

**RANCANG BANGUN OPTIMALISASI *PHOTOVOLTAIC*
CHARGER MENGGUNAKAN *TEMPERATURE CONTROL*
BERBASIS *PROPORTIONAL INTEGRAL DERIVATIVE*
*CONTROLLER***

TUGAS AKHIR



Oleh :

ELSA MARCELA CANDRA DEVI
NIT: 30121031

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

RANCANG BANGUN OPTIMALISASI *PHOTOVOLTAIC CHARGER* MENGGUNAKAN *TEMPERATURE CONTROL* BERBASIS *PROPORTIONAL INTEGRAL DERIVATIVE CONTROLLER*

TUGAS AKHIR

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



Oleh:

ELSA MARCELA CANDRA DEVI
NIT: 30121031

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

HALAMAN PERSETUJUAN

“RANCANG BANGUN OPTIMALISASI *PHOTOVOLTAIC CHARGER*
MENGUNAKAN *TEMPERATURE CONTROL* BERBASIS *PROPORTIONAL*
INTEGRAL DERIVATIVE CONTROLLER”

Oleh:

ELSA MARCELA CANDRA DEVI
NIT: 30121031

Disetujui untuk diujikan pada:
Surabaya, 06 Agustus 2024

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

Pembimbing II : DWIYANTO, S.T., M.Pd.
NIP. 19690420 199103 1 004

HALAMAN PENGESAHAN

“RANCANG BANGUN OPTIMALISASI *PHOTOVOLTAIC CHARGER*
MENGUNAKAN *TEMPERATURE CONTROL* BERBASIS *PROPORTIONAL*
INTEGRAL DERIVATIVE CONTROLLER”

Oleh:

ELSA MARCELA CANDRA DEVI
NIT: 30121031

Telah dipertahankan dan dinyatakan lulus pada Sidang Tugas Akhir Program
Pendidikan Diploma III Teknik Listrik Bandara
Politeknik Penerbangan Surabaya
Pada tanggal: 06 Agustus 2024

Panitia Penguji:

1. Ketua : RIFDIAN I.S., S.T., M.M., M.T.
NIP. 19810629 200912 1 002
2. Sekretaris : DWIYANTO, S.T., M.Pd.
NIP. 19690420 199103 1 004
3. Anggota : Dr. SLAMET HARIYADI, S.T., M.M.
NIP. 19630408 198902 1 001


.....

.....

.....

Ketua Program Studi
D.III TEKNIK LISTRIK BANDARA



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

ABSTRAK

RANCANG BANGUN OPTIMALISASI *PHOTOVOLTAIC CHARGER* MENGUNAKAN *TEMPERATURE CONTROL* BERBASIS *PROPORTIONAL* *INTEGRAL DERIVATIVE CONTROLLER*

Oleh:

ELSA MARCELA CANDRA DEVI
NIT: 30121031

Penelitian ini bertujuan untuk merancang dan membangun sistem *photovoltaic charger* yang dioptimalisasikan menggunakan *temperature control proportional integral derivative controller*. Sistem ini dibuat untuk mengefisiensi pengisian daya *charger* dengan mengontrol suhu *photovoltaic charger* yang sering mempengaruhi kinerja dan daya pakai sistem. Alasan memilih sistem kontrol PID karena kinerja dan kemampuannya cepat dan stabil dalam mengendalikan suhu.

Dalam penelitian ini melibatkan penggunaan *solar cell* untuk mengubah energi matahari menjadi energi listrik, *solar charger controller* untuk mengatur arus dan tegangan yang dihasilkan panel surya lalu menuju *battery* untuk mengisi. Dari *battery* menyuplai *load* dan modul *stepdown* untuk menurunkan tegangan lalu digunakan untuk menyuplai arduino. Arduino akan membaca data tegangan dan arus dari sensor *ina219* dan suhu dari sensor *ds18b20*. Lalu menampilkannya di *lcd20x4*. Arduino juga akan mengontrol modul *relay* yang menuju pompa untuk mengaktifkan serta menonaktifkan pompa dengan algoritma PID berdasarkan input suhu yang terbaca untuk mengatur suhu *solar cell* dengan media air melalui semprotan air dari pompa.

Hasil penelitian ini menunjukkan bahwa sistem yang dirancang dapat meningkatkan efisiensi pengisian dan memperpanjang umur pakai *photovoltaic charger* dan mampu menjaga temperatur suhu dalam batas yang diinginkan.

Kata Kunci : *Photovoltaic charger, Proportional Integral Derivative controller, temperature control*

ABSTRACT

PHOTOVOLTAIC CHARGER OPTIMIZATION DESIGN USING TEMPERATURE CONTROL BASED ON PROPORTIONAL INTEGRAL DERIVATIVE CONTROLLER

By:

ELSA MARCELA CANDRA DEVI

NIT: 30121031

This research aims to design and build a photovoltaic charger system that is optimized using a temperature control proportional integral derivative controller. This system was created to streamline charger charging by controlling the temperature of the photovoltaic charger which often affects system performance and usability. The reason for choosing a PID control system is because of its fast and stable performance and ability to control temperature.

This research involves the use of solar cells to convert solar energy into electrical energy, a solar charger controller to regulate the current and voltage produced by the solar panels and then go to the battery to charge. From the battery it supplies the load and kr stepdown module to reduce the voltage and then it is used to supply the Arduino. Arduino will read voltage and current data from the ina219 sensor and temperature from the ds18b20 sensor. Then display it on LCD20x4. Arduino will also control the relay module that goes to the pump to turn on and turn off the pump using a PID algorithm based on the temperature input that is read to regulate the temperature of the solar cell using water as a medium through water spray from the pump.

The results of this research show that the system designed can increase charging efficiency and extend the service life of the photovoltaic charger and is able to maintain temperatures within the desired limits.

Key words : Photovoltaic charger, Proportional Integral Derivative controller, temperature control

PERNYATAAN KEASLIAN DAN HAK CIPTA

Saya yang bertanda tangan di bawah ini :

Nama : Elsa Marcela Candra Devi
NIT : 30121031
Program Studi : D3 Teknik Listrik Bandara
Judul Tugas Akhir : RANCANG BANGUN OPTIMALISASI
PHOTOVOLTAIC CHARGER
MENGGUNAKAN TEMPERATURE
CONTROL BERBASIS PROPORTIONAL
INTEGRAL DERIVATIVE CONTROLLER

Dengan ini menyatakan bahwa :

1. Tugas 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, mengubah instalasi, mengelola, merawat, dan mempublikasikan Tugas 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, 06 Agustus 2024
Yang membuat pernyataan



10000
REPUBLIK INDONESIA
TEL. 20
METERAL
TEMPEL
41FAAJX111496318

Elsa Marcela Candra Devi
NIT.30121031

KATA PENGANTAR

Segala puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas segala Rahmat dan Karunia-Nya yang telah memberikan kesehatan, pengetahuan, keterampilan, dan pengalaman dengan baik, sehingga Tugas Akhir yang berjudul “RANCANG BANGUN OPTIMALISASI *PHOTOVOLTAIC CHARGER* MENGGUNAKAN *TEMPERATURE CONTROL* BERBASIS *PROPORTIONAL INTEGRAL DERIVATIVE CONTROLLER*” ini dapat diselesaikan dengan baik.

Proses penyusunan Tugas Akhir ini dimaksudkan sebagai salah satu syarat menyelesaikan pendidikan di Politeknik Penerbangan Surabaya dan memperoleh gelar Ahli Madya (A.Md.).

Maka pada kesempatan ini, ucapan terima kasih tak henti-hentinya disampaikan kepada segenap pihak yang telah membantu selama proses penyusunan Tugas Akhir ini, terutama kepada:

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 Bandara di Politeknik Penerbangan Surabaya.
3. Bapak Dr. Slamet Hariyadi, S.T., M.M., selaku Dosen Pembimbing materi yang senantiasa membimbing dan membantu dalam penyusunan Tugas Akhir.
4. Bapak Dwiyanto, S.T., M.Pd., selaku Dosen Pembimbing penulisan yang senantiasa membimbing dan membantu dalam penyusunan Tugas Akhir.
5. Seluruh Dosen dan *civitas* akademika Program Studi Teknik Listrik Bandara Politeknik Penerbangan Surabaya.
6. Kepada orang tua, serta saudara yang telah memberikan doa, kasih sayang, dukungan moril dan materiil, serta bantuan dan dorongan sampai terselesaikannya penulisan Tugas Akhir ini.
7. Rekan – rekan Diploma III Teknik Listrik Bandara angkatan XVI yang selalu memberikan motivasi.

Tentunya karya tulis ini masih jauh dari sempurna. Semoga Tugas Akhir ini dapat memberikan manfaat khususnya bagi taruna Politeknik Penerbangan Surabaya. Semoga penulisan ini dapat dikembangkan dan dapat bermanfaat bagi semua pihak.

Surabaya, 03 Agustus 2024



Elsa Marcela Candra Devi
NIT.30121031

DAFTAR ISI

	Halaman
HALAMAN JUDUL.....	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN.....	iv
ABSTRAK	v
<i>ABSTRACT</i>	vi
PERNYATAAN KEASLIAN DAN HAK CIPTA	vii
KATA PENGANTAR	viii
DAFTAR ISI	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL.....	xiii
DAFTAR LAMPIRAN.....	xiv
 BAB I PENDAHULUAN	
1.1. Latar Belakang Masalah	1
1.2. Rumusan Masalah.....	2
1.3. Batasan Masalah	2
1.4. Tujuan Penelitian	3
1.4.1. Tujuan Umum	3
1.4.2. Tujuan Khusus	3
1.5. Manfaat Penelitian	3
1.6. Sistematika Penulisan	4
 BAB II LANDASAN TEORI	
2.1. Teori Penunjang	5
2.1.1. Energi Matahari	5
2.1.2. Energi Terbarukan	6
2.1.3. Pembangkit Listrik Tenaga Surya	7
2.1.4. Panel Surya.....	7
2.1.5. Baterai.....	10
2.1.6. PID	12
2.1.7. Sistem Pendingin Solar Cell	13
2.1.8. Pompa Air DC	13
2.1.9. Arduino UNO	14
2.1.10. INA 219	14
2.1.11. Modul Relay	15
2.1.12. Aki 12V	16
2.1.13. LM 2596	17
2.1.14. Lampu	18
2.1.15. <i>Solar Charger Controller</i>	18
2.1.16. Ds18b20.....	19

2.1.17. LCD i2c 20x4	20
2.1.18. Selang	20
2.1.19. Pipa	20
2.1.20. <i>Logic Local Converter</i>	21
2.1.21. ESP 32	21
2.2. Kajian Penelitian Terdahulu yang Relevan	22
BAB III METODE PENELITIAN	
3.1. Desain Penelitian	23
3.1.1. Alur Penelitian	24
3.1.2. Metode Penelitian	24
3.1.3. Studi Literatur	24
3.1.4. Konsultasi	25
3.2. Perencanaan dan Perancangan Alat	25
3.2.1. Desain Alat	25
3.2.2. Cara Kerja Alat	27
3.2.3. Komponen Alat	28
3.3. Pengujian Alat	30
3.4. Pengumpulan dan Analisis Data	30
3.5. Tempat dan Waktu Pelaksanaan	31
BAB IV HASIL DAN PEMBAHASAN	
4.1. Hasil Penelitian	32
4.1.1. Komponen Perangkat Keras	32
4.2. Pembuatan Perangkat Keras	33
4.2.1. Perakitan PCB	33
4.2.2. Komponen Perangkat Keras	33
4.2.3. Komponen Perangkat Keras	35
4.3. Pengujian Perangkat Lunak	35
4.3.1. Arduino IDE dan Aplikasi	35
4.3.2. Aplikasi Virtuino IoT	36
4.3.3. Sinkronisasi Perangkat Keras ke Aplikasi	37
4.4. Hasil Pengujian	38
4.4.1. Pengujian Panel Surya	38
4.4.2. Pengujian ESP 32	39
4.4.3. <i>Buck Converter</i> LM2596	40
4.4.4. Sensor INA 219	41
4.4.5. Relay	43
4.4.6. Pompa Motor DC	43
4.4.7. Arduino IDE	45
4.4.8. Pengujian Sensor Suhu	47
4.4.9. Pengujian LCD	49
4.4.10. Pengujian Baterai	50
4.4.11. Pengujian Tegangan <i>Output Solar Panel</i>	52
4.4.12. Pengukuran Debit Air pada Sistem Pendingin Solar Sel... ..	52
4.4.13. PID	53

BAB V PENUTUP	
5.1. Kesimpulan	57
5.2. Saran	57
DAFTAR PUSTAKA	58
LAMPIRAN.....	59



DAFTAR GAMBAR

Gambar 2. 1 Lapisan Panel Solar Cell Type n, Lapisan Pembatas Type p.....	7
Gambar 2. 2 Jenis Panel <i>Polycrystalline</i>	8
Gambar 2. 3 Jenis Panel <i>Monocrystalline</i>	9
Gambar 2. 4 Sel Surya <i>Amorphous Silicone</i>	9
Gambar 2. 5 Cara Kerja Panel Sel Surya	10
Gambar 2. 6 Konstruksi Baterai Sel Surya	11
Gambar 2. 7 Baterai SMT-POWER 12V	12
Gambar 2. 8 Pompa Air DC	14
Gambar 2. 9 Arduino Uno.....	14
Gambar 2. 10 INA 219.....	15
Gambar 2. 11 Modul Relay	16
Gambar 2. 12 Aki 12V	17
Gambar 2. 13 LM2596.....	17
Gambar 2. 14 Lampu	18
Gambar 2. 15 <i>Solar Charger Controller</i>	19
Gambar 2. 16 Ds18b20	19
Gambar 2. 17 LCD i2c 20x4.....	20
Gambar 2. 18 Pipa.....	21
Gambar 2. 19 Modul ESP 32	21
Gambar 3. 1 Diagram Alur Penelitian.....	24
Gambar 3. 2 Diagram alat	25
Gambar 3. 3 <i>Flowchart</i> Sistem Alat	27
Gambar 4. 1 PCB	33
Gambar 4. 2 Rangkaian Komponen Pada PCB	34
Gambar 4. 3 Pipa Penyemprot	35
Gambar 4. 4 Tampilan Awal Aplikasi Arduino IDE	36
Gambar 4. 5 Aplikasi <i>Virtuino</i>	36
Gambar 4. 6 Tampilan Aplikasi.....	37
Gambar 4. 7 Desain Alat.....	38
Gambar 4. 8 Panel Surya <i>Polycrystalline</i>	38
Gambar 4. 9 Hasil Pengukuran ESP 32	40
Gambar 4. 10 Pengukuran <i>Buck Converter</i> LM2596.....	40
Gambar 4. 11 Sensor INA219.....	42
Gambar 4. 12 Rangkaian Sensor INA219.....	42
Gambar 4. 13 Rangkaian Pompa Motor DC	44
Gambar 4. 14 Pengujian Tegangan Pompa DC	44
Gambar 4. 15 Pengujian Pompa Air DC.....	45
Gambar 4. 16 Kode program <i>Arduino</i>	46
Gambar 4. 17 Kode program <i>Arduino</i>	46
Gambar 4.18 <i>Done Compiling</i>	47
Gambar 4. 19 Gambar Rangkaian Alat	49
Gambar 4. 20 Tampilan LCD.....	50
Gambar 4.21 Hasil Pengukuran Baterai Menggunakan <i>Avometer</i>	51

Gambar 4.22 Hasil Pengukuran Baterai Menggunakan <i>Batterytester</i>	51
Gambar 4. 23 Program PID	54



DAFTAR TABEL

Tabel 2. 1 Tabel Kajian Penelitian Terdahulu	22
Tabel 3. 1 Jadwal Pembuatan Proyek Tugas Akhir	31
Tabel 4. 1 Hasil Pengujian Panel Surya	39
Tabel 4. 2 Hasil pengujian <i>Buck Converter</i> LM2596	41
Tabel 4. 3 Hasil Pengujian Sensor INA219	42
Tabel 4. 4 Hasil Pengukuran Suhu	48
Tabel 4. 5 Hasil tegangan output solar panel sel surya	52
Tabel 4. 6 Hasil Pengukuran Debit air	53
Tabel 4. 7 Hasil Pengujian PID	54



DAFTAR LAMPIRAN

Lampiran A. Standar Operasional Prosedur (SOP).....	A-1
Lampiran B. Coding Alat.....	B-1



DAFTAR PUSTAKA

- Ardian Nurahhman, Nazori A. Z. (2020). Sistem Kontrol Suhu Dengan PID dan Monitoring Daya *Output* Pada Panel *Photovoltaic Portable* Dengan *Virtual Private Server*. *Jurnal Maestro*, Vol. 3, No. 2, 403-412.
- Ahmad Nurul Huda, Ilmi R. I, Hilman S. I. (2020). Perancangan *Solar Charge Controler* Menggunakan *Control Proportional Integral Derivative*(PID) Pada *Prototype Traffic Light*. *Jurnal of Electrical Control and Automotive Engineering*. Vol. 5, No. 2, 9-15.
- Enggal Indra Gunawan, Machrus Ali, Hidayatul Nurohmah. (2022). Optimalisasi Thermal Oil Heater Menggunakan PSO Sebagai PID Controller. *Jurnal Sain dan Tekonologi*, 53-64.
- Habib Satria, Syafii. (2016). Sistem *Monitoring Online* dan Analisa Performansi PLTS *Rooftop* Terhubung ke Grid PLN. *Jurnal Rekayasa ElektriKa*, Vol. 14, No. 2, 136-144.
- Muhammad Ariq Kautsar, Ardianto Pranata, Masyuni Hutasuhut. (2023). Implementasi Metode *Fuzzy Logic* Pada Sistem Peringatan Kondisi Air Radiator Sepeda Motor Berbasis Mikrokontroler. *Jurnal Sistem Komputer TGD*, Vol. 2, No. 5, 236-245.
- Novie Ayub Windarko, Irianto, Agus Tami. (2021). Rancang Bangun *Solar Charger* dengan *Maximum Power Point Tracking* (MPPT) dan Kontrol *Proportional Integral Derivative* (PID) untuk Pengisian Baterai *Lithium-Ion*. *Jurnal Ecotipe*, Vol. 8, No. 2, 120-130.
- Riyana Prima Dewi, Saepul Rahmat, Hendi Purnata. (2023). Sistem Pendingin Panel Surya Otomatis Untuk Meningkatkan daya Keluaran Panel Surya. *Jurnal SIMETRIS*, Vol. 14, No. 1, 1-10.
- Cahyono, S. (2017). Perencanaan Sistem *Solar Charger* Menggunakan *Buck Converter* Dengan Metode PID. Skripsi. Jember: Fakultas Teknik Universitas Jember.
- Simamora, Kristianingsih. (2015). Desain Kendali PID Pada *Plant* Debit Air Dengan Metode *Ziegler-Nichols* dan *Cohen Coon* Menggunakan Matlab dan Arduino. Laporan. Bandung: Politeknik Negeri Bandung.
- Mahendra, A. R. (2016). Rancang Bangun *Solar Charger Portable* Berbasis Arduino UNO R3. Universitas Jember.
- Undang-Undang Republik Indonesia. (2007). Nomor 30 Tahun 2007 tentang Energi.

LAMPIRAN

Lampiran A. Standar Operasional Prosedur

Standar Operasional Prosedur (SOP) merupakan suatu panduan atau pedoman dalam melakukan suatu kegiatan atau pekerjaan. SOP ini berisi urutan dan tata cara atau instruksi yang dituangkan secara tertulis agar pekerjaan dapat dijalankan dengan baik, efisien, dan optimal. SOP harus dilaksanakan dengan baik untuk menjaga keselamatan pekerja, mencegah kecelakaan, dan menjaga peralatan agar selalu dalam kondisi yang prima.



Berikut adalah prosedur pengoperasian alat dalam penelitian Proyek Tugas Akhir yang berjudul “Rancang Bangun Optimalisasi *Photovoltaic Charger* Menggunakan *Temperature Control* Berbasis *Proportional Integral Derivative* (PID)”.

1. Siapkan peralatan, alat Proyek Tugas Akhir dan keperluan lain yang dibutuhkan dalam pengoperasian alat.
2. Letakkan alat, terutama komponen panel surya, di bawah sinar matahari langsung.
3. Hubungkan sistem pada baterai sebagai *power supply*.
4. Isi air ke dalam wadah penampungan air.
5. Amati data hasil pengukuran sensor suhu Ds18B20, sensor tegangan, dan arus INA219 serta pompa air DC pada tampilan LCD dan Aplikasi *Virtuino IoT*.
6. Kontrol dan monitoring suhu *setpoint* melalui aplikasi android pada *handphone*.

Lampiran B. Pemrograman (Coding) Alat

Coding Arduino Uno

```
#include <WiFi.h>
#include <ArduinoJson.h>
#include <PubSubClient.h>

const char* ssid    = "Punten";
const char* password = "wifinyabuatapa";
//const char* mqtt_server = "broker.hivemq.com";
const char* mqtt_server = "broker.emqx.io";

DynamicJsonDocument doc(JSON_OBJECT_SIZE(5));
char json[200];

uint8_t notif_status = 0;
uint8_t notif_state = 0;
bool wifi_connected;
bool mqtt_connected;

uint8_t pump_status;

String msg;
String message;
bool new_msg_state;

float voltage;
float current;
float power;
//float energy;
float temperature;

unsigned long cekMsg_millis;
//unsigned long notif_control_millis;
unsigned long push_millis;

WiFiClient espClient;
PubSubClient client(espClient);

void setup(){
  Serial.begin(115200);
  Serial2.begin(4800);
  Serial2.setTimeout(1000);

  client.setServer(mqtt_server, 1883);
  client.setCallback(callback);

  init_task();
}

void loop(){

  if(millis()-cekMsg_millis>500){
    cek_msg();
```



```

        cekMsg_millis=millis();
    }
    if(millis()-push_millis>1000){
        push_data();
        push_millis = millis();
    }
    /*
    if(millis()-notif_control_millis>500){
        notif_control();
        notif_control_millis = millis();
    }
    */
    client.loop();

}

void push_data(){
    client.publish("tx_topic_voltage_28052024", String(voltage,2).c_str());
    client.publish("tx_topic_current_28052024", String(current,3).c_str());
    client.publish("tx_topic_power_28052024", String(power,3).c_str());
    client.publish("tx_topic_pump_28052024", String(pump_status).c_str());
    client.publish("tx_topic_temperature_28052024", String(temperature,2).c_str());
}

/*
void notif_control(){
    if(pump_status==1 && notif_state==0 && notif_status==1){
        String txt = "suhu melebihi batas\n pompa dinyalakan";
        bot.sendMessage(chat_id,txt, "");
        Serial.println(txt);
        notif_state=1;
    }
    if(pump_status==0 && notif_state==1 && notif_status==1){
        String txt = "suhu normal\n pompa dimatikan";
        bot.sendMessage(chat_id,txt, "");
        Serial.println(txt);
        notif_state=0;
    }
}
*/

void cek_msg(){
    message = "";

    if(message.startsWith("cek sensor")){
        String txt = "Voltage = " + String(voltage,2) + "V\n";
        txt += "Current = " + String(current,3) + "A\n";
        txt += "Power = " + String(power,3) + "W\n";
        txt += "Solar Temp = " + String(temperature,2) + "°C";
        //bot.sendMessage(chat_id,txt, "");
    }
    if(message.startsWith("notif on")){
        notif_state=pump_status;
        notif_status=1;
    }
}

```

```

// bot.sendMessage(chat_id,"notifikasi aktif", "");
}
if(message.startsWith("notif off")){
    notif_state=pump_status;
    notif_status=0;
    //bot.sendMessage(chat_id,"notifikasi mati", "");
}
Serial.println(message);

}

void Task0(void *pvParameters){
    while(true){
        while(wifi_connected==false){
            vTaskDelay(500);
        }
        if(Serial2.available(>0){

            msg = Serial2.readStringUntil('/r/n');
            doc.clear();
            msg.toCharArray(json, 200);
            //deserializeJson(doc, json);
            DeserializationError error = deserializeJson(doc, json);
            if (error) {
                //Serial.print(F("deserializeJson() failed: "));
                //Serial.println(error.c_str());
                vTaskDelay(10);
                continue;
            }
            temperature = doc["t"].as<String>().toFloat();
            voltage = doc["v"].as<String>().toFloat();
            current = doc["c"].as<String>().toFloat();
            power = doc["p"].as<String>().toFloat();
            pump_status = doc["pump"].as<String>().toInt();
            //energy = doc["e"].as<String>().toFloat();
            Serial.println("voltage = " + String(voltage,2));
            Serial.println("current = " + String(current,3));
            Serial.println("power = " + String(power,3));
            //Serial.println("energy = " + String(energy,3));
            Serial.println("temperature = " + String(temperature,2));
            Serial.println("pump status = " + String(pump_status));
        }
        vTaskDelay(10);
    }
}

void Task1(void *pvParameters){
    while(true){
        cek_wifi();
        if(wifi_connected==true){
            mqtt_connect();
        }
        if(wifi_connected==false){
            vTaskDelay(500);
        }else{

```



```

        vTaskDelay(3000);
    }
}

void init_task(){
    TaskHandle_t task0;
    TaskHandle_t task1;
    xTaskCreatePinnedToCore(Task0, "task0", 2056, NULL, 3, &task0, 0);
    xTaskCreatePinnedToCore(Task1, "task1", 2056, NULL, 3, &task1, 0);
}

void cek_wifi(){
    static int count = 0;
    static bool connect_state;
    if(connect_state == false && WiFi.status() != WL_CONNECTED){
        connect_state = true;
        wifi_connected = false;
        mqtt_connected = false;
        Serial.println();
        Serial.print("Connecting to ");
        Serial.println(ssid);
        WiFi.mode(WIFI_STA);
        WiFi.begin(ssid, password);
    }
    if(connect_state == true){
        if(WiFi.status() != WL_CONNECTED){
            Serial.print(".");
            count++;
            if(count >= 60){
                ESP.restart();
            }
        }else{
            Serial.print("Connected to ");
            Serial.println(ssid);
            connect_state = false;
            wifi_connected = true;
            count = 0;
        }
    }
}

void callback(char* topic, byte* payload, unsigned int length) {
    String mqtt_msg="";
    char buffer[50];
    auto Topic = [&buffer](const char *Topic){
        memset(buffer, 0, sizeof(buffer)-1);//clear buffer
        strcpy(buffer, Topic);
    };
    Topic("rx_topic_28052024");
    if(strcmp(topic,buffer)==0){
        //whatever you want for this topic
        for (int i = 0; i < length; i++) {
            Serial.print((char)payload[i]);
            mqtt_msg += (char)payload[i];
        }
    }
}

```

```

    }
    new_msg_state=true;
    message = mqtt_msg;
  }
}

void mqtt_connect() {
  if(!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    // Create a random client ID
    String clientId = "ELSAClient-";
    clientId += String(random(0xffff), HEX);
    // Attempt to connect
    if(client.connect(clientId.c_str())) {
      Serial.println("connected");
      client.subscribe("rx_topic_28052024");
      mqtt_connected = true;
    }else{
      mqtt_connected = false;
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
    }
  }
}

void publish(const char *Topic, const char *Msg){
  char buffer[500];
  memset(buffer, 0, sizeof(buffer)); //clear buffer
  strcpy(buffer, Topic);
  client.beginPublish(buffer, strlen(String(Msg).c_str()), false);
  client.print(Msg);
  client.endPublish();
}

void subscribe(const char *Topic){
  char buffer[50];
  memset(buffer, 0, sizeof(buffer)); //clear buffer
  strcpy(buffer, Topic);
  client.subscribe(buffer)
#include <SoftwareSerial.h>
#include <LiquidCrystal_I2C.h>
#include <ArduinoJson.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include "INA219.h"
#include "ArduPID.h"

#define pump_pin 3
#define pump_on digitalWrite(pump_pin,0)
#define pump_off digitalWrite(pump_pin,1)

ArduPID myController;

double input;

```

```

double output;

// Arbitrary setpoint and gains - adjust these as fit for your project:
double p = 5;
double i = 5;
double d = 5;

double temperature_setpoint = 32.0;

DynamicJsonDocument doc(JSON_OBJECT_SIZE(8));
char json[200];

SoftwareSerial swSerial(8, 9); // RX, TX
LiquidCrystal_I2C lcd(0x27, 20, 4);
INA219 INA(0x40);

#define ONE_WIRE_BUS 7
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature temperature_sensor(&oneWire);

float voltage;
float current;
float power;
//float energy;
float temperature;

uint8_t pump_status = 0;

unsigned long lcd_millis;
unsigned long sensor_millis;
unsigned long sendData_millis;
unsigned long control_millis;

int control_delay = 3000;

void setup() {
  pinMode(pump_pin, OUTPUT);
  Serial.begin(115200); // Initialize serial communication
  swSerial.begin(4800);
  lcd.init();
  if(!INA.begin()){
    Serial.println("Could not connect. Fix and Reboot");
  }
  INA.setMaxCurrentShunt(3.4, 0.1);
  temperature_sensor.begin();
  lcd.clear();
  lcd.backlight();

  myController.begin(&input, &output, &temperature_setpoint, p, i, d);
  myController.reverse();
  // myController.setSampleTime(10);    // OPTIONAL - will ensure at least 10ms have past
  // between successful compute() calls
  myController.setOutputLimits(0, 30);
  myController.setBias(30.0 / 3.0);

```

```

    myController.setWindUpLimits(-1, 1); // Groth bounds for the integral term to prevent integral
    wind-up
    myController.start();
}
void loop() {
    if(millis() - sensor_millis>20){
        read_sensor();
        sensor_millis=millis();

        if(millis() - lcd_millis>500){
            lcd_update();
            lcd_millis=millis();
        }
        if(millis() - sendData_millis>1000){
            send_data();
            sendData_millis = millis();
        }
        if(millis() - control_millis>control_delay){
            input = temperature;
            myController.compute();
            control_delay = output*1000;
            if(pump_status==1){
                pump_off;
                pump_status = 0;
                control_delay=3000;
                //Serial.println("pompa menyala");
            }
            else if(pump_status==0 && control_delay>1000){
                pump_on;
                pump_status = 1;
                //Serial.println("pompa mati");
            }
            control_millis = millis();
        }
    }
}
void send_data(){
    doc.clear();
    doc["v"] = String(voltage,2);
    doc["c"] = String(current,3);
    doc["p"] = String(power,3);
    //doc["e"] = String(energy,3);
    doc["t"] = String(temperature,2);
    doc["pump"] = String(pump_status);
    serializeJson(doc, json);
    swSerial.println(json);
    Serial.println(json);
}
void lcd_update(){
    String text = "Voltage = " + String(voltage,2) + "V";
    while(text.length()<20){text+=" ";}
    lcd.print(text);
    text = "Current = " + String(current,3) + "A";
    while(text.length()<20){text+=" ";}
    lcd.setCursor(0, 1);
    lcd.print(text);
}

```

```

text = "Power = " + String(power,3) + "W";
  while(text.length()<20){text+=" ";}
lcd.setCursor(0, 2);
lcd.print(text);
//text = "Energy = " + String(energy,3) + "Wh";
text = "Solar Temp = " + String(temperature,2) + "C";
  while(text.length()<20){text+=" ";}
lcd.setCursor(0, 3);
lcd.print(text);
}
void read_sensor(){
  static int num = 0;
  const int rate_val = 10;
  static float voltage_rate[rate_val];
  static float current_rate[rate_val];
  static float temperature_rate[rate_val];
  voltage_rate[num] = INA.getBusVoltage();
  for(int i=0; i<rate_val; i++){
    if(voltage_rate[i]<=0){
      voltage += voltage_rate[num];
    }else{
      voltage += voltage_rate[i];
    }
  }
  voltage/=rate_val;
  if(voltage<0){voltage=0;}
  current_rate[num] = INA.getCurrent();
  for(int i=0; i<rate_val; i++){
    if(current_rate[i]<=0){
      current += current_rate[num];
    }else{
      current += current_rate[i];
    }
  }
  current/=rate_val;
  if(current<0){current=0;}
  power = voltage*current;
  //double tmp = (voltage_rate[num]*current_rate[num])/3600000.0;
  //tmp*=20.0;
  //energy += tmp;
  // if(energy<0){energy=0;}

  temperature_sensor.requestTemperatures();
  temperature = temperature_sensor.getTempCByIndex(0);
  if(temperature<0){temperature=0;}

  num++;
  if(num>rate_val-1){num=0;}
}

```

DAFTAR RIWAYAT HIDUP



ELSA MARCELA CANDRA DEVI, lahir di Malang pada tanggal 11 Februari 2001, anak kedua dari dua bersaudara, dilahirkan dari pasangan Yuswan Tri Cahyono dan Siti Khofikyah. Bertempat tinggal di Jalan Dorowati Utara RT 03 RW 13, Desa Mulyoarjo, Kecamatan Lawang, Kabupaten Malang. Pada tahun 2007 Memulai pendidikan sekolah dasar di SD Negeri Mulyoarjo 3 Lawang sampai lulus pada tahun 2013.

Melanjutkan pendidikan di SMP Negeri 1 Singosari pada tahun 2013 dan lulus pada tahun 2016. Setelah itu masuk ke SMA Negeri 1 Singosari pada tahun 2016 dan lulus pada tahun 2019.

Sempat *Gap Year* selama dua tahun, kemudian pada tahun 2021 diterima sebagai taruni di Politeknik Penerbangan Surabaya pada Program Studi Diploma III Teknik Listrik Bandara Angkatan XVI Bravo sampai dengan saat ini. Selama mengikuti pendidikan di Politeknik Penerbangan Surabaya, telah mengikuti *On the Job Training* (OJT) di UPBU H. Asan di Sampit pada bulan Mei sampai dengan September Tahun 2024 dan di Bandar Udara Internasional Lombok di Lombok Praya pada bulan Oktober sampai dengan Februari Tahun 2024.

Harapan saya setelah menyelesaikan pendidikan di Politeknik Penerbangan Surabaya, bisa menjadi seorang insan perhubungan yang bertanggung jawab, disiplin dan bisa berguna bagi Bangsa dan Negara. Rasa syukur kepada Tuhan Yang Maha Esa yang selalu memberikan Kasih dan Anugerah-Nya serta tidak lupa kepada orang tua yang selalu mendukung di setiap kegiatan. Allah tidak semata-mata mendatangkan cobaan dan musibah tanpa ada suatu pembelajaran dan sesuai dengan kemampuan hamba-Nya. Berkaitan dengan segala keinginan, harapan, ketakutan atau apapun kepada Allah pemilik segala-galanya, penentu segalanya, niscaya kita akan tenang dan akan ditolong.