

**RANCANG BANGUN DAN MONITORING KEAMANAN RUANGAN
DENGAN PENANGKAPAN GAMBAR BERBASIS *INTERNET OF
THINGS (IOT)* DAN *ARTIFICIAL INTELLIGENCE (AI)***

PROYEK AKHIR



**Oleh:
RICKY HENDRADYANTO PUTRO
NIT. 30221019**

**PROGRAM STUDI DIPLOMA 3 TEKNIK NAVIGASI UDARA
POLITEKNIK PENERBANGAN SURABAYA
2024**

**RANCANG BANGUN DAN MONITORING KEAMANAN RUANGAN
DENGAN PENANGKAPAN GAMBAR BERBASIS *INTERNET OF
THINGS (IOT)* DAN *ARTIFICIAL INTELLIGENCE (AI)***

PROYEK AKHIR

Diajukan Sebagai Syarat Menempuh Mata Kuliah Tugas Akhir
Pada Program Studi Diploma 3 Teknik Navigasi Udara



Oleh:

RICKY HENDRADYANTO PUTRO

NIT. 30221019

**PROGRAM STUDI DIPLOMA 3 TEKNIK NAVIGASI UDARA
POLITEKNIK PENERBANGAN SURABAYA
2024**

HALAMAN PERSETUJUAN

**RANCANG BANGUN DAN MONITORING KEAMANAN RUANGAN
DENGAN PENANGKAPAN GAMBAR BERBASIS *INTERNET OF
THINGS* (IOT) DAN *ARTIFICIAL INTELLIGENCE* (AI)**

Oleh:

RICKY HENDRADYANTO PUTRO

NIT. 30221019

Disetujui untuk diujikan pada:
Surabaya, 2 Juli 2024

Pembimbing I

: **BAMBANG BAGUS H., S.SiT, M.M, M.T**
NIP. 19810915 200502 1 001

Pembimbing II

: **Dr. SUDRAJAT, S.E, M.M**
NIP. 19600514 197912 1 001

HALAMAN PENGESAHAN

**RANCANG BANGUN DAN MONITORING KEAMANAN RUANGAN
DENGAN PENANGKAPAN GAMBAR BERBASIS *INTERNET OF
THINGS (IOT)* DAN *ARTIFICIAL INTELLIGENCE (AI)***

Oleh:

RICKY HENDRADYANTO PUTRO
NTT. 30221019

Telah dipertahankan dan dinyatakan lulus pada Ujian Proyek Akhir
Program Pendidikan Diploma 3 Teknik Navigasi Udara
Politeknik Penerbangan Surabaya

Pada tanggal: 2 Juli 2024
Panitia Penguji:

Ketua : **NYARIS PAMBUDIYATNO, S.SiT., M.MTr.**
NIP. 19820525 200502 1 001

Sekretaris : **Dr. PRASETYO I., S.T, M.M**
NIP. 19730916 199703 1 004

Anggota : **BAMBANG BAGUS H., S.SiT, M.M, MT**
NIP. 19810915 200502 1 001

Ketua Program Studi
D.3 Teknik Navigasi Udara

NYARIS PAMBUDIYATNO, S.SiT, M.MTr.
NIP. 19820525 200502 1 001

HALAMAN PERSEMBAHAN

MOTTO :

“This too Shall Pass”

"Ini pun akan berlalu" yang menunjukkan bahwa apapun yang ada di dunia ini bersifat sementara.

PERSEMBAHAN :

Allah SWT yang telah melimpahkan rahmat dan hidayahNya.

Orang tua 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.

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa, karena berkat limpahan Rahmat dan Hidayah-Nya yang telah memberikan kesehatan, pengetahuan, ketrampilan, pengalaman yang senantiasa diberikan kepada penulis, sehingga penulis dapat menyelesaikan Proyek Akhir yang berjudul “Rancang Bangun dan Monitoring Keamanan Ruangan Dengan Penangkapan Gambar Berbasis *Internet Of Things* (IoT) dan *Artificial Intelligence* (AI)”.

Pada kesempatan ini penulis ingin menyampaikan terima kasih kepada pihak - pihak yang telah membantu penulis dalam penyusunan Proyek Akhir diantaranya :

1. Bapak Ahmad Bahrawi, S.E., M.T. selaku Direktur Politeknik Penerbangan Surabaya yang telah memberikan kesempatan penulis untuk menempuh jenjang pendidikan Vokasi di Politeknik Penerbangan Surabaya.
2. Bapak Nyaris Pambudiyatno, S.SiT, M.MTr. sebagai Ketua Program Studi Diploma III Teknik Navigasi Udara.
3. Bapak Bambang Bagus H., S.SiT, M.M, M.T. sebagai dosen pembimbing pertama yang telah meluangkan waktu untuk membimbing, serta memberi saran.
4. Bapak Dr. Sudrajat, S.E, M.M sebagai dosen pembimbing kedua yang telah meluangkan waktu untuk membimbing, serta memberi saran.
5. Dosen-dosen penguji sidang proyek akhir sebagai penguji penelitian proyek akhir penulis yang telah meluangkan waktu, ide dan masukannya.
6. Seluruh Dosen dan *civitas* akademik Diploma III Teknik Navigasi Udara atas ilmu dan pengetahuan yang telah diberikan.
7. Keluarga tercinta yang selalu memberikan dukungan serta motivasi penulis selama studi di Politeknik Penerbangan Surabaya.
8. Rekan-rekan dari Airman 21 terutamanya pada *course* Diploma III Teknik Navigasi Udara Angkatan 14 yang telah menjadi rekan baik saya.

Tentunya karya tulis ini masih jauh dari sempurna. Penulis berharap semoga Proyek Akhir ini dapat memberikan manfaat khususnya bagi taruna Politeknik kami memohon maaf. Saran dan kritik membangun penulis harapkan demi karya yang lebih baik di masa mendatang.

Surabaya, 2 Juli 2024



Penulis

ABSTRAK

RANCANG BANGUN DAN MONITORING KEAMANAN RUANGAN DENGAN PENANGKAPAN GAMBAR BERBASIS *INTERNET OF THINGS* (IOT) DAN *ARTIFICIAL INTELLIGENCE* (AI)

Oleh:

RICKY HENDRADYANTO PUTRO

NIT. 30221019

Sistem keamanan adalah mekanisme yang menghilangkan rasa takut, cemas, dan khawatir terkait dengan meninggalkan barang berharga, sehingga membebaskan seseorang dari segala risiko. Sistem keamanan pengenalan wajah atau identitas seseorang ketika memasuki ruangan yang memiliki nilai berharga dengan ditambah dengan adanya pemberitahuan melalui ponsel cerdas. Dengan adanya sistem tersebut, dapat memudahkan dan mengingatkan seseorang dalam mengawasi barang berharganya. Sehingga ruangan tersebut dapat dimonitoring dengan berbasis *Internet Of Things* (IoT) dengan di kombinasikan dengan teknologi *Artificial Intelligence* (AI) dan pemberitahuan melalui aplikasi Telegram.

Metode yang digunakan untuk mendukung dalam melaksanakan rancang bangun sistem keamanan dengan penangkapan gambar adalah metode kuantitatif deskriptif dengan model *Prototyping*. *Prototyping* tersebut adalah metode pendekatan di mana model atau *Prototype* awal dari sistem yang akan dikembangkan dibuat dan diuji sebelum pengembangan penuh dimulai.

Hasil penelitian menunjukkan bahwa sistem keamanan ruangan berbasis *Internet Of Things* (IoT) dan *Artificial Intelligence* (AI) dapat dilakukan dengan baik. Akan tetapi, masih ada beberapa perbaikan dan pengembangan supaya perancangan tersebut dapat digunakan atau difungsikan secara maksimal oleh pengguna.

Kata Kunci : Sistem Keamanan, *Internet Of Things* (IoT), *Artificial Intelligence* (AI)

ABSTRACT

DESIGN AND MONITORING OF ROOM SECURITY WITH IMAGE CAPTURE BASED ON INTERNET OF THINGS (IOT) AND ARTIFICIAL INTELLIGENCE (AI)

By:

RICKY HENDRADYANTO PUTRO

NIT. 30221019

A security system is a mechanism that eliminates fear, anxiety, and worry associated with leaving valuables behind, thereby freeing a person from all risks. The security system recognizes a person's face or identity when entering a room that has valuable value coupled with a notification via smartphone. With this system, it can facilitate and remind someone to keep an eye on their valuables. So that the room can be monitored based on the Internet Of Things (IoT) in combination with Artificial Intelligence (AI) technology and notification via the Telegram application.

The method used to support in carrying out the design of a security system with image capture is a descriptive quantitative method with a Prototyping model. Prototyping is an approach method in which an initial model or Prototype of the system to be developed is created and tested before full development begins.

The results showed that the room security system based on Internet Of Things (IoT) and Artificial Intelligence (AI) can be done well. However, there are still some improvements and developments so that the design can be used or functioned optimally by users.

Keywords : Security System, Internet Of Things (IoT), Artificial Intelligence (AI)

PERNYATAAN KEASLIAN DAN HAK CIPTA

Saya yang bertanda tangan di bawah ini :

Nama : Ricky Hendradyanto Putro
NIT : 30221019
Program Studi : D.3 Teknik Navigasi Udara XIV
Judul Tugas Akhir : Rancang Bangun Dan Monitoring Keamanan Ruangan Dengan Penangkapan Gambar Berbasis *Internet Of Things* (IoT) Dan *Artificial Intelligence* (AI)

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 lainnya, 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 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 tugas akhir saya dengan tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya. Apabila dikemudian hari terdapat penyimpangan dan tidak benaran. 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, 2 Juli 2024
Yang membuat pernyataan



Ricky Hendradyanto Putro

Ricky Hendradyanto Putro
NIT. 30221019

DAFTAR ISI

	Halaman
HALAMAN PERSETUJUAN.....	ii
HALAMAN PENGESAHAN.....	iii
HALAMAN PERSEMBAHAN	iv
KATA PENGANTAR	v
ABSTRAK	vii
<i>ABSTRACT</i>	viii
PERNYATAAN KEASLIAN DAN HAK CIPTA.....	ix
DAFTAR ISI.....	x
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiv
DAFTAR LAMPIRAN.....	xv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	4
1.6 Sistematika Penulisan.....	5
BAB 2 LANDASAN TEORI.....	6
2.1 Tinjauan Teori	6
2.1.1 Metode <i>Prototype</i>	6
2.1.2 Rancang Bangun	8
2.1.3 <i>Monitoring</i>	10
2.1.4 Sistem Keamanan.....	12
2.1.5 <i>Internet Of Things</i> (IoT).....	13
2.1.6 <i>Artificial Intelligence</i> (AI).....	20
2.1.7 <i>Face Recognition</i>	24
2.1.8 Kamera	25
2.1.9 ESP32.....	26

2.1.10	Motor Servo	31
2.1.11	<i>Software Arduino IDE</i>	33
2.1.12	Telegram	37
2.2	Kajian Penelitian Terdahulu yang Relevan	38
BAB 3 METODE PENELITIAN.....		43
3.1	Desain Penelitian	43
3.2	Perancangan Alat.....	45
3.2.1	Diagram Rancangan	45
3.2.2	<i>Flowchart</i> cara kerja	46
3.2.3	Komponen Alat	47
3.3	Tempat dan Waktu Penelitian	48
BAB 4 HASIL DAN PEMBAHASAN.....		50
4.1	Identifikasi Kebutuhan	50
4.2	Desain Rancangan <i>Mock up</i>	50
4.3	Pembuatan <i>Mock up</i>	51
4.4	Uji Coba Mockup	68
4.5	Implementasi <i>Mock up</i>	74
4.6	Evaluasi Sistem	75
4.7	Kelebihan dan Kekurangan <i>Mock up</i>	76
4.8	Keunggulan Mock up dari Kajian Penelitian yang Terdahulu	76
BAB 5 PENUTUP		78
5.1	Kesimpulan.....	78
5.2	Saran.....	78

DAFTAR PUSTAKA

LAMPIRAN

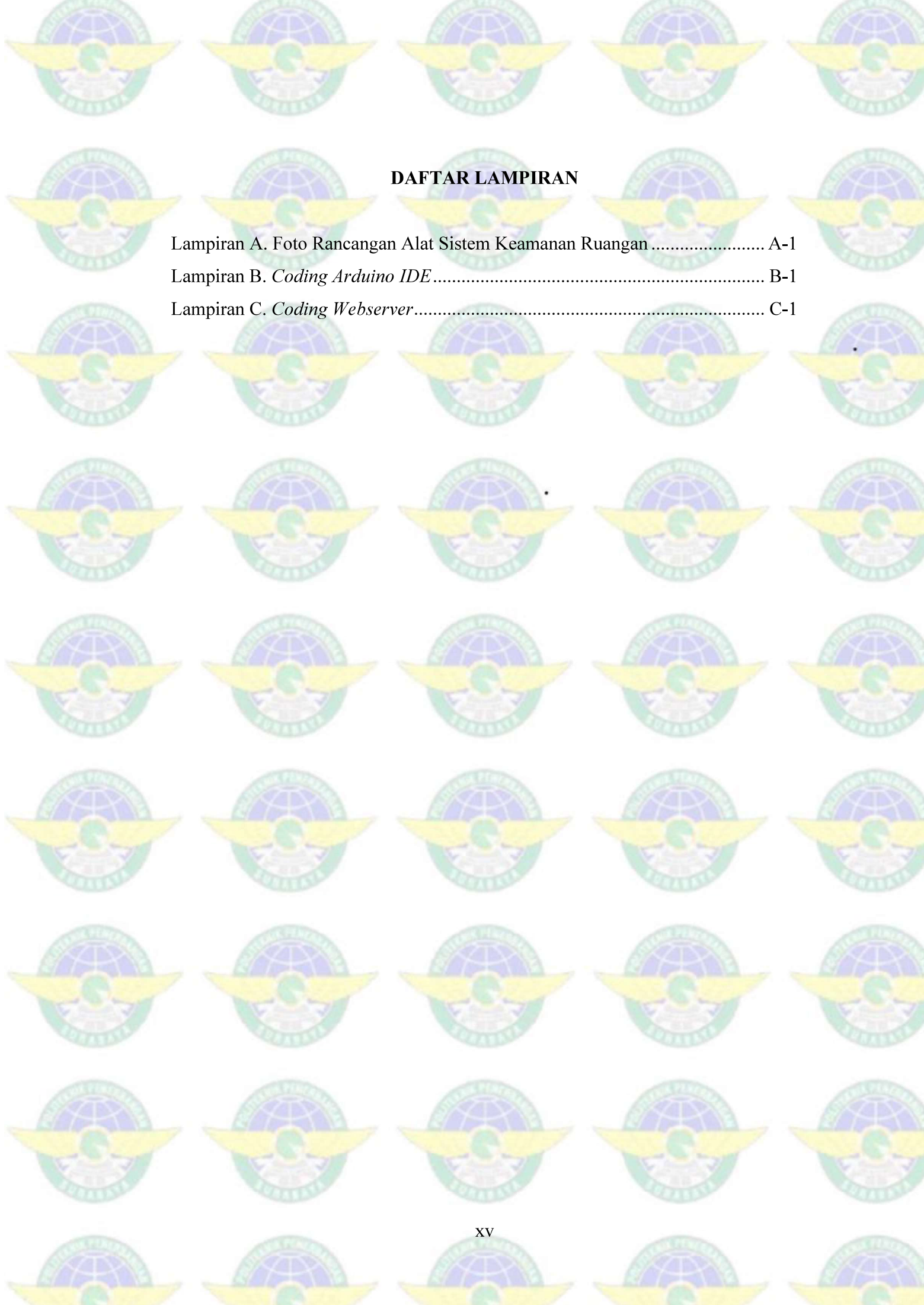
DAFTAR GAMBAR

Gambar 2. 1 Tahapan Pengembangan Prototype	8
Gambar 2. 2 Referensi Model IoT	16
Gambar 2. 3 Proses <i>Face Recognition</i>	24
Gambar 2. 4 ESP32-CAM dengan kamera OV2640	26
Gambar 2. 5 ESP32-WROOM-32U	29
Gambar 2. 6 Motor Servo Arduino	32
Gambar 2. 7 Logo <i>Arduino IDE</i>	33
Gambar 2. 8 <i>Display</i> Perangkat Lunak <i>Arduino IDE</i>	34
Gambar 2. 9 Logo Aplikasi Telegram	37
Gambar 3. 1 Desain Penelitian	44
Gambar 3. 2 Diagram Rancangan	45
Gambar 3. 3 Flowchart Cara Kerja	46
Gambar 4. 1 Desain rancangan alat	51
Gambar 4. 2 Rancangan alat sebelum jadi	51
Gambar 4. 3 Hasil rancangan alat	52
Gambar 4. 4 Tampilan <i>Arduino IDE</i>	52
Gambar 4. 5 Instalasi Library	53
Gambar 4. 6 Input URL ESP 32 Boards	54
Gambar 4. 7 Tampilan <i>Adobe Dream Weaver</i>	54
Gambar 4. 8 <i>Coding</i> untuk judul tab	54
Gambar 4. 9 <i>Coding</i> untuk pengaturan data laman <i>webserver</i>	55
Gambar 4. 10 <i>Codingan</i> tampilan dalam <i>webserver</i>	55
Gambar 4. 11 <i>Coding</i> perintah <i>webserver</i> terhadap <i>hardware</i>	56
Gambar 4. 12 Hasil dari tampilan <i>Codingan</i> dari <i>webserver</i>	56
Gambar 4. 13 <i>BotFather</i> Pada Telegram	57
Gambar 4.14 Tampilan Awal Pada <i>BotFather</i>	57
Gambar 4.15 Display Awal Akan Membuat Bot Baru	58
Gambar 4. 16 Nama Bot dan Token sudah Valid	58
Gambar 4. 17 Membuat <i>IDBot</i>	59

Gambar 4. 18 <i>Coding</i> untuk menyambungkan wifi	59
Gambar 4. 19 <i>Coding</i> untuk mengaktifkan kamera	60
Gambar 4. 20 <i>Coding</i> untuk PIN LED.....	60
Gambar 4. 21 <i>Coding</i> untuk menginstal LED pada void setup.....	60
Gambar 4. 22 <i>Coding</i> untuk PIN servo	60
Gambar 4. 23 <i>Coding</i> menambahkan objek servo	61
Gambar 4. 24 <i>Coding</i> menambahkan posisi awal servo	61
Gambar 4. 25 <i>Coding</i> menginstal servo pada void setup.....	61
Gambar 4. 26 Menambahkan pembacaan kode untuk <i>webserver</i>	61
Gambar 4. 27 <i>Coding</i> pengaktifan <i>webserver</i>	61
Gambar 4. 28 Menginstal <i>webserver</i> supaya dapat diakses	62
Gambar 4. 29 Penyimpanan pengenalan wajah dan daftar wajah.....	62
Gambar 4. 30 <i>Coding</i> perintah pengenalan data	62
Gambar 4. 31 Menginstal data pengenalan wajah	63
Gambar 4. 32 Penangkapan data pengenalan wajah	63
Gambar 4. 33 Penyimpanan data pengenalan wajah.....	63
Gambar 4. 34 Menghapus data pengenalan wajah.....	63
Gambar 4. 35 Perintah dari <i>webserver</i>	64
Gambar 4. 36 <i>Coding string</i> pada <i>token</i> untuk bot dan <i>chatid</i>	65
Gambar 4. 37 Perintah untuk mengirimkan pesan ke Telegram.....	65
Gambar 4. 38 Perintah untuk mengirimkan gambar ke Telegram	66
Gambar 4. 39 <i>Coding Void Setup</i>	67
Gambar 4. 40 <i>Coding Void Loop</i>	67
Gambar 4. 41 Tampilan pesan Telegram ketika memulai perangkat.....	68
Gambar 4. 42 Tampilan <i>webserver</i>	69
Gambar 4. 43 Pesan pada <i>webserver</i> ketika wajah terdeteksi.....	70
Gambar 4. 44 Pesan pada <i>webserver</i> ketika wajah dikenali	71
Gambar 4. 45 Pesan pada <i>webserver</i> apabila wajah tidak dikenali.....	71
Gambar 4. 46 Hasil pengiriman pesan dan gambar ke Telegram	73

DAFTAR TABEL

Tabel 2. 1 Perbedaan Kecerdasan Buatan dan Kecerdasan Alami.....	20
Tabel 2. 2 Kajian Pustaka Terdahulu yang Relevan	39
Tabel 3. 1 Tempat dan Waktu Penelitian.....	48
Tabel 4. 1 Pengujian pengiriman pesan saat memulai perangkat	68
Tabel 4. 2 Pengujian fungsi tombol pada <i>webserver</i>	69
Tabel 4. 3 Pengujian pada deteksi wajah dalam kondisi cahaya matahari.....	70
Tabel 4. 4 Pengujian pada deteksi wajah dalam kondisi cahaya ruangan.....	70
Tabel 4. 5 Pengujian akses keamanan dalam kondisi cahaya matahari	72
Tabel 4. 6 Pengujian akses keamanan dalam kondisi cahaya ruangan	72
Tabel 4. 7 Rentang pengambilan foto pada <i>webserver</i> ke Telegram	73
Tabel 4. 8 Rentang pengiriman pesan ketika wajah dikenali ke Telegram.....	73
Tabel 4. 9 Rentang pengiriman pesan dan pengambilan foto ke Telegram	74

The page is decorated with a repeating pattern of the Politeknik Pemasangan Surabaya logo. The logo is circular, featuring a globe with a green 'S' in the center, and the text 'POLITEKNIK PEMASANGAN SURABAYA' around the perimeter. The background of the page is white, and the logos are arranged in a grid-like pattern.

DAFTAR LAMPIRAN

Lampiran A. Foto Rancangan Alat Sistem Keamanan Ruangan A-1

Lampiran B. *Coding Arduino IDE* B-1

Lampiran C. *Coding Webserver* C-1

DAFTAR PUSTAKA

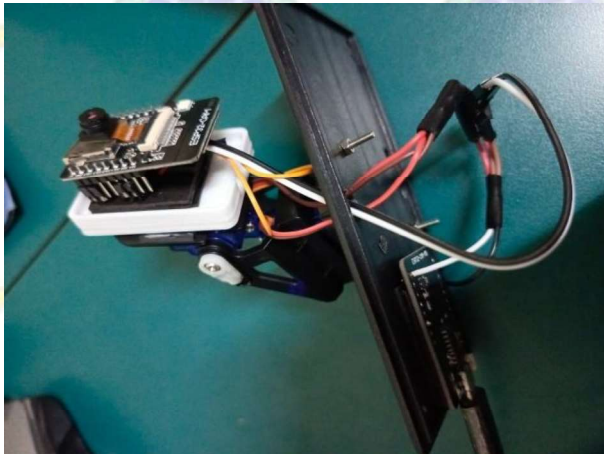
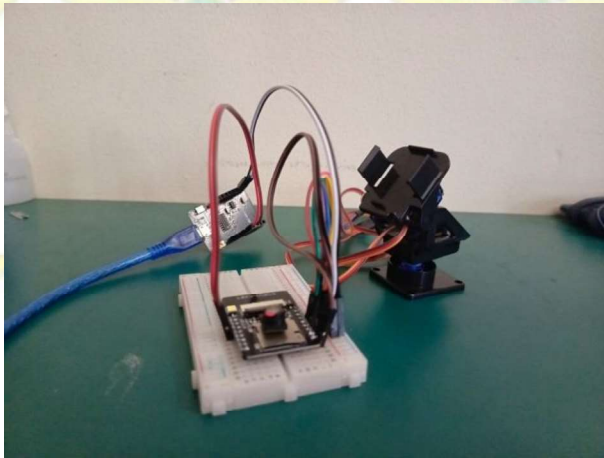
- Ahmad Hania, A. (2017). Mengenal Artificial Intelligence, Machine Learning, & Deep Learning. *Jurnal Teknologi Indonesia*, 1(June), 1–6. <https://amt-it.com/mengenal-perbedaan-artificial-intelligence-machine-learning-deep-learning/>
- Alfarisi Salman. (2019). Aplikasi Media Pengenalan Jenis Kamera dan Lensa Berbasis Android. *Jurnal Sisfotek Global*, 9(1), 1–7.
- Arrohman, O., & Wahyuni, C. T. (2022). Rancang Bangun Aplikasi Animasi Pembelajaran Matakuliah Teori Bahasa Otomata Pada Sekolah Tinggi Teknologi Pagar Alam. *Jurnal Nasional Ilmu Komputer*, 3(1), 9–17. <https://doi.org/10.47747/jurnalnuk.v3i1.615>
- Aseffa, D. T., Science, A., Kalla, H., Science, A., Mishra, S., & Science, A. (2021). Implementation of Iot Based Smart Surveillance Security System Using Face Recognition. *Adalya Journal*, 10(1). <https://doi.org/10.37896/aj10.1/013>
- Budiyanti, R. T. (2021). *Buku Ajar Internet of Things*.
- Efendi, Y. (2018). Internet Of Things (Iot) Sistem Pengendalian Lampu Menggunakan Raspberry Pi Berbasis Mobile. *Jurnal Ilmiah Ilmu Komputer*, 4(1), 19–26. <https://doi.org/10.35329/jiik.v4i1.48>
- Eriana, E. S., & Zein, D. A. (2023). *Artificial Intelligence (Ai) Penerbit Cv. Eureka Media Aksara*. 24–32.
- Fridayanthie, E. W., Haryanto, H., & Tsabitah, T. (2021). Penerapan Metode Prototype Pada Perancangan Sistem Informasi Penggajian Karyawan (Persis Gawan) Berbasis Web. *Paradigma - Jurnal Komputer dan Informatika*, 23(2), 151–157. <https://doi.org/10.31294/p.v23i2.10998>
- Hafni Sahir, S. (2021). *Metodologi Penelitian* (M. S. Dr. Ir. Try Koryati (ed.)). PENERBIT KBM INDONESIA.
- Hilal, A., & Manan, S. (2015). Pemanfaatan Motor Servo Sebagai Penggerak Cctv Untuk Melihat Alat-Alat Monitor Dan Kondisi Pasien Di Ruang Icu. *Gema Teknologi*, 17(2), 95–99. <https://doi.org/10.14710/gt.v17i2.8924>
- Himawan, F., Pressa Perdana, & Yoedo Ageng Surya. (2021). Rancang Bangun

- Purwarupa Smart Garden Menggunakan Kamera, Sensor Suhu Dan Kelembaban Tanah Berbasis Internet Of Things (IOT) Dengan ESP8266. *Jurnal JEETech*, 2(2), 78–83. <https://doi.org/10.48056/jeetech.v2i2.171>
- Hindarto, Sumarno, & Rosid, M. A. (n.d.). *Buku Ajar Kecerdasan Buatan/Artificial Intelegent (AI)* (M. T. Multazam & M. Darmawan (ed.); Nomor Agustus 2022). UMSIDA Press.
- Imran, A., & Rasul, M. (2020). Pengembangan Tempat Sampah Pintar Menggunakan Esp32. *Jurnal Media Elektrik*, 17(2), 2721–9100. <https://ojs.unm.ac.id/mediaelektrik/article/view/14193>
- Jamaaluddin, & Indah, S. (2021). Buku Ajar Kecerdasan Buatan. *Umsida Press*, 121.
- Megawaty, D. A., & Putra, M. E. (2020). Aplikasi Monitoring Aktivitas Akademik Mahasiswa Program Studi Informatika Universitas Xyz Berbasis Android. *Jurnal Informatika dan Rekayasa Perangkat Lunak*, 1(1), 65–74. <https://doi.org/10.33365/jatika.v1i1.177>
- Purnomo, D. (2017). Penerapan Metode Prototype Pada Pengembangan Sistem Informasi Tugas Akhir Mahasiswa. *JIMP-Jurnal Informatika Merdeka Pasuruan*, 2(2), 54–61.
- Rahmat Gunawan, Arif Maulana Yusuf, & Lysa Nopitasari. (2021). Rancang Bangun Sistem Presensi Mahasiswa Dengan Menggunakan Qr Code Berbasis Android. *Elkom : Jurnal Elektronika dan Komputer*, 14(1), 47–58. <https://doi.org/10.51903/elkom.v14i1.369>
- Ramayanti, D., Jumaryadi, Y., Gufron, D. M., & Ramadha, D. D. (2023). Sistem Keamanan Perumahan Menggunakan Face Recognition. *TIN: Terapan Informatika Nusantara*, 3(12), 486–496. <https://doi.org/10.47065/tin.v3i12.4145>
- Renaningtias, N., & Apriliani, D. (2021). Penerapan Metode Prototype Pada Pengembangan Sistem Informasi Tugas Akhir Mahasiswa. *Rekursif: Jurnal Informatika*, 9(1). <https://doi.org/10.33369/rekursif.v9i1.15772>
- Salahuddin, N. S., Iramadhan, N., Sari, S. P., & Saptariani, T. (2022). Prototipe Sistem Keamanan Pintu Inkubator Bayi melalui Pengenalan Wajah

- menggunakan Kamera Web dan OpenCV berbasis Raspberry Pi. *Techno.Com*, 21(3), 579–595. <https://doi.org/10.33633/tc.v21i3.6273>
- Saramuddin. (2018). Cara Mudah Kuasai Mikrokontroler Arduino Teori Dan Praktek. *Angewandte Chemie International Edition*, 6(11), 951–952., 3(1), 10–27. <https://medium.com/@arifwicaksanaa/pengertian-use-case-a7e576e1b6bf>
- Sary, Y. (2022). Sistem Internet Of Things. In *Program Studi Sistem Informasi Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Muhammadiyah Sumatera Utara* (Vol. 2022, Nomor Januari 2022). http://link.springer.com/10.1007/978-3-319-77492-3_16
- Septian, R., & Mufti Prasetyo, S. (2022). Sistem Keamanan Menggunakan Kamera dan Sensor Gerak Berbasis Internet of Things (IoT). *OKTAL : Jurnal Ilmu Komputer dan Science*, 1(9), 1330–1339. <https://journal.mediapublikasi.id/index.php/oktal>
- Soeroso, H., Zuhri Arfianto, A., Eka Mayangsari, N., Taali, M., Teknik Bangunan Kapal, J., Teknik Kelistrikan Kapal, J., Teknik Permesinan Kapal, J., Perkapalan Negeri Surabaya Prodi Administrasi Bisnis, P., & Negeri Madiun, P. (2017). Penggunaan Bot Telegram Sebagai Announcement System pada Intansi Pendidikan. *Seminar Master PPNS, Vol 2 No 1*, 45–48.
- Sulistiyawati, W., Wahyudi, & Trinuryono, S. (2022). Analisis Motivasi Belajar Siswa Dengan Menggunakan Model Pembelajaran Blended Learning Saat Pandemi Covid-19 (Deskriptif Kuantitatif Di Sman 1 Babadan Ponorogo). *Kadikma*, 13(1), 68. <https://doi.org/10.19184/kdma.v13i1.31327>
- Surya, M. A., Susanto, M., Setyawan, A., Fitriawan, H., Elektro, T., Lampung, U., Prof, J., Dr, J., Sumantri, I., No, B., & Lampung, B. (2024). *Sistem Keamanan Ruang Dengan Human Detection Menggunakan Sensor Kamera Berbasis Deep Learning*. 18(1), 182–192. <https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index>
- Yusro, M. (2016). Modul Singkat Teori & Praktik Mikrokontroler Platform Arduino. In *Dialog* (Vol. 44, Nomor 1). Program Studi Pendidikan Teknik Elektronika FT UNJ. <https://doi.org/10.47655/dialog.v44i1.470>

LAMPIRAN

Lampiran A. Foto Rancangan Alat Sistem Keamanan Ruangan



Lampiran B. *Coding Arduino IDE*

```
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <ArduinoWebsockets.h>
#include "esp_http_server.h"
#include "esp_timer.h"
#include "esp_camera.h"
#include "camera_index.h"
#include "Arduino.h"
#include "soc/soc.h"
#include "soc/rtc_cntl_reg.h"
#include <ESP32Servo.h>
#include "fd_forward.h"
#include "fr_forward.h"
#include "fr_flash.h"

const char* ssid = "HOTSPOT POLTEKBANG";
const char* password = "";

String myToken = "7149301594:AAGVz_1LchYF95GfWi4lUCDTECj_9p-
T5KE";
String myChatId = "921626561";

#define ENROLL_CONFIRM_TIMES 5
#define FACE_ID_SAVE_NUMBER 7

#define CAMERA_MODEL_AI_THINKER
//ESP32-CAM
#define PWDN_GPIO_NUM    32
#define RESET_GPIO_NUM   -1
#define XCLK_GPIO_NUM     0
#define SIOD_GPIO_NUM     26
#define SIOC_GPIO_NUM     27

#define Y9_GPIO_NUM       35
#define Y8_GPIO_NUM       34
#define Y7_GPIO_NUM       39
#define Y6_GPIO_NUM       36
#define Y5_GPIO_NUM       21
#define Y4_GPIO_NUM       19
#define Y3_GPIO_NUM       18
#define Y2_GPIO_NUM       5
#define VSYNC_GPIO_NUM    25
```

```

#define HREF_GPIO_NUM      23
#define PCLK_GPIO_NUM      22

#define LED_FLASH_PIN 4    // Pin untuk LED flash pada ESP32-
CAM
// Variabel untuk menyimpan status LED
bool ledState = LOW;

// Servo Pins
#define SERVO_1            14
#define SERVO_2            15
#define SERVO_STEP         5

// Servo Objects
Servo servoN1;
Servo servoN2;
Servo servo1;
Servo servo2;

// Servo Initial Positions
int servo1Pos = 90;
int servo2Pos = 90;

// WebSocket Server
using namespace websockets;
WebsocketsServer socket_server;

// Camera Frame Buffer
camera_fb_t * fb = NULL;

// Timing Variables
long current_millis;
long last_detected_millis = 0;
unsigned long safe_millis = 0;
long interval = 5000;           // open lock for ...
milliseconds
bool face_recognised = false;

// Function Prototypes
void app_facenet_main();
void app_httpserver_init();
void handle_message(WebsocketsClient &client,
WebsocketsMessage msg);
String sendCapturedImage2Telegram(String token, String
chat_id);

```



```
String sendMessage2Telegram(String token, String chat_id,  
String text);
```

```
typedef struct  
{  
    uint8_t *image;  
    box_array_t *net_boxes;  
    dl_matrix3d_t *face_id;  
} http_img_process_result;
```

```
face_id_name_list st_face_list;  
static dl_matrix3du_t *aligned_face = NULL;
```

```
httpd_handle_t camera_httpd = NULL;
```

```
typedef enum  
{  
    START_STREAM,  
    START_DETECT,  
    SHOW_FACES,  
    START_RECOGNITION,  
    START_ENROLL,  
    ENROLL_COMPLETE,  
    DELETE_ALL,  
} en_fsm_state;  
en_fsm_state g_state;
```

```
typedef struct  
{  
    char enroll_name[ENROLL_NAME_LEN];  
} httpd_resp_value;  
httpd_resp_value st_name;
```

```
static inline mtmn_config_t app_mtmn_config()  
{  
    mtmn_config_t mtmn_config = {0};  
    mtmn_config.type = FAST;  
    mtmn_config.min_face = 80;  
    mtmn_config.pyramid = 0.707;  
    mtmn_config.pyramid_times = 4;  
    mtmn_config.p_threshold.score = 0.6;  
    mtmn_config.p_threshold.nms = 0.7;  
    mtmn_config.p_threshold.candidate_number = 20;  
    mtmn_config.r_threshold.score = 0.7;  
    mtmn_config.r_threshold.nms = 0.7;
```

```

mtmn_config.r_threshold.candidate_number = 10;
mtmn_config.o_threshold.score = 0.7;
mtmn_config.o_threshold.nms = 0.7;
mtmn_config.o_threshold.candidate_number = 1;
return mtmn_config;
}
mtmn_config_t mtmn_config = app_mtmn_config();

static esp_err_t index_handler(httpd_req_t *req) {
    httpd_resp_set_type(req, "text/html");
    return httpd_resp_send(req, (const char
*)SISTEM_KEAMANAN_HTML, strlen(SISTEM_KEAMANAN_HTML));
}

httpd_uri_t index_uri = {
    .uri      = "/",
    .method   = HTTP_GET,
    .handler  = index_handler,
    .user_ctx = NULL
};

void setup() {
    WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0); // Disable
brownout detector

    // Mengatur pin LED flash sebagai output
    pinMode(LED_FLASH_PIN, OUTPUT);
    digitalWrite(LED_FLASH_PIN, LOW); // Memulai dengan LED
mati

    // Servo initialization
    servo1.setPeriodHertz(50); // standard 50 hz servo
    servo2.setPeriodHertz(50); // standard 50 hz servo
    servoN1.attach(2, 1000, 2000);
    servoN2.attach(13, 1000, 2000);
    servo1.attach(SERVO_1, 1000, 2000);
    servo2.attach(SERVO_2, 1000, 2000);
    servo1.write(servo1Pos);
    servo2.write(servo2Pos);

    // Serial initialization
    Serial.begin(115200);
    Serial.setDebugOutput(true);
    Serial.println();

```



```

// Camera configuration
camera_config_t config;
config.ledc_channel = LEDC_CHANNEL_0;
config.ledc_timer = LEDC_TIMER_0;
config.pin_d0 = Y2_GPIO_NUM;
config.pin_d1 = Y3_GPIO_NUM;
config.pin_d2 = Y4_GPIO_NUM;
config.pin_d3 = Y5_GPIO_NUM;
config.pin_d4 = Y6_GPIO_NUM;
config.pin_d5 = Y7_GPIO_NUM;
config.pin_d6 = Y8_GPIO_NUM;
config.pin_d7 = Y9_GPIO_NUM;
config.pin_xclk = XCLK_GPIO_NUM;
config.pin_pclk = PCLK_GPIO_NUM;
config.pin_vsync = VSYNC_GPIO_NUM;
config.pin_href = HREF_GPIO_NUM;
config.pin_sscb_sda = SIOD_GPIO_NUM;
config.pin_sscb_scl = SIOC_GPIO_NUM;
config.pin_pwdn = PWDN_GPIO_NUM;
config.pin_reset = RESET_GPIO_NUM;
config.xclk_freq_hz = 20000000;
config.pixel_format = PIXFORMAT_JPEG;
//init with high specs to pre-allocate larger buffers
if (psramFound()) {
    config.frame_size = FRAMESIZE_UXGA;
    config.jpeg_quality = 10;
    config.fb_count = 2;
} else {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 12;
    config.fb_count = 1;
}

// camera init
esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x",
err);
    return;
}

sensor_t * s = esp_camera_sensor_get();
s->set_framesize(s, FRAMESIZE_QVGA);

#ifdef CAMERA_MODEL_M5STACK_WIDE

```

```

s->set_vflip(s, 1);
s->set_hmirror(s, 1);
#endif

// Connect to WiFi
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");

// Initialize HTTP server and WebSocket server
app_httpserver_init();
app_facenet_main();
socket_server.listen(82);

// Print IP address
Serial.print("Camera Ready! Use 'http://");
Serial.print(WiFi.localIP());
Serial.println("' to connect");

// Send IP address to Telegram
String ipAddress = WiFi.localIP().toString();
sendMessage2Telegram(myToken, myChatId, "Hello, Your IP is
http://" + ipAddress);
}

// HTTP server initialization
void app_httpserver_init ()
{
    httpd_config_t config = HTTPD_DEFAULT_CONFIG();
    if (httpd_start(&camera_httpd, &config) == ESP_OK)
        Serial.println("httpd_start");
    {
        httpd_register_uri_handler(camera_httpd, &index_uri);
    }
}

// Facenet initialization
void app_facenet_main()
{
    face_id_name_init(&st_face_list, FACE_ID_SAVE_NUMBER,
ENROLL_CONFIRM_TIMES);

```



```

    aligned_face = dl_matrix3du_alloc(1, FACE_WIDTH,
FACE_HEIGHT, 3);
    read_face_id_from_flash_with_name(&st_face_list);
}

// Enrollment function
static inline int do_enrollment(face_id_name_list
*face_list, dl_matrix3d_t *new_id)
{
    ESP_LOGD(TAG, "START ENROLLING");
    int left_sample_face =
enroll_face_id_to_flash_with_name(face_list, new_id,
st_name.enroll_name);
    ESP_LOGD(TAG, "Face ID %s Enrollment: Sample %d",
st_name.enroll_name,
ENROLL_CONFIRM_TIMES - left_sample_face);
    return left_sample_face;
}

// Send face list to client
static esp_err_t send_face_list(WebsocketsClient &client)
{
    client.send("delete_faces"); // tell browser to delete all
faces
    face_id_node *head = st_face_list.head;
    char add_face[64];
    for (int i = 0; i < st_face_list.count; i++) // loop
current faces
    {
        sprintf(add_face, "listface:%s", head->id_name);
        client.send(add_face); //send face to browser
        head = head->next;
    }
}

static esp_err_t delete_all_faces(WebsocketsClient &client)
{
    delete_face_all_in_flash_with_name(&st_face_list);
    client.send("delete_faces");
}

// Handle WebSocket messages
void handle_message(WebsocketsClient &client,
WebsocketsMessage msg)
{

```

```

if (msg.data() == "stream") {
    g_state = START_STREAM;
    client.send("Streaming");
}
if (msg.data() == "detect") {
    g_state = START_DETECT;
    client.send("Melakukan Pendeteksian Wajah");
}
if (msg.data().substring(0, 8) == "capture:") {
    g_state = START_ENROLL;
    char person[FACE_ID_SAVE_NUMBER * ENROLL_NAME_LEN] =
{0,};
    msg.data().substring(8).toCharArray(person,
sizeof(person));
    memcpy(st_name.enroll_name, person, strlen(person) + 1);
    client.send("Mendaftarkan Wajah");
}
if (msg.data() == "recognise") {
    g_state = START_RECOGNITION;
    client.send("Mengamati");
}
if (msg.data() == "servoup") {
    if(servo1Pos >= 10) {
        servo1Pos -= 10;
        servo1.write(servo1Pos);
    }
    Serial.println(servo1Pos);
    Serial.println("Up");
}
if (msg.data() == "servoleft") {
    if(servo2Pos <= 170) {
        servo2Pos += 10;
        servo2.write(servo2Pos);
    }
    Serial.println(servo2Pos);
    Serial.println("Left");
}
if (msg.data() == "servoright") {
    if(servo2Pos >= 10) {
        servo2Pos -= 10;
        servo2.write(servo2Pos);
    }
    Serial.println(servo2Pos);
    Serial.println("Right");
}
}

```



```

if (msg.data() == "servodown") {
    if(servo1Pos <= 170) {
        servo1Pos += 10;
        servo1.write(servo1Pos);
    }
    Serial.println(servo1Pos);
    Serial.println("Down");
}
if (msg.data() == "ledflash") {
    // Baca status LED saat ini
    ledState = digitalRead(LED_FLASH_PIN);

    // Jika LED dalam keadaan LOW, Menyalakan LED
    if (ledState == LOW) {
        ledState = HIGH;
        digitalWrite(LED_FLASH_PIN, HIGH);
        Serial.println("LED flash menyala");
    } else {
        // Jika LED dalam keadaan HIGH, Mematikan LED
        digitalWrite(LED_FLASH_PIN, LOW);
        ledState = HIGH;
        Serial.print("LED flash mati");
    }
}
if (msg.data() == "takephoto") {
    sendCapturedImage2Telegram(myToken, myChatId);
}
if (msg.data().substring(0, 7) == "remove:") {
    char person[ENROLL_NAME_LEN * FACE_ID_SAVE_NUMBER];
    msg.data().substring(7).toCharArray(person,
sizeof(person));
    delete_face_id_in_flash_with_name(&st_face_list,
person);
    send_face_list(client); // reset faces in the browser
}
if (msg.data() == "delete_all") {
    delete_all_faces(client);
}
}

void loop() {
    auto client = socket_server.accept();
    client.onMessage(handle_message);
    dl_matrix3du_t *image_matrix = dl_matrix3du_alloc(1, 320,
240, 3);

```

```

http_img_process_result out_res = {0};
out_res.image = image_matrix->item;

send_face_list(client);
g_state = START_RECOGNITION;
client.send("RECOGNISING");

while (client.available()) {
    client.poll();

    fb = esp_camera_fb_get();

    if (g_state == START_DETECT || g_state == START_ENROLL
|| g_state == START_RECOGNITION)
    {
        out_res.net_boxes = NULL;
        out_res.face_id = NULL;

        fmt2rgb888(fb->buf, fb->len, fb->format,
out_res.image);

        out_res.net_boxes = face_detect(image_matrix,
&mtmn_config);

        if (out_res.net_boxes)
        {
            if (align_face(out_res.net_boxes, image_matrix,
aligned_face) == ESP_OK)
            {
                out_res.face_id = get_face_id(aligned_face);
                last_detected_millis = millis();
                if (g_state == START_DETECT) {
                    client.send("Wajah terdeteksi");
                }

                if (g_state == START_ENROLL)
                {
                    int left_sample_face =
do_enrollment(&st_face_list, out_res.face_id);
                    char enrolling_message[64];
                    sprintf(enrolling_message, "Pendaftaran %d Untuk
%s", ENROLL_CONFIRM_TIMES - left_sample_face,
st_name.enroll_name);
                    client.send(enrolling_message);

```



```

        if (left_sample_face == 0)
        {
            ESP_LOGI(TAG, "Enrolled Face ID: %s",
st_face_list.tail->id_name);
            g_state = START_STREAM;
            char captured_message[64];
            sprintf(captured_message, "Wajah %s
Didaftarkan", st_face_list.tail->id_name);
            client.send(captured_message);
            send_face_list(client);
        }
    }

    if (g_state == START_RECOGNITION &&
(st_face_list.count > 0))
    {
        face_id_node *f =
recognize_face_with_name(&st_face_list, out_res.face_id);
        if (f)
        {
            String nama = f->id_name;
            char recognised_message[64];
            sprintf(recognised_message, "Selamat Datang
%s", f->id_name);
            Serial.print("Welcome" + nama);
            sendMessage2Telegram(myToken, myChatId,
"Selamat datang " + nama);
            client.send(recognised_message);
            delay(20000);
        }
        else
        {
            client.send("Wajah tidak dikenali");
            Serial.print("Wajah tidak dikenali");
            sendMessage2Telegram(myToken, myChatId,
"Seseorang Tidak Dikenali");
            sendCapturedImage2Telegram(myToken, myChatId);
        }
    }
    dl_matrix3d_free(out_res.face_id);
}
}
else

```

```

    {
        if (g_state != START_DETECT) {
            client.send("Tidak ada Wajah yang Terdeteksi");
        }
    }

    if (g_state == START_DETECT && millis() -
last_detected_millis > 500) { // Detecting but no face
detected
        client.send("Melakukan Pendeteksian Wajah");
    }

}

client.sendBinary((const char *)fb->buf, fb->len);
esp_camera_fb_return(fb);
fb = NULL;
}

}

// Send captured image to Telegram
String sendCapturedImage2Telegram(String token, String
chat_id) {
    const char* myDomain = "api.Telegram.org";
    String getAll="", getBody = "";

    camera_fb_t * fb = NULL;
    fb = esp_camera_fb_get();
    if(!fb) {
        Serial.println("Camera capture failed");
        delay(1000);
        ESP.restart();
        return "Camera capture failed";
    }

    Serial.println("Connect to " + String(myDomain));

    WiFiClientSecure client_tcp;
    client_tcp.setInsecure(); //run version 1.0.5 or above

    if (client_tcp.connect(myDomain, 443)) {
        Serial.println("Connection successful");
    }
}

```



```

String head = "--Taiwan\r\nContent-Disposition: form-
data; name=\"chat_id\"; \r\n\r\n" + chat_id + "\r\n--
Taiwan\r\nContent-Disposition: form-data; name=\"photo\";
filename=\"esp32-cam.jpg\"\r\nContent-Type:
image/jpeg\r\n\r\n";
String tail = "\r\n--Taiwan--\r\n";

uint16_t imageLen = fb->len;
uint16_t extraLen = head.length() + tail.length();
uint16_t totalLen = imageLen + extraLen;

client_tcp.println("POST /bot"+token+"/sendPhoto
HTTP/1.1");
client_tcp.println("Host: " + String(myDomain));
client_tcp.println("Content-Length: " +
String(totalLen));
client_tcp.println("Content-Type: multipart/form-data;
boundary=Taiwan");
client_tcp.println();
client_tcp.print(head);

uint8_t *fbBuf = fb->buf;
size_t fbLen = fb->len;
for (size_t n=0;n<fbLen;n=n+1024) {
    if (n+1024<fbLen) {
        client_tcp.write(fbBuf, 1024);
        fbBuf += 1024;
    } else if (fbLen%1024>0) {
        size_t remainder = fbLen%1024;
        client_tcp.write(fbBuf, remainder);
    }
}

client_tcp.print(tail);

esp_camera_fb_return(fb);

int waitTime = 10000; // timeout 10 seconds
long startTime = millis();
boolean state = false;

while ((startTime + waitTime) > millis()) {
    Serial.print(".");
    delay(100);
    while (client_tcp.available()) {

```

```

        char c = client_tcp.read();
        if (state==true) getBody += String(c);
        if (c == '\n') {
            if (getAll.length()==0) state=true;
            getAll = "";
        }
        else if (c != '\r')
            getBody += String(c);
            startTime = millis();
        }
        if (getBody.length()>0) break;
    }
    client_tcp.stop();
    Serial.println();
    Serial.println(getBody);
} else {
    getBody="Connected to api.Telegram.org failed.";
    Serial.println("Connected to api.Telegram.org failed.");
}

return getBody;
}

// Send a message to Telegram
String sendMessage2Telegram(String token, String chat_id,
String text) {
    const char* myDomain = "api.Telegram.org";
    String getAll="", getBody = "";

    Serial.println("Connect to " + String(myDomain));

    WiFiClientSecure client_tcp;
    client_tcp.setInsecure(); //run version 1.0.5 or above

    if (client_tcp.connect(myDomain, 443)) {
        Serial.println("Connection successful");

        String message = "chat_id="+chat_id+"&text="+text;
        client_tcp.println("POST /bot"+token+"/sendMessage
HTTP/1.1");
        client_tcp.println("Host: " + String(myDomain));
        client_tcp.println("Content-Length: " +
String(message.length()));
        client_tcp.println("Content-Type: application/x-www-
form-urlencoded");
    }
}

```



```

client_tcp.println();
client_tcp.print(message);

int waitTime = 10000; // timeout 10 seconds
long startTime = millis();
boolean state = false;

while ((startTime + waitTime) > millis()) {
    Serial.print(".");
    delay(100);
    while (client_tcp.available()) {
        char c = client_tcp.read();
        if (state==true) getBody += String(c);
        if (c == '\n') {
            if (getAll.length()==0) state=true;
            getAll = "";
        }
        else if (c != '\r')
            getAll += String(c);
        startTime = millis();
    }
    if (getBody.length()>0) break;
}
client_tcp.stop();
Serial.println();
Serial.println(getBody);
} else {
    getBody="Connected to api.Telegram.org failed.";
    Serial.println("Connected to api.Telegram.org failed.");
}

return getBody;
}

```

Lampiran C. Coding Webserver

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,initial-
scale=1">
<title>SISTEM KEAMANAN RUANGAN</title>
<style>
@media only screen and (min-width: 850px) {
  body {
    display: flex;
  }
  #content-right {
    margin-left: 10px;
  }
}
body {
  font-family: Arial, Helvetica, sans-serif;
  background: #181818;
  color: #EFEFEF;
  font-size: 16px;
}
#content-left {
  max-width: 400px;
  flex: 1;
}
#content-right {
  max-width: 400px;
  flex: 1;
}
#stream {
  width: 100%;
}
#status-display {
  height: 25px;
  border: none;
  padding: 10px;
  font: 18px/22px sans-serif;
  margin-bottom: 10px;
  border-radius: 5px;
  background: #4040F0;
  text-align: center;
}
```



```

#person {
    width:100%;
    height: 25px;
    border: none;
    padding: 20px 10px;
    font: 18px/22px sans-serif;
    margin-bottom: 10px;
    border-radius: 5px;
    resize: none;
    box-sizing: border-box;
}
button {
    display: block;
    margin: 5px 0;
    padding: 0 12px;
    border: 0;
    width: 48%;
    line-height: 28px;
    cursor: pointer;
    color: #fff;
    background: #FE2727;
    border-radius: 5px;
    font-size: 16px;
    outline: 0;
}
.buttons {
    height:40px;
}
button:hover {
    background: #ff494d;
}
button:active {
    background: #f21c21;
}
button:disabled {
    cursor: default;
    background: #a0a0a0;
}
.left {
    float: left;
}
.right {
    float: right;
}
.image-container {

```

```

        position: relative;
    }
    .stream {
        max-width: 400px;
    }
    ul {
        list-style: none;
        padding: 5px;
        margin: 0;
    }
    li {
        padding: 5px 0;
    }
    .delete {
        background: #ff3034;
        border-radius: 100px;
        color: #fff;
        text-align: center;
        line-height: 18px;
        cursor: pointer;
    }
    h3 {
        margin-bottom: 3px;
    }
</style>
</head>
<body>
<div id="content-left">
    <div id="stream-container" class="image-container"> <img
id="stream" src=""> </div>
</div>
<div id="content-right">
    <h1>MONITORING SISTEM KEAMANAN RUANGAN</h1>
    <div id="status-display"> <span id="current-
status"></span> </div>
    <div id="person-name">
        <input id="person" type="text" value=""
placeholder="Masukkan nama disini">
    </div>
    <div class="buttons">
        <button id="button-stream" class="left">MONITOR
KAMERA</button>
        <button id="button-detect" class="right">DETEKSI
WAJAH</button>
    </div>

```



```

<div class="buttons">
  <button id="button-capture" class="left" title="Masukkan
nama sebelum tambah wajah">TAMBAH WAJAH</button>
  <button id="button-recognise" class="right">AKSES
KEAMANAN</button>
</div>
<div class="buttons">
  <button id="button-take" class="left" >AMBIL
FOTO</button>
  <button id="button-led" class="right" >LED
FLASH</button>
</div>
<div class="buttons">
  <button id="button-left" class="left" >KIRI</button>
  <button id="button-right" class="right">KANAN</button>
</div>
<div class="buttons">
  <button id="button-up" class="left">ATAS</button>
  <button id="button-down" class="right">BAWAH</button>
</div>
<div class="people">
  <h3>DAFTAR WAJAH</h3>
  <ul>
  </ul>
</div>
<div class="buttons">
  <button id="delete_all">HAPUS SEMUA</button>
</div>
<script>
document.addEventListener("DOMContentLoaded",
function(event) {
  var baseHost = document.location.origin;
  var streamUrl = baseHost + ":81";
  const WS_URL = "ws://" + window.location.host + ":82";
  const ws = new WebSocket(WS_URL);

  const view = document.getElementById("stream");
  const personFormField = document.getElementById("person");
  const streamButton = document.getElementById("button-
stream");
  const detectButton = document.getElementById("button-
detect");
  const captureButton = document.getElementById("button-
capture");

```



```

const recogniseButton = document.getElementById("button-recognise");
const ledButton = document.getElementById("button-led");
const takeButton = document.getElementById("button-take");
const upButton = document.getElementById("button-up");
const leftButton = document.getElementById("button-left");
const rightButton = document.getElementById("button-right");
const downButton = document.getElementById("button-down");
const deleteAllButton = document.getElementById("delete_all");

```

```

// gain, frequency, duration
a=new AudioContext();
function alertSound(w,x,y){
    v=a.createOscillator();
    u=a.createGain();
    v.connect(u);
    v.frequency.value=x;
    v.type="square";
    u.connect(a.destination);
    u.gain.value=w*0.01;
    v.start(a.currentTime);
    v.stop(a.currentTime+y*0.001);
}

```

```

ws.onopen = () => {
    console.log(`Connected to ${WS_URL}`);
};
ws.onmessage = message => {
    if (typeof message.data === "string") {
        if (message.data.substr(0, 8) == "listface") {
            addFaceToScreen(message.data.substr(9));
        } else if (message.data == "delete_faces") {
            deleteAllFacesFromScreen();
        } else {
            document.getElementById("current-status").innerHTML = message.data;
            document.getElementById("status-display").style.background = "#4040F0";
        }
    }
    if (message.data instanceof Blob) {
        var urlObject = URL.createObjectURL(message.data);
        view.src = urlObject;
    }
}

```



```

    }
  }

  streamButton.onclick = () => {
    ws.send("stream");
  };
  detectButton.onclick = () => {
    ws.send("detect");
  };
  captureButton.onclick = () => {
    person_name = document.getElementById("person").value;
    ws.send("capture:" + person_name);
  };
  recogniseButton.onclick = () => {
    ws.send("recognise");
  };
  ledButton.onclick = () => {
    ws.send("ledflash");
  };
  takeButton.onclick = () => {
    ws.send("takephoto");
  };
  upButton.onclick = () => {
    ws.send("servoup");
  };
  leftButton.onclick = () => {
    ws.send("servoleft");
  };
  rightButton.onclick = () => {
    ws.send("servoright");
  };
  downButton.onclick = () => {
    ws.send("servodown");
  };
  deleteAllButton.onclick = () => {
    ws.send("delete_all");
  };
  personFormField.onkeyup = () => {
    captureButton.disabled = false;
  };

  function deleteAllFacesFromScreen() {
    // deletes face list in browser only
    const faceList = document.querySelector("ul");
    while (faceList.firstChild) {

```

```
        faceList.firstChild.remove();
    }
    personFormField.value = "";
    captureButton.disabled = true;
}

function addFaceToScreen(person_name) {
    const faceList = document.querySelector("ul");
    let listItem = document.createElement("li");
    let closeItem = document.createElement("span");
    closeItem.classList.add("delete");
    closeItem.id = person_name;
    closeItem.addEventListener("click", function() {
        ws.send("remove:" + person_name);
    });
    listItem.appendChild(
        document.createElement("strong")
    ).textContent = person_name;
    listItem.appendChild(closeItem).textContent = "X";
    faceList.appendChild(listItem);
}

captureButton.disabled = true;
});
</script>
</body>
</html>
```


DAFTAR RIWAYAT HIDUP



RICKY HENDRADYANTO PUTRO, lahir di Jombang pada tanggal 24 Oktober 1999. Merupakan anak ketiga dari tiga bersaudara dari pasangan Alm. Untoro Hadi dan Hartini. Bertempat tinggal di Jalan HOS Cokroaminoto 67, Jombatan, Jombang, Jawa Timur. Memulai pendidikan Sekolah Dasar di SD Negeri Jombatan 3 Jombang, lulus tahun 2012. Melanjutkan pendidikan Sekolah Menengah Pertama di SMP Negeri 2 Jombang, lulus tahun 2015. Melanjutkan pendidikan Sekolah Menengah Atas di SMA Negeri Jogoroto Jombang, lulus tahun 2018. Pernah melanjutkan pendidikan di Universitas Airlangga jurusan DIII Akuntansi, lulus tahun 2021. Selanjutnya mencoba mendaftar di Politeknik Penerbangan Surabaya dan diterima sebagai Taruna Politeknik Penerbangan Surabaya program studi DIII Teknik Navigasi Udara pada bulan September 2021. Selama masa Pendidikan di Politeknik Penerbangan Surabaya telah mengikuti *On the Job Training (OJT)* di Perum LPPNPI Cabang Pembantu Batam dan Bandara Internasional Batam (BIB). Aktif sebagai salah satu anggota Dewan Musyawarah Taruna dan Gita Swara Buana XIV.