



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

LAPORAN TUGAS AKHIR



**RANCANG BANGUN SISTEM LOKER PROYEK DOSEN
TELEKOMUNIKASI MENGGUNAKAN MODUL 4G BERBASIS AI DAN
ANDROID**

**“RANCANG BANGUN SISTEM LOKER DOSEN TELEKOMUNIKASI
BERBASIS AI *FACE RECOGNITION*”**

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk memperoleh gelar Diploma Tiga

Rahul Asrialdi

2203332084

**PROGRAM STUDI TEKNIK TELEKOMUNIKASI
JURUSAN TEKNIK ELEKTRO
POLITEKNIK NEGERI JAKARTA**

2025



HALAMAN PERNYATAAN ORISINALITAS

Tugas akhir ini adalah hasil karya saya sendiri dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.

Nama : Rahul Asrialdi

NIM : 2203332084

Tanda Tangan :

Tanggal : 1 Juli 2025



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

LEMBAR PENGESAHAN

TUGAS AKHIR

Tugas Akhir diajukan oleh :

Nama : Rahul Asrialdi

NIM : 2203332084

Program Studi : Telekomunikasi

Judul Tugas Akhir : Rancang Bangun Sistem Loker Proyek
Dosen Telekomunikasi Menggunakan
Modul 4G Berbasis AI Dan Android

Telah diuji oleh tim penguji dalam Sidang Tugas Akhir pada 24 Juli 2025 dan dinyatakan LULUS.

Pembimbing : Toto Supriyanto, S.T., M.T.
NIP. 196603061990031001 (.....)

POLITEKNIK
NEGERI
JAKARTA

Depok, 24 Juli 2025

Disahkan oleh
Ketua Jurusan Teknik Elektro



Dr. Murie Dwiyaniti, S.T., M.T.
NIP. 197803312003122002



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

KATA PENGANTAR

Puji dan syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas rahmat dan karunia-Nya, penulis dapat menyelesaikan tugas akhir. Penulisan tugas akhir ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Diploma Tiga Politeknik.

Tugas akhir ini berjudul “Rancang Bangun Sistem Loker Proyek Dosen Telekomunikasi Menggunakan Modul 4G Berbasis AI Dan Android”. Penulis menyadari bahwa terselesaikannya tugas akhir ini tanpa bimbingan dan bantuan dari beberapa pihak tidak akan selesai pada waktunya.

Oleh karena itu, penulis mengucapkan terimakasih kepada :

1. Toto Supriyanto, S.T., M.T., selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan dalam penyusunan tugas akhir ini.
2. Orang tua dan keluarga penulis yang memberikan doa serta kasih sayang dan dukungan moral maupun material.
3. Muhammad Zatiero Adrafa dan Revalina Serlidyia selaku rekan dalam mengerjakan tugas akhir serta teman-teman dari Program Studi Telekomunikasi 2022 terkhusus kelas B yang telah mendukung serta bekerja sama untuk menyelesaikan tugas akhir.
4. Rani Herlina, yang selalu mendukung dan menghibur penulis dalam mengerjakan tugas akhir.

Akhir kata, penulis berharap Tuhan Yang Maha Esa berkenan membalas kebaikan semua pihak yang telah membantu. Semoga tugas akhir ini membawa manfaat bagi pengembangan ilmu.

Depok, 4 Juli 2025

Rahul Asrialdi



Rancang Bangun Sistem Loker Proyek Dosen Telekomunikasi Menggunakan Modul 4G Berbasis AI Dan Android

ABSTRAK

Pengembangan sistem loker cerdas berbasis AI dilakukan untuk menjawab kebutuhan penyimpanan yang aman dan efisien di lingkungan akademik. Sistem ini memanfaatkan teknologi Face Recognition sebagai metode autentikasi, yang memungkinkan akses otomatis ketika wajah pengguna terdeteksi. Raspberry Pi 5 digunakan sebagai unit pemrosesan utama untuk menangani proses pengenalan wajah serta mengelola komunikasi data secara real-time. Modul 4G SIM7600 M.2 ditambahkan untuk memastikan konektivitas internet yang stabil tanpa bergantung pada jaringan lokal kampus. Integrasi antara AI, Raspberry Pi, dan modul 4G memungkinkan sistem bekerja secara mandiri, akurat, dan dapat diakses jarak jauh melalui aplikasi pendukung, sehingga memberikan solusi penyimpanan yang adaptif dan modern. Sistem AI yang digunakan telah menunjukkan kinerja yang optimal dalam pengenalan wajah, khususnya pada kondisi dengan sudut pandang frontal (0 derajat) dan jarak sekitar 40 cm dari kamera. Pada posisi ini, sistem mampu mengenali wajah dengan sangat akurat dan merespons dengan cepat tanpa keterlambatan. Berdasarkan pengujian performa jaringan menggunakan kartu By.U, diperoleh throughput sebesar 624.689 bps, packet loss 0%, delay 9,9 ms, kecepatan unduh rata-rata 18 Mbps, unggah 8,4 Mbps, dan ping sebesar 40 ms. Hasil ini menunjukkan bahwa kualitas jaringan sangat mendukung pengoperasian sistem face recognition berbasis artificial intelligence secara real-time dan stabil.

Kata kunci: Face Recognition, Raspberry Pi, Artificial Intelligence, Konektivitas 4G, Loker cerdas.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

POLITEKNIK
NEGERI
JAKARTA



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

*Design and Development of a Telecommunication Lecturer Project Locker System
Using a 4G Module Based on AI and Android*

ABSTRACT

The development of a smart locker system based on Artificial Intelligence (AI) was carried out to address the need for secure and efficient storage in academic environments. This system utilizes Face Recognition technology as an authentication method, allowing automatic access when a user's face is detected. Raspberry Pi 5 is used as the main processing unit to handle facial recognition processes and manage real-time data communication. The 4G SIM7600 M.2 module is added to ensure stable internet connectivity without relying on the campus's local network. The integration of AI, Raspberry Pi, and the 4G module enables the system to operate independently, accurately, and be remotely accessed through a supporting application, offering an adaptive and modern storage solution. The AI system has demonstrated optimal performance in face recognition, particularly under conditions with a frontal face angle (0 degrees) and a distance of approximately 40 cm from the camera. At this position, the system can accurately identify faces and respond quickly without delay. Based on network performance testing using a By.U SIM card, the results showed a throughput of 624.689 bps, 0% packet loss, 9.9 ms delay, an average download speed of 18 Mbps, an upload speed of 8.4 Mbps, and a ping of 40 ms. These results indicate that the network quality fully supports the real-time and stable operation of the AI-based face recognition system.

Keywords: *Face Recognition, Raspberry Pi, Artificial Intelligence, 4G Connectivity, Smart Locker.*

**POLITEKNIK
NEGERI
JAKARTA**



DAFTAR ISI

HALAMAN PERNYATAAN ORISINALITAS	iii
LEMBAR PENGESAHAN TUGAS AKHIR.....	iv
KATA PENGANTAR.....	v
ABSTRAK	vi
DAFTAR ISI	viii
DAFTAR TABEL.....	x
DAFTAR GAMBAR	xi
DAFTAR RUMUS.....	xii
DAFTAR LAMPIRAN	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Tujuan	2
1.4 Luaran	2
BAB II TINJAUAN PUSTAKA	3
2.1 Raspberry Pi	3
2.2 Sistem Operasi Raspberry Pi.....	4
2.3 Robotik.....	4
2.4 <i>Computer Vision</i>	5
2.5 <i>Artificial Intelligence</i>	5
2.5.1 <i>Machine learning</i>	6
2.5.2 <i>FaceRecognition</i>	6
2.6 Bahasa Pemrograman Python	7
2.6.1 OpenCV	7
2.6.2 <i>Face Recognition</i>	8
2.6.3 <i>Firebase Admin</i>	8
2.7 <i>Firebase</i>	9
2.7.1 <i>Realtime Database</i>	9
2.7.2 <i>Storage</i>	10
2.8 Loker Penyimpanan	10
2.9 <i>Webcam</i>	11
2.10 <i>Software Visual Studio Code</i>	11
2.11 <i>Quality of Service</i>	12
BAB III RANCANGAN DAN REALISASI	15
3.1 Rancangan	15
3.1.1 Deskripsi	15
3.1.2 Cara Kerja	15
3.1.3 Spesifikasi.....	17
3.1.4 Diagram Blok.....	17
3.2 Realisasi	18
3.2.1 Pembuatan <i>Database</i> pada <i>Firebase</i>	18
3.2.2 Pengkoneksian <i>Artificial Intelligence</i> dengan <i>Firebase</i>	21
3.2.3 Pembuatan <i>Face Recognition</i>	22

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

BAB IV PEMBAHASAN.....	35
4.1 Pengujian <i>Face Recognition</i>	35
4.1.1 Deskripsi Pengujian	35
4.1.2 Alat dan Peralatan	36
4.1.3 <i>Set-Up</i> Sistem	36
4.1.4 Prosedur Pengujian	37
4.1.5 Data Hasil Pengukuran	38
4.1.6 Analisa Data	54
4.2 Pengujian <i>Quality of Service</i>	56
4.2.1 Deskripsi Pengujian	56
4.2.2 Alat dan Peralatan	56
4.2.3 <i>Set-Up</i> Sistem	56
4.2.4 Prosedur Pengujian	57
4.2.5 Data Hasil Pengukuran	57
4.2.6 Analisa Data	59
4.3 Pengujian <i>Speed Test</i>	60
4.3.1 Prosedur Pengujian	60
4.3.2 Alat dan Peralatan	60
4.3.3 <i>Set-Up</i> Sistem	60
4.3.4 Data Hasil Pengukuran	60
4.3.5 Analisa Data	62
BAB V KESIMPULAN.....	63
DAFTAR PUSTAKA.....	64
DAFTAR RIWAYAT HIDUP.....	67
LAMPIRAN.....	66

**POLITEKNIK
NEGERI
JAKARTA**



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR TABEL

Tabel 2.1 Kategori <i>Throughput</i>	12
Tabel 2.2 Kategori <i>Packet Loss</i>	13
Tabel 2.3 Kategori <i>Delay</i>	14
Tabel 3.1 Spesifikasi Perangkat	17
Tabel 4.1 Pengujian wajah terdaftar sesuai	39
Tabel 4.2 Pengujian wajah belum terdaftar.....	42
Tabel 4.3 Pengujian wajah dengan masker	48
Tabel 4.4 Pengujian wajah dengan kacamata.....	51
Tabel 4.5 Pengujian jarak wajah terdaftar.....	52
Tabel 4.6 Hasil QoS	59





Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR GAMBAR

Gambar 2.1	Raspberry Pi Model 5.....	3
Gambar 2.2	Sistem Operasi Raspbian.....	4
Gambar 2.3	<i>Machine Learning</i>	6
Gambar 2.4	<i>Webcam</i>	11
Gambar 3.1	<i>Flowchart</i> cara kerja system pendeteksi wajah.....	16
Gambar 3.2	Diagram Blok alat penyimpanan loker otomatis.....	18
Gambar 3.3	Tampilan awal <i>Firebase</i>	19
Gambar 3.4	Tampilan <i>Real-time Database</i>	20
Gambar 3.5	Tampilan <i>Database Storage</i>	20
Gambar 3.6	Tampilan <i>Authentication</i>	21
Gambar 3.7	Tampilan <i>Face Recognition</i>	34
Gambar 4.1	Loker yang ditaruh pada lantai.....	36
Gambar 4.2	Wajah yang didekatkan pada area deteksi kamera.....	37
Gambar 4.3	Wajah yang terdeteksi dan sesuai.....	38
Gambar 4.4	Data wajah pada <i>Firebase</i>	39
Gambar 4.5	Wajah yang belum terdaftar	41
Gambar 4.6	Wajah belum terdaftar dikenali sebagai terdaftar	43
Gambar 4.7	Data wajah pada <i>Firebase</i> (2)	44
Gambar 4.8	Beberapa wajah tidak terdeteksi dan dikenali	45
Gambar 4.9	Wajah terdaftar menggunakan masker	47
Gambar 4.10	Wajah terdaftar menggunakan kacamata	50
Gambar 4.11	Rangkaian Raspberry Pi dengan Modul 4G LTE.....	57
Gambar 4.12	Hasil <i>Capture Wireshark</i>	58
Gambar 4.13	Pengujian Speed Test pertama.....	61
Gambar 4.14	Pengujian <i>Speed Test</i> kedua	61
Gambar 4.15	Pengujian Speed Test ketiga	62



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR RUMUS

(2.1) <i>Throughput</i>	12
(2.2) <i>Packet Loss</i>	13
(2.3) <i>Delay</i>	13





Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR LAMPIRAN

L-1 Luaran Sistem Utama Alat	68
L-2 <i>Source Code</i>	69





Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

BAB I PENDAHULUAN

1.1 Latar Belakang

Pada era modern, teknologi seperti *internet of things* (IoT) hadir untuk mempermudah pekerjaan manusia dengan sistemnya yang handal dan termutakhir, ditambah dengan adanya teknologi *artificial intelligence* (AI) menjadi kunci dalam menciptakan sistem keamanan yang kompleks dan sulit dibobol. Salah satu penerapannya adalah sistem loker berbasis *face recognition* dan android, yang memungkinkan autentikasi cepat dan akurat untuk melindungi serta menyimpan barang dengan aman. Dosen di bidang telekomunikasi sering menghadapi kendala dalam menyimpan barang pribadi yang penting, seperti dokumen untuk proses pembelajaran dan penilaian mahasiswa. Barang-barang ini rentan hilang jika ditinggalkan di kampus atau merepotkan jika harus dibawa pulang setiap hari. Kendala ini dapat mengganggu efisiensi proses mengajar.

Untuk itulah *internet of things* (IoT) hadir untuk mempermudah pekerjaan manusia dengan sistemnya yang handal dan termutakhir. Selain itu, dengan perkembangan zaman yang pesat ini, Penggunaan IoT dapat diperluas dengan menggunakan *artificial intelligence* (AI). AI mampu bekerja untuk mempelajari dan membuat keputusan sendiri, sehingga IoT mampu menjadi lebih cerdas dan responsif terhadap lingkungannya. Dengan data multimedia yang dikirim, Long-Term Evolution (LTE) dapat dijadikan sebagai media data, karena kecepatan, ketersediaan, dan kemampuannya yang dapat diandalkan dalam menyediakan koneksi internet cepat dan stabil. Berdasarkan permasalahan tersebut, munculah ide untuk membuat sistem loker penyimpanan otomatis.

Dengan memanfaatkan IoT yang dilengkapi AI, membuat alat ini mampu bekerja mandiri dan memaksimalkan keamanan loker dengan fitur AI untuk mendeteksi wajah dan pengenalan wajah. Dengan adanya sistem keamanan pada loker dosen telekomunikasi ini dapat mempermudah dosen dalam menyimpan dokumen serta tetap menjaga barang tetap aman. Berdasarkan permasalahan tersebut, dibuatlah sistem dengan judul "Rancang Bangun Sistem Loker Dosen Telekomunikasi Menggunakan Modul 4G Berbasis AI dan Android." Sistem ini



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

mengintegrasikan keamanan AI dan kemudahan akses melalui aplikasi Android untuk mendukung efisiensi dosen dalam pembelajaran.

1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah diuraikan diatas, terdapat beberapa permasalahan yang akan dibahas dalam tugas akhir ini ialah sebagai berikut:

1. Bagaimana cara merancang sistem keamanan dengan pendeteksi wajah berbasis *Artificial Intelligence* ?
2. Bagaimana cara merealisasikan sistem keamanan dengan pendeteksi wajah berbasis *Artificial Intelligence*?
3. Bagaimana kinerja pendeteksi wajah dengan jaringan internet modul SIM 4G LTE?

1.3 Tujuan dan Kegunaan

Tujuan yang dapat tercapai dari tugas akhir ini ialah sebagai berikut:

1. Mampu merancang sistem keamanan dengan pengenalan wajah pada loker otomatis.
2. Mampu merealisasikan sistem keamanan dengan pengenalan wajah pada loker otomatis.
3. Mampu melakukan pengujian pengenalan wajah serta pengujian kinerja dan kecepatan jaringan internet saat mengirim data pada loker otomatis.

1.4 Luaran

Adapun bentuk luaran dari tugas akhir ini adalah:

1. Alat sistem pendeteksi wajah.
2. Laporan Tugas Akhir.
3. Artikel ilmiah.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

BAB V

KESIMPULAN

Berdasarkan hasil perancangan, realisasi, dan pengujian terhadap sistem loker otomatis berbasis *Face Recognition*, dapat disimpulkan beberapa hal berikut:

1. Sistem keamanan berbasis pengenalan wajah pada loker otomatis berhasil dirancang menggunakan teknologi *artificial intelligence* (AI) yang diimplementasikan pada Raspberry Pi 5 dengan dukungan modul 4G SIM7600 M.2 untuk memastikan konektivitas internet yang stabil. Sistem mampu melakukan inisialisasi penuh dalam waktu rata-rata 12 detik, serta menyelesaikan proses deteksi wajah dan pembukaan loker dalam waktu kurang dari 5 detik setelah wajah dikenali. Rangkaian perangkat keras bekerja stabil selama 24 jam pengujian nonstop tanpa kendala fungsional.
2. Sistem yang direalisasikan mampu mendeteksi dan mengenali wajah pengguna secara real-time, membuka akses secara otomatis hanya untuk wajah yang terdaftar, wajah terdeteksi dan dikenali secara optimal pada sudut nol derajat (0°) dan jarak 40cm, sistem menolak akses untuk wajah yang tidak dikenali. Mekanisme autentikasi ini membuktikan bahwa sistem bekerja efektif, akurat, dan mandiri.
3. Pengujian performa pengenalan wajah dan kinerja jaringan menunjukkan hasil throughput sebesar 624.689 bps, packet loss 0%, delay 9,9 ms, kecepatan unduh rata-rata 18 Mbps, kecepatan unggah 8,4 Mbps, dan ping sebesar 40 ms. Hasil ini membuktikan bahwa kualitas jaringan mendukung pengiriman data dan pengoperasian sistem face recognition secara optimal dan real-time.

Sistem loker berbasis *Face Recognition* ini masih dapat ditingkatkan, terutama dalam hal akurasi deteksi wajah dengan memperkaya data wajah dan menggunakan model AI yang lebih canggih seperti menggunakan pengenalan retina mata. Penambahan fitur keamanan seperti verifikasi dua langkah atau notifikasi ke aplikasi Android juga disarankan untuk mencegah akses tidak sah. Selain itu, penempatan kamera yang tepat dan pencahayaan yang cukup akan membantu meningkatkan hasil deteksi. Ke depan, sistem ini dapat diperluas untuk keperluan lain seperti absensi, pintu otomatis, atau loker umum berbasis autentikasi wajah.



DAFTAR PUSTAKA

- Clinton, R. M., & Sengkey, R. (2019). Purwarupa Sistem Daftar Pelanggaran Lalulintas Berbasis Mini-Komputer Raspberry Pi. *Jurnal Teknik Elektro dan Komputer*, 8(3), 181-192.
- Berliani, D., Saragih, Y., & Latifa, U. (2021). Pemanfaatan 4G LTE dengan mikrokomputer Raspberry pi. *JET (Journal of Electrical Technology)*, 6(1), 1-6
- Budiawan, I., & Andriana, A. (2019). Pengujian Pengenalan Wajah Menggunakan Raspberry Pi. *Jurnal Otomasi Kontrol dan Instrumentasi*, 6(2), 485680.
- Rahman, S., Sembiring, A., Siregar, D., Prahmana, I. G., Puspadini, R., & Zen, M.(2023). *Python: Dasar dan Pemrograman Berorientasi Objek*. Penerbit Tahta Media.
- Upton, E., & Halfacree, G. (2019). *Raspberry Pi user guide*. John Wiley & Sons
- Riza, H. (2022). *The Artificial Intelligence National Strategy: Acceleration Innovation Toward Indonesia 2045*. Diakses dari dcs.binus.ac.id.
- Mulyanto, T. A., Habiby, M., Kusnadi, K., & Adam, R. (2021). Home Automation Sistem Dengan Menggunakan Raspberry Pi 4. *Jurnal Digit: Digital of Information Technology*, 11(1), 60-73.
- Carion, N., Yun, W., & Chang, E. (2020). End-to-end Object Detection with transformers. *Proceedings of the European Conference on Computer Vision*, 214-229.
- Russell, S., & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson.
- Syamjovanka, R. P., Firdausi, P. A. C., Ana, V. R., & Sari, A. P. (2024). Implementasi Sistem Pengenalan Wajah dengan Integrasi Firebase untuk Manajemen Kehadiran Real-Time.
- Raharjo, A. S., Saputra, A., & Irianto, S. Y. (2019, November). Pengembangan Pengolahan Citra Face Recognition, Face Counting dan Age Gender Detection Secara Real Time di Python. In *Prosiding Seminar Nasional Darmajaya* (Vol. 1, pp. 68-77).
- Sofi, N., & Dharmawan, R. (2022). Perancangan aplikasi bengkel CSM berbasis Android menggunakan framework Flutter (bahasa Dart). *Jurnal Teknik dan Science*, 1(2), 53-64.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR RIWAYAT HIDUP



Rahul Asrialdi Lahir di Depok, 15 Oktober 2003. Lulus dari SDN Abadijaya 1 Tahun 2016, SMPN 16 Depok Tahun 2019, dan SMKN 1 Depok Tahun 2022. Gelar Diploma Tiga (D3) diperoleh Tahun 2025 dari Program Studi Telekomunikasi, Jurusan Teknik Elektro, Politeknik Negeri Jakarta.



LAMPIRAN



L-1 Luaran Sistem Utama Alat

	PROGRAM STUDI TELEKOMUNIKASI JURUSAN TEKNIK ELEKTRO POLITEKNIK NEGERI JAKARTA			Digambar :	Rahul Asrialdi
				Diperiksa :	Toto Supriyanto S.T., M.T.
				Tanggal:	15 Juli 2025

- Hak Cipta :**
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
 2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Hak Cipta:

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
import cv2
import os
import pickle
import face_recognition
import numpy as np
import Firebase_admin
from Firebase_admin import credentials, Storage, db
from datetime import datetime
import threading
import time
import hashlib

# ===== KONFIGURASI SISTEM =====
# Path ke file kredensial Firebase Anda
FIREBASE_CREDENTIALS = "serviceAccountKey.json"
# Nama bucket penyimpanan Firebase Anda
STORAGE_BUCKET = "tararero-1a7c6.FirebaseStorage.app"
# URL database Firebase Realtime Database Anda
DATABASE_URL = "https://tararero-1a7c6-default-rtdb.firebaseio.com/"

# Folder lokal untuk menyimpan gambar wajah yang diunduh
FOLDER_WAJAH_LOKAL = "local_faces"
# Folder lokal untuk menyimpan hasil encoding wajah
FOLDER_ENCODING = "face_encodings"
# Folder di Firebase Storage tempat gambar wajah disimpan
FOLDER_STORAGE = "faces"
# Interval (dalam detik) untuk memeriksa pembaruan di Firebase
INTERVAL_PEMERIKSAAN = 30 # detik

# Variabel global
encoding_terdaftar = [] # Daftar encoding wajah yang sudah terdaftar
id_terdaftar = [] # Daftar ID pengguna yang sesuai dengan encoding
kamera_aktif = False # Status kamera (aktif/tidak)
last_update_time = 0 # Waktu terakhir pembaruan riwayat ke database
UPDATE_INTERVAL = 5 # Interval (dalam detik) untuk mencegah spam database

# Buat folder penyimpanan jika belum ada
os.makedirs(FOLDER_WAJAH_LOKAL, exist_ok=True)
os.makedirs(FOLDER_ENCODING, exist_ok=True)

def calculate_file_hash(filepath):
    """
    Menghitung hash MD5 dari sebuah file.
    Digunakan untuk membandingkan integritas file.
    """
    hash_md5 = hashlib.md5()
    with open(filepath, "rb") as f:
```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta:

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
# Variabel global
encoding_terdaftar = [] # Daftar encoding wajah yang sudah
terdaftar
id_terdaftar = [] # Daftar ID pengguna yang sesuai
dengan encoding
kamera_aktif = False # Status kamera (aktif/tidak)
last_update_time = 0 # Waktu terakhir pembaruan riwayat ke
database
UPDATE_INTERVAL = 5 # Interval (dalam detik) untuk
mencegah spam database

# Buat folder penyimpanan jika belum ada
os.makedirs(FOLDER_WAJAH_LOKAL, exist_ok=True)
os.makedirs(FOLDER_ENCODING, exist_ok=True)

def calculate_file_hash(filepath):
    """
    Menghitung hash MD5 dari sebuah file.
    Digunakan untuk membandingkan integritas file.
    """
    hash_md5 = hashlib.md5()
    with open(filepath, "rb") as f:
        # Baca file dalam potongan kecil untuk efisiensi memori
        for chunk in iter(lambda: f.read(4096), b''):
            hash_md5.update(chunk)
    return hash_md5.hexdigest()

def inisialisasi_Firebase():
    """
    Menginisialisasi koneksi ke Firebase.
    Memeriksa apakah Firebase sudah diinisialisasi untuk
    menghindari inisialisasi ganda.
    """
    try:
        if not Firebase_admin._apps:
            cred =
credentials.Certificate(FIREBASE_CREDENTIALS)
            Firebase_admin.initialize_app(cred, {
                'databaseURL': DATABASE_URL,
                'StorageBucket': STORAGE_BUCKET
            })
            print("[INFO] Firebase berhasil terhubung.")
            return True
    except Exception as e:
        print(f"[ERROR] Gagal terhubung ke Firebase: {str(e)}")
        return False

def buat_encoding(path_gambar, path_encoding, user_id):
    """
    Membuat encoding wajah dari sebuah gambar dengan validasi.
    Memastikan hanya ada satu wajah dalam gambar dan menyimpan
    encoding-nya.
    """
```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta:

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
"""
try:
    gambar = cv2.imread(path_gambar)
    # Periksa apakah gambar valid (tidak kosong)
    if gambar is None:
        print(f"[WARNING] Gambar tidak valid:
{path_gambar}")
        os.remove(path_gambar) # Hapus gambar yang tidak
valid

        return False

    rgb_gambar = cv2.cvtColor(gambar, cv2.COLOR_BGR2RGB)
    face_locations =
face_recognition.face_locations(rgb_gambar)

    # Pastikan hanya ada tepat satu wajah yang terdeteksi
    if len(face_locations) != 1:
        print(f"[WARNING] Harus tepat 1 wajah dalam gambar:
{path_gambar}")
        os.remove(path_gambar) # Hapus gambar yang tidak
memenuhi kriteria
        return False

    # SELALU buat encoding baru untuk gambar baru/perbaruan
    # Ambil encoding dari wajah yang terdeteksi
    encoding = face_recognition.face_encodings(rgb_gambar,
face_locations)[0]
    # Simpan encoding ke file menggunakan pickle
    with open(path_encoding, 'wb') as f:
        pickle.dump({
            'id': user_id,
            'encoding': encoding.tolist() # Simpan sebagai
list agar bisa di-pickle
        }, f)

    print(f"[SUCCESS] Encoding berhasil dibuat/diperbarui
untuk user {user_id}")
    return True
except Exception as e:
    print(f"[ERROR] Gagal memproses {path_gambar}:
{str(e)}")
    # Hapus gambar jika ada kesalahan dalam pemrosesan
    if os.path.exists(path_gambar):
        os.remove(path_gambar)
    return False

def muat_encoding():
    """
    Memuat semua encoding wajah yang tersimpan secara lokal
    dengan validasi.
    Membersihkan encoding yang tidak memiliki gambar referensi.
    """
    global encoding_terdaftar, id_terdaftar
```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta:

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
# Pastikan semua encoding memiliki gambar referensi yang
valid
for enc_file in os.listdir(FOLDER_ENCODING):
    if enc_file.endswith('.enc'):
        user_id = enc_file.split('.')[0]
        # Cari file gambar yang sesuai dengan user_id
        img_files = [f for f in
os.listdir(FOLDER_WAJAH_LOKAL) if f.startswith(f"{user_id}.")]

        # Jika tidak ada gambar yang ditemukan untuk
encoding ini, hapus encodingnya
        if not img_files: # Periksa apakah ada gambar untuk
user_id ini
            print(f"[CLEANUP] Encoding tanpa gambar untuk
{user_id}, menghapus...")
            os.remove(os.path.join(FOLDER_ENCODING,
enc_file))
            continue

        try:
            # Muat data encoding dari file
            with open(os.path.join(FOLDER_ENCODING,
enc_file), 'rb') as f:
                data = pickle.load(f)
                encoding_wajah.append(np.array(data['encodi
ng'])) # Konversi kembali ke numpy array
                daftar_id.append(data['id'])
            except Exception as e:
                print(f"[ERROR] Gagal memuat encoding
{enc_file}: {str(e)}")
                continue

encoding_terdaftar = encoding_wajah
id_terdaftar = daftar_id
print(f"[INFO] {len(encoding_terdaftar)} encoding wajah
valid berhasil dimuat.")

def unduh_wajah():
    """
    Mengunduh gambar wajah dari Firestore Storage dengan
pemeriksaan yang ditingkatkan.
    Membandingkan hash file untuk menghindari pengunduhan ulang
file yang sama.
    """
    try:
        bucket = Storage.bucket()
        # Dapatkan daftar semua blob (file) di folder 'faces'
        blobs =
list(bucket.list_blobs(prefix=f"{FOLDER_STORAGE}/"))
        file_terunduh = []

        for blob in blobs:
```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta:

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
# Pastikan hanya memproses file JPG
if nama_file.endswith('.jpg'):
    user_id = nama_file.split('.')[0]
    path_lokal = os.path.join(FOLDER_WAJAH_LOKAL,
nama_file)
    path_encoding = os.path.join(FOLDER_ENCODING,
f"{user_id}.enc")

    # Selalu download gambar baru terlebih dahulu
    ke file temporer
    temp_path = os.path.join(FOLDER_WAJAH_LOKAL,
f"temp_{nama_file}")
    blob.download_to_filename(temp_path)

    # Cek apakah file sudah ada secara lokal
    if os.path.exists(path_lokal):
        # Jika sudah ada, bandingkan hash-nya
        existing_hash =
calculate_file_hash(path_lokal)
        new_hash = calculate_file_hash(temp_path)

        if existing_hash != new_hash:
            print(f"[UPDATE] Gambar berbeda untuk
user {user_id}, memperbarui...")
            os.remove(path_lokal) # Hapus file lama
            # Hapus encoding lama jika gambar
berubah untuk memastikan encoding baru dibuat
            if os.path.exists(path_encoding):
                os.remove(path_encoding)
            os.rename(temp_path, path_lokal) #
Ganti dengan file baru
            file_terunduh.append((path_lokal,
path_encoding, user_id))
        else:
            print(f"[SKIP] Gambar sama untuk user
{user_id}")
            os.remove(temp_path) # Hapus file
temporer karena sama
            # TETAP memproses encoding jika belum
ada, meskipun gambarnya sama
            if not os.path.exists(path_encoding):
                file_terunduh.append((path_lokal,
path_encoding, user_id))
            else:
                # Jika gambar baru (belum ada di lokal),
langsung proses
                os.rename(temp_path, path_lokal) #
Pindahkan dari temp ke lokasi akhir
                file_terunduh.append((path_lokal,
path_encoding, user_id))
                print(f"[NEW] Gambar baru untuk user
{user_id}")
```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta:

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
        return file_terunduh
    except Exception as e:
        print(f"[ERROR] Saat mengunduh dari Storage: {str(e)}")
        return []

def catat_riwayat(user_id, is_recognized):
    """
    Mencatat riwayat pengenalan ke Firebase Realtime Database.
    Mencegah penulisan berlebihan dengan interval
    UPDATE_INTERVAL.
    """
    global last_update_time

    current_time = time.time()
    # Periksa apakah sudah waktunya untuk memperbarui riwayat
    if current_time - last_update_time < UPDATE_INTERVAL:
        return # Jika belum, keluar dari fungsi

    last_update_time = current_time # Perbarui waktu terakhir
    update

    try:
        timestamp = datetime.now().strftime("%Y-%m-%d
%H:%M:%S")

        riwayat_data = {
            "status": "YES" if is_recognized else "NO", #
            Status pengenalan
            "waktu": timestamp #
            Waktu kejadian
        }

        # Dapatkan referensi ke lokasi database untuk user_id
        ini
        ref = db.reference(f"status_pengenalan/{user_id}")
        ref.set(riwayat_data) # Set data ke database (akan
        menerima jika sudah ada)

        print(f"[HISTORY] Riwayat diperbarui untuk user
        {user_id}: {riwayat_data}")
    except Exception as e:
        print(f"[ERROR] Gagal mencatat riwayat: {str(e)}")

class FirebaseMonitor(threading.Thread):
    """
    Thread terpisah untuk memantau pembaruan gambar wajah di
    Firebase Storage.
    Secara berkala mengunduh gambar baru/yang diperbarui dan
    membuat ulang encoding.
    """
    def __init__(self):
```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta:

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
self.daemon = True # Set sebagai daemon agar thread
berhenti saat program utama berhenti

def run(self):
    """Loop utama pemantauan dengan pemeriksaan yang
    ditingkatkan."""
    while self.running:
        try:
            file_terunduh = unduh_wajah() # Unduh gambar
            dari Firebase
            if file_terunduh:
                print(f"[PROCESS] Memproses
                {len(file_terunduh)} wajah baru/diperbarui")

                for path_gambar, path_encoding, user_id in
                file_terunduh:
                    # Buat encoding hanya jika gambar berhasil
                    diproses
                    if buat_encoding(path_gambar,
                    path_encoding, user_id):
                        muat_encoding() # Muat ulang semua
                        encoding setelah ada perubahan
                    except Exception as e:
                        print(f"[MONITOR ERROR] {str(e)}")

                time.sleep(INTERVAL_PEMERIKSAAN) # Jeda sebelum
                pemeriksaan berikutnya

        def stop(self):
            """Menghentikan thread pemantauan."""
            self.running = False

        def inisialisasi_kamera():
            """
            Menginisialisasi kamera komputer dengan pendekatan multi-
            backend
            untuk meningkatkan kompatibilitas di berbagai sistem
            operasi.
            """
            global kamera_aktif

            # Daftar backend OpenCV yang akan dicoba secara berurutan
            backends = [
                cv2.CAP_V4L2, # Untuk Linux (misalnya Raspberry Pi)
                cv2.CAP_DSHOW, # Untuk Windows (DirectShow)
                cv2.CAP_MSMF, # Untuk Windows (Microsoft Media
                Foundation)
                cv2.CAP_ANY # Auto-detect (generik)
            ]

            for backend in backends:
                try:
```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta:

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
kamera = cv2.VideoCapture(0, backend) # 0 biasanya
adalah kamera default
if kamera.isOpened(): # Periksa apakah kamera
berhasil dibuka
    ret, _ = kamera.read() # Coba baca satu frame
    untuk memastikan kamera berfungsi
    if ret:
        # Atur resolusi kamera
        kamera.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
        kamera.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)
        kamera_aktif = True # Set status kamera
menjadi aktif
        print(f"[INFO] Kamera berhasil
diinisialisasi dengan backend: {backend}")
        return kamera # Kembalikan objek kamera
        kamera.release() # Lepaskan kamera jika tidak
bisa membaca frame
    except Exception as e:
        print(f"[WARNING] Kesalahan dengan backend
{backend}: {str(e)}")
        continue # Lanjutkan ke backend berikutnya

print("[ERROR] Semua backend kamera gagal menginisialisasi
kamera.")
return None # Kembalikan None jika semua backend gagal

def proses_pengenalan(kamera):
    """
    Fungsi utama untuk melakukan proses pengenalan wajah secara
    real-time dari kamera.
    """
    global kamera_aktif
    while kamera_aktif: # Loop selama kamera aktif
        try:
            ret, frame = kamera.read() # Baca frame dari kamera
            if not ret:
                print("[WARNING] Gagal membaca frame dari
kamera")
                time.sleep(0.1)
                continue

            # Ubah ukuran frame untuk mempercepat deteksi wajah
            small_frame = cv2.resize(frame, (0, 0), fx=0.25,
fy=0.25)

            rgb_frame = cv2.cvtColor(small_frame,
cv2.COLOR_BGR2RGB) # Konversi ke RGB (face_recognition butuh
RGB)

            face_locations =
face_recognition.face_locations(rgb_frame) # Deteksi lokasi
wajah
```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta:

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
(top, right, bottom, left) = face_locations[0]
# Buat encoding untuk wajah yang terdeteksi
face_encoding =
face_recognition.face_encodings(rgb_frame,
[face_locations[0]]) [0]

if encoding_terdaftar: # Pastikan ada encoding
yang terdaftar
    # Bandingkan wajah yang terdeteksi dengan
encoding yang terdaftar
    matches =
face_recognition.compare_faces(encoding_terdaftar,
face_encoding)
    face_distances =
face_recognition.face_distance(encoding_terdaftar,
face_encoding)
    best_match = np.argmin(face_distances) #
Dapatkan kecocokan terbaik (jarak terdekat)
    is_recognized = matches[best_match] #
Tentukan apakah wajah dikenali
    user_id = id_terdaftar[best_match] if
is_recognized else "unknown" # Dapatkan ID jika dikenali

    # Skala kembali koordinat kotak wajah ke
ukuran frame asli
    top *= 4; right *= 4; bottom *= 4; left *=
4
    # Tentukan warna kotak (hijau jika
dikenali, merah jika tidak)
    color = (0, 255, 0) if is_recognized else
(0, 0, 255)
    cv2.rectangle(frame, (left, top), (right,
bottom), color, 2) # Gambar kotak di wajah
    label = user_id if is_recognized else
"Tidak Dikenal"
    cv2.putText(frame, label, (left + 6, bottom
- 6),
cv2.FONT_HERSHEY_SIMPLEX, 0.7,
(255, 255, 255), 1) # Tambahkan label

    catat_riwayat(user_id, is_recognized) #
Catat riwayat pengenalan

# Jika terdeteksi lebih dari satu wajah
elif len(face_locations) > 1:
    print("[WARNING] Lebih dari satu wajah
terdeteksi. Melewatkan frame.")

cv2.imshow('Face Recognition', frame) # Tampilkan
frame di jendela
# Tekan 'q' untuk keluar
if cv2.waitKey(1) & 0xFF == ord('q'):
```