.
- [17] Caniou, J. (2013). *Passive infrared detection: theory and applications*. Springer Science & Business Media.
- [18] K. Ogata, *Modern Control Engineering* 5ed, New Jersey: Prentice-Hall, 2010.
- [19] Arulmozhiyal, R., & Kandiban, R. (2012, January). *Design of fuzzy PID controller for brushless DC motor*. In *2012 International Conference on Computer Communication and Informatics* (pp. 1-7). IEEE.
- [20] Abadi, Imam. (2008). *Simulasi Pengendalian Level Steam Drum Dengan Pengendali Pid Berbasis Fuzzy Gain Scheduling*. Jurnal Sains dan Teknologi EMAS, Vol. 18, No. 4.
- [21] Pedrycz, Witold. "Fuzzy Control and Fuzzy Systems (2 ed.)," Research Studies Press Ltd. 1993
- [22] Ahmed, N Udin., Rabbi, M F., Rahaman, S A M Matiur., Sundaraj, Kenneth., "Fuzzy Logic Controller Design for Intelligent Air-Conditioning System," 2nd International Conference on Control Science and Systems Engineering. 2016.
- [23] Zhao, Z. Y., Tomizuka, M., & Isaka, S. (1993). *Fuzzy gain scheduling of PID controllers*. *IEEE transactions on systems, man, and cybernetics*, 23(5), 1392-1398.
- [24] Hu, B. G., Mann, G. K., & Gosine, R. G. (2001). *A systematic study of fuzzy PID controllers-function-based evaluation approach*. *IEEE transactions on fuzzy systems*, 9(5), 699-712.

- 25] Xu, J. X., Hang, C. C., & Liu, C. (2000). *Parallel structure and tuning of a fuzzy PID controller. Automatica*, 36(5), 673-684.
- [26] Handbook-Fundamentals, A. S. H. R. A. E. (2009). *American society of Heating. Refrigerating and Air-Conditioning Engineers*.

LAMPIRAN 1

Pengambilan Data Primer

a. Dimensi Ruang

Instrumen Ukur : Meteran Pita

Dimensi	Ukuran (m)
Panjang	3,55
Lebar	3,55
Tinggi	2,52

b. Suhu Internal dan Eksternal Ruang

Instrumen Ukur : Termokopel Tipe-K *APPA51*

Suhu	Percobaan ke-					Rata-Rata
	1	2	3	4	5	
Internal*	26,6	26,2	26,7	26,5	26,3	26,46
Eksternal	28,5	28,4	28,6	27,8	27,5	28,16

* pengukuran suhu internal dilakukan saat beban panas maksimum (6 orang, 1 TLD, 2 laptop, 6 telepon selular)

c. Pencahayaan Alami pada Ruang P-212

Instrumen Ukur : Luxmeter *DEKKO*

21,6	51,0	64,8	32,3
21,3	52,3	60,8	30,3
21,6	51,3	62,0	29,8
20,2	25,5	46,0	33,4
20,2	26,5	45,1	35,1
20,3	25,7	45,6	34,6
19,3	20,3	32,0	20,5
19,4	20,7	35,2	22,5
19,2	20,5	31,5	22,2
13,5	19,5	23,0	19,1
13,6	19,2	22,2	19,2
13,8	19,1	22,0	20,1

d. Pengujian Sensor DHT22

Suhu Ref (°C)	Suhu Sensor (°C)					Rerata	σ	Ua1	Koreksi
	1	2	3	4	5				
30	29.1	29.1	29.1	29.2	29.1	29.12	0.044721	0.020000	0.88
29	27.9	27.9	28.0	28.0	28.0	27.96	0.054772	0.024495	1.04
28	26.8	26.9	26.9	27.0	27.0	26.92	0.083666	0.037417	1.08
27	26.0	26.0	26.0	26.1	26.0	26.02	0.044721	0.020000	0.98
26	24.9	24.9	25.0	24.9	24.9	24.92	0.044721	0.020000	1.08
25	24.0	24.1	24.0	24.0	24.0	24.02	0.044721	0.020000	0.98
24	23.1	23.1	23.2	23.2	23.1	23.14	0.054772	0.024495	0.86
23	21.9	22.0	22.2	22.1	22.1	22.06	0.114018	0.050990	0.94
22	21.0	20.9	20.9	20.9	20.8	20.90	0.070711	0.031623	1.10
21	20.0	20.0	19.9	19.9	19.9	19.94	0.054772	0.024495	1.06
20	18.9	19.0	19.0	19.0	19.0	18.98	0.044721	0.020000	1.02
19	18.0	18.0	17.9	17.9	18.0	17.96	0.054772	0.024495	1.04
18	17.1	17.1	17.2	17.1	17.3	17.16	0.089443	0.040000	0.84

e. Pencahayaan Smart Room System pada Ruangan P-212

Instrumen Ukur : Luxmeter DEKKO

160,6	161,0	164,8	162,3
160,3	162,3	160,8	160,3
160,6	161,3	162,0	161,8
162,2	170,2	166,0	163,4
162,5	166,5	170,1	165,1
162,3	165,7	165,6	164,6
159,3	160,3	163,7	160,5
159,4	160,7	165,2	162,5
159,2	160,5	161,5	162,2
153,5	159,5	163,0	160,1
153,6	159,2	162,2	160,2
153,8	159,1	162,0	162,1

LAMPIRAN 2

Source Code Arduino

```
/*
 * Arduino sketch: Smart Room System with KLF-PID
 *
 * Author   : Ardyas Nur Aufa
 * e-mail   : ardyasnuraufa@gmail.com
 */

//Menambahkan Lib & Header ke Program
#include <IRremote.h> //Lib untuk Wireless-IR
#include <DHT.h> // Lib untuk Sensor DHT22
#include "fis_header.h" // Header untuk FIS

//Inisialisasi Sensor DHT22
#define DHTPIN 13
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);

// Menentukan Pin Input: Sensor PIR
#define PIR_1 5
#define PIR_2 4
#define PIR_3 3
#define PIR_4 11

// Menentukan Pin Output: Relay
#define RELAY_1 6
#define RELAY_2 7
#define RELAY_3 8
#define RELAY_4 10

// Menentukan RAW Data setiap Fungsi AC
```

```
unsigned int AC18[] = {3208, 1576, 408, 388, 404, 1180, 408, 388,
404, 388, 404, 1180, 408, 388, 404, 1180, 404, 388, 408, 384, 408,
1180, 404, 1180, 408, 1180, 404, 388, 408, 1176, 408, 388, 404,
1180, 408, 1176, 408, 1180, 408, 388, 404, 388, 404, 388, 408,
388, 404, 1180, 404, 1180, 408, 388, 404, 1180, 408, 1176, 408,
388, 404, 388, 408, 1176, 408, 388, 408, 384, 408, 1176, 408, 388,
404, 388, 408, 1176, 408, 1180, 408, 384, 408, 1180, 404, 1180,
408, 1180, 404, 1180, 408, 1180, 404, 1180, 408, 1180, 404, 1180,
412, 1176, 404, 1180, 408, 388, 404, 388, 404, 388, 408, 384, 408,
388, 404, 388, 404, 388, 408, 384, 408, 1176, 408, 1180, 408, 1176,
408, 1180, 408, 388, 404, 1180, 404, 1180, 408, 1180, 408, 388,
404, 388, 404, 388, 404, 388, 408, 1176, 408, 388, 404, 388, 404,
388, 408, 388, 404, 1180, 408, 1176, 408, 388, 404, 388, 408, 1176,
408, 1184, 400, 1180, 408, 1180, 404, 388, 408, 388, 404, 1180,
404, 1180, 408, 388, 404, 388, 404, 388, 408};
```

```
unsigned int AC19[] = {3208, 1576, 404, 388, 404, 1184, 404, 388,
404, 392, 404, 1180, 404, 388, 404, 1180, 408, 388, 404, 388, 404,
1180, 408, 1180, 404, 1180, 408, 388, 404, 1180, 408, 388, 404,
1180, 404, 1180, 408, 1180, 404, 388, 404, 392, 404, 388, 404,
388, 404, 1184, 404, 1180, 404, 392, 404, 1180, 404, 1180, 408,
388, 404, 388, 404, 1180, 404, 392, 404, 388, 404, 1180, 408, 388,
404, 388, 404, 1180, 408, 1180, 404, 388, 404, 1180, 408, 1180,
404, 1184, 404, 1180, 404, 1180, 408, 1180, 404, 1180, 408, 1180,
404, 1180, 408, 1180, 404, 388, 404, 392, 404, 388, 404, 388, 404,
392, 404, 388, 404, 388, 404, 388, 404, 1180, 408, 1180, 404, 1184,
404, 1180, 404, 392, 404, 1180, 404, 1180, 404, 1184, 404, 388,
404, 388, 404, 392, 404, 388, 404, 1180, 408, 388, 404, 388, 404,
388, 404, 392, 404, 1180, 404, 1180, 404, 392, 404, 1180, 404,
392, 404, 1180, 404, 1180, 408, 1180, 404, 388, 404, 392, 400,
1184, 404, 388, 404, 1180, 408, 388, 404, 388, 404};
```

```
unsigned int AC20[] = {3208, 1576, 404, 392, 400, 1184, 404, 388,
404, 392, 400, 1184, 404, 388, 404, 1184, 404, 388, 404, 388, 404,
1180, 408, 1180, 404, 1180, 408, 388, 404, 1180, 404, 392, 404,
1180, 404, 1180, 408, 1180, 404, 388, 404, 392, 404, 388, 404,
```

388, 404, 1180, 408, 1180, 404, 392, 400, 1184, 404, 1180, 404,
 392, 404, 388, 404, 1180, 404, 392, 404, 388, 404, 1180, 404, 392,
 404, 388, 404, 1180, 404, 1184, 404, 388, 404, 1180, 408, 1180,
 404, 1180, 408, 1180, 404, 1180, 408, 1180, 404, 1180, 408, 1180,
 404, 1180, 408, 1180, 404, 388, 404, 392, 404, 388, 404, 388, 404,
 388, 404, 392, 404, 388, 404, 388, 404, 1180, 408, 1180, 404, 1180,
 408, 1180, 404, 392, 404, 1176, 408, 1180, 408, 1180, 404, 388,
 404, 388, 404, 392, 404, 388, 404, 1180, 404, 392, 404, 388, 404,
 388, 404, 392, 404, 1176, 408, 1180, 404, 392, 404, 388, 404, 388,
 404, 1180, 408, 1180, 404, 1180, 408, 388, 404, 388, 404, 1180,
 408, 1180, 404, 1180, 408, 388, 404, 388, 404, 388, 404};
 unsigned int AC21[] = {3208, 1576, 404, 388, 404, 1180, 408, 388,
 404, 388, 404, 1180, 408, 388, 404, 1180, 404, 392, 404, 388, 404,
 1180, 408, 1180, 404, 1180, 408, 388, 404, 1180, 404, 392, 404,
 1180, 404, 1180, 404, 1184, 404, 392, 400, 392, 404, 388, 404,
 388, 404, 1180, 404, 1184, 404, 388, 404, 1180, 408, 1180, 404,
 388, 404, 392, 404, 1180, 404, 392, 404, 388, 404, 1180, 404, 392,
 400, 392, 404, 1180, 404, 1184, 404, 388, 404, 1180, 408, 1180,
 404, 1180, 404, 1184, 404, 1180, 404, 1184, 404, 1180, 404, 1184,
 404, 1180, 404, 1184, 404, 388, 404, 388, 408, 388, 404, 388, 404,
 388, 404, 392, 404, 388, 404, 388, 404, 1180, 404, 1184, 404, 1180,
 404, 1184, 404, 388, 404, 1184, 404, 1180, 404, 1180, 408, 388,
 404, 388, 404, 392, 400, 392, 404, 1180, 404, 392, 400, 392, 404,
 388, 404, 388, 404, 1180, 408, 1180, 404, 392, 404, 1180, 404,
 1180, 404, 392, 404, 1180, 404, 1180, 408, 388, 404, 388, 404,
 1180, 408, 388, 404, 388, 404, 1180, 408, 388, 404};
 unsigned int AC22[] = {3208, 1576, 408, 388, 404, 1180, 404, 392,
 404, 388, 404, 1180, 408, 388, 404, 1180, 404, 388, 404, 392, 404,
 1180, 404, 1180, 408, 1180, 404, 388, 404, 1184, 404, 388, 404,
 1180, 408, 1180, 404, 1180, 408, 388, 404, 388, 404, 388, 404,
 392, 404, 1180, 404, 1180, 408, 388, 404, 1180, 408, 1180, 404,
 388, 404, 392, 400, 1184, 404, 388, 404, 388, 408, 1180, 404, 388,
 404, 392, 400, 1184, 404, 1180, 404, 388, 408, 1180, 404, 1180,
 408, 1180, 404, 1180, 408, 1180, 404, 1180, 408, 1180, 404, 1180,

```
408, 1180, 404, 1180, 408, 388, 404, 388, 404, 388, 404, 392, 404,
388, 404, 388, 404, 388, 404, 392, 404, 1180, 404, 1180, 408, 1180,
404, 1180, 408, 388, 404, 1180, 404, 1180, 408, 1180, 404, 392,
404, 388, 404, 388, 404, 388, 404, 1180, 408, 388, 404, 388, 404,
392, 404, 388, 404, 1180, 404, 1180, 408, 388, 404, 388, 408, 1176,
408, 388, 404, 1180, 408, 1180, 404, 388, 404, 392, 404, 1180,
404, 1180, 408, 388, 404, 1180, 404, 392, 404};
unsigned int AC23[] = {3208, 1576, 408, 388, 404, 1180, 404, 392,
404, 388, 404, 1180, 404, 392, 404, 1180, 404, 392, 400, 392, 404,
1180, 404, 1180, 408, 1180, 404, 392, 400, 1184, 404, 388, 404,
1180, 408, 1180, 404, 1184, 404, 388, 404, 388, 404, 388, 404,
392, 404, 1180, 404, 1180, 408, 388, 404, 1180, 404, 1180, 408,
388, 404, 392, 400, 1184, 404, 388, 404, 392, 400, 1180, 408, 388,
404, 388, 404, 1180, 408, 1180, 404, 392, 404, 1180, 404, 1180,
404, 1184, 404, 1180, 408, 1180, 404, 1180, 408, 1180, 404, 1180,
408, 1180, 404, 1180, 408, 388, 404, 388, 404, 388, 404, 388, 408,
388, 404, 388, 404, 388, 404, 392, 404, 1180, 404, 1180, 408, 1176,
408, 1180, 408, 388, 404, 1180, 404, 1180, 408, 1180, 404, 392,
400, 392, 404, 388, 404, 392, 400, 1180, 408, 388, 404, 388, 404,
392, 404, 388, 404, 1180, 404, 1184, 404, 388, 404, 1180, 408,
388, 404, 388, 404, 1180, 408, 1180, 404, 388, 404, 388, 408, 1180,
404, 388, 404, 1180, 408, 1180, 404, 388, 408};
unsigned int AC24[] = {3208, 1572, 408, 388, 404, 1180, 408, 388,
404, 388, 404, 1180, 408, 388, 404, 1184, 404, 388, 404, 388, 404,
1180, 408, 1180, 404, 1180, 408, 388, 404, 1180, 408, 384, 408,
1180, 404, 1180, 408, 1180, 404, 388, 404, 392, 404, 388, 404,
388, 404, 1180, 408, 1180, 404, 388, 404, 1180, 408, 1180, 404,
392, 404, 388, 404, 1180, 404, 392, 404, 388, 404, 1180, 404, 392,
404, 388, 404, 1180, 404, 1180, 408, 388, 404, 1180, 408, 1180,
404, 1180, 404, 1180, 408, 1180, 404, 1180, 408, 1180, 408, 1180,
404, 1180, 408, 1180, 404, 388, 404, 392, 400, 392, 404, 388, 404,
388, 404, 388, 404, 392, 404, 388, 404, 1180, 404, 1180, 408, 1180,
404, 1184, 404, 388, 404, 1180, 408, 1180, 404, 1180, 408, 388,
404, 388, 404, 388, 404, 392, 404, 1180, 404, 388, 404, 392, 404,
```


388, 404, 388, 404, 1180, 408, 1180, 404, 392, 400, 392, 404, 388, 404, 388, 404, 1180, 408, 1180, 404, 392, 400, 392, 404, 1180, 404, 1180, 408, 1180, 404, 1180, 408, 388, 404};

unsigned int AC25[] = {3208, 1576, 408, 388, 404, 1180, 404, 392, 404, 388, 404, 1180, 404, 392, 404, 1180, 404, 388, 404, 392, 400, 1184, 404, 1184, 404, 1180, 404, 392, 400, 1180, 408, 388, 404, 1180, 408, 1180, 404, 1180, 408, 388, 404, 388, 404, 388, 404, 392, 404, 1180, 404, 1180, 408, 388, 404, 1180, 404, 1184, 404, 388, 404, 388, 404, 1184, 404, 388, 404, 388, 404, 1180, 408, 388, 404, 388, 404, 1184, 404, 1180, 404, 392, 404, 1180, 404, 1180, 408, 1180, 404, 1180, 408, 1180, 404, 1180, 408, 1180, 404, 1180, 408, 1180, 404, 1180, 408, 388, 404, 388, 404, 388, 404, 388, 408, 388, 404, 388, 404, 388, 404, 392, 404, 1180, 404, 1180, 408, 1180, 404, 1180, 404, 392, 404, 1180, 404, 1180, 408, 1180, 404, 388, 404, 392, 404, 388, 404, 388, 404, 1180, 408, 388, 404, 388, 404, 1176, 408, 1180, 404, 392, 404, 1180, 404, 388, 404, 392, 404, 1180, 404, 392, 400, 392, 404, 388, 404, 1180, 404};

unsigned int AC26[] = {3208, 1576, 408, 388, 404, 1180, 404, 392, 404, 388, 404, 1180, 404, 392, 404, 1180, 404, 392, 400, 392, 404, 1180, 404, 1180, 408, 1180, 404, 392, 404, 1176, 408, 388, 404, 1180, 408, 1180, 404, 1180, 408, 388, 404, 388, 404, 392, 400, 392, 404, 1180, 404, 1180, 408, 388, 404, 1180, 404, 1184, 404, 388, 404, 388, 404, 1180, 408, 388, 404, 388, 404, 1180, 408, 388, 404, 388, 404, 1180, 408, 388, 404, 1180, 404, 1180, 408, 1180, 404, 1180, 408, 1180, 404, 1180, 408, 1180, 404, 1180, 408, 1180, 404, 1180, 408, 1180, 404, 1180, 408, 1180, 404, 1180, 408, 1180, 404, 392, 404, 388, 404, 388, 404, 392, 400, 392, 404, 388, 404, 388, 404, 392, 400, 1184, 404, 1180, 404, 1184, 404, 1180, 404, 392, 404, 1180, 404, 1180, 408, 1180, 404, 392, 400, 388, 408, 388, 404, 388, 404, 1180, 408, 388, 404, 388, 404, 388, 408, 388, 404, 1180, 404, 1180, 408, 388, 404, 388, 404, 1180, 408, 1180, 404, 388, 404, 1184, 404, 388, 404, 388, 404, 1184, 404, 1180, 404, 392, 404, 388, 404, 1180, 404};

```
unsigned int AC27[] = {3208, 1576, 408, 388, 404, 1180, 404, 392,
404, 388, 404, 1180, 404, 392, 404, 1180, 404, 388, 404, 392, 404,
1180, 404, 1180, 408, 1180, 404, 388, 404, 1180, 408, 388, 404,
1180, 408, 1180, 404, 1180, 408, 388, 404, 388, 404, 388, 404,
392, 404, 1180, 404, 1180, 408, 388, 404, 1180, 404, 1180, 408,
388, 404, 388, 404, 1184, 404, 388, 404, 388, 404, 1180, 408, 388,
404, 388, 404, 1180, 408, 1180, 404, 392, 404, 1176, 408, 1180,
404, 1184, 404, 1180, 408, 1180, 404, 1180, 404, 1184, 404, 1180,
404, 1184, 404, 1180, 404, 392, 404, 388, 404, 388, 404, 388, 404,
388, 408, 388, 404, 388, 404, 392, 400, 1184, 404, 1180, 404, 1180,
408, 1180, 404, 392, 404, 1180, 404, 1180, 404, 1184, 404, 388,
404, 392, 404, 388, 404, 388, 404, 1180, 408, 388, 404, 388, 404,
388, 404, 392, 404, 1180, 404, 1180, 408, 388, 404, 1180, 404,
388, 408, 1180, 404, 388, 404, 1180, 408, 388, 404, 388, 404, 1180,
408, 388, 404, 1180, 408, 388, 404, 1180, 404};
```

```
unsigned int AC28[] = {3208, 1576, 404, 388, 404, 1184, 404, 388,
404, 388, 404, 1184, 404, 388, 404, 1180, 404, 392, 404, 388, 404,
1180, 408, 1176, 408, 1180, 404, 392, 404, 1180, 404, 392, 400,
1184, 404, 1180, 404, 1180, 408, 388, 404, 388, 404, 392, 404,
388, 404, 1180, 404, 1184, 404, 388, 404, 1180, 408, 1180, 404,
388, 404, 392, 404, 1180, 404, 388, 404, 392, 404, 1180, 404, 392,
400, 392, 404, 1180, 404, 1180, 408, 388, 404, 1180, 404, 1184,
404, 1180, 404, 1180, 408, 1180, 404, 1184, 404, 1180, 408, 1180,
404, 1180, 404, 1180, 408, 388, 404, 392, 400, 392, 404, 388, 404,
388, 404, 388, 404, 392, 404, 388, 404, 1180, 404, 1180, 408, 1180,
404, 1180, 408, 388, 404, 1180, 408, 1180, 404, 1180, 408, 388,
404, 388, 404, 392, 400, 392, 404, 1180, 404, 388, 404, 388, 408,
388, 404, 388, 404, 1180, 404, 1184, 404, 392, 400, 392, 400, 392,
404, 1180, 404, 392, 404, 1176, 408, 388, 404, 392, 404, 1180,
404, 1180, 404, 1184, 404, 388, 404, 1180, 408};
```

```
unsigned int AC29[] = {3208, 1576, 404, 392, 404, 1180, 404, 392,
404, 388, 404, 1180, 404, 392, 400, 1180, 408, 392, 400, 392, 400,
1184, 404, 1180, 404, 1184, 404, 388, 404, 1180, 408, 388, 404,
1180, 408, 1180, 404, 1180, 404, 392, 404, 388, 404, 392, 400,
```

```

392, 400, 1180, 408, 1180, 408, 388, 404, 1180, 404, 1184, 408,
384, 404, 388, 404, 1180, 408, 388, 404, 388, 404, 1180, 408, 388,
404, 388, 404, 1180, 408, 1180, 404, 388, 404, 1184, 404, 1180,
404, 1184, 404, 1180, 404, 1184, 404, 1180, 404, 1184, 404, 1180,
404, 1184, 404, 1180, 404, 388, 408, 388, 404, 388, 404, 388, 404,
392, 404, 388, 404, 388, 404, 388, 404, 1184, 404, 1184, 400, 1180,
408, 1180, 404, 392, 404, 1180, 404, 1180, 408, 1180, 404, 388,
404, 392, 404, 388, 404, 388, 404, 1180, 408, 388, 404, 388, 404,
388, 404, 388, 408, 1180, 404, 1180, 408, 388, 404, 1180, 404,
1180, 408, 388, 404, 388, 404, 1184, 404, 388, 404, 388, 404, 1180,
408, 388, 404, 388, 404, 1180, 408, 1180, 404};
unsigned int AC30[] = {3208, 1576, 408, 388, 404, 1180, 408, 384,
404, 392, 404, 1180, 404, 392, 404, 1180, 404, 388, 404, 392, 404,
1180, 404, 1180, 408, 1180, 404, 388, 404, 1180, 408, 388, 404,
1180, 404, 1184, 404, 1180, 408, 388, 404, 388, 404, 388, 404,
388, 408, 1180, 404, 1180, 404, 392, 404, 1180, 404, 1184, 404,
388, 404, 388, 404, 1180, 408, 388, 404, 388, 404, 1180, 408, 388,
404, 388, 404, 1180, 408, 1180, 404, 392, 400, 1184, 404, 1180,
404, 1184, 404, 1180, 404, 1180, 408, 1180, 404, 1180, 408, 1180,
404, 1184, 404, 1180, 404, 392, 404, 388, 404, 388, 404, 388, 404,
392, 404, 388, 404, 388, 404, 388, 404, 1180, 408, 1180, 404, 1184,
404, 1180, 404, 392, 404, 1180, 404, 1180, 408, 1180, 404, 388,
404, 388, 408, 388, 404, 388, 404, 1180, 408, 388, 404, 388, 404,
388, 404, 392, 400, 1184, 404, 1180, 404, 392, 404, 388, 404, 1180,
404, 392, 404, 388, 404, 1180, 404, 392, 404, 388, 404, 1180, 408,
1180, 404, 388, 404, 1180, 408, 1180, 404};

```

```

//Menentukan Variabel untuk Kontrol Suhu
IRsend irsend;
int khz = 38; // 38kHz frekuensi karier untuk protokol NEC
int AC_Temp, suhuACint;

```

```

//Menentukan Variabel untuk Kontrol PID
double Kp, Ki, Kd, Kpp, Kdp, alpha;

```

```
double Ku=4.6737, Tu=3.26;
double Kpmin=0.32*Ku, Kpmax=0.6*Ku, Kdmin=0.08*Ku*Tu,
Kdmax=0.15*Ku*Tu;
unsigned long currentTime, previousTime;
double elapsedTime;
double error, delta_error;
double lastError;
double input, output, setpoint;
double cumError, rateError;

// Jumlah Input FIS
const int fis_gcI = 2;
// Jumlah Output FIS
const int fis_gcO = 3;
// Jumlah Rules FIS
const int fis_gcR = 49;
FIS_TYPE g_fisInput[fis_gcI];
FIS_TYPE g_fisOutput[fis_gcO];

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

  // inialisasi input&output sub-sistem pencahayaan
  Init_cahaya();

  // memulai pengambilan data DHT22
  dht.begin();

  // menentukan setpoint suhu ruangan
  setpoint = 25;

  // inialisasi PLX-DAQ
  Init_PlxDaq();}
```

```

void loop() {
//Lampu 1
  if(deteksi1())
    lampuON1();
  else
    lampuOFF1();
//Lampu 2
  if(deteksi2())
    lampuON2();
  else
    lampuOFF2();
//Lampu 3
  if(deteksi3())
    lampuON3();
  else
    lampuOFF3();
//Lampu 4
  if(deteksi4())
    lampuON4();
  else
    lampuOFF4();

  input = dht.readTemperature()-2; // Membaca suhu dari sensor
DHT22
  if (isnan(input)) {           // Antisipasi kegagalan di awal
pembacaan suhu
    input=setpoint;
    return;}
// Membaca Input: e
  g_fisInput[0] = error;
// Membaca Input: delta error
  delta_error=error-lastError;
  g_fisInput[1] = delta_error;

```

```

// Variabel untuk menyimpan nilai Output
g_fisOutput[0] = 0;
g_fisOutput[1] = 0;
g_fisOutput[2] = 0;
// Memproses FIS
fis_evaluate();
// Menentukan Output: kpp
Kpp= g_fisOutput[0];
// Menentukan Output: kdp
Kdp=g_fisOutput[1];
// Menentukan Output: alpha
alpha=g_fisOutput[2];
Kp=Kpmin+(Kpmax-Kpmin)*Kpp; // Menentukan Kp
Ki=(Kp*Kp)/(Kd*alpha); // Menentukan Ki
if(Ki>5){
    Ki=1.5;}
Kd=Kdmin+(Kdmax-Kdmin)*Kdp; // Menentukan Kd

// Komputasi Kontrol PID
output = computePID(input); // Menghitung PID
output=output*0.0001; // Gain=1
if(output>5){
    output=5;}
else if (output<-6){
    output=-6;}
double suhuAC = input+output;
suhuACint = (int) suhuAC; // Konversi Tipe Data ke Int
if(suhuACint>30){
    suhuACint=30;}
else if (suhuACint<18){
    suhuACint=18;}

// Daftar Instruksi Suhu AC
switch (suhuACint){

```

```
case 18:
    irsend.sendRaw(AC18, sizeof(AC18) / sizeof(AC18[0]), khz);
    delay(300);
    break;
case 19:
    irsend.sendRaw(AC19, sizeof(AC19) / sizeof(AC19[0]), khz);
    delay(300);
    break;
case 20:
    irsend.sendRaw(AC20, sizeof(AC20) / sizeof(AC20[0]), khz);
    delay(300);
    break;
case 21:
    irsend.sendRaw(AC21, sizeof(AC21) / sizeof(AC21[0]), khz);
    delay(300);
    break;
case 22:
    irsend.sendRaw(AC22, sizeof(AC22) / sizeof(AC22[0]), khz);
    delay(300);
    break;
case 23:
    irsend.sendRaw(AC23, sizeof(AC23) / sizeof(AC23[0]), khz);
    delay(300);
    break;
case 24:
    irsend.sendRaw(AC24, sizeof(AC24) / sizeof(AC24[0]), khz);
    delay(300);
    break;
case 25:
    irsend.sendRaw(AC25, sizeof(AC25) / sizeof(AC25[0]), khz);
    delay(300);
    break;
case 26:
    irsend.sendRaw(AC26, sizeof(AC26) / sizeof(AC26[0]), khz);
```

```

    delay(300);
    break;
case 27:
    irsend.sendRaw(AC27, sizeof(AC27) / sizeof(AC27[0]), khz);
    delay(300);
    break;
case 28:
    irsend.sendRaw(AC28, sizeof(AC28) / sizeof(AC28[0]), khz);
    delay(300);
    break;
case 29:
    irsend.sendRaw(AC29, sizeof(AC29) / sizeof(AC29[0]), khz);
    delay(300);
    break;
case 30:
    irsend.sendRaw(AC30, sizeof(AC30) / sizeof(AC30[0]), khz);
    delay(300);
    break;}

// menyimpan data ke PLX-DAQ
Write_PlxDaq();}

// fungsi untuk kontrol PID
double computePID(double input){
    currentTime = millis();
    elapsedTime = (double)(currentTime - previousTime);
    error = setpoint - input;
    cumError += error * elapsedTime;
    rateError = (error - lastError)/elapsedTime;
    double output = Kp*error + Ki*cumError + Kd*rateError;
    lastError = error;
    previousTime = currentTime;
    return output;}

```



```
// inialisasi input&output kontrol pencahayaan
```

```
void Init_cahaya(){
  pinMode(PIR_1, INPUT);
  pinMode(PIR_2, INPUT);
  pinMode(PIR_3, INPUT);
  pinMode(PIR_4, INPUT);
  pinMode(RELAY_1,OUTPUT);
  pinMode(RELAY_2,OUTPUT);
  pinMode(RELAY_3,OUTPUT);
  pinMode(RELAY_4,OUTPUT);}

```

```
//Lampu 1
```

```
void lampuON1(){
  digitalWrite(RELAY_1,HIGH);}
void lampuOFF1(){
  digitalWrite(RELAY_1,LOW);}
boolean deteksi1(){
  int sensorValue1 = digitalRead(PIR_1);
  if(sensorValue1 == LOW)//if the sensor value is HIGH?
    return true;//yes,return ture }
  else{
    return false;//no,return false } }

```

```
//Lampu 2
```

```
void lampuON2(){
  digitalWrite(RELAY_2,HIGH);}
void lampuOFF2(){
  digitalWrite(RELAY_2,LOW)}
boolean deteksi2(){
  int sensorValue2 = digitalRead(PIR_2);
  if(sensorValue2 == LOW)//if the sensor value is HIGH?
    return true;//yes,return ture }
  else{
    return false;//no,return false } }

```

```

//Lampu 3
void lampuON3(){
  digitalWrite(RELAY_3,HIGH);}
void lampuOFF3(){
  digitalWrite(RELAY_3,LOW);}
boolean deteksi3(){
  int sensorValue3 = digitalRead(PIR_3);
  if(sensorValue3 == LOW)//if the sensor value is HIGH?{
    return true;//yes,return ture }
  else{
    return false;//no,return false } }

//Lampu 4
void lampuON4(){
  digitalWrite(RELAY_4,HIGH);}
void lampuOFF4(){
  digitalWrite(RELAY_4,LOW);}
boolean deteksi4(){
  int sensorValue4 = digitalRead(PIR_4);
  if(sensorValue4 == LOW)//if the sensor value is HIGH?{
    return true;//yes,return ture }
  else{
    return false;//no,return false } }
// inialisasi PLX-DAQ
void Init_PlxDaq(){
  Serial.println("CLEARDATA");
  Serial.println("LABEL, Date, Time, Kp, Ki, Kd, Suhu Ruangan,
  Output FPID, Suhu AC");}

// menyimpan data ke PLX-DAQ
void Write_PlxDaq(){
  Serial.print("DATA");
  Serial.print(",");
  Serial.print("DATE");

```

```

Serial.print(",");
Serial.print("TIME");
Serial.print(",");
Serial.print(Kp);
Serial.print(",");
Serial.print(Ki);
Serial.print(",");
Serial.print(Kd);
Serial.print(",");
Serial.print(input);
Serial.print(",");
Serial.print(output);
Serial.print(",");
Serial.print(suhuACint);
Serial.print(",");
Serial.println();}

//*****
// Support functions for Fuzzy Inference System
//*****
// Trapezoidal Member Function
FIS_TYPE fis_trapmf(FIS_TYPE x, FIS_TYPE* p){
    FIS_TYPE a = p[0], b = p[1], c = p[2], d = p[3];
    FIS_TYPE t1 = ((x <= c) ? 1 : ((d < x) ? 0 : ((c != d) ? ((d - x) /
(d - c)) : 0)));
    FIS_TYPE t2 = ((b <= x) ? 1 : ((x < a) ? 0 : ((a != b) ? ((x - a) /
(b - a)) : 0)));
    return (FIS_TYPE) min(t1, t2);}

// Triangular Member Function
FIS_TYPE fis_trimf(FIS_TYPE x, FIS_TYPE* p){
    FIS_TYPE a = p[0], b = p[1], c = p[2];
    FIS_TYPE t1 = (x - a) / (b - a);
    FIS_TYPE t2 = (c - x) / (c - b);

```

```

if ((a == b) && (b == c)) return (FIS_TYPE) (x == a);
if (a == b) return (FIS_TYPE) (t2*(b <= x)*(x <= c));
if (b == c) return (FIS_TYPE) (t1*(a <= x)*(x <= b));
t1 = min(t1, t2);
return (FIS_TYPE) max(t1, 0);}

```

```
// Gaussian Member Function
```

```

FIS_TYPE fis_gaussmf(FIS_TYPE x, FIS_TYPE* p){
    FIS_TYPE s = p[0], c = p[1];
    FIS_TYPE t = (x - c) / s;
    return exp(-(t * t) / 2);}

```

```

FIS_TYPE fis_min(FIS_TYPE a, FIS_TYPE b){
    return min(a, b);}

```

```

FIS_TYPE fis_max(FIS_TYPE a, FIS_TYPE b){
    return max(a, b);}

```

```

FIS_TYPE fis_array_operation(FIS_TYPE *array, int size,
    _FIS_ARR_OP pfnOp){

```

```

    int i;
    FIS_TYPE ret = 0;

```

```

    if (size == 0) return ret;
    if (size == 1) return array[0];

```

```

    ret = array[0];
    for (i = 1; i < size; i++){
        ret = (*pfnOp)(ret, array[i]);}

```

```

    return ret;}

```

```

//*****
// Data for Fuzzy Inference System
//*****

```

```

// Pointers to the implementations of member functions
_FIS_MF fis_gMF[] = {
    fis_trapmf, fis_trimf, fis_gaussmf};

// Count of member function for each input
int fis_gIMFCount[] = { 7, 7 };

// Count of member function for each Output
int fis_gOMFCount[] = { 2, 2, 4 };

// Coefficients for the input Member Functions
FIS_TYPE fis_gMFI0Coeff1[] = { -1e+17, -1e+17, -0.9667, -0.7
};
FIS_TYPE fis_gMFI0Coeff2[] = { -1, -0.6667, -0.3333 };
FIS_TYPE fis_gMFI0Coeff3[] = { -0.6667, -0.3333, -5.551e-17 };
FIS_TYPE fis_gMFI0Coeff4[] = { -0.3333, 0, 0.3333 };
FIS_TYPE fis_gMFI0Coeff5[] = { -5.551e-17, 0.3333, 0.6667 };
FIS_TYPE fis_gMFI0Coeff6[] = { 0.3333, 0.6667, 1 };
FIS_TYPE fis_gMFI0Coeff7[] = { 0.7, 0.9667, 1e+17, 1e+17 };
FIS_TYPE* fis_gMFI0Coeff[] = { fis_gMFI0Coeff1,
    fis_gMFI0Coeff2, fis_gMFI0Coeff3, fis_gMFI0Coeff4,
    fis_gMFI0Coeff5, fis_gMFI0Coeff6, fis_gMFI0Coeff7 };
FIS_TYPE fis_gMFI1Coeff1[] = { -1e+17, -1e+17, -0.9667, -0.7
};
FIS_TYPE fis_gMFI1Coeff2[] = { -1, -0.6667, -0.3333 };
FIS_TYPE fis_gMFI1Coeff3[] = { -0.6667, -0.3333, -5.551e-17 };
FIS_TYPE fis_gMFI1Coeff4[] = { -0.3333, 0, 0.3333 };
FIS_TYPE fis_gMFI1Coeff5[] = { -5.551e-17, 0.3333, 0.6667 };
FIS_TYPE fis_gMFI1Coeff6[] = { 0.3333, 0.6667, 1 };
FIS_TYPE fis_gMFI1Coeff7[] = { 0.7, 0.9667, 1e+17, 1e+17 };
FIS_TYPE* fis_gMFI1Coeff[] = { fis_gMFI1Coeff1,
    fis_gMFI1Coeff2, fis_gMFI1Coeff3, fis_gMFI1Coeff4,
    fis_gMFI1Coeff5, fis_gMFI1Coeff6, fis_gMFI1Coeff7 };

```

```
FIS_TYPE**  fis_gMFICoeff[]  =  {  fis_gMFI0Coeff,
fis_gMFI1Coeff  };
```

```
// Coefficients for the Output Member Functions
```

```
FIS_TYPE fis_gMFO0Coeff1[] = { 0.4247, 0 };
```

```
FIS_TYPE fis_gMFO0Coeff2[] = { 0.4247, 1 };
```

```
FIS_TYPE*  fis_gMFO0Coeff[]  =  {  fis_gMFO0Coeff1,
fis_gMFO0Coeff2  };
```

```
FIS_TYPE fis_gMFO1Coeff1[] = { 0.4247, 0 };
```

```
FIS_TYPE fis_gMFO1Coeff2[] = { 0.4247, 1 };
```

```
FIS_TYPE*  fis_gMFO1Coeff[]  =  {  fis_gMFO1Coeff1,
fis_gMFO1Coeff2  };
```

```
FIS_TYPE fis_gMFO2Coeff1[] = { 1.95, 2, 2.05 };
```

```
FIS_TYPE fis_gMFO2Coeff2[] = { 2.95, 3, 3.05 };
```

```
FIS_TYPE fis_gMFO2Coeff3[] = { 3.95, 4, 4.05 };
```

```
FIS_TYPE fis_gMFO2Coeff4[] = { 4.95, 5, 5.05 };
```

```
FIS_TYPE*  fis_gMFO2Coeff[]  =  {  fis_gMFO2Coeff1,
fis_gMFO2Coeff2, fis_gMFO2Coeff3, fis_gMFO2Coeff4  };
```

```
FIS_TYPE**  fis_gMFOCoeff[]  =  {  fis_gMFO0Coeff,
fis_gMFO1Coeff, fis_gMFO2Coeff  };
```

```
// input membership function set
```

```
int fis_gMFI0[] = { 0, 1, 1, 1, 1, 1, 0 };
```

```
int fis_gMFI1[] = { 0, 1, 1, 1, 1, 1, 0 };
```

```
int* fis_gMFI[] = { fis_gMFI0, fis_gMFI1 };
```

```
// Output membership function set
```

```
int fis_gMFO0[] = { 2, 2 };
```

```
int fis_gMFO1[] = { 2, 2 };
```

```
int fis_gMFO2[] = { 1, 1, 1, 1 };
```

```
int* fis_gMFO[] = { fis_gMFO0, fis_gMFO1, fis_gMFO2 };
```

```
// Rule Weights
```



```
int fis_gRI24[] = { 4, 4 };
int fis_gRI25[] = { 4, 5 };
int fis_gRI26[] = { 4, 6 };
int fis_gRI27[] = { 4, 7 };
int fis_gRI28[] = { 5, 1 };
int fis_gRI29[] = { 5, 2 };
int fis_gRI30[] = { 5, 3 };
int fis_gRI31[] = { 5, 4 };
int fis_gRI32[] = { 5, 5 };
int fis_gRI33[] = { 5, 6 };
int fis_gRI34[] = { 5, 7 };
int fis_gRI35[] = { 6, 1 };
int fis_gRI36[] = { 6, 2 };
int fis_gRI37[] = { 6, 3 };
int fis_gRI38[] = { 6, 4 };
int fis_gRI39[] = { 6, 5 };
int fis_gRI40[] = { 6, 6 };
int fis_gRI41[] = { 6, 7 };
int fis_gRI42[] = { 7, 1 };
int fis_gRI43[] = { 7, 2 };
int fis_gRI44[] = { 7, 3 };
int fis_gRI45[] = { 7, 4 };
int fis_gRI46[] = { 7, 5 };
int fis_gRI47[] = { 7, 6 };
int fis_gRI48[] = { 7, 7 };
int* fis_gRI[] = { fis_gRI0, fis_gRI1, fis_gRI2, fis_gRI3, fis_gRI4,
fis_gRI5, fis_gRI6, fis_gRI7, fis_gRI8, fis_gRI9, fis_gRI10,
fis_gRI11, fis_gRI12, fis_gRI13, fis_gRI14, fis_gRI15, fis_gRI16,
fis_gRI17, fis_gRI18, fis_gRI19, fis_gRI20, fis_gRI21, fis_gRI22,
fis_gRI23, fis_gRI24, fis_gRI25, fis_gRI26, fis_gRI27, fis_gRI28,
fis_gRI29, fis_gRI30, fis_gRI31, fis_gRI32, fis_gRI33, fis_gRI34,
fis_gRI35, fis_gRI36, fis_gRI37, fis_gRI38, fis_gRI39, fis_gRI40,
fis_gRI41, fis_gRI42, fis_gRI43, fis_gRI44, fis_gRI45, fis_gRI46,
fis_gRI47, fis_gRI48 };
```



```
// Rule Outputs
int fis_gRO0[] = { 2, 1, 1 };
int fis_gRO1[] = { 2, 1, 1 };
int fis_gRO2[] = { 2, 1, 1 };
int fis_gRO3[] = { 2, 1, 1 };
int fis_gRO4[] = { 2, 1, 1 };
int fis_gRO5[] = { 2, 1, 1 };
int fis_gRO6[] = { 2, 1, 1 };
int fis_gRO7[] = { 1, 2, 2 };
int fis_gRO8[] = { 2, 2, 2 };
int fis_gRO9[] = { 2, 1, 1 };
int fis_gRO10[] = { 2, 1, 1 };
int fis_gRO11[] = { 2, 1, 1 };
int fis_gRO12[] = { 2, 2, 2 };
int fis_gRO13[] = { 2, 2, 2 };
int fis_gRO14[] = { 1, 2, 3 };
int fis_gRO15[] = { 1, 2, 2 };
int fis_gRO16[] = { 2, 2, 2 };
int fis_gRO17[] = { 2, 1, 1 };
int fis_gRO18[] = { 2, 2, 2 };
int fis_gRO19[] = { 1, 2, 2 };
int fis_gRO20[] = { 1, 2, 3 };
int fis_gRO21[] = { 1, 2, 4 };
int fis_gRO22[] = { 1, 2, 3 };
int fis_gRO23[] = { 1, 2, 2 };
int fis_gRO24[] = { 2, 2, 2 };
int fis_gRO25[] = { 1, 2, 2 };
int fis_gRO26[] = { 1, 2, 3 };
int fis_gRO27[] = { 1, 2, 4 };
int fis_gRO28[] = { 1, 2, 3 };
int fis_gRO29[] = { 1, 2, 2 };
int fis_gRO30[] = { 2, 2, 2 };
int fis_gRO31[] = { 2, 1, 1 };
int fis_gRO32[] = { 2, 2, 2 };
```

```
int fis_gRO33[] = { 1, 2, 2 };
int fis_gRO34[] = { 1, 2, 3 };
int fis_gRO35[] = { 1, 2, 2 };
int fis_gRO36[] = { 2, 2, 2 };
int fis_gRO37[] = { 2, 1, 1 };
int fis_gRO38[] = { 2, 1, 1 };
int fis_gRO39[] = { 2, 1, 1 };
int fis_gRO40[] = { 2, 2, 2 };
int fis_gRO41[] = { 1, 2, 2 };
int fis_gRO42[] = { 2, 1, 1 };
int fis_gRO43[] = { 2, 1, 1 };
int fis_gRO44[] = { 2, 1, 1 };
int fis_gRO45[] = { 2, 1, 1 };
int fis_gRO46[] = { 2, 1, 1 };
int fis_gRO47[] = { 2, 1, 1 };
int fis_gRO48[] = { 2, 1, 1 };
int* fis_gRO[] = { fis_gRO0, fis_gRO1, fis_gRO2, fis_gRO3,
fis_gRO4, fis_gRO5, fis_gRO6, fis_gRO7, fis_gRO8, fis_gRO9,
fis_gRO10, fis_gRO11, fis_gRO12, fis_gRO13, fis_gRO14,
fis_gRO15, fis_gRO16, fis_gRO17, fis_gRO18, fis_gRO19,
fis_gRO20, fis_gRO21, fis_gRO22, fis_gRO23, fis_gRO24,
fis_gRO25, fis_gRO26, fis_gRO27, fis_gRO28, fis_gRO29,
fis_gRO30, fis_gRO31, fis_gRO32, fis_gRO33, fis_gRO34,
fis_gRO35, fis_gRO36, fis_gRO37, fis_gRO38, fis_gRO39,
fis_gRO40, fis_gRO41, fis_gRO42, fis_gRO43, fis_gRO44,
fis_gRO45, fis_gRO46, fis_gRO47, fis_gRO48 };
```

```
// input range Min
```

```
FIS_TYPE fis_gIMin[] = { -1, -1 };
```

```
// input range Max
```

```
FIS_TYPE fis_gIMax[] = { 1, 1 };
```

```
// Output range Min
```

```

FIS_TYPE fis_gOMin[] = { 0, 0, 1 };

// Output range Max
FIS_TYPE fis_gOMax[] = { 1, 1, 6 };

//*****
// Data dependent support functions for Fuzzy Inference System
//*****
FIS_TYPE fis_MF_out(FIS_TYPE** fuzzyRuleSet, FIS_TYPE x,
int o){
    FIS_TYPE mfOut;
    int r;

    for (r = 0; r < fis_gcR; ++r){
        int index = fis_gRO[r][o];
        if (index > 0){
            index = index - 1;
            mfOut = (fis_gMF[fis_gMFO[o][index]])(x,
fis_gMFOCoeff[o][index]);
        }
        else if (index < 0){
            index = -index - 1;
            mfOut = 1 - (fis_gMF[fis_gMFO[o][index]])(x,
fis_gMFOCoeff[o][index]);
        }
        else{
            mfOut = 0;
        }

        fuzzyRuleSet[0][r] = fis_min(mfOut, fuzzyRuleSet[1][r]);
    }
    return fis_array_operation(fuzzyRuleSet[0], fis_gcR, fis_max);
}

FIS_TYPE fis_defuzz_centroid(FIS_TYPE** fuzzyRuleSet, int o)
{
    FIS_TYPE step = (fis_gOMax[o] - fis_gOMin[o]) /
(FIS_RESOLUTION - 1);

```

```

FIS_TYPE area = 0;
FIS_TYPE momentum = 0;
FIS_TYPE dist, slice;
int i;

// calculate the area under the curve formed by the MF outputs
for (i = 0; i < FIS_RESOLUTION; ++i){
    dist = fis_gOMin[o] + (step * i);
    slice = step * fis_MF_out(fuzzyRuleSet, dist, o);
    area += slice;
    momentum += slice*dist;}

return ((area == 0) ? ((fis_gOMax[o] + fis_gOMin[o]) / 2) :
(momentum / area));}

/*****
// Fuzzy Inference System
*****/
void fis_evaluate(){
    FIS_TYPE fuzzyInput0[] = { 0, 0, 0, 0, 0, 0, 0 };
    FIS_TYPE fuzzyInput1[] = { 0, 0, 0, 0, 0, 0, 0 };
    FIS_TYPE* fuzzyInput[fis_gcI] = { fuzzyInput0, fuzzyInput1,
};
    FIS_TYPE fuzzyOutput0[] = { 0, 0 };
    FIS_TYPE fuzzyOutput1[] = { 0, 0 };
    FIS_TYPE fuzzyOutput2[] = { 0, 0, 0, 0 };
    FIS_TYPE* fuzzyOutput[fis_gcO] = { fuzzyOutput0,
fuzzyOutput1, fuzzyOutput2, };
    FIS_TYPE fuzzyRules[fis_gcR] = { 0 };
    FIS_TYPE fuzzyFires[fis_gcR] = { 0 };
    FIS_TYPE* fuzzyRuleSet[] = { fuzzyRules, fuzzyFires };
    FIS_TYPE sW = 0;

// Transforming input to fuzzy Input

```

```

int i, j, r, o;
for (i = 0; i < fis_gcI; ++i){
    for (j = 0; j < fis_gIMFCount[i]; ++j){
        fuzzyInput[i][j] =
            (fis_gMF[fis_gMFI[i][j]])(g_fisInput[i],
fis_gMFICoeff[i][j]);} }

int index = 0;
for (r = 0; r < fis_gcR; ++r){
    if (fis_gRType[r] == 1){
        fuzzyFires[r] = FIS_MAX;
        for (i = 0; i < fis_gcI; ++i){
            index = fis_gRI[r][i];
            if (index > 0)
                fuzzyFires[r] = fis_min(fuzzyFires[r],
fuzzyInput[i][index - 1]);
            else if (index < 0)
                fuzzyFires[r] = fis_min(fuzzyFires[r], 1 -
fuzzyInput[i][-index - 1]);
            else
                fuzzyFires[r] = fis_min(fuzzyFires[r], 1);} }
    else{
        fuzzyFires[r] = FIS_MIN;
        for (i = 0; i < fis_gcI; ++i){
            index = fis_gRI[r][i];
            if (index > 0)
                fuzzyFires[r] = fis_max(fuzzyFires[r],
fuzzyInput[i][index - 1]);
            else if (index < 0)
                fuzzyFires[r] = fis_max(fuzzyFires[r], 1 -
fuzzyInput[i][-index - 1]);
            else
                fuzzyFires[r] = fis_max(fuzzyFires[r], 0);} }

```

```
fuzzyFires[r] = fis_gRWeight[r] * fuzzyFires[r];  
sW += fuzzyFires[r];}  
  
if (sW == 0){  
    for (o = 0; o < fis_gcO; ++o){  
        g_fisOutput[o] = ((fis_gOMax[o] + fis_gOMin[o]) / 2);}  
    else{  
        for (o = 0; o < fis_gcO; ++o){  
            g_fisOutput[o] = fis_defuzz_centroid(fuzzyRuleSet, o);}}}
```

BIODATA PENULIS



Penulis bernama lengkap Ardyas Nur Afa atau biasa dipanggil Ardyas, dilahirkan di Surabaya, 19 Juni 1998. Penulis merupakan anak pertama dari tiga bersaudara. Penulis telah menyelesaikan pendidikan di MIN Malang 1 pada tahun 2010, pendidikan di SMPN 1 Malang pada tahun 2012, pendidikan di SMAN 10 Malang pada tahun 2015 dan sedang menempuh pendidikan

Sarjana Strata-1 di Departemen Teknik Fisika ITS. Penulis memiliki pengalaman program *internship* selama satu bulan di PT. Tirta Investama (AQUA) Pandaan, Jawa Timur. Selama menjadi mahasiswa, penulis aktif di kegiatan mahasiswa seperti menjadi Koordinator Rayon Malang Raya *Engineering Physics Competition* (2016), Kepala Divisi Penelitian dan Pengembangan JMMI ITS (2017-2018), Kepala Badan Kaderisasi JMMI ITS (2018), penerima beasiswa mahasiswa berprestasi Rumah Kepemimpinan (2016-2018), sukarelawan gerakan mengajar Laskar Pahlawan (2016-2018), dan menjadi kontributor di Selasar.com. Pada bulan Desember 2019 penulis telah menyelesaikan Tugas Akhir dengan judul **Perancangan *Smart Room System* dengan Indikator Kenyamanan Suhu dan Pencahayaan Berbasis Kontrol Logika *Fuzzy-PID* (KLF-PID)**. Bagi pembaca yang memiliki kritik, saran, atau ingin berdiskusi lebih lanjut mengenai tugas akhir ini, maka dapat menghubungi melalui email: *ardyasnuraufa@gmail.com*

“Halaman ini sengaja dikosongkan”