

**RANCANG BANGUN KONTROL DAN MONITORING
KUALITAS UDARA DI LABORATORIUM *MACHINARY*
SEBAGAI PENUNJANG PRAKTEK TARUNA BERBASIS IOT
DENGAN METODE *FUZZY LOGIC***

PROYEK AKHIR



Oleh :

YOGA YANUAR NUGROHO
NIT : 30121047

**PROGRAM STUDI DIPLOMA 3 TEKNIK LISTRIK BANDARA
POLITEKNIK PENERBANGAN SURABAYA**

2024

**RANCANG BANGUN KONTROL DAN MONITORING
KUALITAS UDARA DI LABORATORIUM *MACHINARY*
SEBAGAI PENUNJANG PRAKTEK TARUNA BERBASIS IOT
DENGAN METODE *FUZZY LOGIC***

PROYEK AKHIR

Diajukan sebagai Salah Satu Syarat untuk Mendapatkan Gelar Ahli Madya
(A.Md.) pada Program Studi Diploma 3 Teknik Listrik Bandar Udara



Oleh :

YOGA YANUAR NUGROHO
NIT : 30121047

**PROGRAM STUDI DIPLOMA 3 TEKNIK LISTRIK BANDARA
POLITEKNIK PENERBANGAN SURABAYA**

2024

LEMBAR PERSETUJUAN

**RANCANG BANGUN KONTROL DAN MONITORING KUALITAS
UDARA DI LABORATORIUM MACHINERY SEBAGAI PENUNJANG
PRAKTEK TARUNA BERBASIS IOT DENGAN
METODE *FUZZY LOGIC***

Oleh :

YOGA YANUAR NUGROHO
NIT : 30121047

Disetujui untuk diujikan pada :
Surabaya, 7 Agustus 2024

Pembimbing I : **RIFDIAN I.S., S.T, M.M, M.T.**
NIP. 19810629 200912 1 002

Pembimbing II : **Dr. LAILA ROCHMAWATI, SS, M. Pd**
NIP. 19810723 200502 2 001

LEMBAR PENGESAHAN

**RANCANG BANGUN KONTROL DAN MONITORING KUALITAS
UDARA DI LABORATORIUM MACHINARY SEBAGAI PENUNJANG
PRAKTEK TARUNA BERBASIS IOT DENGAN
METODE *FUZZY LOGIC***

Oleh :

YOGA YANUAR NUGROHO
NIT : 30121047

Telah dipertahankan dan dinyatakan lulus pada Ujian Proyek Akhir
Program Pendidikan Diploma 3 Teknik Listrik Bandara
Politeknik Penerbangan Surabaya
pada tanggal : 7 Agustus 2024

Panitia Penguji :

Ketua : **Dr. SLAMET HARIYADI, ST, MM**
NIP. 19630408 198902 1 001

Sekretaris : **Dr. LAILA ROCHMAWATI, SS, M.Pd**
NIP. 19810723 200502 2 001

Anggota : **RIFDIAN I.S, ST, MM, MT**
NIP. 19810629 200912 1 002

Mengetahui,
Ketua Program Studi
D3 Teknik Listrik Bandara



Dr. GUNAWAN SAKTI, S.T., M.T.
NIP. 19881001 200912 1 003

HALAMAN PERSEMBAHAN

MOTTO

**“TIDAK ADA ORANG GAGAL DI DUNIA INI, ORANG YANG GAGAL
IALAH ORANG YANG TIDAK MAU MENCOBA SAMA SEKALI”**

PERSEMBAHAN :

Allah SWT yang telah melimpahkan Rahmat dan HidayahNya.
Bapak Agus Priyanto dan Ibu Hariyati Orang Tua terhebat dan terkuat yang selalu
memberikan doa dan semangat untuk kesuksesan Putera nya.
Pihak-pihak yang telah membantu dan mendukung dalam penyusunan Proyek
Akhir ini.
Teman-teman seperjuangan yang penuh dengan cerita.
Terimakasih semua

PERNYATAAN KEASLIAN DAN HAK CIPTA

Saya yang bertanda tangan di bawah ini:

Nama : YOGA YANUAR NUGROHO

NIT : 30121047

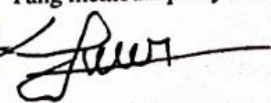
Program Studi : D-III Teknik Listrik Bandar Udara

Judul Proyek Akhir : Rancang Bangun Kontrol Dan Monitoring Kualitas Udara Di Laboratorium *Machinery* Sebagai Penunjang Praktek Taruna Berbasis Iot Dengan Metode *Fuzzy Logic*

dengan ini menyatakan bahwa :

1. Proyek Akhir ini merupakan karya asli dan belum pernah diajukan untuk mendapatkan gelar akademik, baik di Politeknik Penerbangan Surabaya maupun di Perguruan Tinggi lain, serta dipublikasikan, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
2. Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan Hak Bebas Royalti Non Eksklusif (*Non-Exclusive Royalty-Free Right*) kepada Politeknik Penerbangan Surabaya beserta perangkat yang ada (jika diperlukan). Dengan hak ini, Politeknik Penerbangan Surabaya berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan proyek akhir saya dengan tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya. Apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Politeknik Penerbangan Surabaya.

Surabaya, 7 Agustus 2024
Yang membuat pernyataan

YOGA YANUAR NUGROHO



ABSTRAK

RANCANG BANGUN KONTROL DAN MONITORING KUALITAS UDARA DI LABORATORIUM *MACHINERY* SEBAGAI PENUNJANG PRAKTEK TARUNA BERBASIS IOT DENGAN METODE *FUZZY LOGIC*

Oleh :
Yoga Yanuar Nugroho
NIT. 30121047

Pada saat kualitas udara buruk di sebuah Laboratorium *Machinery* dapat disebabkan oleh beberapa faktor, diantaranya adalah saat praktek pengelasan, praktek bubut atau frais, praktek AC dan praktek lainnya yang berdampak pada kualitas udara serta kelalaian manusia (*human error*) dalam menggunakan peralatan elektronik yang berbahaya. Dalam upaya untuk meminimalisir kejadian tersebut maka dalam Proyek Akhir ini berfokus pada pembuatan protoipe monitoring dan kontrol kualitas udara dengan menggunakan empat sensor yang berfungsi sebagai pendeteksi gas amonia, karbondioksida (CO), debu atau partikel – partikel halus, suhu dan kelembapan pada ruangan serta juga pendeteksi adanya kadar gas ataupun gumpalan asap pada ruangan.

Sistem pada prototipe ini menggunakan empat sensor yaitu sensor gas amonia yaitu MQ135 , sensor gas CO yaitu MQ7, sensor debu yaitu Dust dan sensor DHT 11 yaitu suhu berguna untuk memonitoring suhu dan kelembapan ruangan jika terdapat kenaikan. Keempat sensor tersebut berguna untuk memonitoring adanya kadar gas yang meningkat pada suatu ruangan. Sistem ini menggunakan ESP32 sebagai mikrokontroler sehingga data dari kedua sensor tersebut diolah dan dikirim melalui kabel data dan divisualisasikan melalui aplikasi Matlab. Data yang dikirim dan di tampilkan pada aplikasi Ubidots berupa grafik dan LCD Nextion secara realtime. Ketika grafik yang muncul menunjukkan adanya kenaikan suhu dan kadar gas melebihi batas yang telah ditentukan, maka pada prototipe ini seperti blower atau *fan* bekerja secara otomatis.

Hasil dari sistem monitoring dan kontrol ini diharapkan dapat meminimalisir terjadinya kualitas udara yang buruk dan juga meminimalisir kerugian kesehatan bagi taruna. Dengan aplikasi Ubidots maka data yang muncul pada aplikasi berupa data realtime sehingga informasi kualitas udara dapat diketahui secara cepat dan penanganan akan lebih mudah diatasi.

Kata kunci : ESP32, Matlab, MQ135 , MQ7, Dust, DHT 11

ABSTRACT

DESIGN AND DEVELOPMENT OF AIR QUALITY CONTROL AND MONITORING IN MACHINERY LABORATORY TO SUPPORT CADET PRACTICE BASED ON IOT WITH FUZZY LOGIC METHOD

By :

Yoga Yanuar Nugroho
NIT. 30121047

When air quality in a Machinery Laboratory is poor, it can be caused by several factors, including welding practices, lathe or milling practices, AC practices, and other activities that impact air quality, as well as human error in using hazardous electronic equipment. To minimize these occurrences, this Final Project focuses on creating a prototype for monitoring and controlling air quality using four sensors that detect ammonia gas, carbon dioxide (CO), dust or fine particles, temperature, and humidity in the room, as well as detecting gas levels or smoke in the room.

The system in this prototype uses four sensors: the ammonia gas sensor (MQ135), the CO gas sensor (MQ7), the dust sensor (Dust), and the temperature and humidity sensor (DHT 11). These four sensors are useful for monitoring increased gas levels in a room. The system uses an ESP32 microcontroller to process the data from the sensors, which is then transmitted via data cable and visualized through the Matlab application. The data sent is displayed on the Ubidots application in the form of real-time graphs and on the LCD Nextion. When the graph shows an increase in temperature and gas levels exceeding the set limits, components like blowers or fans in the prototype work automatically.

The results of this monitoring and control system are expected to minimize the occurrence of poor air quality and reduce health risks for the cadets. With the Ubidots application, the data appears in real-time, allowing air quality information to be quickly known and managed more easily.

Key word : ESP32, Matlab, MQ135 , MQ7, Dust, DHT 11

KATA PENGANTAR

Segala puji bagi Allah SWT, karena berkat limpahan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Proyek Akhir yang berjudul “Rancang Bangun Kontrol Dan Monitoring Kualitas Udara Di Laboratorium *Machinery* Sebagai Penunjang Praktek Taruna Berbasis Iot Dengan Metode *Fuzzy Logic*” ini dapat terselesaikan sesuai waktu yang telah ditentukan. Penyusunan Proyek Akhir ini diajukan guna memenuhi syarat untuk menyelesaikan pendidikan Program Studi Diploma 3 Teknik Listrik Bandara di Politeknik Penerbangan Surabaya.

Penulis mengucapkan terima kasih kepada segenap pihak yang telah membantu selama proses penyusunan Proyek Akhir ini, lebih khusus kepada :

1. Tuhan Yang Maha Esa.
2. Untuk orang tua, Mas, dan Adik yang telah memberikan doa dan semangat kepada penulis.
3. Bapak Ahmad Bahrawi, S.E., M.T. selaku Direktur Politeknik Penerbangan Surabaya.
4. Bapak Dr. Gunawan Sakti, S.T., M.T. selaku Ketua Program Studi Teknik Listrik Bandar Udara di Politeknik Penerbangan.
5. Bapak Rifdian Indrianto Sudjoko, S.T., M.M., M.T. selaku dosen pembimbing I yang telah memberikan ilmu, pemahaman, dan membimbing penulis selama penyusunan Proyek Akhir ini.
6. Mam Laila Rochmawati, SS, M.Pd selaku dosen pembimbing II yang telah memberikan ilmu, pemahaman, dan membimbing penulis selama penyusunan Proyek Akhir ini.
7. Seluruh dosen dan *civitas* akademi Prodi Diploma 3 Teknik Listrik Bandara Politeknik Penerbangan Surabaya yang telah memberikan banyak ilmu kepada penulis.
8. Rekan – rekan TLB XVI, senior alumni TLB XIV dan XV, serta adik – adik TLB XVII yang telah membantu dan memberikan semangat.

Penulis menyadari bahwa masih banyak kekurangan dalam pembuatan Proyek Akhir ini. Oleh karena itu, perlunya kritik dan saran yang bersifat membangun diharapkan dapat menyempurnakan rancangan ini kedepannya.

Penulis berharap penulisan Proyek Akhir ini dapat bermanfaat bagi penulis khususnya, dan bagi pembaca.

Surabaya, 7 Agustus 2024



Yoga Yanuar Nugroho



DAFTAR LAMPIRAN

Lampiran A Standard Operational Procedure (SOP)	A-1
Lampiran B Dokumentasi Alat	B-1
Lampiran C Coding Alat.....	C-2
Lampiran D Daftar Riwayat Hidup.....	D-1



DAFTAR ISI

LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
HALAMAN PERSEMBAHAN.....	iv
PERNYATAAN KEASLIAN DAN HAK CIPTA.....	v
ABSTRAK	vi
ABSTRACT	vii
KATA PENGANTAR.....	viii
DAFTAR LAMPIRAN	x
DAFTAR ISI.....	xi
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL.....	xv
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	4
1.6 Sistematika Penulisan.....	5
BAB 2 LANDASAN TEORI	6
2.1 Rancang Bangun Kontrol dan Monitoring.....	6
2.1.1 Sensor MQ 135	6
2.1.2 Sensor Dust	7
2.1.3 Sensor MQ 7	9
2.1.4 Sensor DHT 11.....	11
2.1.5 NodeMCU ESP32	12
2.1.6 Fan DC 12V DC	14
2.1.7 Liquid Crystal Display (LCD) NEXTION.....	16
2.1.8 Adaptor 12 volt	17
2.1.9 Buck Converter LM 2596.....	18
2.1.10 Driver Motor L298N.....	19
2.1.11 ADS 1115.....	20
2.1.12 Arduino IDE.....	21
2.1.13 Ubidots	21
2.2 Kualitas Udara.....	22
2.3 Laboratorium Machinery	23
2.4 Praktek Taruna	23
2.5 Internet Of Things	24
2.6 Metode Fuzzy Logic	24
2.7 Kajian Penelitian Terdahulu yang Relevan.....	26
BAB 3 METODE PENELITIAN	29
3.1 Desain Penelitian.....	29
3.2 Perancangan Alat	31
3.2.1 Desain Alat.....	31
3.2.2 Cara Kerja Alat	32

3.2.3	Komponen Perangkat Keras.....	34
3.2.4	Komponen Perangkat Lunak (<i>Software</i>).....	36
3.3	Teknik Pengujian	37
3.2.1	Pengujian Kinerja Sensor sensor MQ 135, Sensor Dust, sensor MQ7, sensor DHT 11	37
3.3.2	Pengujian <i>Fan</i>	38
3.3.3	Pengujian Kinerja ESP 32.....	38
3.3.4	Pengujian Tampilan Data ke LCD	38
3.3.5	Pengujian Pengiriman Data.....	38
3.3.6	Pengujian Website Ubidots	38
3.3.7	Pengujian Integrasi.....	38
3.4.8	Teknik Analisis Data.....	38
3.3	Tempat dan Waktu Pelaksanaan	39
BAB 4	HASIL DAN PEMBAHASAN	41
4.1	Hasil Penelitian	41
4.1.1	Potensi dan Masalah.....	41
4.1.2	Pengumpulan Data	41
4.1.3	Desain Produk	42
4.1.4	Validasi Desain	44
4.1.5	Revisi Desain	47
4.1.6	Uji Coba Produk.....	47
4.1.7	Revisi Produk	80
4.1.8	Validasi Desain	80
4.1.9	Revisi Produk Final.....	83
4.1.10	Produksi Terbatas.....	83
4.2.1	Kelebihan Alat	84
4.2.2	Kekurangan alat	84
BAB 5	PENUTUP.....	85
5.1	Kesimpulan	85
5.2	Saran.....	85
	DAFTAR PUSTAKA.....	86
	LAMPIRAN.....	A-1

DAFTAR GAMBAR

Gambar 2.1 Sensor MQ 135.....	6
Gambar 2.2 Sensor Dust	7
Gambar 2.3 Prinsip kerja sensor Dust.....	8
Gambar 2.4 Keterangan pin sensor Dust.....	8
Gambar 2.5 Sensor MQ 7.....	9
Gambar 2.6 Sensor DHT 11.....	11
Gambar 2.7 Mapping Pin Nodemcu Dekvit V3 Lolin	12
Gambar 2.8 Node MCU ESP32	13
Gambar 2.9 <i>Fan</i> DC.....	14
Gambar 2.10 LCD NEXTION	16
Gambar 2.11 Adaptor 12 Volt.....	17
Gambar 2.12 Buck Converter LM 2596	18
Gambar 2.13 Driver Motor L298N	19
Gambar 2.14 ADS 5111.....	20
Gambar 2.15 Arduino IDE.....	21
Gambar 2.16 Ubidots	21
Gambar 3.1 Langkah-langkah Penelitian.....	29
Gambar 3.2 Desain Blok Diagram	31
Gambar 3.3 <i>Flowchart</i> Sistem Alat	32
Gambar 4.1 Desain Produk Alat	42
Gambar 4.2 Desain Rangkaian PCB Sistem Perangkat Keras	43
Gambar 4.3 Rangkaian Komponen Pada PCB.....	44
Gambar 4.4 Revisi Desain Alat.....	47
Gambar 4.5 Mengupload Program.....	48
Gambar 4.6 Proses <i>Compling</i> Program.....	49
Gambar 4.7 Done <i>Compling</i> Program	49
Gambar 4.8 Tampilan Awal MATLAB.....	50
Gambar 4.9 Tampilan Design Fuzzy Logic	51
Gambar 4.10 <i>Membership Fuction</i> Sensor MQ135 (amonia).....	51
Gambar 4.11 <i>Membership Fuction</i> Sensor MQ7 (CO).....	52
Gambar 4.12 <i>Membership Fuction</i> Sensor Dust (Debu).....	53
Gambar 4.13 <i>Membership Fuction</i> Sensor DHT11 (Suhu)	53
Gambar 4.14 <i>Membership Fuction</i> Sensor DHT11 (Kelembapan)	54
Gambar 4.15 <i>Membership Fuction</i> Kualitas Udara	55
Gambar 4.16 <i>Rule Editor</i>	57
Gambar 4.17 <i>Rule Viewer</i>	58
Gambar 4.18 Tampilan Awal Ubidots	59
Gambar 4.19 Tampilan Ubidots Pada Saat Sensor Bekerja.....	59
Gambar 4.20 Hasil Pengukuran Adaptor 12 VDC.....	61
Gambar 4.21 Pengujian ESP32.....	62
Gambar 4.22 Hasil Pengukuran <i>Buck Converter</i> LM2596	63
Gambar 4.23 Hasil Pengukuran <i>Driver Motor</i> Kondisi Normal.....	64
Gambar 4.24 Hasil Pengukuran <i>Driver Motor</i> Kondisi Sedang	65
Gambar 4.25 Hasil Pengukuran <i>Driver Motor</i> L298N Kondisi Bahaya.....	65

Gambar 4.26 Penguploadan Tampilan LCD Nextion	66
Gambar 4.27 Tampilan LCD Nextion Ketika Semua Sensor Aktif.....	67
Gambar 4.28 Pengujian Sensor MQ 135 Dengan Amonia	68
Gambar 4.29 Pengujian Cairan Amonia Menggunakan NH3 Tester.....	69
Gambar 4.30 Grafik Hasil Pengukuran Sensor MQ135.....	69
Gambar 4.31 Pengujian Gas CO Dengan Asap Dari Bus, Mobil, Dan Genset.....	70
Gambar 4.32 Grafik Hasil Pengukuran Sensor MQ7 (CO)	71
Gambar 4.33 Grafik Hasil Pengukuran Sensor Dust (debu)	73
Gambar 4.34 Pengujian Suhu Menggunakan Alat Ukur.....	74
Gambar 4.35 Grafik Hasil Pengukuran Sensor DHT11 (suhu).....	76
Gambar 4.36 Pengukuran Sensor DHT 11 (Kelembapan).....	77
Gambar 4.37 Grafik Hasil Pengukuran Sensor DHT11 (Kelembapan).....	78
Gambar 4.38 Menguji Kecepatan <i>Fan</i> dengan Tachometer	79
Gambar 4.39 <i>Mock Up</i> Alat	83



DAFTAR TABEL

Tabel 2.1 Spesifikasi Sensor MQ 135.....	7
Tabel 2.2 Keterangan pin sensor Dust	9
Tabel 2.3 Spesifikasi Sensor Dust.....	9
Tabel 2.4 Spesifikasi Sensor MQ 7.....	10
Tabel 2.5 Spesifikasi Sensor DHT 11	11
Tabel 2.6 Spesifikasi <i>Fan</i> 12v DC	15
Tabel 2.7 Keterangan pin LCD Nextion 2.8 inch	16
Tabel 2.8 Spesifikasi LCD NEXTION	17
Tabel 2.9 Spesifikasi Adaptor 12V DC.....	18
Tabel 2.10 Spesifikasi Buck Converter LM 2596.....	19
Tabel 2.11 Spesifikasi Driver Motor L298N	20
Tabel 2.12 Ambang Batas Kualitas Udara	23
Tabel 2.13 Kajian Penelitian Terdahulu.....	26
Tabel 3.1 Tabel Waktu Penelitian 2024.....	40
Tabel 4.1 Ambang Batas Kualitas Udara Menurut KLKH	42
Tabel 4.2 Validasi Rancangan Alat.....	45
Tabel 4.3 Range Membership Fuction Sensor MQ135 (amonia)	51
Tabel 4.4 Range Membership Fuction Sensor MQ7 (CO)	52
Tabel 4.5 Range Membership Fuction Sensor Dust (Debu)	52
Tabel 4.6 Range Membership Fuction Sensor DHT11 (Suhu)	53
Tabel 4.7 Range Membership Fuction Sensor DHT11 (Kelembapan).....	54
Tabel 4.8 Tabel Hasil Pengujian Adaptor 12 VDC.....	61
Tabel 4.9 Hasil Pengujian Buck Converter LM2596.....	63
Tabel 4.10 Hasil Pengujian Driver Motor Kondisi Normal.....	64
Tabel 4.11 Hasil Pengujian Driver Motor Kondisi Sedang	65
Tabel 4.12 Hasil Pengujian Driver Motor L298N kondisi Bahaya.....	66
Tabel 4.13 Pengujian Sensor MQ135 Dengan Cairan Amonia	68
Tabel 4.14 Pengujian Sensor MQ7 Dengan Gas Buang Dari Bus, Mobil, Dan Genset.....	70
Tabel 4.15 Pengujian Partikel Debu Dengan Beberapa Benda.....	72
Tabel 4.16 Pengujian Sensor Dust Dengan Memasukkan Beberapa Jenis Benda	72
Tabel 4.17 Pengujian Sensor Suhu (DHT11).....	75
Tabel 4.18 Pengujian Sensor Kelembapan (DHT11).....	77
Tabel 4.19 Pengujian Kinerja <i>Fan</i> 12 VDC.....	79
Tabel 4.20 Validasi Rancangan Alat.....	81

DAFTAR PUSTAKA

- (KLHK), K. L. (2021, Maret 5). Diambil kembali dari www.menlhk.go.id.
- Datasheets, N. P. (2021, 9 10). Diambil kembali dari <https://nextion.tech/datasheets/>
- Evert Nebath, D. P. (2014). Rancang Bangun Alat Pengukur Gas Berbahaya CO Dan CO₂ di Lingkungan Industri. *Journal Teknik Elektro dan Komputer*.
- Grace C. Rumampuk, V. C. (2021). nternet of Things-Based Indoor Air Quality Monitoring System Design. *Jurnal Teknik Informatika* .
- Komang Wahyudi Suardika, G. G. (2018). Perbandingan Metode Tsukatmoto Metode Mamdani Dan Metode Sugeno Umtuk Menentukan Produksi Dupa. *E-Jurnal Matematika*.
- Kumalawati, D. A. (2022, 8 23). *DIGITAL LAB & SIMULATOR*. Diambil kembali dari Lab Machinery: absim.poltekbangsby.ac.id
- Muchamad Seno Sahisnu Virdaus, E. I. (2021). Rancang Bangun Monitoring Dan Kontrol Kualitas Udara Dengan Metode Fuzzy Logic Berbasis Wemos. *Jurnal Teknologi Elektro*.
- Muflihah, N. (2013). *Metodologi Penelitian*. Diambil kembali dari <http://nurmuflihah.blogspot.co.id/2013/04/bab-iii-metodologi-penelitian.html>
- Purbakawaca, R. (2019). SENSOR DEBU GP2Y1010AU0F GP2Y1010AU0F dust sensor.
- Rangan, A. Y. (2020). Sistem Monitoring berbasis Internet of things pada Suhu dan KelembabanUdara di Laboratorium Kimia XYZ. *Jurnal E-KOMTEK (Elektro-Komputer-Teknik)*, 172.
- Rosa, A. A. (2020). Sistem Pendeteksi Pencemaran Udara Portabel Menggunakan Sensor MQ-7 dan MQ-135. *Ultima : Jurnal Sistem Komputer*, 23-28.
- Slamet Purwo Santoso, F. W. (2022). RANCANG BANGUN AKSES PINTU DENGAN SENSOR SUHU DAN. *Jurnal Elektro* , 21.
- SURABAYA, P. P. (2023). *POLITEKNIK PENERBANGAN SURABAYA*. Diambil kembali dari DIGITAL LABS & SIMULATOR.

Udezue, C. (2016). Project Developement Institute. *12v Portable Battrry Charging System*.

Abadi, R. (n.d.). *ABSTRACT CARBON MONOXIDE (CO) AND CARBON DIOXIDE (CO₂) TELEMETRY SYSTEM WEB-BASED IN UNIVERSITY OF LAMPUNG*.

Arifin, J., Dewanti, I. E., & Kurnianto, D. (2017). Prototipe Pendingin Perangkat Telekomunikasi Sumber Arus DC menggunakan Smartphone. *Media ElektriKa*, 10(1).

Cistri, A. W., Darmawan, D., & Iskandar, R. F. (2016). Penerapan Kontrol Adaptif Dahlin Proporsional Pada Buck Converter Dengan Gangguan Medan Magnet Eksternal. *eProceedings of Engineering*, 3(3).

Datasheets, N. P. (2021, 9 10). Retrieved from <https://nextion.tech/datasheets/>

Hakiki, MI, Darusalam, U., & Nathasia, ND (2020). Konfigurasi Arduino IDE Untuk Monitoring Pendeteksi Suhu dan Kelembapan Pada Ruang Data Center Menggunakan Sensor DHT11. *Jurnal Media Informatika Budidarma*, 4 (1), 150-156.

Komang Wahyudi Suardika, G. G. (2018). Perbandingan Metode Tsukamoto Metode Mamdani Dan Metode Sugeno Umtuk Menentukan Produksi Dupa. *E-Jurnal Matematika*.

Kumalawati, D. A. (2022, 8 23). DIGITAL LAB & SIMULATOR. Retrieved from Lab Machinery: absim.poltekbangsby.ac.id

Kurnia, A. (2008). *LANGKAH LANGKAH PENELITIAN*. From <https://skripsimahasiswa.blogspot.co.id/2008/10/langkah-langkah-penelitian.html>

Lasmana, D. S., & Fitriani, E. (2020, October). Rancang Bangun Prototype Robot Penghisap Debu Menggunakan Optical Dust Sensor GP2Y1010AU0F. In *Bina Darma Conference on Engineering Science (BDCES)* (Vol. 2, No. 1, pp. 20-29).

Lim, MT, Tan, RH, Tan, GA, Hew, CH, Lee, WK, & Mercha, M. (Desember 2023). Pengembangan stasiun pengukuran iradiasi surya berbasis IoT

- dengan piranometer berbiaya rendah. Dalam Seri Konferensi IOP: Ilmu Bumi dan Lingkungan (Vol. 1281, No. 1, hlm. 012012). Penerbitan IOP.
- Maria Yus Trinity Irsan, M. I. (2019). PENGGUNAAN FUZZY LOGIC & METODE MAMDANI UNTUK MENGHITUNG PEMBELIAN, PENJUALAN DAN PERSEDIAAN. *journal of applied accounting and finance*.
- Muchamad Seno Sahisnu Virdaus. (2021). "Rancang Bangun Monitoring Dan Kontrol Kualitas Udara Dengan Metode Fuzzy Logic Berbasis Wemos". Teknik Elektro, Universitas Mercu Buana, Jakarta
- Muflihah, N. (2013). Metodologi Penelitian. From <http://nurmuflihah.blogspot.co.id/2013/04/bab-iii-metodologi-penelitian.html>
- Nebath, E., Pang, D., & Wuwung, J. O. (2014). Rancang Bangun Alat Pengukur Gas Berbahaya CO Dan CO₂ di Lingkungan Industri. *Jurnal Teknik Elektro Dan Komputer*, 3(4), 65-72.
- Purbakawaca, R. (2019). SENSOR DEBU GP2Y1010AU0F GP2Y1010AU0F dust sensor.
- Saelan, A. (2009). Logika Fuzzy. *Makalah If2091 Struktur Diskrit Tahun 2009*, 1(13508029), 1-5.
- Safrudin & Bachrudin. (2020). "Perancangan Sistem Monitoring Tegangan, Arus dan Frekuensi Pada Pembangkit Listrik Tenaga Mikrohidro Berbasis IOT". Program Studi Teknik Elektro. Universitas Nurul Jadid
- Santosa, S. P., & Wijayanto, F. (2022). Rancang Bangun Akses Pintu Dengan Sensor Suhu Dan Handsanitizer Otomatis Berbasis Arduino. *Jurnal Elektro*, 10(1), 20-31.
- Slamet Purwo Santoso, F. W. (2022). RANCANG BANGUN AKSES PINTU DENGAN SENSOR SUHU DAN. *Jurnal Elektro*, 21.
- Sugiyono. (2015). Metode Penelitian Kuantitatif, Kualitatif, dan R&D. Bandung : ALFABETA.
- Sumardi. (2016). "Rancang Bangun Sistem Keamanan Rumah Dengan Menggunakan Modul NODEMCU Berbasis IoT (INTERNET OF

- THINGS)” Program Studi Teknik Informatika. Sekolah Tinggi Manajemen Informatika dan Komputer.
- SURABAYA, P. P. (2023). *POLITEKNIK PENERBANGAN SURABAYA*. Retrieved from DIGITAL LABS & SIMULATOR.
- Trivusi. (2022). Penjelasan Lengkap Mengenai Logika Fuzzy (Fuzzy Logic). Trivusi.<https://www.trivusi.web.id/2022/05/pengertian-fuzzy-logic.html>
- Putra, A. T., & Risfendra, R. (2021). Penggunaan Aplikasi Ubidots untuk Sistem Kontrol dan Monitoring pada Gudang Gula Berbasis Arduino UNO. *JTEIN: Jurnal Teknik Elektro Indonesia*, 2(1), 40-48.
- Rahakbauw, D. L. (2015). PENERAPAN LOGIKA FUZZY METODE SUGENO. *Jurnal Ilmu Matematika dan Terapan* |, 121-134.
- Rahman, A., & Li, B. (2018). "Design and Implementation of a Monitoring and Control System for Home Automation." *IEEE Transactions on Industrial Informatics*, [DOI].
- Rumampuk, G. C., Poekoel, V. C., & Rumagit, A. M. (2022). Perancangan Sistem Monitoring Kualitas Udara Dalam Ruangan Berbasis IoT. *Jurnal Teknik Informatika*, 17(1), 11-18.
- Rangan, A. Y. (2020). Sistem Monitoring berbasis Internet of things pada Suhu dan Kelembaban Udara di Laboratorium Kimia XYZ. *Jurnal E-KOMTEK (Elektro-Komputer-Teknik)*, 172.
- Rosa, A. A. (2020). Sistem Pendeteksi Pencemaran Udara Portabel Menggunakan Sensor MQ-7 dan MQ-135. *Ultima : Jurnal Sistem Komputer*, 23-28.
- Udezue, C. (2016). Project Developement Institute. *12v Portable Battrry Charging System*.
- Virdaus, M. S. (2021). Rancang Bangun Monitoring Dan Kontrol Kualitas Udara Dengan Metode Fuzzy Logic Berbasis Wemos. *JURNAL TEKNOLOGI ELEKTRO*, 22-28.
- Zaini. (2020). “Perancangan Sistem Monitoring Tegangan, Arus dan Frekuensi Pada Pembangkit Listrik Tenaga Mikrohidro Berbasis IOT.” Program Studi Teknik Elektro. Universitas Nurul Jadid.

LAMPIRAN

Lampiran A *Standard Operational Procedure (SOP)*

“RANCANG BANGUN KONTROL DAN MONITORING KUALITAS UDARA DI LABORATORIUM MACHINARY SEBAGAI PENUNJANG PRAKTEK TARUNA BERBASIS IOT DENGAN METODE FUZZY LOGIC”

Oleh:

YOGA YANUAR NUGROHO

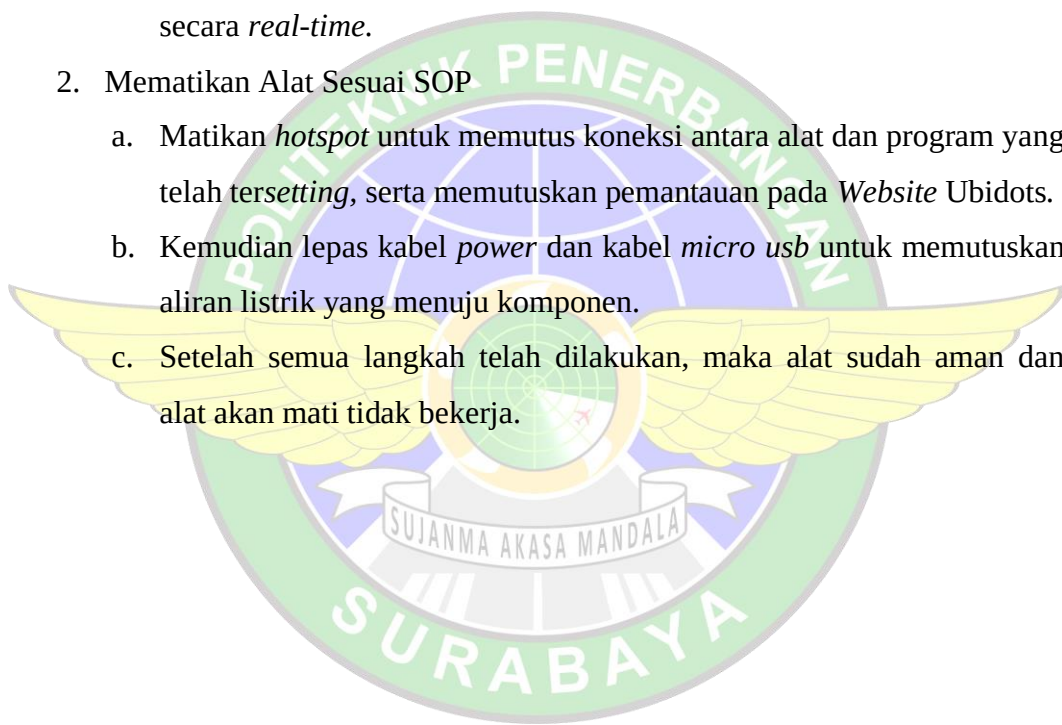
NIT. 30121047

Standard Operational Procedure (SOP) dalam pengoperasian alat adalah panduan yang merinci mengenai langkah-langkah yang harus diikuti untuk menggunakan alat dengan benar dan aman. SOP ini mencakup instruksi rinci tentang cara menghidupkan dan mematikan alat. Dimana dengan menggunakan *Standar Operational Procedure (SOP)* yang telah dibuat bertujuan untuk menjaga alat Proyek Akhir terhindar dari kerusakan atau salah prosedur, setiap pengguna alat akan mengikuti prosedur yang sama, sehingga dapat menjamin konsistensi dan akurasi hasil. Para pengguna harus memahami dan mengikuti SOP ini dengan cermat untuk memastikan operasi yang aman, efisien, dan tepat sesuai dengan tujuan penggunaan alat tersebut.

Berikut merupakan *Standar Operational Procedure (SOP)* untuk menghidupkan dan mematikan alat Proyek Akhir dengan judul “Rancang Bangun Kontrol Dan Monitoring Kualitas Udara Di Laboratorium *Machinary* Sebagai Penunjang Praktek Taruna Berbasis Iot Dengan Metode *Fuzzy Logic*” sebagai berikut:

1. Mengoperasikan Alat Sesuai dengan SOP
 - a. Sambungkan kabel power untuk menghidupkan adaptor 12 VDC dan sambungkan juga kabel untuk menghidupkan ESP32, fan 12 VDC beserta LCD.

- b. Pastikan semua komponen dapat menyala sesuai fungsinya masing-masing.
 - c. Kemudian nyalakan *hotspot* sehingga nantinya dapat termonitoring oleh LCD Nextion untuk menampilkan data sensor MQ135, sensor MQ7, sensor Dust, dan sensor DHT11 untuk memantau kondisi kualitas udara dan dapat juga termonitoring pada *Website Ubidots*.
 - d. Jika alat sudah siap untuk digunakan, maka alat akan bekerja sesuai apa yang telah diperintahkan, kemudian hasil pengukuran dapat secara langsung termonitoring pada LCD Nextion dan *Website Ubidots* secara *real-time*.
2. Mematikan Alat Sesuai SOP
- a. Matikan *hotspot* untuk memutus koneksi antara alat dan program yang telah *tersetting*, serta memutuskan pemantauan pada *Website Ubidots*.
 - b. Kemudian lepas kabel *power* dan kabel *micro usb* untuk memutuskan aliran listrik yang menuju komponen.
 - c. Setelah semua langkah telah dilakukan, maka alat sudah aman dan alat akan mati tidak bekerja.



Lampiran B Dokumentasi Alat

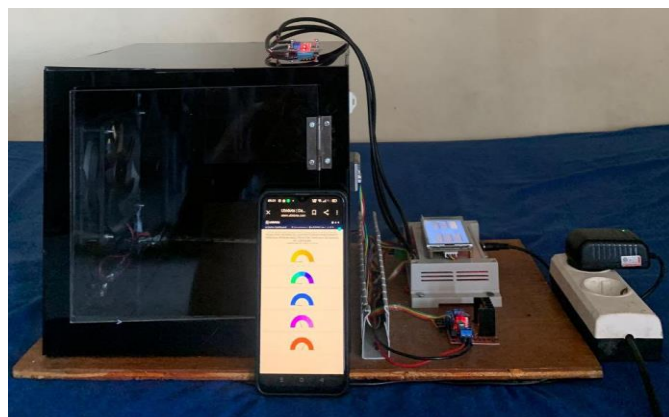
A. Gambar 1



B. Gambar 2



C. Gambar 3



Lampiran C Coding Alat

```
// ubidots
// username: yogaynr
// pw : yoga12345

// set library
#include <DHT.h>
#include <Wire.h>
#include <OneWire.h>
#include <ADS1115.h>
#include <WiFi.h>
#include <LiquidCrystal_I2C.h>
#include "UbidotsEsp32Mqtt.h"
// end

// Define Chanel Driver
const int Kipas1 = 1;

// Setting PWM properties
const int freq = 5000;
const int resolution = 8;
// end

// deklarasi PIN
#define DHTPIN 18
#define DEBUPIN 27
#define BTN_DUA 35
#define PIN_PWM 26
#define BTN_TIGA 34
#define BTN_SATU 32
#define PIN_BUZZER 25
#define DHTTYPE DHT11
#define BUILTIN_LED 2

#define SerialNextion Serial2
#define KIPAS_ON ledcWrite(Kipas1, 0);
#define KIPAS_SEDANG ledcWrite(Kipas1, 50);
#define KIPAS_OFF ledcWrite(Kipas1, 255);
// end

DHT dht(DHTPIN, DHTTYPE);

// set Lib. ADS
ADS1115 adc;
// end

// ----- set val. untuk wifi -----//
const char *WIFI_SSID = "Oddo"; // Put here your Wi-Fi SSID
const char *WIFI_PASS = "smart123"; // Put here your Wi-Fi
password
String status_send = "ERR";
```

```

// ----- end -----//
// Deklarasi for Debu
    unsigned int samplingTime = 280;
    unsigned int deltaTime = 40;
    unsigned int sleepTime = 9680;

    float voMeasured = 0;
    float calcVoltage = 0;
    float dustDensity = 0;
// end

unsigned long prevMilliskirim = 0;

/// Setup Fuzzy

    // deklarasi jumlah memnership / sensor
    #define jml_mf_debu 3
    #define jml_mf_mq135 3
    #define jml_mf_mq7 3
    #define jml_mf_kelembapan 2
    #define jml_mf_suhu 3
// end

    // deklarasi range sensor
    // mq136
    #define min_debu 0
    #define max_debu 20

    // mq135
    #define min_mq135 0
    #define max_mq135 500

    // mq7
    #define min_mq7 0
    #define max_mq7 500

    // kelembapan
    #define min_kelembapan 0
    #define max_kelembapan 100

    // kelembapan
    #define min_suhu 0
    #define max_suhu 60
// end

    // deklarasi nama index
    // selain kelembapan dan suhu
    #define Baik 0
    #define Sedang 1
    #define Bahaya 2

    // suhu
    #define Dingin 0
    #define Normal 1
    #define Panas 2

```



```

// kelembapan
#define Kering 0
#define Lembab 1
// end

// deklarasi nilai range sensor
// Debu
double mf_debu [3][3] = {
    {0, 5, 10}, // Baik
    {5, 10, 15}, // Sedang
    {10, 15, 20} // Bahaya
};

// mq135
double mf_mq135 [3][3] = {
    {0, 125, 250}, // Baik
    {125, 250, 375}, // Sedang
    {250, 375, 500} // Bahaya
};

// mq7
double mf_mq7 [3][3] = {
    {0, 125, 250}, // Baik
    {125, 250, 375}, // Sedang
    {250, 375, 500} // Bahaya
};

// kelembapan
double mf_kelembapan [2][3] = {
    {0, 42.5, 75}, // Kering
    {70, 85, 100} // Lembab
};

// suhu
double mf_suhu [3][3] = {
    {0, 15, 30}, // Dingin
    {15, 30, 45}, // Normal
    {30, 45, 60} // Panas
};
// end

// deklarasi aturan rule base
double rule[81];
double bobot_debu[jml_mf_debu];
double bobot_mq135[jml_mf_mq135];
double bobot_mq7[jml_mf_mq7];
double bobot_kelembapan[jml_mf_kelembapan];
double bobot_suhu[jml_mf_suhu];

double mf_KualitasUdara[] = {100, 200, 300};
String HasilKualitasUdara;
// end

/// end

float humidity, temperature;

```

```

float kalibrasiDebu = 0.0;

const char *UBIDOTS_TOKEN = "BBUS-KNbDCXUx8vbEawvMvJSZA7ZGLuC6m9";
const char *DEVICE_LABEL = "monitoringkualitassudara";
Ubidots ubidots(UBIDOTS_TOKEN);

void callback(char *topic, byte *payload, unsigned int length){
    Serial.print("Message arrived [");
    Serial.print(topic);
    Serial.print("] ");
    for (int i = 0; i < length; i++)
    {
        Serial.print((char)payload[i]);
    }
    Serial.println();
}

void setup() {
    dht.begin();
    Wire.begin();
    Serial.begin(9600);
    SerialNextion.begin(9600);
    adc.setSpeed(ADS1115_SPEED_16SPS);

    pinMode(DEBUPIN, OUTPUT);
    pinMode(PIN_BUZZER, OUTPUT);
    pinMode(BUILTIN_LED, OUTPUT);
    pinMode(BTN_DUA, INPUT_PULLUP);
    pinMode(BTN_SATU, INPUT_PULLUP);
    pinMode(BTN_TIGA, INPUT_PULLUP);

    // set PWM
    ledcSetup(Kipas1, freq, resolution);
    ledcAttachPin(PIN_PWM, Kipas1);
    // end

    ledcWrite(Kipas1, 255);

    ubidots.connectToWifi(WIFI_SSID, WIFI_PASS);
    ubidots.setCallback(callback);
    ubidots.setup();
    ubidots.reconnect();

    float ratiooo = 0.0;

    for (int i = 0; i < 50; i++) {
        float Value = adc.convertAutoScale(ADS1115_CHANNEL3, 3);
        ratiooo = ratiooo + Value;
        delay(10);
    }

    kalibrasiDebu = ratiooo / 50.0;

    Serial.println("Nilai Kalibrasi: " + String(kalibrasiDebu));

    beepBuzzer(2);

```

```

}

void xloop(){
    float defuzzy_KualitasUdara = HasilFuzzy(2.536, 83.33, 68.84,
34.56, 7.721);
    Serial.println(defuzzy_KualitasUdara);

    String HasilFuzzy = "";
    if (defuzzy_KualitasUdara == 0){
        HasilFuzzy = Fillter(2.536, 83.33, 68.84, 34.56, 7.721);
        Serial.println("Masuk Kualitas " + HasilFuzzy);
    }

    else if (defuzzy_KualitasUdara >=90 && defuzzy_KualitasUdara <=
160){
        Serial.println("Masuk Kualitas Normal");
        HasilFuzzy = "Normal";
        KIPAS_OFF;
    }

    else if (defuzzy_KualitasUdara >= 161 && defuzzy_KualitasUdara
<= 260){
        Serial.println("Masuk Kualitas Sedang");
        HasilFuzzy = "Sedang";
        KIPAS_SEDANG;
    }

    else if (defuzzy_KualitasUdara >= 261 && defuzzy_KualitasUdara
<= 320){
        Serial.println("Masuk Kualitas Bahaya");
        HasilFuzzy = "Bahaya";
        KIPAS_ON;
    }

    delay(10000);
}

void loop() {
    GetDHT();
    float Debu = GetValueDebu();
    float ppm135 = MQ135_getPPM();
    float ppm7 = MQ7_getPPM();

    Serial.println("Nilai Debu: " + String(Debu));
    Serial.println("Nilai PPM mq135: " + String(ppm135) + " Nilai
PPM mq7: " + String(ppm7));
    Serial.println("Nilai humidity: " + String(humidity) + " Nilai
temperature: " + String(temperature));

    if (!ubidots.connected()){
        ubidots.reconnect();
    }

    float defuzzy_KualitasUdara = HasilFuzzy(Debu, ppm135, ppm7,
humidity, temperature);
    Serial.println(defuzzy_KualitasUdara);
}

```

```

String HasilFuzzy = "";
if (defuzzy_KualitasUdara == 0){
    HasilFuzzy = Fillter(Debu, ppm135, ppm7, humidity,
temperature);
    Serial.println("Masuk Kualitas " + HasilFuzzy);
}

else if (defuzzy_KualitasUdara >=90 && defuzzy_KualitasUdara <=
160){
    Serial.println("Masuk Kualitas Normal");
    HasilFuzzy = "Normal";
    KIPAS_OFF;
}

else if (defuzzy_KualitasUdara >= 161 && defuzzy_KualitasUdara
<= 260){
    Serial.println("Masuk Kualitas Sedang");
    HasilFuzzy = "Sedang";
    KIPAS_SEDANG;
}

else if (defuzzy_KualitasUdara >= 261 && defuzzy_KualitasUdara
<= 320){
    Serial.println("Masuk Kualitas Bahaya");
    HasilFuzzy = "Bahaya";
    KIPAS_ON;
}

unsigned long currentMillis = millis();
if (currentMillis - prevMilliskirim >= 5000) {
    ubidots.add("debu", Debu);
    ubidots.add("mq135", ppm135);
    ubidots.add("mq7", ppm7);
    ubidots.add("kelembapan", humidity);
    ubidots.add("suhu", temperature);
    ubidots.publish(DEVICE_LABEL);

    Serial.println("Kirim Ke ubidots");

    prevMilliskirim = currentMillis;
}

setTextBox("ValDebu", String(Debu, 2));
setTextBox("ValAmonia", String(ppm135, 2));
setTextBox("ValCO", String(ppm7, 2));
setTextBox("ValHuman", String(humidity, 2));
setTextBox("ValSuhu", String(temperature, 2));
setTextBox("ValFuzzy", HasilFuzzy);

delay(500);
}

void beepBuzzer(int count) {

```

```

    for (int i = 0; i < count; ++i) {
        digitalWrite(PIN_BUZZER, HIGH);
        delay(100);
        digitalWrite(PIN_BUZZER, LOW);
        delay(100);
    }
}

void setTextBox(String objName, String text) {
    SerialNextion.print(objName);
    SerialNextion.print(".txt=\"");
    SerialNextion.print(text);
    SerialNextion.print("\");
    SerialNextion.write(0xff);
    SerialNextion.write(0xff);
    SerialNextion.write(0xff);
}

float GetValueDebu(){
    digitalWrite(DEBUPIN, LOW);
    delayMicroseconds(280);
    float Value = adc.convertAutoScale(ADS1115_CHANNEL3, 3);

    delayMicroseconds(40);
    digitalWrite(DEBUPIN, HIGH);
    delayMicroseconds(9680);

    dustDensity = (Value - kalibrasiDebu) * 1000.0 / 5.0;

    if ( dustDensity < 0)
    {
        dustDensity = 0.00;
    }

    return dustDensity;
}

float MQ135_getPPM() {
    float VMq135 = adc.convertAutoScale(ADS1115_CHANNEL1, 3);

    float rs = ((5.0 * 10.0) - (10.0 * VMq135)) / VMq135;
    float ratio = rs / 76.63;
    float CO2_PPM= (146.15*(2.868-ratio)+10);

    if (CO2_PPM < 0){
        CO2_PPM = 1.0;
    }

    return CO2_PPM;
}

float MQ7_getPPM() {
    float Vmq7 = adc.convertAutoScale(ADS1115_CHANNEL2, 3);
    float RsRL = (5.0 - Vmq7) / Vmq7;
    float readRs = 10 * RsRL;

```



```

float HasilPPM7 = 100 * pow( (readRs/64.66), -1.513);
float ppm7 = map(HasilPPM7, 1.61, 12000, 1, 1000);

return ppm7;
}

void GetDHT(){
    humidity = dht.readHumidity();
    temperature = dht.readTemperature();
}

float Mg_CO2(float ppm) {
    float MolarMass = 44.01;
    float kondisiStandart = 22.414;

    float hasil = ppm * (MolarMass / kondisiStandart);
    return hasil;
}

float Mg_CO(float ppm) {
    float MolarMass = 28.01;
    float kondisiStandart = 22.414;

    float hasil = ppm * (MolarMass / kondisiStandart);
    return hasil;
}

float HasilFuzzy(float V_debu, float V_mq135, float V_mq7, float
V_kelembapan, float V_suhu){
    get_bobot(bobot_debu, mf_debu, V_debu, jml_mf_debu, max_debu,
    "Debu");
    get_bobot(bobot_mq135, mf_mq135, V_mq135, jml_mf_mq135,
    max_mq135, "MQ 135");
    get_bobot(bobot_mq7, mf_mq7, V_mq7, jml_mf_mq7, max_mq7, "MQ
    7");
    get_bobot(bobot_kelembapan, mf_kelembapan, V_kelembapan,
    jml_mf_kelembapan, max_kelembapan, "MQ Kelembapan");
    get_bobot(bobot_suhu, mf_suhu, V_suhu, jml_mf_suhu, max_suhu,
    "Suhu");

    setRule(1, bobot_debu[Baik], bobot_mq135[Baik], bobot_mq7[Baik],
    bobot_kelembapan[Kering], bobot_suhu[Dingin]);
    setRule(2, bobot_debu[Baik], bobot_mq135[Baik], bobot_mq7[Baik],
    bobot_kelembapan[Kering], bobot_suhu[Normal]);
    setRule(3, bobot_debu[Baik], bobot_mq135[Baik], bobot_mq7[Baik],
    bobot_kelembapan[Kering], bobot_suhu[Panas]);
    setRule(4, bobot_debu[Baik], bobot_mq135[Baik], bobot_mq7[Baik],
    bobot_kelembapan[Lembab], bobot_suhu[Dingin]);
    setRule(5, bobot_debu[Baik], bobot_mq135[Baik], bobot_mq7[Baik],
    bobot_kelembapan[Lembab], bobot_suhu[Dingin]);

    setRule(6, bobot_debu[Baik], bobot_mq135[Baik], bobot_mq7[Baik],
    bobot_kelembapan[Lembab], bobot_suhu[Normal]);
    setRule(7, bobot_debu[Baik], bobot_mq135[Baik], bobot_mq7[Baik],
    bobot_kelembapan[Lembab], bobot_suhu[Panas]);
    setRule(8, bobot_debu[Baik], bobot_mq135[Baik],
    bobot_mq7[Sedang], bobot_kelembapan[Kering], bobot_suhu[Dingin]);

```

```

    setRule(9,          bobot_debu[Baik],          bobot_mq135[Baik],
    bobot_mq7[Sedang], bobot_kelembapan[Kering], bobot_suhu[Normal]);
    setRule(10,         bobot_debu[Baik],          bobot_mq135[Baik],
    bobot_mq7[Sedang], bobot_kelembapan[Kering], bobot_suhu[Panas]);

    setRule(11,         bobot_debu[Baik],          bobot_mq135[Baik],
    bobot_mq7[Sedang], bobot_kelembapan[Lembab], bobot_suhu[Dingin]);
    setRule(12,         bobot_debu[Baik],          bobot_mq135[Baik],
    bobot_mq7[Sedang], bobot_kelembapan[Lembab], bobot_suhu[Normal]);
    setRule(13,         bobot_debu[Baik],          bobot_mq135[Baik],
    bobot_mq7[Sedang], bobot_kelembapan[Lembab], bobot_suhu[Panas]);
    setRule(14,         bobot_debu[Baik],          bobot_mq135[Baik],
    bobot_mq7[Bahaya], bobot_kelembapan[Kering], bobot_suhu[Dingin]);
    setRule(15,         bobot_debu[Baik],          bobot_mq135[Baik],
    bobot_mq7[Bahaya], bobot_kelembapan[Kering], bobot_suhu[Normal]);

    setRule(16,         bobot_debu[Baik],          bobot_mq135[Baik],
    bobot_mq7[Bahaya], bobot_kelembapan[Kering], bobot_suhu[Panas]);
    setRule(17,         bobot_debu[Baik],          bobot_mq135[Baik],
    bobot_mq7[Bahaya], bobot_kelembapan[Lembab], bobot_suhu[Dingin]);
    setRule(18,         bobot_debu[Baik],          bobot_mq135[Baik],
    bobot_mq7[Bahaya], bobot_kelembapan[Lembab], bobot_suhu[Normal]);
    setRule(19,         bobot_debu[Baik],          bobot_mq135[Baik],
    bobot_mq7[Bahaya], bobot_kelembapan[Lembab], bobot_suhu[Panas]);
    setRule(20,         bobot_debu[Baik],          bobot_mq135[Sedang],
    bobot_mq7[Baik],   bobot_kelembapan[Kering], bobot_suhu[Dingin]);

    setRule(21,         bobot_debu[Baik],          bobot_mq135[Sedang],
    bobot_mq7[Baik],   bobot_kelembapan[Kering], bobot_suhu[Normal]);
    setRule(22,         bobot_debu[Baik],          bobot_mq135[Sedang],
    bobot_mq7[Baik],   bobot_kelembapan[Kering], bobot_suhu[Panas]);
    setRule(23,         bobot_debu[Baik],          bobot_mq135[Sedang],
    bobot_mq7[Baik],   bobot_kelembapan[Lembab], bobot_suhu[Dingin]);
    setRule(24,         bobot_debu[Baik],          bobot_mq135[Sedang],
    bobot_mq7[Baik],   bobot_kelembapan[Lembab], bobot_suhu[Normal]);
    setRule(25,         bobot_debu[Baik],          bobot_mq135[Sedang],
    bobot_mq7[Baik],   bobot_kelembapan[Lembab], bobot_suhu[Panas]);

    setRule(26,         bobot_debu[Baik],          bobot_mq135[Sedang],
    bobot_mq7[Sedang], bobot_kelembapan[Kering], bobot_suhu[Dingin]);
    setRule(27,         bobot_debu[Baik],          bobot_mq135[Sedang],
    bobot_mq7[Sedang], bobot_kelembapan[Kering], bobot_suhu[Normal]);
    setRule(28,         bobot_debu[Baik],          bobot_mq135[Sedang],
    bobot_mq7[Sedang], bobot_kelembapan[Kering], bobot_suhu[Panas]);
    setRule(29,         bobot_debu[Baik],          bobot_mq135[Sedang],
    bobot_mq7[Sedang], bobot_kelembapan[Lembab], bobot_suhu[Dingin]);
    setRule(30,         bobot_debu[Baik],          bobot_mq135[Sedang],
    bobot_mq7[Sedang], bobot_kelembapan[Lembab], bobot_suhu[Normal]);

    setRule(31,         bobot_debu[Baik],          bobot_mq135[Sedang],
    bobot_mq7[Sedang], bobot_kelembapan[Lembab], bobot_suhu[Panas]);
    setRule(32,         bobot_debu[Baik],          bobot_mq135[Sedang],
    bobot_mq7[Bahaya], bobot_kelembapan[Kering], bobot_suhu[Dingin]);
    setRule(33,         bobot_debu[Baik],          bobot_mq135[Sedang],
    bobot_mq7[Bahaya], bobot_kelembapan[Kering], bobot_suhu[Normal]);

```

```

setRule(34,          bobot_debu[Baik],          bobot_mq135[Sedang],
bobot_mq7[Bahaya], bobot_kelembapan[Kering], bobot_suhu[Panas]);
setRule(35,          bobot_debu[Baik],          bobot_mq135[Sedang],
bobot_mq7[Bahaya], bobot_kelembapan[Lembab], bobot_suhu[Dingin]);

setRule(36,          bobot_debu[Baik],          bobot_mq135[Sedang],
bobot_mq7[Bahaya], bobot_kelembapan[Lembab], bobot_suhu[Normal]);
setRule(37,          bobot_debu[Baik],          bobot_mq135[Sedang],
bobot_mq7[Bahaya], bobot_kelembapan[Lembab], bobot_suhu[Panas]);
setRule(38,          bobot_debu[Baik],          bobot_mq135[Bahaya],
bobot_mq7[Baik], bobot_kelembapan[Kering], bobot_suhu[Dingin]);
setRule(39,          bobot_debu[Baik],          bobot_mq135[Bahaya],
bobot_mq7[Baik], bobot_kelembapan[Kering], bobot_suhu[Normal]);
setRule(40,          bobot_debu[Baik],          bobot_mq135[Bahaya],
bobot_mq7[Baik], bobot_kelembapan[Kering], bobot_suhu[Panas]);

setRule(41,          bobot_debu[Baik],          bobot_mq135[Bahaya],
bobot_mq7[Baik], bobot_kelembapan[Lembab], bobot_suhu[Dingin]);
setRule(42,          bobot_debu[Baik],          bobot_mq135[Bahaya],
bobot_mq7[Baik], bobot_kelembapan[Lembab], bobot_suhu[Normal]);
setRule(43,          bobot_debu[Baik],          bobot_mq135[Bahaya],
bobot_mq7[Baik], bobot_kelembapan[Lembab], bobot_suhu[Panas]);
setRule(44,          bobot_debu[Baik],          bobot_mq135[Bahaya],
bobot_mq7[Sedang], bobot_kelembapan[Kering], bobot_suhu[Dingin]);
setRule(45,          bobot_debu[Baik],          bobot_mq135[Bahaya],
bobot_mq7[Sedang], bobot_kelembapan[Kering], bobot_suhu[Normal]);

setRule(46,          bobot_debu[Baik],          bobot_mq135[Bahaya],
bobot_mq7[Sedang], bobot_kelembapan[Kering], bobot_suhu[Panas]);
setRule(47,          bobot_debu[Baik],          bobot_mq135[Bahaya],
bobot_mq7[Sedang], bobot_kelembapan[Lembab], bobot_suhu[Dingin]);
setRule(48,          bobot_debu[Baik],          bobot_mq135[Bahaya],
bobot_mq7[Sedang], bobot_kelembapan[Lembab], bobot_suhu[Normal]);
setRule(49,          bobot_debu[Baik],          bobot_mq135[Bahaya],
bobot_mq7[Sedang], bobot_kelembapan[Lembab], bobot_suhu[Panas]);
setRule(50,          bobot_debu[Baik],          bobot_mq135[Bahaya],
bobot_mq7[Bahaya], bobot_kelembapan[Kering], bobot_suhu[Dingin]);

setRule(51,          bobot_debu[Baik],          bobot_mq135[Bahaya],
bobot_mq7[Bahaya], bobot_kelembapan[Kering], bobot_suhu[Normal]);
setRule(52,          bobot_debu[Baik],          bobot_mq135[Bahaya],
bobot_mq7[Bahaya], bobot_kelembapan[Kering], bobot_suhu[Panas]);
setRule(53,          bobot_debu[Baik],          bobot_mq135[Bahaya],
bobot_mq7[Bahaya], bobot_kelembapan[Lembab], bobot_suhu[Dingin]);
setRule(54,          bobot_debu[Baik],          bobot_mq135[Bahaya],
bobot_mq7[Bahaya], bobot_kelembapan[Lembab], bobot_suhu[Normal]);
setRule(55,          bobot_debu[Baik],          bobot_mq135[Bahaya],
bobot_mq7[Bahaya], bobot_kelembapan[Lembab], bobot_suhu[Panas]);

setRule(56,          bobot_debu[Sedang],          bobot_mq135[Baik],
bobot_mq7[Baik], bobot_kelembapan[Kering], bobot_suhu[Dingin]);
setRule(57,          bobot_debu[Sedang],          bobot_mq135[Baik],
bobot_mq7[Baik], bobot_kelembapan[Kering], bobot_suhu[Normal]);
setRule(58,          bobot_debu[Sedang],          bobot_mq135[Baik],
bobot_mq7[Baik], bobot_kelembapan[Kering], bobot_suhu[Panas]);

```

```

    setRule(59,          bobot_debu[Sedang],          bobot_mq135[Baik],
    bobot_mq7[Baik], bobot_kelembapan[Lembab], bobot_suhu[Dingin]);
    setRule(60,          bobot_debu[Sedang],          bobot_mq135[Baik],
    bobot_mq7[Baik], bobot_kelembapan[Lembab], bobot_suhu[Normal]);

    setRule(61,          bobot_debu[Sedang],          bobot_mq135[Baik],
    bobot_mq7[Baik], bobot_kelembapan[Lembab], bobot_suhu[Panas]);
    setRule(62,          bobot_debu[Sedang],          bobot_mq135[Baik],
    bobot_mq7[Sedang], bobot_kelembapan[Kering], bobot_suhu[Dingin]);
    setRule(63,          bobot_debu[Sedang],          bobot_mq135[Baik],
    bobot_mq7[Sedang], bobot_kelembapan[Kering], bobot_suhu[Normal]);
    setRule(64,          bobot_debu[Sedang],          bobot_mq135[Baik],
    bobot_mq7[Sedang], bobot_kelembapan[Kering], bobot_suhu[Panas]);
    setRule(65,          bobot_debu[Sedang],          bobot_mq135[Baik],
    bobot_mq7[Sedang], bobot_kelembapan[Lembab], bobot_suhu[Dingin]);

    setRule(66,          bobot_debu[Sedang],          bobot_mq135[Baik],
    bobot_mq7[Sedang], bobot_kelembapan[Lembab], bobot_suhu[Normal]);
    setRule(67,          bobot_debu[Sedang],          bobot_mq135[Baik],
    bobot_mq7[Sedang], bobot_kelembapan[Lembab], bobot_suhu[Panas]);
    setRule(68,          bobot_debu[Sedang],          bobot_mq135[Baik],
    bobot_mq7[Bahaya], bobot_kelembapan[Kering], bobot_suhu[Dingin]);
    setRule(69,          bobot_debu[Sedang],          bobot_mq135[Baik],
    bobot_mq7[Bahaya], bobot_kelembapan[Kering], bobot_suhu[Normal]);
    setRule(70,          bobot_debu[Sedang],          bobot_mq135[Baik],
    bobot_mq7[Bahaya], bobot_kelembapan[Kering], bobot_suhu[Panas]);

    setRule(71,          bobot_debu[Sedang],          bobot_mq135[Baik],
    bobot_mq7[Bahaya], bobot_kelembapan[Lembab], bobot_suhu[Dingin]);
    setRule(72,          bobot_debu[Sedang],          bobot_mq135[Baik],
    bobot_mq7[Bahaya], bobot_kelembapan[Lembab], bobot_suhu[Normal]);
    setRule(73,          bobot_debu[Sedang],          bobot_mq135[Baik],
    bobot_mq7[Bahaya], bobot_kelembapan[Lembab], bobot_suhu[Panas]);
    setRule(74,          bobot_debu[Bahaya],          bobot_mq135[Bahaya],
    bobot_mq7[Bahaya], bobot_kelembapan[Lembab], bobot_suhu[Panas]);
    setRule(75,          bobot_debu[Bahaya],          bobot_mq135[Bahaya],
    bobot_mq7[Bahaya], bobot_kelembapan[Lembab], bobot_suhu[Dingin]);

    setRule(76,          bobot_debu[Bahaya],          bobot_mq135[Bahaya],
    bobot_mq7[Bahaya], bobot_kelembapan[Lembab], bobot_suhu[Normal]);
    setRule(77,          bobot_debu[Bahaya],          bobot_mq135[Bahaya],
    bobot_mq7[Bahaya], bobot_kelembapan[Kering], bobot_suhu[Dingin]);
    setRule(78,          bobot_debu[Bahaya],          bobot_mq135[Bahaya],
    bobot_mq7[Bahaya], bobot_kelembapan[Kering], bobot_suhu[Normal]);
    setRule(79,          bobot_debu[Bahaya],          bobot_mq135[Bahaya],
    bobot_mq7[Bahaya], bobot_kelembapan[Kering], bobot_suhu[Panas]);
    setRule(80,          bobot_debu[Bahaya],          bobot_mq135[Sedang],
    bobot_mq7[Sedang], bobot_kelembapan[Kering], bobot_suhu[Dingin]);

    double total_rule = 0;
    double defuzzy_KualitasUdara = 0;

    for (int xx = 1; xx <= 80; xx++) {
        total_rule = rule[xx] + total_rule;
    }

```



```

if (total_rule != 0 ) {
    defuzzy_KualitasUdara = 0;
    defuzzy_KualitasUdara = rule[1] * mf_KualitasUdara[Baik] +
        rule[2] * mf_KualitasUdara[Baik] +
        rule[3] * mf_KualitasUdara[Sedang] +
        rule[4] * mf_KualitasUdara[Baik] +
        rule[5] * mf_KualitasUdara[Bahaya] +

        rule[6] * mf_KualitasUdara[Bahaya] +
        rule[7] * mf_KualitasUdara[Bahaya] +
        rule[8] * mf_KualitasUdara[Sedang] +
        rule[9] * mf_KualitasUdara[Sedang] +
        rule[10] * mf_KualitasUdara[Sedang] +

        rule[11] * mf_KualitasUdara[Bahaya] +
        rule[12] * mf_KualitasUdara[Bahaya] +
        rule[13] * mf_KualitasUdara[Bahaya] +
        rule[14] * mf_KualitasUdara[Bahaya] +
        rule[15] * mf_KualitasUdara[Bahaya] +

        rule[16] * mf_KualitasUdara[Bahaya] +
        rule[17] * mf_KualitasUdara[Bahaya] +
        rule[18] * mf_KualitasUdara[Bahaya] +
        rule[19] * mf_KualitasUdara[Bahaya] +
        rule[20] * mf_KualitasUdara[Sedang] +

        rule[21] * mf_KualitasUdara[Sedang] +
        rule[22] * mf_KualitasUdara[Sedang] +
        rule[23] * mf_KualitasUdara[Bahaya] +
        rule[24] * mf_KualitasUdara[Bahaya] +
        rule[25] * mf_KualitasUdara[Bahaya] +

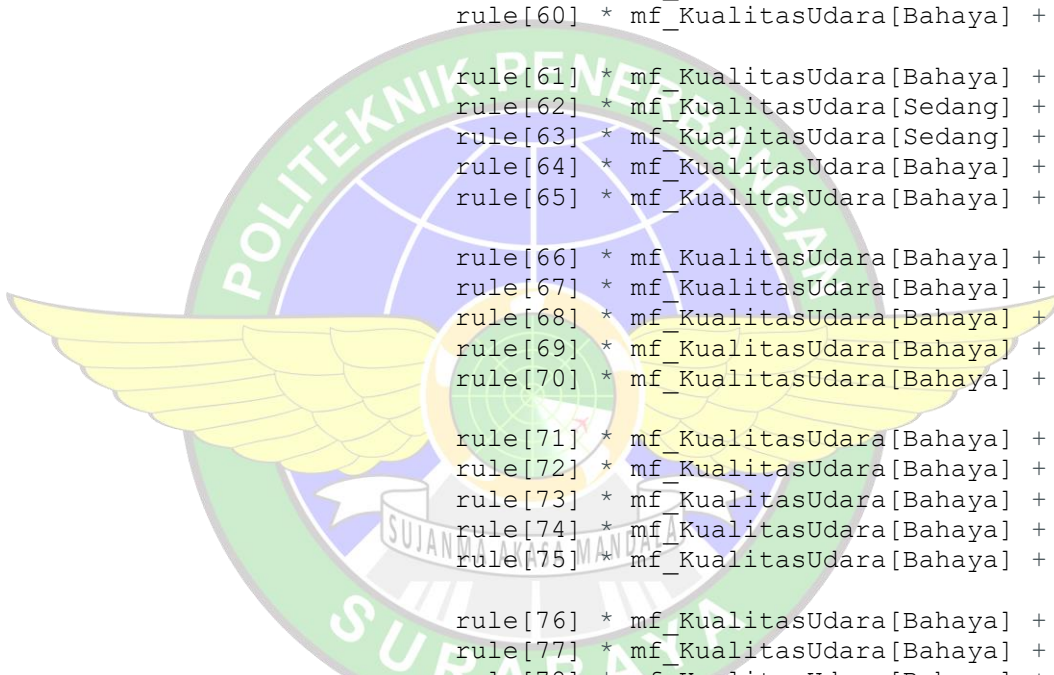
        rule[26] * mf_KualitasUdara[Sedang] +
        rule[27] * mf_KualitasUdara[Sedang] +
        rule[28] * mf_KualitasUdara[Bahaya] +
        rule[29] * mf_KualitasUdara[Bahaya] +
        rule[30] * mf_KualitasUdara[Bahaya] +

        rule[31] * mf_KualitasUdara[Bahaya] +
        rule[32] * mf_KualitasUdara[Bahaya] +
        rule[33] * mf_KualitasUdara[Bahaya] +
        rule[34] * mf_KualitasUdara[Bahaya] +
        rule[35] * mf_KualitasUdara[Bahaya] +

        rule[36] * mf_KualitasUdara[Bahaya] +
        rule[37] * mf_KualitasUdara[Bahaya] +
        rule[38] * mf_KualitasUdara[Bahaya] +
        rule[39] * mf_KualitasUdara[Bahaya] +
        rule[40] * mf_KualitasUdara[Bahaya] +

        rule[41] * mf_KualitasUdara[Bahaya] +
        rule[42] * mf_KualitasUdara[Bahaya] +
        rule[43] * mf_KualitasUdara[Bahaya] +
        rule[44] * mf_KualitasUdara[Bahaya] +
        rule[45] * mf_KualitasUdara[Bahaya] +

```

```

rule[46] * mf_KualitasUdara[Bahaya] +
rule[47] * mf_KualitasUdara[Bahaya] +
rule[48] * mf_KualitasUdara[Bahaya] +
rule[49] * mf_KualitasUdara[Bahaya] +
rule[50] * mf_KualitasUdara[Bahaya] +

rule[51] * mf_KualitasUdara[Bahaya] +
rule[52] * mf_KualitasUdara[Bahaya] +
rule[53] * mf_KualitasUdara[Bahaya] +
rule[54] * mf_KualitasUdara[Bahaya] +
rule[55] * mf_KualitasUdara[Bahaya] +

rule[56] * mf_KualitasUdara[Sedang] +
rule[57] * mf_KualitasUdara[Sedang] +
rule[58] * mf_KualitasUdara[Bahaya] +
rule[59] * mf_KualitasUdara[Bahaya] +
rule[60] * mf_KualitasUdara[Bahaya] +

rule[61] * mf_KualitasUdara[Bahaya] +
rule[62] * mf_KualitasUdara[Sedang] +
rule[63] * mf_KualitasUdara[Sedang] +
rule[64] * mf_KualitasUdara[Bahaya] +
rule[65] * mf_KualitasUdara[Bahaya] +

rule[66] * mf_KualitasUdara[Bahaya] +
rule[67] * mf_KualitasUdara[Bahaya] +
rule[68] * mf_KualitasUdara[Bahaya] +
rule[69] * mf_KualitasUdara[Bahaya] +
rule[70] * mf_KualitasUdara[Bahaya] +

rule[71] * mf_KualitasUdara[Bahaya] +
rule[72] * mf_KualitasUdara[Bahaya] +
rule[73] * mf_KualitasUdara[Bahaya] +
rule[74] * mf_KualitasUdara[Bahaya] +
rule[75] * mf_KualitasUdara[Bahaya] +

rule[76] * mf_KualitasUdara[Bahaya] +
rule[77] * mf_KualitasUdara[Bahaya] +
rule[78] * mf_KualitasUdara[Bahaya] +
rule[79] * mf_KualitasUdara[Bahaya] +
rule[80] * mf_KualitasUdara[Bahaya];

    defuzzy_KualitasUdara = defuzzy_KualitasUdara / total_rule;
}
else {
    defuzzy_KualitasUdara = 0;
}

return defuzzy_KualitasUdara;
}

String Fillter(float V_debu, float V_mq135, float V_mq7, float
V_kelembapan, float V_suhu){
    String Hasil = "Normal";
    if (V_debu >= 12.5 || V_mq135 >= 125 || V_mq7 >= 125 || V_suhu
>= 17.5){

```

```

    Hasil = "Sedang";
}

if (V_debu >= 25 || V_mq135 >= 250 || V_mq7 >= 250 ||
V_kelembapan >= 85 || V_suhu >= 35){
    Hasil = "Bahaya";
}

return Hasil;
}

void get_bobot(double bobot_sensor[], double mf_sensor[][3],
double value, int baris, int max_nilai, String label) {
    double y, m, c;
    double awal, puncak, akhir;
    double konstanta = 0.01;
    double iterasi;
    iterasi = max_nilai / konstanta;
    double nilai_iterasi;

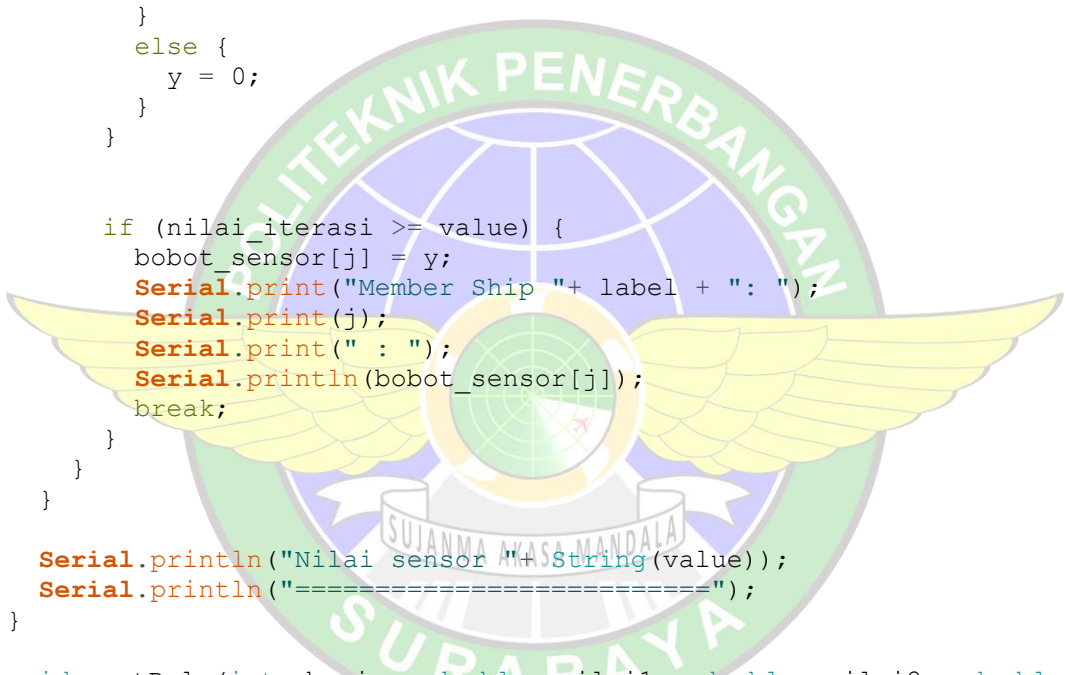
    for (int j = 0; j < baris - 1; j++) {
        bobot_sensor[j] = 0;
    }
    for (int j = 0; j <= baris - 1; j++) {
        awal = mf_sensor[j][0];
        puncak = mf_sensor[j][1];
        akhir = mf_sensor[j][2];
        nilai_iterasi = 0;

        for (int i = 0; i <= iterasi - 1; i++) {
            nilai_iterasi = nilai_iterasi + konstanta;
            if (nilai_iterasi < awal){
                if (j == 0) {
                    y = 1;
                }
                else {
                    y = 0;
                }
            }

            else if (nilai_iterasi >= awal && nilai_iterasi < puncak) {
                if (j == 0) {
                    y = 1;
                }
                else {
                    m = (puncak - awal);
                    m = 1 / m;
                    c = (m * puncak);
                    c = 1 - c;
                    y = (m * nilai_iterasi);
                    y = y + c;
                }
            }

            else if (nilai_iterasi >= puncak && nilai_iterasi < akhir){
                if (j == baris - 1){

```



```

        y = 1;
    }

    else {
        m = (puncak - akhir);
        m = 1 / m;
        c = (m * puncak);
        c = 1 - c;
        y = (m * nilai_iterasi);
        y = y + c;
    }
}

else if (nilai_iterasi >= akhir ) {
    if ( j == baris - 1) {
        y = 1;
    }
    else {
        y = 0;
    }
}

if (nilai_iterasi >= value) {
    bobot_sensor[j] = y;
    Serial.print("Member Ship "+ label + ": ");
    Serial.print(j);
    Serial.print(" : ");
    Serial.println(bobot_sensor[j]);
    break;
}
}
}

Serial.println("Nilai sensor "+ String(value));
Serial.println("=====");
}

void setRule(int baris, double nilai1, double nilai2, double
nilai3, double nilai4, double nilai5){
    rule[baris] = min_val(nilai1, nilai2, nilai3, nilai4, nilai5);
}

double min_val(double sen1, double sen2, double sen3, double sen4,
double sen5){
    double smallest = sen1;
    if (sen2 < smallest) {
        smallest = sen2;
    }

    if (sen3 < smallest) {
        smallest = sen3;
    }

    if (sen4 < smallest) {
        smallest = sen4;
    }
}

```

```
}  
  
if (sen5 < smallest) {  
    smallest = sen5;  
}  
  
return smallest;  
}
```



Lampiran D Daftar Riwayat Hidup

DAFTAR RIWAYAT HIDUP



Yoga Yanuar Nugroho, Lahir di Kediri, Jawa Timur pada tanggal 25 Januari 2003. Anak kedua dari dua bersaudara, dari pasangan Bapak Agus Priyanto dan Ibu Hariyati. Mempunyai 2 saudara kandung kakak Adi Pratama Nugro Ramadhanani dan adik Herlina Candra Cahya Ningrum . Beragama Islam. Bertempat tinggal di Dusun Pajaran Selatan, RT 15 RW 05, Kelurahan Mliriprowo, Kecamatan Tarik, Kabupaten Sidoarjo, Provinsi Jawa Timur.

Dengan pendidikan formal yang pernah diikuti sebagai berikut :

1. SD Negeri Magersari 1 (lulus pada tahun 2015)
2. SMP Negeri 2 Kota Mojokerto (lulus pada tahun 2018)
3. SMA Negeri 1 Bangsal Mojokerto (lulus pada tahun 2021)

Pada tahun 2021 diterima sebagai Taruna di Politeknik Penerbangan Surabaya Program Studi Diploma 3 Teknik Listrik Bandar Udara Angkatan ke-16. Melaksanakan *On the Job Training* pertama di Bandar Udara Sultan Muhammad Salahudin, Bima pada tanggal 08 Mei 2023 sampai 12 September 2023 dan *On the Job Training* kedua di Bandar Udara Depati Amir, Pangkal Pinang pada tanggal 02 Oktober 2023 sampai 29 Februari 2024. Aktif sebagai anggota Resimen Korps Taruna dan Gita Swara Buana.