

**RANCANG BANGUN KONTROL DAN MONITORING POMPA
BAHAN BAKAR GENSET PADA *DAILY TANK* BERBASIS IOT
MENGUNAKAN METODE *NEURAL NETWORK***

PROYEK AKHIR



Oleh :

BRIHAM MAULIDAN
NIT. 30121029

**PROGRAM STUDI DIPLOMA 3 TEKNIK LISTRIK BANDARA
POLITEKNIK PENERBANGAN SURABAYA
2024**

RANCANG BANGUN KONTROL DAN MONITORING POMPA BAHAN BAKAR GENSET PADA *DAILY TANK* BERBASIS IOT MENGUNAKAN METODE *NEURAL NETWORK*

PROYEK AKHIR

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



Oleh :

BRIHAM MAULIDAN
NIT. 30121029

**PROGRAM STUDI DIPLOMA 3 TEKNIK LISTRIK BANDARA
POLITEKNIK PENERBANGAN SURABAYA
2024**

LEMBAR PERSETUJUAN

RANCANG BANGUN KONTROL DAN MONITORING POMPA
BAHAN BAKAR GENSET PADA *DAILY TANK* BERBASIS IOT
MENGUNAKAN METODE *NEURAL NETWORK*

Oleh :
Briham Maulidan
NIT. 30121029

Disetujui untuk diujikan pada :
Surabaya, 6 Agustus 2024

Pembimbing 1 : Dr. SLAMET HARIYADI, S.T., M.M.
NIP. 19630408 198902 1 001

Pembimbing 2 : DEWI RATNA SARI, S.E., M.M.
NIP. 19690609 199303 2 002



LEMBAR PENGESAHAN

RANCANG BANGUN KONTROL DAN MONITORING POMPA BAHAN BAKAR GENSET PADA *DAILY TANK* BERBASIS IOT MENGUNAKAN METODE *NEURAL NETWORK*

oleh :
Briham Maulidan
NIT. 30121029

Telah dipertahankan dan dinyatakan lulus pada Ujian Proyek Akhir
Program Pendidikan Diploma 3 Teknik Listrik Bandar Udara
Politeknik Penerbangan Surabaya
Pada tanggal : 6 Agustus 2024

Panitia Penguji :

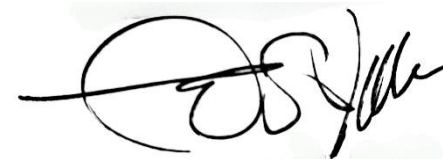
1. Ketua : RIFDIAN I.S., S.T., M.M., M.T.
NIP. 19810629 200912 1 002
2. Sekretaris : DEWI RATNA SARI, S.E., M.M.
NIP. 19690609 199303 2 002
3. Anggota : Dr. SLAMET HARIYADI, S.T., M.M.
NIP. 19630408 198902 1 001


.....

.....

.....

Ketua Program Studi
Diploma 3 Teknik Listrik Bandara



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

ABSTRAK

RANCANG BANGUN KONTROL DAN MONITORING POMPA BAHAN BAKAR GENSET PADA *DAILY TANK* BERBASIS IOT MENGUNAKAN METODE *NEURAL NETWORK*

Oleh:

Briham Maulidan

NIT.30121029

Penelitian ini bertujuan untuk merancang dan membangun sistem kontrol dan monitoring pompa bahan bakar genset pada *daily tank* berbasis Internet of Things (IoT) menggunakan metode *neural network*. Dalam operasional bandara, keandalan daya listrik sangat penting, dan genset berperan sebagai sumber daya cadangan yang krusial. Pengelolaan bahan bakar genset melalui *daily tank* harus efisien untuk menghindari kehabisan bahan bakar yang dapat mengganggu operasional.

Metode penelitian melibatkan penggunaan algoritma *neural network* backpropagation untuk mengatur kecepatan pompa berdasarkan data dari sensor ultrasonik dan sensor flowmeter. Sensor ultrasonik mengukur ketinggian bahan bakar dalam *daily tank*, sementara sensor flowmeter mengukur laju aliran bahan bakar yang digunakan genset. Data ini dikirimkan ke modul ESP32 untuk diproses oleh algoritma *neural network*, yang kemudian mengirimkan sinyal kontrol ke driver BTS7960 untuk mengatur kecepatan pompa. Sistem ini juga menyediakan monitoring real-time melalui LCD dan website.

Hasil penelitian menunjukkan bahwa sistem yang dirancang mampu memastikan pasokan bahan bakar yang kontinu dan efisien, serta meningkatkan keandalan dan keamanan operasional genset. Sistem ini dapat memantau dan mengatur pasokan bahan bakar secara real-time, adaptif terhadap beban kerja, dan mendeteksi potensi kerusakan, sehingga peralatan kritis tetap berfungsi optimal selama pemadaman listrik atau situasi darurat.

Kesimpulannya, sistem kontrol dan monitoring pompa bahan bakar genset berbasis IoT dengan metode *neural network* berhasil dirancang dan diuji, menunjukkan kemampuan untuk meningkatkan efisiensi dan keandalan operasional genset di bandara.

Kata Kunci: Kontrol Pompa, Monitoring Bahan Bakar, Genset, *Daily tank*, IoT, *Neural network*, *Backpropagation*.

ABSTRACT

PUMP CONTROL AND MONITORING DESIGN GENSET FUEL IN IOT BASED DAILY TANKS USING NEURAL NETWORK METHOD

By:

Briham Maulidan
NIT.30121029

This research aims to design and build a control and monitoring system for generator fuel pumps in daily tanks based on the Internet of Things (IoT) using the neural network method. In airport operations, the reliability of electrical power is very important, and generators act as a crucial backup power source. Generator fuel management through the daily tank must be efficient to avoid running out of fuel which can disrupt operations.

The research method involves using a backpropagation neural network algorithm to regulate pump speed based on data from ultrasonik sensors and flowmeter sensors. The ultrasonik sensor measures the fuel level in the daily tank, while the flowmeter sensor measures the fuel flow rate used by the generator. This data is sent to the ESP32 module to be processed by the neural network algorithm, which then sends control signals to the BTS7960 driver to regulate the pump speed. This system also provides real-time monitoring via LCD and website.

The research results show that the designed system is able to ensure continuous and efficient fuel supply, as well as increasing the operational reliability and safety of the generator. This system can monitor and regulate fuel supply in real-time, be adaptive to workloads, and detect potential damage, so that critical equipment continues to function optimally during power outages or emergency situations.

In conclusion, an IoT-based generator fuel pump control and monitoring system using the neural network method was successfully designed and tested, showing the ability to increase the operational efficiency and reliability of generators at airports.

Keywords: Pump Control, Fuel Monitoring, Generator, Daily tank, IoT, Neural network, Backpropagation.

PERNYATAAN KEASLIAN DAN HAK CIPTA

Saya yang bertanda tangan di bawah ini :

Nama : Briham Maulidan
NIT : 30121029
Program Studi : D3 Teknik Listrik Bandara
Judul Proyek Akhir : Rancang Bangun Kontrol Dan Monitoring
Pompabahan Bakar Genset Pada *Daily tank*
Berbasis Iot Menggunakan Metode *Neural network*

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 Perguruan Tinggi dan Akademi Penerbangan.

Surabaya, 6 Agustus 2024
Yang membuat pernyataan



Briham Maulidan
NIT. 30121029

KATA PENGANTAR

Puji syukur kehadiran Allah SWT, yang telah melimpahkan nikmat dan karunia-Nya kepada penulis sehingga dapat menyelesaikan penulisan Proyek Akhir ini dengan baik. Proyek Akhir yang berjudul “RANCANG BANGUN KONTROL DAN MONITORING POMPA BAHAN BAKAR GENSET PADA *DAILY TANK* BERBASIS IOT MENGGUNAKAN ALGORITMA *NEURAL NETWORK*”, Penyusunan Proyek Akhir ini dimaksudkan sebagai salah satu syarat menyelesaikan pendidikan di Perguruan Tinggi dan Akademi Penerbangan dan memperoleh gelar Ahli Madya (A.Md.).

Tak lupa penulis ucapkan terima kasih kepada semua pihak yang telah membantu dalam penyusunan Proyek Akhir ini, antara lain :

1. Bapak Ahmad Bahrawi, S.E., M.T. selaku Direktur Politeknik Penerbangan Surabaya.
2. Bapak Dr. Gunawan Sakti, S.T., M.T. selaku Ketua Program Studi Teknik Listrik Bandar Udara Politeknik Penerbangan Surabaya.
3. Bapak Dr. Slamet Hariyadi, S.T., M.M. Selaku Pembimbing 1 yang senantiasa membimbing dan membantu dalam perancangan alat
4. Ibu Dewi Ratna Sari, S.E., M.M. Selaku Pembimbing 2 yang senantiasa membimbing dan membantu dalam penulisan laporan proyek akhir.
5. Para Dosen dan Instruktur Politeknik Penerbangan Surabaya yang telah memberikan ilmu.
6. Almarhum Bapak Agus Basuki yang menjadi motivasi terbaik dan ibu Qumaidah beserta saudara yang selalu memberikan kasih sayang dan doa.
7. Teman-teman beserta adik tingkat TLB yang telah menyumbangkan pikiran, saran, dan motivasi serta memberikan hiburan.
8. Serta semua pihak yang tidak bisa penulis sebutkan satu persatu yang telah membantu penulisan Proyek Akhir ini.

Penulis menyadari bahwa Proyek Akhir ini masih jauh dari kata sempurna dan masih banyak kekurangan. Hal tersebut dikarenakan keterbatasan ilmu dan pengetahuan yang dimiliki penulis semata. Oleh karena itu penulis mengharapkan kritik dan saran dari pembaca untuk menyempurnakan penulisan Proyek Akhir ini. Semoga Proyek Akhir ini dapat memberikan manfaat bagi semua pihak yang membutuhkan.

Surabaya, 22 Juli 2024
Penulis

BRIHAM MAULIDAN
NIT. 30121029

DAFTAR ISI

	Halaman
HALAMAN JUDUL	ii
LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
ABSTRAK	iv
ABSTRACT	v
PERNYATAAN KEASLIAN DAN HAK CIPTA.....	vi
KATA PENGANTAR.....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	x
DAFTAR TABEL	xi
DAFTAR LAMPIRAN	xii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	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	4
BAB II LANDASAN TEORI	6
2.1 Teori Penunjang	6
2.1.1 <i>Neural network</i>	6
2.1.2 Algoritma Backpropagation	8
2.1.3 <i>Google colab</i>	9
2.1.4 <i>Thingspeak</i>	10
2.1.5 Arduino IDE	12
2.1.6 Generator set (Genset).....	12
2.1.7 Tanki Bahan Bakar (<i>Daily tank</i>)	13
2.1.8 Motor DC	14
2.1.9 Pompa DC	16
2.1.10 ESP32	17
2.1.11 Power Supply	18
2.1.12 BTS7960.....	20
2.1.13 LCD	21
2.1.14 Sensor Ultrasonik HC-SR04	22
2.1.15 Sensor Flowmeter	23
2.2 Kajian Penelitian Terdahulu.....	25
BAB III METODE PENELITIAN	27
3.1 Desain Penelitian.....	27
3.2 Perancangan Alat	29
3.2.1 Desain Alat	29
3.2.1 Cara Kerja Alat.....	29
3.2.2 Komponen Alat	32
3.3 Teknik Pengujian	34
3.3.1 Pengujian Power supply	35

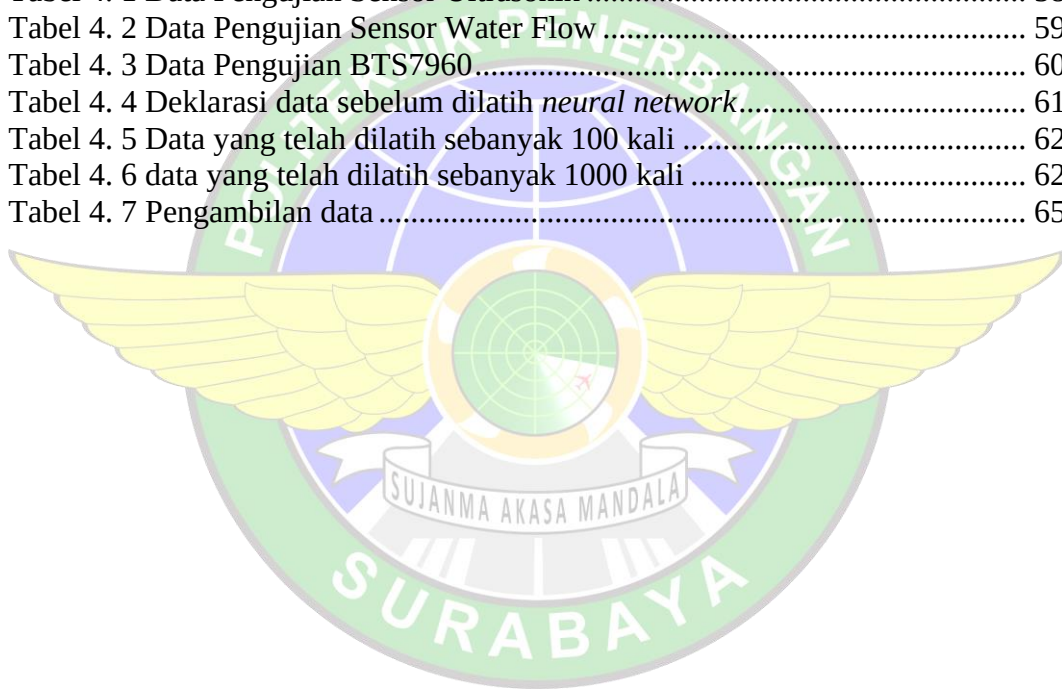
3.3.2	Pengujian ESP32	35
3.3.3	Pengujian Pompa DC	36
3.3.4	Pengujian Sensor Ultrasonik HC-SR04	36
3.3.5	Pengujian Sensor Water Flow	37
3.3.6	Pengujian LCD	37
3.3.7	Pengujian BTS7960.....	38
3.3.8	Pengujian dan Pelatihan Algoritma <i>Neural network</i>	38
3.4	Teknik Analisis Data	38
3.5	Tempat dan Waktu Penelitian	39
BAB IV	HASIL DAN PEMBAHASAN.....	40
4.1	Hasil Penelitian	40
4.2	Analisis (<i>Analyze</i>)	40
4.3	Desain (<i>Design</i>).....	41
4.3.1	Persiapan komponen perangkat keras	41
4.3.2	Desain Alat	42
4.4	Pengembangan (<i>Development</i>).....	42
4.4.1	Pembuatan Perangkat Keras	42
4.4.2	Pembuatan Perangkat Lunak	47
4.5	Implementasi (<i>Implementation</i>)	52
4.5.1	Pengujian Power Supply	52
4.5.2	Pengujian ESP32	54
4.5.3	Pengujian Pompa DC	55
4.5.4	Pengujian LCD 12x2 I2c.....	56
4.5.5	Pengujian Sensor Ultrasonik HC-SR04	57
4.5.6	Pengujian Sensor Water Flow	58
4.5.7	Pengujian BTS7960.....	60
4.5.8	Pengujian Metode <i>Neural Network</i>	61
4.5.9	Pengujian Sistem Terintegrasi.....	63
4.5.10	Pengujian IOT ThingSpeak	65
4.6	Evaluasi (<i>Evaluation</i>).....	67
BAB V	PENUTUP	70
5.1	Simpulan	70
5.2	Saran.....	71
	DAFTAR PUSTAKA	72
	LAMPIRAN.....	A-1

DAFTAR GAMBAR

	Halaman
Gambar 1. 1 Latar belakang	2
Gambar 2. 1 Prinsip <i>neural network</i>	7
Gambar 2. 2 Prinsip Algoritma <i>Backpropagation</i>	9
Gambar 2. 3 <i>Google colab</i>	10
Gambar 2. 4 Generator set	13
Gambar 2. 5 Pompa Motor DC	17
Gambar 2. 6 ESP32	18
Gambar 2. 7 Power Supply DC.....	19
Gambar 2. 8 BTS7960	21
Gambar 2. 9 LCD	21
Gambar 2. 10 Sensor ultrasonik JSN-SR04T.....	23
Gambar 2. 11 Sensor Water Flow YF-S201	25
Gambar 3. 1 Desain Penelitian.....	27
Gambar 3. 2 Diagram blok desain alat.....	29
Gambar 3. 3 Wiring Diagram Alat.....	30
Gambar 3. 4 <i>Flowchart</i> cara kerja alat.....	31
Gambar 4. 1 Sirkuit PCB	43
Gambar 4. 2 PCB setelah dicetak.....	45
Gambar 4. 3 Komponen yang telah dirakit	46
Gambar 4. 4 Tangki bahan bakar buatan dan pompa.....	47
Gambar 4. 5 Deklarasi data <i>neural network</i>	48
Gambar 4. 6 Jumlah pelatihan data <i>neural network</i>	48
Gambar 4. 7 Pelatihan data <i>neural network</i>	49
Gambar 4. 8 Pembuatan program.....	50
Gambar 4. 9 Pemindahan bobot dan bias <i>neural network</i>	50
Gambar 4. 10 Pengisian alamat kode mqtt thingspeak	51
Gambar 4. 11 Pengiriman data ke thingspeak dan LCD.....	52
Gambar 4. 12 Pengujian Power Supply 5VDC.....	53
Gambar 4. 13 Pengujian Power Supply 12VDC	54
Gambar 4. 14 Pengujian ESP32	55
Gambar 4. 15 Pengujian Pompa DC	55
Gambar 4. 16 Pengujian LCD	56
Gambar 4. 17 Pengujian Sensor Ultrasonik	57
Gambar 4. 18 Pembacaan Hasil Sensor Ultrasonik	58
Gambar 4. 19 Pengujian BTS7960.....	60
Gambar 4. 20 Percobaan pelatihan data	61
Gambar 4. 21 Pengujian alat	64
Gambar 4. 22 Pengujian thingspeak.....	66

DAFTAR TABEL

	Halaman
Tabel 2. 1 Kajian Penelitian Terdahulu.....	25
Tabel 3. 1 Pengujian <i>power supply</i>	35
Tabel 3. 2 Pengujian ESP32.....	35
Tabel 3. 3 Pengujian pompa DC	36
Tabel 3. 4 Pengujian Sensor Ultrasonik.....	36
Tabel 3. 5 Pengujian sensor water flow	37
Tabel 3. 6 Pengujian LCD.....	37
Tabel 3. 7 Pengujian BTS7960	38
Tabel 3. 8 Pengujian dan Pelatihan Algoritma <i>neural network</i>	38
Tabel 3. 9 Waktu Penelitian dan perencanaan	39
Tabel 4. 1 Data Pengujian Sensor Ultrasonik	58
Tabel 4. 2 Data Pengujian Sensor Water Flow	59
Tabel 4. 3 Data Pengujian BTS7960.....	60
Tabel 4. 4 Deklarasi data sebelum dilatih <i>neural network</i>	61
Tabel 4. 5 Data yang telah dilatih sebanyak 100 kali	62
Tabel 4. 6 data yang telah dilatih sebanyak 1000 kali	62
Tabel 4. 7 Pengambilan data	65



DAFTAR LAMPIRAN

	Halaman
Lampiran A. Standard Operational Procedure (SOP)	A-1
Lampiran B. Program Alat	B-1
Lampiran C. Daftar Riwayat Hidup	C-1



DAFTAR PUSTAKA

- Aisyah. (2020). SISTEM MONITORING ALAT KETINGGIAN BAHAN BAKAR. *Jurnal Sriwijaya University* .
- Alhanjouri, M. (2015). Speed Control of DC Motor Using Artificial Neural . *International Journal of Science and Research (IJSR)*, 2140-2148.
- Ardiliansyah, A. R. (2019). Rancang Bangun Prototipe Pompa Otomatis Dengan Fitur Monitoring Berbasis IoT Menggunakan Sensor Flow Meter dan Ultrasonik. *Jurnal Keilmuan dan aplikasi teknik Informatika*, 59-67.
- Awalia, S. (2021). Rancangan Pengisian Bahan Bakar Genset Otomatis dalam Meningkatkan Kinerja Teknisi di Bandar Udara Gusti Sjamsir Alam. *Journal of Airport Technology*.
- Damayanti, S. (2017). SISTEM MONITORING VOLUME AIR MENGGUNAKAN SENSOR ULTRASONIK DAN MONITORING OUTPUT VOLUME AIR MENGGUNAKAN FLOW METER BERBASIS ARDUINO. *GEMA TEKNOLOGI*, 6-9.
- Djalal. (2017). Kontrol Kecepatan Motor Induksi menggunakan Algoritma Backpropagation *Neural network*. *Jurnal ELKOMIKA*, 138 - 148.
- kabel, S. (2016). *katalog sutrado kabel* . Diambil kembali dari Sutrado kabel: sutrakabel.com
- Kumar, S. B. (2014). Design and Simulation of Speed Control of DC Motor by Artificial Neural network Technique. *International Journal of Scientific and Research Publications*, .
- Kurniawan, A. (2015). STRATEGI KENDALI KECEPATAN MOTOR INDUKSI MENGGUNAKAN PWM INVERTER BERBASIS JARINGAN SARAF TIRUAN. *Jurnal ITS*.
- Kusumah, H. (2019). PENERAPAN TRAINER INTERFACING MIKROKONTROLER DAN INTERNET OF THINGS BERBASIS ESP32 PADA MATA KULIAH INTERFACING. *Jurnal Cerita*, 120-134.
- MadhusudhanaRao, G. (2009). A Neural network Based Speed Control for DC . *International Journal of Recent Trends in Engineering*, 121-124.

- Muhammad, F. (2020). Sistem Kendali Sliding Roof untuk Smarthome Berbasis Internet of Things. *Jurnal Teknik Elektro Indonesia*, 135-138.
- Nurdiansyah, E. D. (2018). Pengendalian Kecepatan Motor Induksi Satu Fasa Berbasis Backpropagation. *Jurnal Teknik Elektro*, 69-76.
- Nurmila, N. (2010). ALGORITMA BACK PROPAGATION *NEURAL NETWORK* UNTUK PENGENALAN POLA KARAKTER HURUF JAWA . *Jurnal Masyarakat Informatika*,.
- Santoso, R. B. (2021). Rancang Bangun Smarthome Berbasis QR Code Dengan Mikrokontroller Module ESP32. *JASEE*, 47-60.
- Wahyudi. (2008). Pengendalian Motor DC Menggunakan Jaringan Syaraf Tiruan Backpropagation . *seminar nasional informatika*.
- Widiasari, C. (2019). Sistem Monitoring Tangki dan Penghitung RunHour Genset Otomatis Berbasis Internet of Things (IoT). *Jurnal ELEMENTER*, 59-70.
- Wiryadinata, R. (2005). SIMULASI JARINGAN SYARAF TIRUAN BERBASIS METODE BACKPROPAGATION SEBAGAI PENGENDALI KECEPATAN MOTOR DC. *Seminar Nasional Aplikasi Teknologi Informasi*, 29-36.
- Yoski, M. S. (2020). Prototipe Robot Pembersih Lantai Berbasis Mikrokontroller dengan Sensor Ultrasonik. *Jurnal Teknik Elektro Indonesia*, 158-161.

LAMPIRAN

Lampiran A. Standard Operational Procedure (SOP)

RANCANG BANGUN KONTROL DAN MONITORING POMPA BAHAN BAKAR GENSET PADA *DAILY TANK* BERBASIS IOT MENGUNAKAN METODE *NEURAL NETWORK*

Oleh:

BRIHAM MAULIDAN
NIT. 30121029

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 Pompa Bahan Bakar Genset Pada *Daily tank* Berbasis Iot Menggunakan Metode *Neural network* ” sebagai berikut:

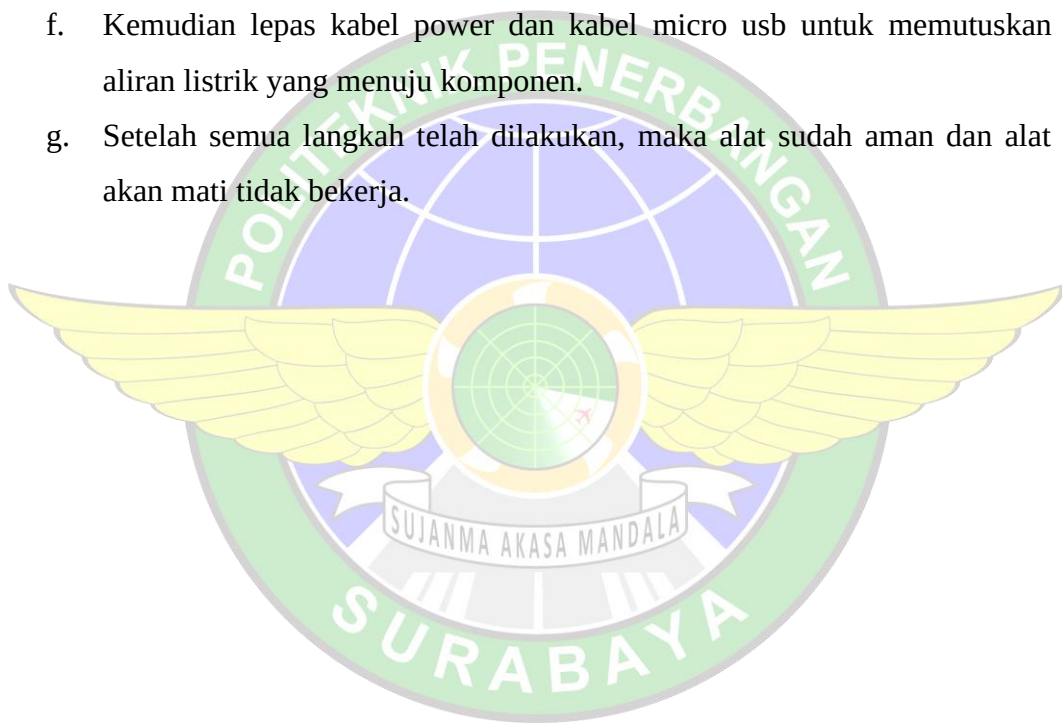
1. Mengoperasikan Alat Sesuai dengan SOP
 - a. Sambungkan kabel power untuk menghidupkan *power supply* dan sambungkan juga kabel *usb type-c* untuk menghidupkan ESP32.
 - b. Pastikan semua komponen dapat menyala sesuai fungsinya masing-masing.
 - c. Kemudian nyalakan *hotspot* sehingga nantinya dapat termonitoring oleh layar LCD untuk menampilkan data kapasitas tangki 1, debit air yang

keluar, kecepatan pompa dan dapat juga termonitoring pada *Website Thingspeak*.

- d. Jika alat sudah siap untuk digunakan, maka alat akan bekerja sesuai apa yang telah diperintahkan, kemudian hasil pengukuran dapat secara langsung termonitoring pada layar LCD dan Website Thingspeak secara real-time.

2. Mematikan Alat Sesuai SOP

- e. Matikan hotspot untuk memutus koneksi antara alat dan program yang telah ter setting, serta memutuskan pemantauan pada Website Thingspeak.
- f. Kemudian lepas kabel power dan kabel micro usb untuk memutuskan aliran listrik yang menuju komponen.
- g. Setelah semua langkah telah dilakukan, maka alat sudah aman dan alat akan mati tidak bekerja.



Lampiran B. Program Alat

Program Alat

Master

```
// Include Library
#include <LiquidCrystal_I2C.h>
#include <PubSubClient.h>
#include <Arduino.h>
#include <WiFi.h>
// end

// deklarasi PIN
#define PINTrig1 5
#define PINEcho1 4
#define PINTrig2 19
#define PINEcho2 18

#define PIN_FLOW_1 25
#define PIN_FLOW_2 26

#define PINMOTOR 27

#define PIN_RELAY 33
#define RELAY_ON digitalWrite(PIN_RELAY, HIGH)
#define RELAY_OFF digitalWrite(PIN_RELAY, LOW)

const int Motor = 1;
const int freq = 30000;
const int resolution = 8;
// end

// thinkspeak
// bmaulidan@gmail.com
// Briham09062002

// Set I2C LCD
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

```

static unsigned long prevMillisLCD = 0;
int lcd_count = 0;
// end

// ----- set val. untuk wifi -----//
const char* ssid = "me";
const char* password = "123456789";
String status_send = "ERR";
// ----- end -----//

// Ganti dengan Channel ID dan MQTT API Key ThingSpeak Anda
const char* mqttServer = "mqtt3.thingspeak.com"; //server mqtt
const int mqttPort = 1883; //port khusus untuk MQTT
const char* mqttUser = "NAAjOgMJBAMbAAoFGTsGNTY"; //client id dari web thingspeak
const char* mqttPassword = "6mxkOefBrc328+L52d3dGHQb"; //dari web thingspeak juga
const char* channelID = "NAAjOgMJBAMbAAoFGTsGNTY";

String topicUltrasonik1 = "channels/2557386/publish/fields/field1"; //kodenya didapat dari web
thingspeak di channel
String topicFlow1 = "channels/2557386/publish/fields/field3";

String topicUltrasonik2 = "channels/2557386/publish/fields/field2";
String topicFlow2 = "channels/2557386/publish/fields/field4";
// end

WiFiClient espClient;
PubSubClient client(espClient); // setup ke thingspeak

// set val. Flow 1
float flowRate;
int pulseCount = 0;
byte pulse1Sec = 0;
int interval = 1000;
unsigned int flowMilliLitres;
float calibrationFactor = 7.5;
unsigned long totalMilliLitres;
unsigned long prevMillisFlow = 0;
// end

```

```

// set val. Flow 2
float flowRateV2;
int pulseCountV2 = 0;
byte pulse1SecV2 = 0;
int intervalV2 = 1000;
unsigned int flowMilliLitresV2;
float calibrationFactorV2 = 7.5;
unsigned long totalMilliLitresV2;
unsigned long prevMillisFlowV2 = 0;
// end

// set NN
// note lakukan pelatihan pada platfrom lain, kita tidak menyediakan program training dengan
// platfrom arduino

// Deklarsi Aturan Layer NN
#define INPUT_NODES 2 //dimensi
#define HIDDEN_NODES_1 10 //jumlah hidden neuron
#define HIDDEN_NODES_2 10
#define OUTPUT_NODES 1
// end

// Deklarasikan bobot dan bias untuk layer 1
const float hidden_layer_1_weights[INPUT_NODES][HIDDEN_NODES_1] = {
    {0.40265924, -0.08351019, 0.56558716, 0.4951658, 0.33686078, -0.1695892, 0.19712876,
0.33150977, 0.4953584, 0.47451574},
    {-0.5395516, 0.6058981, -0.25323987, -0.12827243, 0.5892859, 0.17590149, 0.36011717,
0.36379483, -0.4400219, 0.40448374}
};

const float hidden_layer_1_biases[HIDDEN_NODES_1] = {0.0, -0.07285871, -0.05382881,
0.22757775, 0.11516502, 0.21700531, 0.19211927, -0.00568496, 0.11714573, -0.06883631};
// end

// Deklarasikan bobot dan bias untuk layer 2
const float hidden_layer_2_weights[HIDDEN_NODES_1][HIDDEN_NODES_2] = {

```



```

    {-0.4874986 , -0.18435732, -0.31884363, -0.17403322, 0.07529855, -0.16233116, 0.1141749
    , -0.12946367, 0.48926115, 0.32538658},
    { 0.24591975, -0.49102995, 0.13748175, 0.06210613, 0.3861366 , -0.5420647 ,
    0.41689754, -0.59784555, -0.6337946 , -0.23644125},
    {0.5171232 , 0.44528276, 0.01353866, -0.07103148, -0.5636177 , 0.15167958, -0.28022867,
    0.07526711, -0.5664585 , 0.37159777},
    {0.02645781, -0.5302838 , -0.08838183, -0.49026445, 0.30385402, -0.54102296, 0.4254442
    , -0.2391927 , 0.15068227, -0.5410451},
    {0.59192955, 0.04979539, -0.38393652, 0.1378578 , -0.08714522, 0.13741863, 0.05559803,
    0.19693016, 0.33136103, 0.49326718},
    {0.11497378, 0.01595736, -0.545824 , 0.24241495, 0.4216509 , 0.22039211, -0.32363546,
    0.69439334, 0.5868569 , -0.2982845},
    {0.06783885, -0.23114854, -0.11281365, -0.1022225 , 0.14243616, -0.10414475,
    0.10691227, 0.52596205, -0.39722505, -0.19817802},
    {0.42325673, -0.0544917 , -0.23466331, 0.0319283 , -0.03105006, 0.32362336, 0.24156153,
    -0.158064 , 0.2668916 , 0.2158683},
    {-0.53101784, 0.4331991 , -0.38552788, 0.18129426, 0.41616997, -0.3475023 , -
    0.34796187, -0.34593785, 0.6399918 , 0.20561785},
    {0.47942606, 0.41726512, 0.26097417, 0.03909653, -0.23527269, -0.00476366, -
    0.47715056, 0.09945424, 0.43866044, -0.5287417}
    };

    const float hidden_layer_2_biases[HIDDEN_NODES_2] = {-0.08590125, 0.0, 0.0, 0.0,
    0.1844298 , 0.0, 0.18688062, 0.1808284 , 0.18289629, 0.0};

    // end

    // Deklarasikan bobot dan bias untuk layer ouput
    const float output_layer_weights[HIDDEN_NODES_2][OUTPUT_NODES] = {
        {-0.72145325},
        {-0.3311571},
        {0.18775147},
        {0.29326802},
        {0.625173},
        {-0.6798727},
        {0.32286027},
        {0.70761454},
        {0.3876719},
    }

```

```

    {0.38301057}
};

const float output_layer_biases[OUTPUT_NODES] = {0.17211057};
// end

int countinggg = 0;
int countinglcd = 0;
unsigned long previousMillis = 0;
unsigned long previousMillisLCD = 0;
const long intervalSend = 3000; //waktu untuk pengiriman
float output; //hasil NN
float predicted_speed; //kecepatan PWM

int TinggiTank1 = 26;
int TinggiTank2 = 26;

float maxDataTraining[2] = {25, 25}; //disesuaikan di Google colab

void IRAM_ATTR pulseCounter() {
    pulseCount++; //fungsi sinyal interrupt untuk menghitung flowmeter
}

void IRAM_ATTR pulseCounterV2() {
    pulseCountV2++;
}

void setup() {
    Serial.begin(9600);
    lcd.begin();
    lcd_setup(1);
    pinMode(PINTrig1, OUTPUT);
    pinMode(PINEcho1, INPUT);
    pinMode(PINTrig2, OUTPUT);
    pinMode(PINEcho2, INPUT);
    pinMode(PIN_RELAY, OUTPUT);
    ledcSetup(Motor, freq, resolution);

```

```

ledcAttachPin(PINMOTOR, Motor);

// set flow 1
pinMode(PIN_FLOW_1, INPUT_PULLUP);
attachInterrupt(digitalPinToInterrupt(PIN_FLOW_1), pulseCounter, FALLING);
pulseCount = 0;
flowRate = 0.0;
flowMilliLitres = 0;
totalMilliLitres = 0;
prevMillisFlow = 0;
// end

// set flow 2
pinMode(PIN_FLOW_2, INPUT_PULLUP);
attachInterrupt(digitalPinToInterrupt(PIN_FLOW_2), pulseCounterV2, FALLING);
pulseCountV2 = 0;
flowRateV2 = 0.0;
flowMilliLitresV2 = 0;
totalMilliLitresV2 = 0;
prevMillisFlowV2 = 0;
// end

lcd_setup(2);
WiFi.mode(WIFI_STA);
WiFi.begin(ssid, password);
Serial.print("Connecting to WiFi ..");
while (WiFi.status() != WL_CONNECTED) {
    Serial.print('.');
}
Serial.println("Connected");
lcd_setup(3);
lcd_setup(4);
client.setServer(mqttServer, mqttPort);
client.setCallback(callback);
while (!client.connected()) {

    if (client.connect(channelID, mqttUser, mqttPassword)) {

```

```

    Serial.println("connected");
} else {
    Serial.print("failed with state ");
    Serial.println(client.state());
    delay(2000);
}
}

lcd_setup(5);
lcd_setup(6);
}

void xloop(){ //buat ngetes x nya diilangin
    prosesDataNN(10.0, 10.0);
    delay(1000);
}

void loop() { //main program
    client.loop();
    unsigned long currentMillis = millis(); //pembacaan detik untuk pengiriman data
    unsigned long currentMillisLCD = millis(); // slide
    float tabung_1 = ReadUltra(PINEcho1, PINTrig1, TinggiTank1);
    float tabung_2 = ReadUltra(PINEcho2, PINTrig2, TinggiTank2);
    ReadFlow_1();
    ReadFlow_2();
    if (tabung_1 <= 10){ // nilai minimal pompa menyala
        RELAY_ON;
        ledcWrite(Motor, predicted_speed);
    }
    else {
        RELAY_OFF;
        ledcWrite(Motor, 0);
    }
    if (currentMillisLCD - previousMillisLCD >= 10000) { //pindah slide tiap 10 detik
        previousMillisLCD = currentMillisLCD;
        countinglcd++;
    }
    float nilaicm3_1 = 3.14 * 81 * tabung_1;
    float nilaicm3_2 = 3.14 * 81 * tabung_2;

```

```

if (countinglcd == 1){

    lcd.setCursor(0, 0);
    lcd.print("U1:" + String(nilaicm3_1, 0) + " ");
    lcd.setCursor(8, 0);
    lcd.print("U2:" + String(nilaicm3_2, 0) + " ");
    lcd.setCursor(0, 1);
    lcd.print("F1:" + String(flowRate, 1) + " ");
    lcd.setCursor(8, 1);
    lcd.print("F2:" + String(flowRateV2, 1) + " ");
}
else if (countinglcd == 2){
    lcd.setCursor(0, 0);
    lcd.print("Predict:" + String(output));
    lcd.setCursor(0, 1);
    lcd.print("PWM:" + String(predicted_speed));
}
else if (countinglcd == 3){
    countinglcd = 1;
}
if (currentMillis - previousMillis >= intervalSend) {
    previousMillis = currentMillis;
    countinggg++;
    lcd.begin();
    prosesDataNN(flowRateV2, tabung_1);
    sendData(nilaicm3_1, nilaicm3_2);
}

}

```

Thingspeak

```

void callback(char* topic, byte* payload, unsigned int length) {
    // Fungsi callback jika ada pesan masuk
    Serial.print("Message arrived [");
    Serial.print(topic);
    Serial.print("] ");
    for (int i = 0; i < length; i++) {

```

```

    Serial.print((char)payload[i]);
}
Serial.println();
}
void reconnect() {
    // Loop sampai terhubung kembali ke MQTT
    while (!client.connected()) {
        Serial.print("Attempting MQTT connection...");
        if (client.connect("ESP32Client", mqttUser, mqttPassword)) {
            Serial.println("connected");
        } else {
            Serial.print("failed, rc=");
            Serial.print(client.state());
            delay(2000);
        }
    }
}

void sendData(float tabung_1, float tabung_2){
    if (countinggg == 1){
        Serial.println("Send Data Ultra1");
        client.publish(topicUltrasonik1.c_str(), String(tabung_1).c_str());
    }
    else if (countinggg == 2){
        Serial.println("Send Data Flow1");
        client.publish(topicFlow1.c_str(), String(totalMilliLitres).c_str());
    }
    else if (countinggg == 3){
        Serial.println("Send Data Ultra2");
        client.publish(topicUltrasonik2.c_str(), String(tabung_2).c_str());
    }
    else if (countinggg == 4){
        Serial.println("Send Data Flow2");
        client.publish(topicFlow2.c_str(), String(totalMilliLitresV2).c_str());
    }
    else if (countinggg >= 5){
        Serial.println("Reset Data");
    }
}

```



```

        countinggg = 0;
    }
}

Neural network

float relu(float x) {
    return x > 0 ? x : 0;
}

float inverse_transform(float x, float min_val, float max_val) {
    return x * (max_val - min_val) + min_val;
}

void forward_pass(float input[INPUT_NODES], float &output) {
    // Hidden layer 1 calculations
    float hidden_layer_1_output[HIDDEN_NODES_1];
    for (int i = 0; i < HIDDEN_NODES_1; i++) {
        hidden_layer_1_output[i] = hidden_layer_1_biases[i];
        for (int j = 0; j < INPUT_NODES; j++) {
            hidden_layer_1_output[i] += input[j] * hidden_layer_1_weights[j][i];
        }
        hidden_layer_1_output[i] = relu(hidden_layer_1_output[i]);
    }

    // Hidden layer 2 calculations
    float hidden_layer_2_output[HIDDEN_NODES_2];
    for (int i = 0; i < HIDDEN_NODES_2; i++) {
        hidden_layer_2_output[i] = hidden_layer_2_biases[i];
        for (int j = 0; j < HIDDEN_NODES_1; j++) {
            hidden_layer_2_output[i] += hidden_layer_1_output[j] * hidden_layer_2_weights[j][i];
        }
        hidden_layer_2_output[i] = relu(hidden_layer_2_output[i]);
    }

    // Output layer calculations
    output = output_layer_biases[0];
    for (int i = 0; i < HIDDEN_NODES_2; i++) {
        output += hidden_layer_2_output[i] * output_layer_weights[i][0];
    }
}

```

```

float convertToRange(float x, float minVal, float maxVal) {
    return (x - minVal) / (maxVal - minVal);
}

void prosesDataNN(float debit, float tinggi){
    float input[INPUT_NODES] = {debit, tinggi}; //mengelompokkan inputan
    float newData[INPUT_NODES];

    for (int i = 0; i < INPUT_NODES; i++) {
        newData[i] = convertToRange(input[i], 0, maxDataTraining[i]); // dinormalisasi atau merubah
        nilai range 0-25 menjadi 0-1
    }
    forward_pass(newData, output);
    if (output < 0) output = 0.0;
    else if (output > 1) output = 1.0;
    Serial.println("Nilai Predict: " + String(output));
    predicted_speed = inverse_transform(output, 10, 255); // diubah dari 0-1 menjadi 10-255 sesuai
    kecepatan motor
    if (predicted_speed < 0) predicted_speed = 0.0;
    else if (predicted_speed > 255) predicted_speed = 255;
    Serial.println("Nilai RPM: " + String(predicted_speed));
}

```

Sensor

```

float ReadUltra(int echoPin, int trigPin, int tank){
    long duration;
    float distance;
    digitalWrite(trigPin, LOW);
    delayMicroseconds(5);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    duration = pulseIn(echoPin, HIGH);
    distance = (duration * 0.034) / 2;
    float tinggiAir = tank - distance;
    if (tinggiAir <= 0){
        tinggiAir = 0;
    }
}

```

```

}

return tinggiAir;
}

void ReadFlow_1(){
    unsigned long currentMillisFlow = millis();
    if (currentMillisFlow - prevMillisFlow > interval) {
        pulse1Sec = pulseCount;
        pulseCount = 0;
        flowRate = ((1000.0 / (millis() - prevMillisFlow)) * pulse1Sec) / calibrationFactor;
        prevMillisFlow = millis();
        flowMilliLitres = (flowRate / 60) * 1000;
        totalMilliLitres += flowMilliLitres;
        // Serial.print("Flow rate: ");
        // Serial.print(int(flowRate));
        // Serial.print("L/min");
        // Serial.print("\t");

        // Serial.print("Output Liquid Quantity: ");
        // Serial.print(totalMilliLitres);
        // Serial.print("mL / ");
        // Serial.print(totalMilliLitres / 1000);
        // Serial.println("L");
    }
}

void ReadFlow_2(){
    unsigned long currentMillisFlowV2 = millis();
    if (currentMillisFlowV2 - prevMillisFlowV2 > intervalV2) {
        pulse1SecV2 = pulseCountV2;
        pulseCountV2 = 0;

        flowRateV2 = ((1000.0 / (millis() - prevMillisFlowV2)) * pulse1SecV2) / calibrationFactorV2;
        prevMillisFlowV2 = millis();
    }
}

```

```

flowMilliLitresV2 = (flowRateV2 / 60) * 1000;
totalMilliLitresV2 += flowMilliLitresV2;

// Serial.print("Flow rate V2: ");
// Serial.print(int(flowRateV2));
// Serial.print("L/min");
// Serial.print("\t");

// Serial.print("Output Liquid Quantity: ");
// Serial.print(totalMilliLitresV2);
// Serial.print("mL / ");
// Serial.print(totalMilliLitresV2 / 1000);
// Serial.println("L");
}
}

```

LCD

```

void lcd_mean(int a, String data) {
    int jml = data.length();
    int index_kosong = (16 - jml) / 2;
    if (a > 1) {
        index_kosong = index_kosong; //-4;
    }
    lcd.setCursor(index_kosong, a);
    lcd.print(data);
}

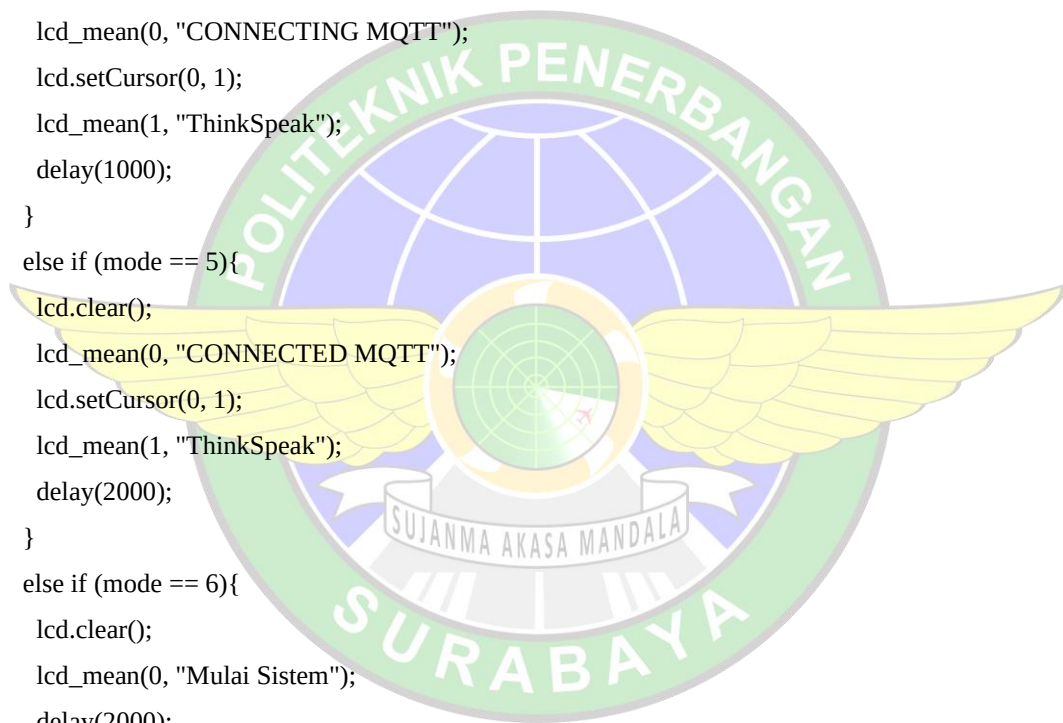
void lcd_setup(int mode){
    if (mode == 1){
        lcd_mean(0, "BHN BAKAR GENSET");
        lcd_mean(1, "MONITORING");
        delay(3000);
    }
    else if (mode == 2){
        lcd.clear();
        lcd_mean(0, "CONNECTING WIFI");
        lcd_mean(1, ssid);
    }
}

```

```

else if (mode == 3){
    lcd.clear();
    lcd_mean(0, "CONNECTED " + String(ssid));
    lcd.setCursor(0, 1);
    lcd.print("IP:");
    lcd.print(WiFi.localIP());
    delay(2000);
}
else if (mode == 4){
    lcd.clear();
    lcd_mean(0, "CONNECTING MQTT");
    lcd.setCursor(0, 1);
    lcd_mean(1, "ThinkSpeak");
    delay(1000);
}
else if (mode == 5){
    lcd.clear();
    lcd_mean(0, "CONNECTED MQTT");
    lcd.setCursor(0, 1);
    lcd_mean(1, "ThinkSpeak");
    delay(2000);
}
else if (mode == 6){
    lcd.clear();
    lcd_mean(0, "Mulai Sistem");
    delay(2000);
}
}
void lcd_Main(int menu){
}

```



Lampiran C. Daftar Riwayat Hidup

DAFTAR RIWAYAT HIDUP



Nama : Briham Maulidan
Nama Panggilan : Briham
Tempat, Tanggal Lahir : Jombang, 9 Juni 2002
Agama : Islam
Orang Tua : Agus Basuki dan Qumaidah
Saudara : Vylde Pavela dan Afry Rabel
Alamat : Perum Plandi Permai blok L no
9-10 RT 028/RW 006
Kecamatan Jomban Kabupaten
Jombang , Jawa Timur
Hobi : Bermain Game

Latar Belakang Pendidikan :

- 2009 – 2015 : SD Negeri Jombatan 3
- 2015 – 2018 : SMP Negeri 2 Jombang
- 2018 – 2021 : SMA Negeri 2 Jombang
- 2021 – 2024 : Politeknik Penerbangan Surabaya

On the Job Training

- (Mei 2023 – Sempetmber 2023) UPBU APT Pranoto, Samarinda
- (Oktober 2023 – Februari 2024) Bandara Internasional I Gusti Ngurah Rai, Bali

Sertifikat Kompetensi :

- ACS
- Transmisi Distribusi
- *Airfield Lighting System*