

**RANCANG BANGUN *TOOLBOX* OTOMATIS DENGAN
METODE *SMART VENDING MACHINE* BERBASIS
*INTERNET OF THINGS***

PROYEK AKHIR



Oleh:

RIZQI MAULA HAMDANI
NIT. 30121022

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

**RANCANG BANGUN *TOOLBOX* OTOMATIS DENGAN
METODE *SMART VENDING MACHINE* BERBASIS
*INTERNET OF THINGS***

PROYEK AKHIR

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



Oleh:

RIZQI MAULA HAMDANI
NIT. 30121022

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

LEMBAR PERSETUJUAN

RANCANG BANGUN *TOOLBOX* OTOMATIS DENGAN METODE *SMART VENDING MACHINE* BERBASIS *INTERNET OF THINGS*

Oleh :

Rizqi Maula Hamdani
NIT. 30121022

Disetujui untuk diujikan pada :

Surabaya, 09 Agustus 2024

Pembimbing I : YUDHIS THIRO KABUL YUNIOR, S.T., M.Kom.
NIP. 19870224 202203 1 003

Pembimbing II : Ir. BAMBANG JUNIPITOYO, S.T., M.T.
NIP. 19780626 200912 1 001



LEMBAR PENGESAHAN

RANCANG BANGUN *TOOLBOX* OTOMATIS DENGAN METODE *SMART VENDING MACHINE* BERBASIS *INTERNET OF THINGS*

Oleh :

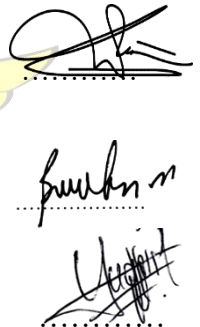
Rizqi Maula Hamdani

NIT. 30121022

Telah dipertahankan dan dinyatakan lulus pada Ujian Proyek Akhir
Program Pendidikan Diploma 3 Teknik Listrik Bandara
Politeknik Penerbangan Surabaya
Pada tanggal : 09 Agustus 2024

Panitia Penguji :

1. Ketua : Drs. HARTONO. S.T, M.Pd., M.M.
NIP. 19630408 198902 1 001
2. Sekertaris : Ir. BAMBANG JUNIPITOYO, S.T., M.T.
NIP. 19780626 200912 1 001
3. Anggota : YUDHIS THIRO KABUL YUNIOR, S.T., M.Kom.
NIP. 19870224 202203 1 003



Mengetahui,
Ketua Program Studi
D3 Teknik Listrik Bandara



Dr. GUNAWAN SAKTI, S.T., M.T.

NIP. 19881001 200912 1 003

HALAMAN PERSEMBAHAN

MOTTO :

“Untuk mendapatkan sesuatu yang kau inginkan kau harus bersabar dengan sesuatu yang kau benci” – Imam Ghazali

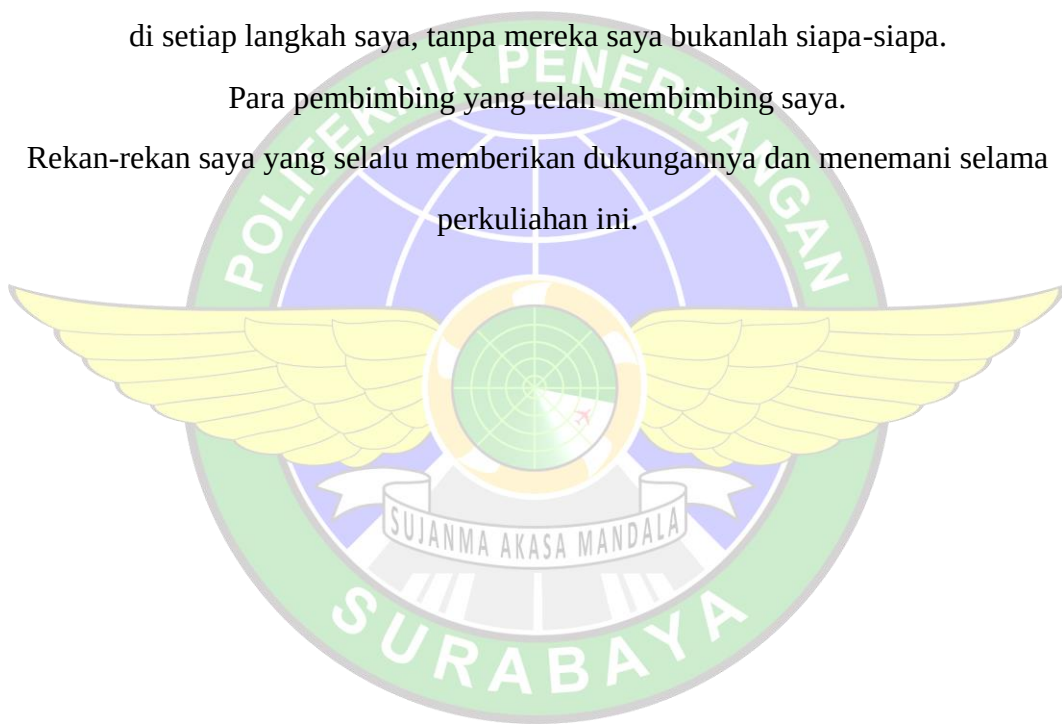
PERSEMBAHAN :

Allah SWT yang telah melimpahkan rahmat dan hidayahNya.

Orangtua yang tercinta, untuk setiap do’a yang selalu dipanjatkan,serta dukungan di setiap langkah saya, tanpa mereka saya bukanlah siapa-siapa.

Para pembimbing yang telah membimbing saya.

Rekan-rekan saya yang selalu memberikan dukungannya dan menemani selama perkuliahan ini.



PERNYATAAN KEASLIAN DAN HAK CIPTA

Saya yang bertanda tangan di bawah ini :

Nama : Rizqi Maula Hamdani
NIT : 30121022
Program Studi : D3 Teknik Listrik Bandara
Judul Proyek akhir : Rancang Bangun *Toolbox* Otomatis Menggunakan Metode *Smart Vending Machine* Berbasis *Internet of Things*.

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, mengalih media/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 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, 16 Juli 2024

Yang membuat pernyataan



Rizqi Maula Hamdani
NIT.30121022

ABSTRAK

RANCANG BANGUN *TOOLBOX* OTOMATIS DENGAN METODE *SMART VENDING MACHINE* BERBASIS *INTERNET OF THINGS*

Oleh :

Rizqi Maula Hamdani
NIT. 30121022

Toolbox merupakan penyimpanan dari beberapa peralatan yang berguna untuk memudahkan teknisi dalam mengambil suatu peralatan. Sistem ini masih sering ditemukan menggunakan sistem manual, yang berdampak jika barang hilang maka tidak ada identifikasi yang akurat dengan terakhir pemakaian. Pada bidang industri maupun bandara, perkembangan teknologi sudah mencapai revolusi industri 4.0, teknologi dengan gabungan *Internet of Things* merupakan suatu inovasi yang bisa memudahkan pekerjaan manusia.

Penelitian ini dengan merancang serta membuat *toolbox* otomatis menggunakan metode *smart vending machine* dilengkapi dengan komponen *Internet of Things* yang dapat mendeteksi wajah jika ingin menggunakan *toolbox*. *Webcam* berfungsi sebagai mengidentifikasi *user* kemudian mengirimkan respon ke *orange pi* dan *ESP32* sebagai mikrokontroller serta diteruskan di *web server*. Kemudian motor stepper berjalan atas perintah *Arduino Nano* dan *driver motor* yang akan bergerak maju dan mundur sesuai perintah *user*.

Rancang bangun ini berjalan dengan optimal sebagai sarana pengambilan *toolbox* otomatis yang dilengkapi pengaman yaitu identifikasi wajah dan login *web server* dengan keakuratan yang sesuai dengan gambar. Setelah dilakukan pengujian keseluruhan bisa disimpulkan bahwa untuk mendeteksi wajah *user* dibutuhkan waktu 5 sampai 10 detik setelah itu akan masuk ke *web server* untuk melakukan pengambilan atau pengembalian alat.

Kata kunci: *Toolbox, Smart Vending Machine, Arduino Nano, User, ESP32*

ABSTRACT

DESIGN OF AUTOMATIC TOOLBOX WITH SMART VENDING MACHINE METHOD BASED ON INTERNET OF THINGS

By:

Rizqi Maula Hamdani
NIT. 30121022

Toolbox is the storage of several tools that are useful to facilitate technicians in taking a tool. This system is still often found using a manual system, which has an impact if the item is lost, there is no accurate identification with the last use. In the industrial and airport fields, technological developments have reached the industrial revolution 4.0, technology with the combination of the Internet of Things is an innovation that can facilitate human work.

In this study by designing and making an automatic toolbox using the smart vending machine method equipped with Internet of Things components that can detect faces if you want to use the toolbox. The webcam functions as identifying the user then sending a response to the orange pi and ESP32 as a microcontroller and forwarded on the web server. Then the stepper motor runs on the arduino nano command and the motor driver will move forward and backward according to the user's request.

This design runs optimally as a means of automatic toolbox retrieval equipped with security, namely face identification and web server login with accuracy in accordance with the image. After overall testing it can be concluded that to detect the user's face it takes 5 to 10 seconds after which it will enter the web server to retrieve or return the tool.

Keywords: Toolbox, Smart Vending Machine, Arduino Nano, User, ESP32

KATA PENGANTAR

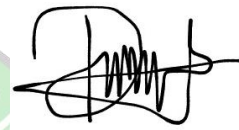
Puji syukur kehadiran Allah SWT, yang telah melimpahkan nikmat dan karunia-Nya kepada penulis sehingga dapat menyelesaikan penulisan Proyek Akhir ini dengan baik. Proyek akhir yang berjudul “RANCANG BANGUN *TOOLBOX* OTOMATIS DENGAN METODE *SMART VENDING MACHINE* BERBASIS *INTERNET OF THINGS*” diajukan untuk memenuhi salah satu persyaratan untuk mendapatkan gelar Ahli Madya (A.Md.) Program studi Diploma 3 Teknik Listrik Bandara pada Politeknik Penerbangan Surabaya.

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 Bandara Politeknik Penerbangan Surabaya.
3. Bapak Yudhis Thiro Kabul Yuniar, S.T., M.Kom., selaku dosen pembimbing I yang senantiasa membimbing dalam penyusunan Proyek Akhir.
4. Bapak Ir. Bambang Junipitoyo, S.T., M.T. selaku dosen pembimbing II yang senantiasa membimbing dalam penyusunan proyek akhir.
5. Para Dosen dan Instruktur Politeknik Penerbangan Surabaya yang telah memberikan ilmu.
6. Keluarga yang memberikan kasih sayang, dukungan, dan doa kepada saya kapanpun dan dimanapun berada.
7. Teman-teman TLB XVI yang telah menyumbangkan pikiran, saran, dan motivasi.
8. Sahabat-sahabat yang membantu dalam memberikan dukungan dan doa.
9. 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, 29 Juli 2024



Rizqi Maula Hamdani



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 ISI	x
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiii
DAFTAR LAMPIRAN	xiv
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
1.6 Sistematika Penulisan.....	3
BAB 2 TINJAUAN TEORI.....	5
2.1 Teori Penunjang.....	5
2.1.1 Metode Rancang Bangun.....	5
2.1.2 Vending Machine.....	5
2.1.3 Smart Vending Machine	6
2.1.4 ESP32.....	6
2.1.5 Arduino IDE	8
2.1.6 Arduino Nano	10
2.1.7 Internet Of Things (IoT)	11
2.1.8 Motor Stepper	11
2.1.9 Modul Driver Motor TB6600	12
2.1.10 Modul StepDown LM2596.....	13
2.1.11 <i>Liquid Crystal Dimension</i> (LCD 16 x 2)	14
2.1.12 Orange Pi	15
2.1.13 Kamera Webcam.....	16

2.1.14 <i>Web Server</i>	17
2.1.15 XAMPP.....	18
2.1.16 Adaptor	19
2.2 Kajian Penelitian Terdahulu yang Relavan	19
BAB 3 METODE PENELITIAN.....	23
3.1 Desain Penelitian.....	23
3.2 Perancangan Alat.....	25
3.2.1 Desain Alat	25
3.2.2 Cara Kerja Alat	27
3.2.3 Komponen Alat.....	28
3.3 Teknik Pengujian.....	31
3.4 Teknik Analisis Data	31
3.5 Tempat dan Waktu Pelaksanaan.....	32
BAB 4 HASIL DAN PEMBAHASAN.....	33
4.1 Uraian Rencana Penelitian	33
4.2 Pengujian Motor stepper.....	33
4.2.1 Rangkaian Motor Stepper	34
4.2.2 Hasil Pengujian Motor Stepper.....	35
4.3 Pengujian <i>Face Recognition</i>	37
4.3.1 Rangkaian <i>Face Recognition</i>	37
4.3.2 Program <i>Face Recognition</i>	39
4.3.3 Hasil Pengujian <i>Face Recognition</i>	39
4.4 Rangkaian Keseluruhan Sistem.....	42
BAB 5 PENUTUP.....	43
5.1 Kesimpulan.....	43
5.2 Saran	43
DAFTAR PUSTAKA	45
LAMPIRAN	

DAFTAR GAMBAR

Gambar 2. 1 ESP32	6
Gambar 2. 2 Tampilan Arduino IDE.....	9
Gambar 2. 3 Arduino Nano.....	11
Gambar 2. 4 Motor Stepper.....	12
Gambar 2. 5 Driver Motor TB6600	13
Gambar 2. 6 Modul Stepdown LM2596	14
Gambar 2. 7 <i>Liquid Crystal Dimension 16 x 2</i>	14
Gambar 2. 8 Orange Pi.....	16
Gambar 2. 9 Webcam	16
Gambar 2. 10 Web server	17
Gambar 2. 11 Tampilan software XAMPP	18
Gambar 2. 12 Adaptor 12V	19
Gambar 3. 1 Desain Penelitian Alat	23
Gambar 3. 2 Desain Blok Diagram Perancangan Alat.....	25
Gambar 3. 3 Desain Alat <i>Toolbox</i> Otomatis	25
Gambar 3. 4 Cara Kerja Alat.....	27
Gambar 4. 1 Pengujian Motor Stepper.....	33
Gambar 4. 2 Rangkaian Motor Stepper.....	34
Gambar 4. 3 Alur Motor Stepper	34
Gambar 4. 4 Motor Stepper Menentukan Posisi Pergerakan	35
Gambar 4. 5 Motor Stepper Berjalan	36
Gambar 4. 6 Motor Stepper Berjalan Mundur	36
Gambar 4. 7 Rangkaian <i>Face Recognize</i>	37
Gambar 4. 8 Alur Pengujian <i>Face Recognize</i>	38
Gambar 4. 9 Program Hasil Pengujian.....	39
Gambar 4. 10 Rangkaian Keseluruhan Sistem.....	42

DAFTAR TABEL

Tabel 2. 1 Spesifikasi ESP32	8
Tabel 2. 2 Spesifikasi Arduino Nano	10
Tabel 2. 3 Spesifikasi Orange Pi Zero	15
Tabel 2. 4 Kajian Penelitian Terdahulu yang Relevan.....	19
Tabel 3. 1 Jadwal Pembuatan Proyek Akhir	32
Tabel 4. 1 Tabel Pengujian Motor Stepper	36
Tabel 4. 3 Hasil Pengujian <i>Face Recognize</i>	40



DAFTAR LAMPIRAN

Lampiran A. Standard Operasional Prosedure (SOP).....	A-1
Lampiran B. Coding Alat.....	B-1
Lampiran C. Daftar Riwayat Hidup	C-1



DAFTAR PUSTAKA

- Arfian Habib Patonra, S. M. (2020). Rancang Bangun Media Pembelajaran Praktik Motor Stepper . *Mechatronics Journal in Professional and Entrepreneur*, 5.
- Arief Selay, G. D. (2022). INTERNET OF THINGS. *Karimah Tauhid, Volume 1 Nomor 6* (2022), e-ISSN 2963-590X| Selay et al., 3-4.
- Efendi, Y. (2018). INTERNET OF THINGS (IOT) SISTEM PENGENDALIAN LAMPU . *Jurnal Ilmiah Ilmu Komputer, Vol. 4, No. 1, 2*.
- Frindi Mangimbulude, P. D. (2023). Implementasi Metode Local Binary Pattern Histogram Dan Facial Landmark Pada Keamanan Akses Login. *Cogito Smart*, 6.
- Hafizhuddin, S. (2019). DESIGN AND IMPLEMENTATION OF AUTOMATIC BREAD VENDING MACHINE BASED INTERNET OF THINGS. *FINAL PROJECT – EE 184801*, 29.
- Heryanda, R. (2017). PROTOTIPE VENDING MACHINE MINUMAN KALENG. *TEKNIK ELEKTRO FAKULTAS TEKNIK UNIVERSITAS NEGERI JAKARTA*, 15.
- Iksal, S. S. (2018). Perancangan Sistem Kendali Otomatisasi On-Off Lampu. *Seminar Nasional Rekayasa Teknologi Informasi*, 3.
- Jati Widyo Leksono, H. K. (2019). Modul Belajar Arduino Uno. In H. K. Jati Widyo Leksono, *Modul Belajar Arduino Uno* (p. 6). Jombang: Humaidillah.
- Lulu Fikriyah, A. R. (2018). SISTEM KONTROL PENDINGIN RUANGAN MENGGUNAKAN ARDUINO WEB SERVER DAN EMBEDDED . *SIMANTIKA Vol. 3 No. 1, 2*.
- M. Bagus Arifudin, Y. M. (2023). SIMULASI SMART VENDING MACHINE MINUMAN KOPI. *JITET (Jurnal Informatika dan Teknik Elektro Terapan)*, 3.
- Maier, A. S. (2017). Comparative analysis and practical implementation of the. *Glyndŵr University Research Online*, 3.
- Maulana, A. (2019). Sistem Simulasi Kendali Traffic Light dengan Metode Pendeteksian. *Institut Teknologi Indonesia*, 10-11.
- Muliadi, A. I. (2020). PENGEMBANGAN TEMPAT SAMPAH PINTAR MENGGUNAKAN ESP32. *Jurnal MEDIA ELEKTRIK, Vol. 17, No. 2,* 2.
- Pressman, R. S. (2010). *Rekayasa perangkat lunak*. yogyakarta: Andi Offset.

Salahuddin, I. A. (2020). Rancang Bangun Vending Machine Jajanan Tradisional. *Universitas Dinamika*, 31.

Wicaksono, S. R. (Jakarta). *Konsep Dasar Rekayasa Perangkat Lunak*. 2011: Prestasi Pustaka.

Yoga Satriawan, Y. S. (2022). Sistem Kontrol Penggerak Motor Stepper Molding Microplastic Metode Fuzzy Logic Berbasis Myrio. *Jurnal Elkolind Volume 9, Nomor 1, 2*.



LAMPIRAN

Lampiran A. Standard Operasional Prosedure (SOP)

RANCANG BANGUN *TOOLBOX* OTOMATIS DENGAN METODE *SMART VENDING MACHINE* BERBASIS *INTERNET OF THINGS*

Oleh :

Rizqi Maula Hamdani

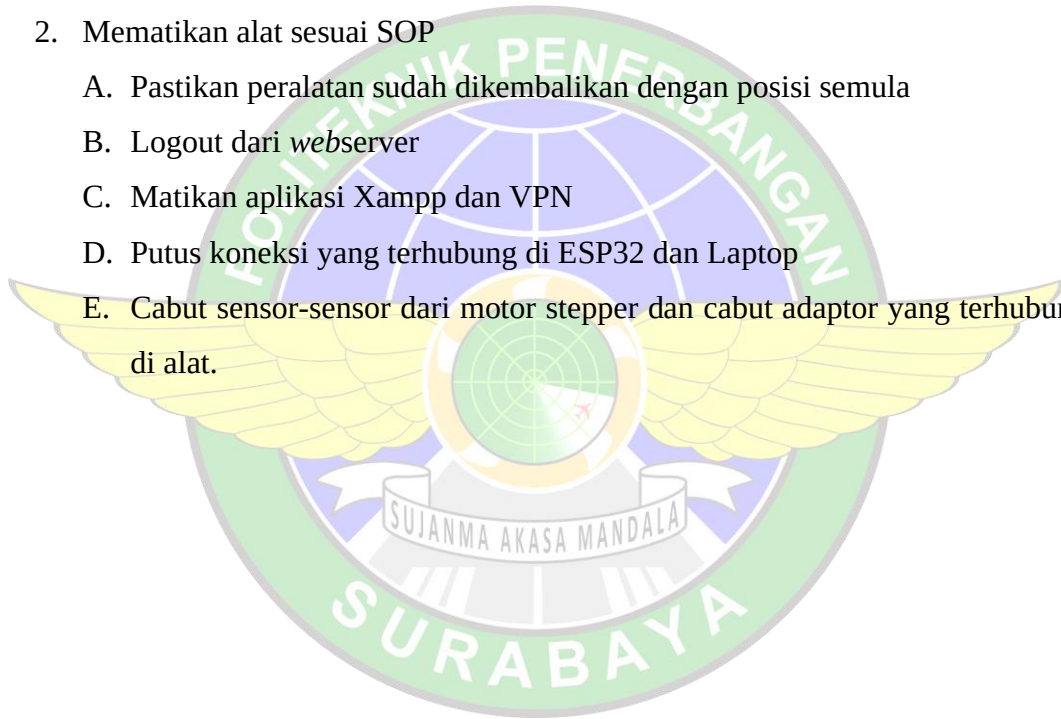
NIT. 30121022

Standard Operational Procedure (SOP) dalam pengoperasian alat adalah panduan yang merinci 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 *TOOLBOX* OTOMATIS DENGAN METODE *SMART VENDING MACHINE* BERBASIS *INTERNET OF THINGS*” sebagai berikut :

1. Mengoperasikan alat sesuai SOP
 - A. Pasang dan sambungkan semua sensor ke motor stepper.
 - B. Pasang USB dari ESP32 dan sambungkan ke laptop.
 - C. Pastikan modul ESP32 terhubung ke jaringan wifi yang memiliki akses internet.

- D. Pastikan alat sudah terkoneksi pada jaringan wifi yang sudah terdaftar di Arduino IDE
 - E. Aktifkan aplikasi Xampp dan Hubungkan VPN melalui Ip yang sudah di program dari Arduino IDE
 - F. Masukkan IP di *webserver* untuk melakukan login dan pastikan username dan password sudah benar.
 - G. Pantau pergerakan motor stepper disaat meminjam dan mengembalikan alat.
2. Mematikan alat sesuai SOP
- A. Pastikan peralatan sudah dikembalikan dengan posisi semula
 - B. Logout dari *webserver*
 - C. Matikan aplikasi Xampp dan VPN
 - D. Putus koneksi yang terhubung di ESP32 dan Laptop
 - E. Cabut sensor-sensor dari motor stepper dan cabut adaptor yang terhubung di alat.



Lampiran B. Coding Alat

```
#include <WiFi.h>
#include <HTTPClient.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
const char* ssid = "Rizqi Maula";
const char* password = "poltekngabb";
//const char* ssid = "jisa technology";
//const char* password = "3990000000";
String server = "http://202.158.40.156:9000/update.php";
String server_command;
String face_name;
String face_name_compare;
bool wifi_connected;
unsigned long lcd_update_timing;
bool recognize_status;
bool recognized_status;
int task_running = -1;
__NOINIT_ATTR int32_t box_1_status;
__NOINIT_ATTR int32_t box_2_status;
__NOINIT_ATTR int32_t box_3_status;
__NOINIT_ATTR int32_t first_start;
void process_command();
void init_task();
void setup(){
    delay(500);
    Serial.begin(115200);
    Serial1.begin(4800, SERIAL_8N1, 4, 15); //rx=4, tx=15
    Serial1.setTimeout(1000);
    Serial2.begin(115200, SERIAL_8N1, 16, 17);
    Serial2.setTimeout(1000);
    lcd.begin();
    lcd.clear();
    lcd.backlight();
    if(first_start!=11111){
        first_start = 11111;
        box_1_status=1;
        box_2_status=1;
        box_3_status=1;
    }
    init_task();
```



```

}
void loop(){
    process_command();
    delay(1000);
}
void process_command(){
    if(millis()-lcd_update_timing>3000 || recognized_status==true){
        update_lcd();
    }
    if(server_command.indexOf("recognize")!= -1){
        face_name_compare = server_command.substring(12,server_command.length());
        TaskHandle_t task_recognize_face;
        xTaskCreatePinnedToCore(Task_recognize_face, "task_recognize_face", 2096,
        NULL, 3, &task_recognize_face, 1);
        TaskHandle_t task_server_hw_update;
        xTaskCreatePinnedToCore(Task_server_hw_update, "task_server_hw_update",
        4096, NULL, 3, &task_server_hw_update, 1);
        server_command = "";
    }
    if(server_command.indexOf("button1 pinjam")!= -1 && box_1_status==1){
        box_1_status=0;
        take(1);
        server_command = "";
    }
    if(server_command.indexOf("button2 pinjam")!= -1 && box_2_status==1){
        box_2_status=0;
        take(2);
        server_command = "";
    }
    if(server_command.indexOf("button3 pinjam")!= -1 && box_3_status==1){
        box_3_status=0;
        take(3);
        server_command = "";
    }
    if(server_command.indexOf("button1 kembalikan")!= -1 && box_1_status==0){
        box_1_status=1;
        Return(1);
        server_command = "";
    }
    if(server_command.indexOf("button2 kembalikan")!= -1 && box_2_status==0){
        box_2_status=1;
        Return(2);
    }
}

```

```

server_command = "";
}
if(server_command.indexOf("button3 kembalikan")!= -1 && box_3_status==0){
    box_3_status=1;
    Return(3);
    server_command = "";
}
}
void Task_recognize_face(void *pvParameters){
    recognize_status=true;
    while(1){
        Serial2.println("recog="+face_name_compare);
        String rx = Serial2.readStringUntil('\r\n');
        if(rx.indexOf("recognize face")!= -1){
            Serial.println("recognizing face");
            break;
        }
        vTaskDelay(1000);
    }
    while(1){
        Serial2.setTimeout(30000);
        String rx = Serial2.readStringUntil('\r\n');
        if(rx.indexOf("face_name:")!= -1){
            face_name = rx.substring(11,rx.length()-1);
            Serial.println("face recognized");
            break;
        }
        Serial2.setTimeout(1000);
        Serial.println("timeout");
        face_name="";
        break;
        vTaskDelay(10);
    }
    recognize_status=false;
    vTaskDelete(NULL);
}
void Task_server_hw_update(void *pvParameters){
    while (1){
        while(recognize_status==true){
            vTaskDelay(500);
        }
        while(wifi_connected==false){vTaskDelay(1000);}
        WiFiClient clientHTTPClient http;

```

```

String data;
if(face_name_compare == face_name){
    data = "hardware_command=recognized";
    recognized_status = true;
}else{
    data = "hardware_command=not recognized";
}
http.begin(client, server); // Replace with your server URL
http.addHeader("Content-Type", "application/x-www-form-urlencoded"); // Set
content type
http.setTimeout(5000); // Set the timeout
int httpResponseCode = http.POST(data);
if (httpResponseCode > 0) {
    Serial.print("POST request response: ");
    Serial.println(httpResponseCode);
    String payload = http.getString();
    Serial.println(payload);
    break;
} else {
    Serial.println("Error sending POST request");
}
http.end();
vTaskDelay(1000);
}
vTaskDelete(NULL);
}
void Task_server_update(void *pvParameters){
    while(1){
        while(wifi_connected==false || recognize_status==true){vTaskDelay(1000);}
        WiFiClient client;
        HTTPClient http;
        String data = "update_box=1&box_1_status=" + String(box_1_status)
            + "&box_2_status=" + String(box_2_status)
            + "&box_3_status=" + String(box_3_status);
        http.begin(client, server); // Replace with your server URL
        http.addHeader("Content-Type", "application/x-www-form-urlencoded"); // Set
        content type
        http.setTimeout(5000); // Set the timeout
        int httpResponseCode = http.POST(data);
        if (httpResponseCode > 0) {
            Serial.print("POST request response: ");
            Serial.println(httpResponseCode)
            String payload = http.getString();

```

```

        Serial.println(payload);
    } else {
        Serial.println("Error sending POST request");
    }
    http.end();
    data = "update_command=1";
    http.begin(client, server); // Replace with your server URL
    http.addHeader("Content-Type", "application/x-www-form-urlencoded"); // Set
content type
    http.setTimeout(5000); // Set the timeout

    httpStatusCode = http.POST(data);
    if (httpStatusCode > 0) {
        Serial.print("POST request response: ");
        Serial.println(httpStatusCode);
        String payload = http.getString();
        Serial.println(payload);
        server_command = payload;
    } else {
        Serial.println("Error sending POST request");
        http.end();
        continue;
    }
    http.end();
    data = "reset_command=1";
    http.begin(client, server); // Replace with your server URL
    http.addHeader("Content-Type", "application/x-www-form-urlencoded"); // Set
content type
    http.setTimeout(5000); // Set the timeout
    httpStatusCode = http.POST(data);
    if (httpStatusCode > 0) {
        Serial.print("POST request response: ");
        Serial.println(httpStatusCode);
        String payload = http.getString();
        Serial.println(payload);
    } else {
        Serial.println("Error sending POST request");
    }
    http.end();
    Serial.println(uxTaskGetStackHighWaterMark(NULL));
    Serial.print("Free heap: ");
    Serial.println(ESP.getFreeHeap())
    vTaskDelay(2000);

```

```

    }
}
void reset_pos(){
    TaskHandle_t task0;
    xTaskCreatePinnedToCore(Task0, "task0", 2056, NULL, 3, &task0, 0);
}
void take(int num){
    if(num==1){
        TaskHandle_t task1;
        xTaskCreatePinnedToCore(Task1, "task1", 2056, NULL, 3, &task1, 0);
    }
    if(num==2){
        TaskHandle_t task2;
        xTaskCreatePinnedToCore(Task2, "task2", 2056, NULL, 3, &task2, 0);
    }
    if(num==3){
        TaskHandle_t task3;
        xTaskCreatePinnedToCore(Task3, "task3", 2056, NULL, 3, &task3, 0);
    }
}
void Return(int num){
    if(num==1){
        TaskHandle_t task4;
        xTaskCreatePinnedToCore(Task4, "task4", 2056, NULL, 3, &task4, 0);
    }
    if(num==2){
        TaskHandle_t task5;
        xTaskCreatePinnedToCore(Task5, "task5", 2056, NULL, 3, &task5, 0);
    }
    if(num==3){
        TaskHandle_t task6;
        xTaskCreatePinnedToCore(Task6, "task6", 2056, NULL, 3, &task6, 0);
    }
}
void Task0(void *pvParameters){
    while(true){
        while(task_running!=0){
            if(task_running==-1){
                task_running=0;
            }
            vTaskDelay(500);
        }
        if(motor_driver_ready_cek()==true){

```

```

while(1){
    Serial.println("reset_pos");
    Serial1.println("reset_pos");
    String rx = Serial1.readStringUntil('\r\n');
    if(rx.indexOf("reset_pos ok")!=-1){
        break;
    }
    vTaskDelay(500);
}
break;
}
vTaskDelay(500);
}
task_running=-1;
vTaskDelete(NULL);
}

void Task1(void *pvParameters){
    while(true){
        while(task_running!=1){
            if(task_running==-1){
                task_running=1;
            }
            vTaskDelay(500);
        }
        if(motor_driver_ready_cek()==true){
            while(1){
                Serial.println("take_1");
                Serial1.println("take_1");
                String rx = Serial1.readStringUntil('\r\n');
                if(rx.indexOf("take_1 ok")!=-1){
                    break;
                }
                vTaskDelay(500);
            }
            break;
        }
        vTaskDelay(500);
    }
    task_running=-1;
    vTaskDelete(NULL);
}

void Task2(void *pvParameters){

```



```

while(true){
    while(task_running!=2){
        if(task_running===-1){
            task_running=2;
        }
        vTaskDelay(500);
    }
    if(motor_driver_ready_cek()==true){
        while(1){
            Serial.println("take_2");
            Serial1.println("take_2");
            String rx = Serial1.readStringUntil('\r\n');
            if(rx.indexOf("take_2 ok")!= -1){
                break;
            }
            vTaskDelay(500);
        }
        break;
    }
    vTaskDelay(500);
}
task_running=-1;
vTaskDelete(NULL);
}

void Task3(void *pvParameters){
    while(true){
        while(task_running!=3){
            if(task_running===-1){
                task_running=3;
            }
            vTaskDelay(500);
        }
        if(motor_driver_ready_cek()==true){
            while(1){
                Serial.println("take_3");
                Serial1.println("take_3");
                String rx = Serial1.readStringUntil('\r\n');
                if(rx.indexOf("take_3 ok")!= -1){
                    break;
                }
                vTaskDelay(500);
            }
            break;
        }
    }
}

```

```

    }
    vTaskDelay(500);
}
task_running=-1;
vTaskDelete(NULL);
}
void Task4(void *pvParameters){
    while(true){
        while(task_running!=4){
            if(task_running===-1){
                task_running=4;
            }
            vTaskDelay(500);
        }

        if(motor_driver_ready_cek()==true){
            while(1){
                Serial.println("Return_1");
                Serial1.println("Return_1");
                String rx = Serial1.readStringUntil('\r\n');
                if(rx.indexOf("Return_1 ok")!= -1){
                    break;
                }
                vTaskDelay(500);
            }
            break;
        }
        vTaskDelay(500);
    }
    task_running=-1;
    vTaskDelete(NULL);
}
void Task5(void *pvParameters){
    while(true){
        while(task_running!=5){
            if(task_running===-1){
                task_running=5;
            }
            vTaskDelay(500);
        }
        if(motor_driver_ready_cek()==true){
            while(1){
                Serial.println("Return_2");

```

```

        Serial1.println("Return_2");
        String rx = Serial1.readStringUntil('\r\n');
        if(rx.indexOf("Return_2 ok")!=-1){
            break;
        }
        vTaskDelay(500);
    }
    break;
}
vTaskDelay(500);
}
task_running=-1;
vTaskDelete(NULL);
}
void Task6(void *pvParameters){
    while(true){
        while(task_running!=6){
            if(task_running===-1){
                task_running=6;
            }
            vTaskDelay(500);
        }
        if(motor_driver_ready_cek()==true){
            while(1){
                Serial.println("Return_3");
                Serial1.println("Return_3");
                String rx = Serial1.readStringUntil('\r\n');
                if(rx.indexOf("Return_3 ok")!=-1){
                    break;
                }
                vTaskDelay(500);
            }
            break;
        }
        vTaskDelay(500);
    }
    task_running=-1;
    vTaskDelete(NULL);
}
bool motor_driver_ready_cek(){
    Serial1.println("ping");
    String rx = Serial1.readStringUntil('\r\n');
    if(rx.indexOf("ping ok") != -1){

```

```

        return true;
    }else{
        return false;
    }
}

void init_task(){
    TaskHandle_t task_cek_wifi;
    xTaskCreatePinnedToCore(Task_cek_wifi, "task_cek_wifi", 4096, NULL, 1,
    &task_cek_wifi, 1);
    TaskHandle_t task_server_update;
    xTaskCreatePinnedToCore(Task_server_update, "task_server_update", 4096,
    NULL, 3, &task_server_update, 1);
}

void update_lcd(){
    if(server_command=="\r\n" || server_command==""){
        lcd.clear();
        lcd.print("  Informasi  ");
        lcd.setCursor(0,1);
        lcd.print(" S1:"+String(box_1_status)+" S2:"+String(box_2_status)+"
S3:"+String(box_3_status));
    }else{
        lcd_update_timing=millis();
    }

    if(server_command.indexOf("recognize")!= -1){
        lcd.clear();
        lcd.print("Mengenal Wajah");
    }
    if(recognized_status==true){
        recognized_status=false;
        lcd.clear();
        lcd.print("Halo " + face_name);
    }
    if(server_command.indexOf("button1 pinjam")!= -1){
        lcd.clear();
        lcd.print("Pinjam S1");
    }
    if(server_command.indexOf("button2 pinjam")!= -1){
        lcd.clear();
        lcd.print("Pinjam S2");
    }
    if(server_command.indexOf("button3 pinjam")!= -1)
        lcd.clear();
}

```

```

    lcd.print("Pinjam S3");
}
if(server_command.indexOf("button1 kembalikan")!= -1){
    lcd.clear();
    lcd.print("Kembalikan S1");
}
if(server_command.indexOf("button2 kembalikan")!= -1){
    lcd.clear();
    lcd.print("Kembalikan S2");
}
if(server_command.indexOf("button3 kembalikan")!= -1){
    lcd.clear();
    lcd.print("Kembalikan S3");
}
}
void Task_cek_wifi(void *pvParameters){
    while(true){
        cek_wifi();
        if(wifi_connected==false){
            vTaskDelay(500);
        }else{
            vTaskDelay(3000);
        }
    }
}
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;
        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;  
  }  
}  
}
```



Lampiran C. Daftar Riwayat Hidup



RIZQI MAULA HAMDANI lahir di Jember pada tanggal 25 November 2001. Anak kedua dari pasangan Bapak Haryanto dan Ibu Murti Yuliantika, mempunyai satu saudara perempuan bernama Alief Qurrotu Ainin. Tinggal di Jl. Letjend Suprpto Gang 14 Nomor 10 Kecamatan Sumbersari, Kabupaten Jember. Dengan Pendidikan formal:

1. Sekolah Dasar Negeri 1 Kencong lulus tahun 2014.
2. Sekolah Menengah Pertama Negeri 1 Kencong lulus tahun 2017.
3. Madrasah Aliyah Negeri 3 Jember lulus tahun 2020.

Pada Tahun 2021 diterima sebagai Taruna di Politeknik Penerbangan Surabaya, Jurusan Teknik Listrik Bandara Angkatan ke XVI. Melaksanakan *On The Job Training* I di Bandar Udara Mutiara Sis Al-Jufrie Palu yang terhitung mulai tanggal 8 Mei 2023 – 22 September 2023 dan *On The Job Training* II di Bandar Udara Internasional Supadio Pontianak yang terhitung mulai tanggal 2 Oktober 2023 – 29 Februari 2024. Telah melaksanakan proyek akhir sebagai syarat kelulusan dalam pendidikan di Politeknik Penerbangan Surabaya