

**PERANCANGAN SISTEM LOKER HELM PINTAR
BERBASIS INTERNET OF THINGS (IOT) DENGAN
OTENTIKASI APLIKASI MOBILE DAN PREDIKSI
KETERSEDIAAN MENGGUNAKAN MACHINE
LEARNING**

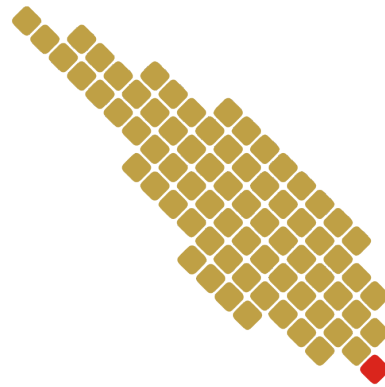
TUGAS AKHIR

Diajukan sebagai syarat menyelesaikan jenjang strata Satu
(S-1) di Program Studi Teknik Informatika, Fakultas
Teknologi Industri, Institut Teknologi Sumatera

Oleh:

-

-



ITERA

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SUMATERA
LAMPUNG SELATAN**

2025

DAFTAR ISI

DAFTAR ISI	ii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian	3
1.4 Batasan Masalah	3
1.5 Kontribusi Penelitian	3
BAB II TINJAUAN PUSTAKA	5
2.1 Penelitian Terkait	5
2.1.1 Sistem <i>Smart Locker</i> dan Akses Keamanan	5
2.1.2 Integrasi IoT di Lingkungan Kampus	6
2.1.3 Penerapan <i>Machine Learning</i> pada Fasilitas Publik	7
2.1.4 Posisi Penelitian (<i>Research Gap</i>)	8
2.2 Landasan Teori	9
2.2.1 <i>Internet of Things</i> (IoT) dan ESP32	9
2.2.2 <i>Solenoid Door Lock</i>	10
2.2.3 <i>Machine Learning: Support Vector Regression</i> (SVR)	11
2.2.4 <i>System Usability Scale</i> (SUS)	13
2.3 Kerangka Pemikiran	14
BAB III METODE PENELITIAN	16
3.1 Alur Penelitian	16
3.2 Alat dan Bahan	18
3.2.1 Perangkat Keras (<i>Hardware</i>)	18
3.2.2 Perangkat Lunak (<i>Software</i>)	18
3.3 Metode Pengumpulan Data	19
3.4 Perancangan Sistem	20

3.4.1	Arsitektur IoT	20
3.4.1.1	A. Layer Perangkat Keras (<i>Hardware Layer</i>)	21
3.4.1.2	B. Layer Komunikasi (<i>Communication Layer</i>).....	21
3.4.1.3	C. Layer Backend (<i>Server Layer</i>)	23
3.4.2	Aplikasi Mobile	23
3.4.3	Model <i>Machine Learning</i>	25
3.4.3.1	A. <i>Feature Engineering</i>	25
3.4.3.2	B. <i>Training</i> dan Validasi	26
3.4.3.3	C. <i>Deployment</i>	27
3.5	Rencana Pengujian.....	28
3.5.1	Pengujian Fungsional	28
3.5.2	Pengujian Kinerja (Performance)	28
3.5.2.1	A. Response Time (Latency)	28
3.5.2.2	B. System Reliability	29
3.5.2.3	C. Concurrent User Handling.....	29
3.5.3	Evaluasi Akurasi Machine Learning	29
3.5.3.1	A. Mean Absolute Error (MAE)	29
3.5.3.2	B. Root Mean Squared Error (RMSE)	29
3.5.3.3	C. Baseline Comparison	29
3.5.3.4	D. Interpretasi Hasil	29
3.5.4	Pengujian Usability	30
3.5.4.1	A. Responden.....	30
3.5.4.2	B. Prosedur Pengujian	30
3.5.4.3	C. Perhitungan Skor SUS	30
3.5.4.4	D. Kriteria Keberhasilan	30
3.5.5	Dokumentasi Hasil Pengujian	30
DAFTAR PUSTAKA		32
LAMPIRAN		34
A	Worksheet Proses Review Proposal Awal.....	34

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi, khususnya *Internet of Things* (IoT), telah membawa transformasi signifikan dalam menciptakan lingkungan kampus yang lebih cerdas (*Smart Campus*). Salah satu aspek penting dalam lingkungan kampus adalah penyediaan fasilitas yang aman dan nyaman bagi civitas akademika. Sejalan dengan upaya peningkatan ketertiban dan keselamatan, beberapa perguruan tinggi menerapkan kebijakan wajib penggunaan helm bagi pengendara sepeda motor di area kampus. Kebijakan ini bertujuan positif untuk menanamkan budaya keselamatan berkendara.

Namun, implementasi kebijakan wajib helm seringkali belum diimbangi dengan penyediaan fasilitas penyimpanan helm yang memadai. Mahasiswa sering kali harus meninggalkan helm di sepeda motor atau tempat yang tidak terjamin keamanannya, sehingga menimbulkan kerawanan terhadap pencurian. Permasalahan ini menunjukkan adanya kebutuhan akan solusi penyimpanan helm yang aman, mudah diakses, dan efisien di lingkungan kampus.

Penelitian terkait sistem penyimpanan pintar (*smart locker*) berbasis IoT telah banyak dilakukan untuk berbagai kebutuhan, seperti penitipan barang umum, pengiriman paket [1], hingga fasilitas loker di lingkungan kampus [2], [3]. Beberapa sistem memanfaatkan teknologi otentikasi seperti RFID [4], [5], OTP, QR Code [6], [7], biometrik, maupun kombinasi multi-otentikasi [8]. Integrasi dengan aplikasi mobile juga umum diterapkan untuk meningkatkan kemudahan akses dan monitoring [2], [4], [8], [9]. Penelitian oleh Suciningtyas dkk. secara spesifik telah mengembangkan kabinet penyimpanan helm berbasis IoT dengan akses menggunakan QR Code melalui aplikasi [6]. Sementara itu, Wu dkk. merancang sistem loker pintar IoT khusus untuk kampus dengan fitur berbagi akses antar pengguna [2]. Penelitian lain seperti Chandrappa dkk. juga menekankan pentingnya

evaluasi kuantitatif dalam pengembangan smart locker [9].

Meskipun teknologi *smart locker* terus berkembang, terdapat celah penelitian (*research gap*) yang signifikan. Belum banyak penelitian yang secara spesifik menangani masalah penyimpanan helm mahasiswa di lingkungan kampus dengan mengintegrasikan tidak hanya IoT dan aplikasi mobile untuk keamanan dan kemudahan akses, tetapi juga memanfaatkan *Machine Learning* (ML) untuk memberikan fitur prediktif [2], [6], [7], [8], [9]. Fitur prediksi, seperti prediksi tingkat ketersediaan loker berdasarkan data historis, dapat memberikan nilai tambah yang signifikan bagi pengguna dalam merencanakan penggunaan fasilitas. Penggunaan ML untuk prediksi dalam konteks parkir pintar telah ditunjukkan dalam penelitian Anitha dkk. [10], namun belum diterapkan pada kasus loker helm kampus.

Ketiadaan fasilitas penitipan helm yang aman dan terstruktur tidak hanya menimbulkan kerugian finansial dan rasa cemas bagi mahasiswa akibat risiko kehilangan helm, tetapi juga berpotensi mengurangi efektivitas penerapan kebijakan wajib helm itu sendiri. Oleh karena itu, pengembangan sebuah sistem solusi yang komprehensif menjadi krusial untuk mendukung kebijakan kampus dan meningkatkan kenyamanan serta keamanan bagi mahasiswa.

Berdasarkan latar belakang dan celah penelitian yang telah diidentifikasi, penelitian ini mengusulkan perancangan dan implementasi sebuah prototipe sistem loker helm pintar. Sistem ini akan memanfaatkan teknologi IoT untuk kontrol perangkat keras, aplikasi mobile untuk otentikasi pengguna yang aman dan mudah, serta *Machine Learning* untuk menyediakan fitur prediksi ketersediaan loker, sehingga menawarkan solusi yang lebih cerdas dan efisien dibandingkan sistem yang ada saat ini.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka permasalahan penelitian dirumuskan sebagai berikut:

1. Bagaimana perancangan dan implementasi sistem penitipan helm pintar (*smart helmet locker*) berbasis *Internet of Things* (IoT) dapat menjadi solusi yang aman, efisien, dan mudah diakses bagi mahasiswa di lingkungan

kampus?

2. Seberapa efektif prototipe sistem yang diusulkan setelah dibangun dalam hal keamanan (otentikasi), kecepatan proses penitipan/pengambilan, dan kemudahan penggunaan?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah di atas, maka tujuan dari penelitian ini adalah:

1. Merancang arsitektur dan membangun prototipe fungsional sistem loker helm pintar berbasis IoT, aplikasi mobile, dan fitur prediksi *Machine Learning*, dimulai dari analisis kebutuhan pengguna hingga integrasi sistem.
2. Mengevaluasi tingkat efektivitas prototipe berdasarkan metrik keamanan otentikasi, kecepatan transaksi, kemudahan penggunaan (melalui kuesioner *usability*), dan akurasi model prediksi pada sejumlah responden mahasiswa.

1.4 Batasan Masalah

Agar penelitian ini tetap fokus dan sesuai dengan sumber daya yang tersedia, ditetapkan batasan masalah sebagai berikut:

1. Prototipe fisik yang dibangun hanya mencakup satu unit loker dengan beberapa pintu (misal: 2-4 pintu) sebagai demonstrasi konsep.
2. Sistem otentikasi dan kontrol akses loker sepenuhnya menggunakan aplikasi mobile (Android) melalui mekanisme login dan tombol buka loker terotentikasi via internet.
3. Model *Machine Learning* yang dikembangkan difokuskan pada prediksi tingkat ketersediaan loker berdasarkan data historis waktu penggunaan.
4. Penelitian tidak membahas aspek komersialisasi, sistem pembayaran, atau integrasi dengan sistem informasi akademik (SIKAD) kampus.
5. Pengujian prototipe tidak mencakup uji ketahanan fisik terhadap vandalisme atau kondisi cuaca ekstrem.

1.5 Kontribusi Penelitian

Penelitian ini diharapkan dapat memberikan kontribusi sebagai berikut:

1. Menghasilkan sebuah artefak berupa prototipe sistem loker helm pintar yang fungsional, mengintegrasikan IoT, aplikasi mobile, dan fitur prediksi ML, yang dirancang spesifik untuk menjawab masalah nyata di lingkungan kampus.
2. Menyediakan analisis kebutuhan pengguna (mahasiswa) terkait fasilitas penitipan helm pintar yang dapat menjadi acuan bagi pihak universitas dalam pengembangan fasilitas serupa skala besar.
3. Memberikan bukti empiris (melalui evaluasi kuantitatif) mengenai efektivitas penerapan teknologi IoT dan ML untuk meningkatkan keamanan, efisiensi, dan kemudahan penggunaan fasilitas penyimpanan helm di kampus.

BAB II

TINJAUAN PUSTAKA

Bab ini menguraikan tinjauan terhadap penelitian-penelitian terdahulu untuk mengidentifikasi celah penelitian (*research gap*), serta menjelaskan landasan teori yang menjadi dasar pengembangan sistem. Tinjauan pustaka disusun secara sistematis untuk membangun argumentasi yang kuat mengenai urgensi dan kebaruan penelitian ini.

2.1 Penelitian Terkait

Penelitian mengenai sistem penyimpanan pintar (*smart locker*) dan penerapan IoT di lingkungan kampus telah banyak dilakukan dalam beberapa tahun terakhir. Bagian ini memetakan posisi penelitian ini terhadap studi-studi sebelumnya dengan mengidentifikasi kekuatan dan keterbatasan dari solusi yang telah ada.

2.1.1 Sistem *Smart Locker* dan Akses Keamanan

Aspek keamanan dan kemudahan akses merupakan fokus utama dalam pengembangan sistem *smart locker*. Beberapa penelitian telah mengeksplorasi berbagai mekanisme otentikasi dengan pendekatan yang berbeda-beda.

Pawar dkk. [1] mengembangkan sistem loker pintar untuk pengiriman paket yang menggunakan kombinasi *One-Time Password* (OTP) dan keypad fisik sebagai mekanisme otentikasi. Sistem ini memungkinkan fleksibilitas waktu pengambilan barang karena OTP dapat dikirimkan kepada penerima meskipun mereka tidak hadir saat pengiriman. Namun, kelemahan signifikan dari pendekatan ini adalah keharusan interaksi fisik yang relatif lama di depan loker untuk memasukkan kode OTP melalui keypad, yang dapat menimbulkan antrean saat jam sibuk dan mengurangi efisiensi, terutama dalam konteks lingkungan kampus dengan mobilitas tinggi.

Permana dkk. [7] mengimplementasikan sistem loker dengan autentikasi QR

Code dinamis yang terintegrasi dengan aplikasi web. QR Code dihasilkan secara unik untuk setiap transaksi dan memiliki masa berlaku terbatas untuk meningkatkan keamanan. Meskipun lebih cepat dibandingkan input manual OTP, sistem ini masih memerlukan pemindaian fisik menggunakan kamera atau scanner yang terpasang di loker, sehingga pengguna harus berada tepat di depan perangkat scanner dan dalam kondisi pencahayaan yang memadai. Hal ini dapat menjadi kendala jika terjadi masalah teknis pada kamera atau dalam kondisi cahaya buruk.

Suciningtyas dkk. [6] secara spesifik telah mengembangkan kabinet penyimpanan helm berbasis IoT dengan akses menggunakan QR Code melalui aplikasi mobile, bahkan dilengkapi dengan panel surya untuk efisiensi energi. Penelitian ini sangat relevan dengan konteks penyimpanan helm di kampus. Namun, keterbatasan utama terletak pada mekanisme akses yang masih bergantung pada pemindaian QR Code di lokasi fisik loker, yang tidak berbeda jauh dari pendekatan Permana dkk. Selain itu, penelitian ini tidak mengeksplorasi aspek prediksi ketersediaan yang dapat membantu pengguna merencanakan waktu penggunaan fasilitas.

Penelitian ini mengusulkan pendekatan yang berbeda dengan menggunakan otentikasi satu tombol via aplikasi mobile tanpa memerlukan pemindaian atau input kode di lokasi fisik. Pengguna cukup membuka aplikasi dan menekan tombol "Buka Kunci" yang telah terotentikasi melalui login, sehingga proses pembukaan loker dapat dilakukan lebih cepat (target < 3 detik) dan mengurangi waktu antrean di depan loker, terutama saat jam sibuk perkuliahan.

2.1.2 Integrasi IoT di Lingkungan Kampus

Penerapan teknologi IoT untuk meningkatkan fasilitas kampus (*Smart Campus*) telah menjadi tren penelitian yang berkembang pesat. Wu dkk. [2] merancang sistem loker pintar IoT khusus untuk kampus dengan fitur unik berupa *sharing access* antar pengguna melalui platform WeChat. Sistem ini memungkinkan mahasiswa untuk berbagi akses loker dengan teman, misalnya untuk menyimpan dokumen atau barang bersama. Penelitian ini menunjukkan relevansi konteks kampus di mana kolaborasi dan fleksibilitas penggunaan menjadi penting.

Alqahtani dkk. [3] mengembangkan sistem loker otomatis untuk perguruan tinggi dengan menggunakan teknologi Bluetooth untuk komunikasi antara aplikasi mobile dan perangkat loker. Pendekatan Bluetooth memiliki keunggulan dalam hal konsumsi daya yang rendah dan tidak bergantung pada koneksi internet. Namun, keterbatasan jangkauan Bluetooth (umumnya ; 10 meter) mengharuskan pengguna berada sangat dekat dengan loker untuk dapat berkomunikasi, yang dapat menjadi kendala jika pengguna ingin memeriksa ketersediaan loker dari jarak jauh.

Meskipun penelitian-penelitian ini relevan secara konteks (lingkungan kampus) dan teknologi (IoT), mereka belum menyentuh aspek prediksi ketersediaan fasilitas. Dalam lingkungan kampus dengan jumlah mahasiswa yang jauh lebih banyak dibanding fasilitas loker yang tersedia, informasi prediktif mengenai kapan loker cenderung penuh atau kosong menjadi sangat penting. Tanpa fitur prediksi, mahasiswa harus datang ke lokasi loker terlebih dahulu untuk mengetahui ketersediaan, yang tidak efisien dalam hal waktu dan tenaga.

2.1.3 Penerapan *Machine Learning* pada Fasilitas Publik

Penerapan *Machine Learning* untuk prediksi ketersediaan fasilitas publik telah menunjukkan hasil yang menjanjikan di berbagai domain, meskipun masih terbatas dalam konteks *smart locker*.

Anitha dkk. [10] membuktikan efektivitas algoritma *Support Vector Regression* (SVR) untuk memprediksi harga parkir dinamis dan ketersediaan slot parkir berdasarkan data historis penggunaan. Penelitian ini menunjukkan bahwa SVR mampu menangani pola data *time-series* yang non-linear dengan baik, seperti fluktuasi penggunaan fasilitas yang berbeda-beda sepanjang hari (jam sibuk vs jam sepi). Hasil evaluasi menunjukkan bahwa model SVR mencapai akurasi prediksi yang tinggi (MAE ; 0.8 untuk prediksi jumlah slot tersedia) dan dapat memberikan informasi yang berguna bagi pengguna untuk merencanakan waktu parkir mereka.

Chandrapa dkk. [9] dalam penelitian "*Smart Locker 2.0*" menyebutkan penggunaan *Machine Learning* untuk meningkatkan efisiensi sistem loker publik dan disebutkan bahwa ML dapat digunakan untuk "optimasi alokasi loker" dan "prediksi pola penggunaan". Namun, penelitian ini tidak merinci implementasi teknis

dari komponen ML tersebut, seperti algoritma yang digunakan, fitur input yang dipilih, metrik evaluasi, atau bagaimana hasil prediksi disajikan kepada pengguna. Ketidakjelasan ini menunjukkan bahwa aspek prediktif ML dalam konteks *smart locker* masih merupakan area yang belum matang dan memerlukan eksplorasi lebih lanjut.

Hal ini menunjukkan bahwa meskipun potensi ML untuk prediksi ketersediaan fasilitas fisik telah terbukti di domain lain (seperti parkir), integrasi ML untuk prediksi ketersediaan loker — khususnya loker helm di kampus — adalah area yang potensial namun belum banyak dieksplorasi secara komprehensif dengan implementasi dan evaluasi yang rinci.

2.1.4 Posisi Penelitian (*Research Gap*)

Berdasarkan tinjauan sistematis terhadap penelitian-penelitian terdahulu, dapat diidentifikasi beberapa celah penelitian yang signifikan:

1. Gap Mekanisme Akses: Penelitian sebelumnya masih bergantung pada interaksi fisik yang memerlukan waktu (OTP keypad, pemindaian QR Code) yang dapat menimbulkan antrean. Belum ada solusi yang mengoptimalkan kecepatan akses melalui otentikasi aplikasi satu tombol tanpa interaksi fisik tambahan.
2. Gap Fitur Prediktif: Tidak ada penelitian yang mengintegrasikan fitur prediksi ketersediaan berbasis ML dalam sistem loker helm kampus, padahal fitur ini dapat memberikan nilai tambah signifikan untuk perencanaan pengguna.
3. Gap Konteks Spesifik: Penelitian khusus loker helm di kampus (seperti Suciningtyas dkk.) belum mengeksplorasi kombinasi lengkap: otentikasi cepat via aplikasi + prediksi ML + evaluasi komprehensif (keamanan, kinerja, usability, akurasi).

Tabel 2.1 merangkum perbandingan penelitian ini dengan penelitian terdahulu berdasarkan kriteria kunci:

Dapat disimpulkan bahwa belum ada penelitian yang secara komprehensif mengintegrasikan sistem loker helm khusus untuk kampus dengan otentikasi aplikasi mobile yang cepat dan fitur prediksi ketersediaan berbasis *Machine Learning*.

Tabel 2.1 Perbandingan dengan Penelitian Terdahulu

Peneliti	Fokus	Teknologi Akses	Fitur ML	Keterbatasan Utama
Suciningtyas dkk. [6]	Loker Helm	QR Code, Solar Panel	Tidak Ada	Perlu scan fisik di alat, tidak ada prediksi
Wu dkk. [2]	Loker Kampus	WeChat, Sharing	Tidak Ada	Tidak ada prediksi ketersediaan
Pawar dkk. [1]	Loker Paket	OTP, Keypad	Tidak Ada	Interaksi fisik lambat, tidak untuk helm
Chandrappa dkk. [9]	Loker Publik	Multi-autentikasi	Disebutkan	Implementasi ML tidak detail
Penelitian Ini	Loker Helm Kampus	App Mobile (1-tap)	Prediksi SVR	-

Penelitian ini dirancang untuk mengisi celah tersebut dengan memberikan solusi yang lebih adaptif, informatif, dan efisien bagi mahasiswa.

2.2 Landasan Teori

Bagian ini menjelaskan teori-teori fundamental dan teknologi yang menjadi dasar perancangan dan implementasi sistem loker helm pintar dalam penelitian ini.

2.2.1 *Internet of Things (IoT)* dan ESP32

Internet of Things (IoT) adalah paradigma komputasi yang memungkinkan objek fisik untuk saling berkomunikasi dan bertukar data melalui internet tanpa memerlukan interaksi manusia secara langsung [7]. Dalam konteks sistem loker helm pintar, IoT berperan sebagai *backbone* yang menghubungkan perangkat keras (loker fisik) dengan perangkat lunak (aplikasi mobile dan server) untuk menciptakan ekosistem yang terintegrasi.

ESP32 Development Board dipilih sebagai mikrokontroler utama dalam penelitian ini karena beberapa keunggulan teknis:

- Konektivitas Wi-Fi dan Bluetooth Terintegrasi: ESP32 memiliki modul Wi-Fi 802.11 b/g/n dan Bluetooth 4.2 yang sudah built-in, sehingga tidak memerlukan modul eksternal tambahan. Ini menyederhanakan desain

hardware dan mengurangi biaya.

- **Dual-Core Processor:** ESP32 menggunakan prosesor Xtensa dual-core 32-bit dengan clock hingga 240 MHz, yang cukup powerful untuk menangani multiple task secara bersamaan (membaca sensor, komunikasi HTTP, kontrol aktuator).
- **GPIO yang Memadai:** ESP32 memiliki 34 pin GPIO yang dapat dikonfigurasi untuk berbagai keperluan (digital input/output, ADC, PWM, I2C, SPI), sehingga fleksibel untuk integrasi dengan berbagai sensor dan aktuator.
- **Konsumsi Daya Rendah:** ESP32 dirancang dengan fitur *deep sleep mode* yang memungkinkan konsumsi daya sangat rendah ($< 5 \mu\text{A}$) saat idle, penting untuk aplikasi IoT yang mungkin beroperasi dengan baterai atau solar panel.
- **Ekosistem Pengembangan yang Matang:** ESP32 mendukung pemrograman melalui Arduino IDE, ESP-IDF, dan MicroPython, dengan dokumentasi lengkap dan komunitas developer yang besar.
- **Harga Terjangkau:** Dengan harga sekitar Rp 40.000 - Rp 80.000 per unit, ESP32 sangat cost-effective untuk prototyping dan bahkan produksi skala kecil.

Kombinasi fitur-fitur ini membuat ESP32 sangat cocok untuk aplikasi IoT yang membutuhkan koneksi internet terus-menerus ke server, seperti sistem loker pintar yang harus responsif terhadap perintah dari aplikasi mobile secara real-time.

2.2.2 Solenoid Door Lock

Solenoid door lock adalah aktuator kunci elektronik yang bekerja berdasarkan prinsip elektromagnetik untuk mengontrol mekanisme penguncian secara otomatis [1]. Pemahaman mengenai cara kerja solenoid penting untuk merancang sistem kontrol yang reliable.

Struktur dan Prinsip Kerja:

Komponen solenoid lock terdiri dari beberapa bagian utama:

- **Kumparan Kawat (Coil):** Kawat tembaga yang dililitkan membentuk kumparan elektromagnet.
- **Plunger (Batang Besi):** Batang logam ferromagnetik yang dapat bergerak

masuk-keluar dari kumparan.

- Pegas (Spring): Memberikan gaya balik untuk mengembalikan plunger ke posisi awal.
- Housing: Casing yang melindungi komponen internal.

Ketika arus listrik DC (biasanya 12V, 500-800 mA) dialirkan melalui kumparan, medan magnet yang dihasilkan akan menarik plunger ke dalam kumparan (prinsip elektromagnetik). Gerakan ini menyebabkan bolt atau latch yang terhubung dengan plunger tertarik, sehingga kunci terbuka dan pintu dapat dibuka. Saat arus listrik diputus, medan magnet menghilang dan pegas akan mendorong plunger kembali ke posisi awal, sehingga kunci tertutup kembali.

Keunggulan untuk Sistem Loker:

- Kontrol Digital: Dapat dikontrol langsung oleh mikrokontroler melalui relay, memungkinkan integrasi dengan sistem IoT.
- Response Time Cepat: Solenoid dapat mengunci/membuka dalam hitungan milidetik (≤ 100 ms).
- Fail-Safe Options: Tersedia dalam mode normally-closed (terkunci saat tidak ada daya) atau normally-open (terbuka saat tidak ada daya), dapat disesuaikan dengan kebutuhan keamanan.
- Durabilitas: Solenoid lock dapat menahan ratusan ribu cycle operasi, cocok untuk penggunaan berulang dalam loker publik.

Dalam penelitian ini, solenoid lock 12V dengan mode normally-closed dipilih untuk memastikan loker tetap terkunci meskipun terjadi gangguan listrik, memberikan keamanan fisik yang dapat dikontrol secara digital oleh sistem IoT.

2.2.3 Machine Learning: Support Vector Regression (SVR)

Untuk fitur prediksi ketersediaan loker, penelitian ini menggunakan metode *Support Vector Regression* (SVR), yang merupakan adaptasi dari algoritma *Support Vector Machine* (SVM) untuk kasus regresi (prediksi nilai kontinu).

Konsep Dasar SVR:

Berbeda dengan regresi linear biasa yang berusaha meminimalkan error secara keseluruhan, SVR berusaha menyesuaikan garis regresi (atau hyperplane dalam

dimensi tinggi) yang berada dalam batas toleransi error ϵ (epsilon). SVR hanya memperhitungkan data points yang berada di luar batas toleransi ini (disebut *support vectors*), sehingga lebih robust terhadap outlier.

Fungsi keputusan SVR untuk prediksi dapat dinyatakan sebagai:

$$y = \sum_{i=1}^N (\alpha_i - \alpha_i^*) K(x_i, x) + b \quad (\text{Rumus 2.1})$$

dimana:

- y = nilai prediksi (jumlah loker terisi)
- N = jumlah support vectors
- α_i, α_i^* = koefisien Lagrange multiplier yang dihitung selama training
- $K(x_i, x)$ = fungsi kernel
- b = bias term
- x_i = support vectors
- x = input data baru yang akan diprediksi

Fungsi Kernel:

Kunci kekuatan SVR terletak pada penggunaan *kernel trick* yang memungkinkan algoritma menangani pola data yang non-linear tanpa harus melakukan transformasi eksplisit ke dimensi tinggi. Fungsi kernel $K(x_i, x)$ menghitung similarity antara dua data point. Dalam penelitian ini, digunakan RBF (Radial Basis Function) kernel:

$$K(x_i, x) = \exp(-\gamma \|x_i - x\|^2) \quad (\text{Rumus 2.2})$$

dimana γ adalah parameter yang mengontrol seberapa jauh pengaruh setiap

training example. RBF kernel sangat cocok untuk data penggunaan loker karena mampu menangani fluktuasi yang tidak rata sepanjang hari (misalnya, lonjakan penggunaan pada jam-jam tertentu seperti sebelum kelas dimulai).

Relevansi untuk Prediksi Ketersediaan:

Pola penggunaan loker helm di kampus bersifat non-linear dan bergantung pada banyak faktor (hari, jam, cuaca, jadwal kuliah). SVR dengan RBF kernel telah terbukti efektif untuk kasus serupa, seperti prediksi ketersediaan parkir [10], karena kemampuannya:

- Menangani pola yang kompleks dan non-linear
- Robust terhadap outlier (misalnya, hari libur mendadak atau event khusus)
- Generalisasi yang baik pada data yang belum pernah dilihat

2.2.4 System Usability Scale (SUS)

System Usability Scale (SUS) adalah kuesioner standar yang dikembangkan oleh John Brooke pada tahun 1986 untuk mengukur persepsi kemudahan penggunaan (usability) suatu sistem dari perspektif pengguna **brooke1996sus**. SUS telah menjadi salah satu instrumen evaluasi usability yang paling banyak digunakan dalam penelitian HCI (*Human-Computer Interaction*) dan pengembangan produk karena kesederhanaan dan reliabilitasnya yang tinggi.

Struktur Kuesioner:

SUS terdiri dari 10 pertanyaan dengan skala Likert 1-5, dimana pertanyaan ganjil (1,3,5,7,9) bernada positif dan pertanyaan genap (2,4,6,8,10) bernada negatif. Desain ini bertujuan untuk mengurangi bias responden yang cenderung memilih jawaban yang sama tanpa membaca pertanyaan dengan teliti.

Perhitungan Skor:

Skor akhir SUS dihitung dengan rumus:

$$\text{Skor akhir SUS} = \left[\sum (R_{\text{ganjil}} - 1) + \sum (5 - R_{\text{genap}}) \right] \times 2.5 \quad (\text{Rumus 2.3})$$

dimana R adalah rating yang diberikan responden (1-5).

Skor akhir berkisar antara 0-100, meskipun bukan persentase. Skor di atas 68 secara umum dikategorikan sebagai usability yang baik dan dapat diterima (*acceptable*), berdasarkan studi benchmark oleh Bangor dkk. yang menganalisis lebih dari 2.300 survei SUS **bangor2008empirical**.

Keunggulan SUS:

- Cepat dan Efisien: Hanya 10 pertanyaan, dapat diselesaikan dalam 2-3 menit.
- Reliabilitas Tinggi: Cronbach's alpha ≥ 0.9 , menunjukkan konsistensi internal yang sangat baik.
- Valid dan Terstandarisasi: Telah divalidasi melalui ribuan penelitian selama lebih dari 35 tahun.
- Technology-Agnostic: Dapat digunakan untuk mengevaluasi berbagai jenis sistem (aplikasi mobile, website, hardware, dll).
- Memungkinkan Komparasi: Skor SUS dapat dibandingkan dengan benchmark industri atau sistem lain.

Dalam penelitian ini, SUS dipilih sebagai instrumen evaluasi usability karena

kesesuaiannya untuk mengevaluasi aplikasi mobile baru yang dikembangkan, kemudahan administrasi, dan kemampuan untuk memberikan metrik kuantitatif yang dapat dibandingkan dengan standar industri.

2.3 Kerangka Pemikiran

Kerangka pemikiran penelitian ini menggambarkan alur logis bagaimana masalah keamanan dan efisiensi penyimpanan helm diselesaikan melalui integrasi tiga blok teknologi utama: Node IoT (ESP32 + sensor + solenoid lock), Aplikasi Mobile (antarmuka pengguna), dan Model Machine Learning (prediksi ketersediaan). Ketiga blok ini saling berkomunikasi melalui *Backend Server* yang bertindak sebagai orchestrator sistem.

Gambar 2.1 Kerangka Pemikiran Penelitian

Gambar 2.1 menunjukkan hubungan antar komponen:

1. Input Layer (Masalah):

- Kebijakan wajib helm di kampus tidak didukung fasilitas penyimpanan memadai
- Risiko pencurian helm
- Ketidakpastian ketersediaan loker

2. Processing Layer (Solusi Teknologi):

- Node IoT: ESP32 membaca sensor IR untuk deteksi keberadaan helm, mengontrol solenoid lock berdasarkan perintah dari server, mengirim status real-time ke database.
- Backend Server: Menangani autentikasi pengguna (JWT), menyimpan log transaksi, menjalankan model ML untuk prediksi, menjembatani komunikasi app-ESP32.
- Aplikasi Mobile: Menyediakan interface untuk login, melihat status loker, mengakses prediksi, dan mengirim perintah unlock yang terautentikasi.
- Model ML (SVR): Dilatih dengan data historis penggunaan, memprediksi tingkat ketersediaan loker pada waktu tertentu.

3. Output Layer (Evaluasi):

Sistem dievaluasi berdasarkan empat metrik utama yang menjawab rumusan masalah:

- Keamanan: Autentikasi JWT + kontrol akses berbasis user
- Kecepatan: Response time ; 3 detik
- Usability: Skor SUS ≥ 68
- Akurasi Prediksi: MAE ; 1 loker, RMSE ; 1.5 loker

Kerangka pemikiran ini menegaskan bahwa penelitian ini tidak hanya mengembangkan sistem loker helm pintar secara parsial, tetapi mengintegrasikan seluruh komponen (IoT, aplikasi, ML) menjadi solusi holistik yang memberikan nilai tambah signifikan dibanding sistem yang ada saat ini: lebih aman (otentikasi digital), lebih cepat (tanpa interaksi fisik lama), lebih informatif (fitur prediksi), dan mudah digunakan (aplikasi intuitif).

BAB III

METODE PENELITIAN

Bab ini menjelaskan tahapan sistematis penelitian, mulai dari alur penelitian, alat dan bahan, metode pengumpulan data, perancangan sistem, hingga skenario pengujian dan metrik evaluasi yang digunakan untuk memastikan sistem yang dibangun memenuhi kriteria keamanan, efisiensi, kemudahan penggunaan, dan akurasi prediksi.

3.1 Alur Penelitian

Penelitian ini menggunakan metode *Prototyping*, yang memungkinkan pengembangan sistem secara iteratif berdasarkan evaluasi dari setiap tahap pengembangan. Pendekatan ini dipilih karena sesuai dengan karakteristik penelitian yang memerlukan validasi dan penyempurnaan desain berdasarkan umpan balik pengguna dan hasil pengujian. Tahapan penelitian digambarkan pada Gambar 3.1 dan dijelaskan sebagai berikut:

Gambar 3.1 Diagram Alir Tahapan Penelitian

1. Studi Literatur dan Analisis Kebutuhan

Mengkaji penelitian terdahulu mengenai sistem *smart locker* berbasis IoT, mekanisme otentikasi, dan algoritma prediksi ketersediaan fasilitas. Selain itu, dilakukan survei kebutuhan kepada mahasiswa untuk mengidentifikasi fitur-fitur yang dibutuhkan dan ekspektasi pengguna terhadap sistem penitipan helm pintar.

2. Perancangan Sistem

Merancang arsitektur sistem secara komprehensif yang meliputi: (a) arsitektur IoT (perangkat keras dan skema komunikasi ESP32 dengan server), (b) desain basis data untuk menyimpan informasi pengguna dan log penggunaan, (c) antarmuka aplikasi mobile yang intuitif, dan (d) arsitektur model prediksi

Machine Learning.

3. Implementasi Prototipe

Membangun prototipe fisik loker dengan mengintegrasikan komponen ESP32, solenoid lock, sensor inframerah, dan relay module. Secara paralel, mengembangkan aplikasi mobile berbasis Android dan *backend server* untuk menangani autentikasi pengguna serta kontrol akses loker secara real-time.

4. Pengumpulan Data untuk *Machine Learning*

Mengoperasikan prototipe selama periode minimal 4 minggu untuk mengumpulkan data log penggunaan (meliputi timestamp check-in/check-out, ID loker, durasi penggunaan, dan pola waktu) yang akan digunakan sebagai data training untuk model prediksi ketersediaan.

5. Pengembangan Model Prediksi

Melatih model *Machine Learning* menggunakan algoritma *Support Vector Regression* (SVR) dengan data historis yang telah dikumpulkan. Tahap ini mencakup *feature engineering*, pemilihan hyperparameter optimal melalui *Grid Search Cross-Validation*, dan evaluasi performa model menggunakan metrik MAE dan RMSE.

6. Integrasi dan Pengujian Sistem

Mengintegrasikan model prediksi ke dalam aplikasi mobile sehingga pengguna dapat melihat estimasi ketersediaan loker secara real-time. Selanjutnya, melakukan serangkaian pengujian meliputi: pengujian fungsional (*black box testing*), pengujian kinerja (*response time* dan *reliability*), evaluasi akurasi model ML, dan pengujian *usability* menggunakan kuesioner SUS.

7. Analisis Hasil dan Penyusunan Laporan

Menganalisis hasil pengujian dengan membandingkannya terhadap target metrik yang telah ditetapkan. Hasil analisis ini kemudian disintesis untuk menjawab rumusan masalah penelitian dan menyusun dokumentasi penelitian secara lengkap.

3.2 Alat dan Bahan

3.2.1 Perangkat Keras (*Hardware*)

Komponen perangkat keras yang digunakan dalam pembangunan prototipe sistem loker helm pintar adalah sebagai berikut:

- ESP32 Development Board: Mikrokontroler utama dengan modul Wi-Fi dan Bluetooth terintegrasi yang berfungsi sebagai otak sistem IoT untuk membaca sensor dan mengontrol aktuator.
- Sensor Inframerah (IR): Sensor untuk mendeteksi keberadaan helm di dalam loker. Setiap pintu loker dilengkapi dengan satu unit sensor IR.
- Solenoid Door Lock 12V: Aktuator pengunci pintu elektronik yang bekerja berdasarkan prinsip elektromagnetik. Ketika diberi tegangan, plunger akan tertarik masuk sehingga kunci terbuka.
- Relay Module 4-Channel: Saklar elektronik untuk mengontrol tegangan tinggi (12V) dari ESP32 yang beroperasi pada tegangan 3.3V. Setiap channel mengontrol satu solenoid lock.
- Power Supply 12V 5A: Sumber daya listrik untuk menyuplai seluruh sistem, khususnya solenoid lock yang memerlukan arus cukup besar saat aktivasi.
- Komponen Pendukung: Kabel jumper, PCB prototype board, casing loker dari kayu atau akrilik, dan komponen elektronik lainnya (resistor, LED indikator).

3.2.2 Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan dalam penelitian ini meliputi:

- Arduino IDE v2.0 atau lebih baru: Lingkungan pengembangan terintegrasi untuk pemrograman firmware ESP32 menggunakan bahasa C/C++.
- Visual Studio Code v1.80+: Editor kode sumber yang digunakan untuk pengembangan aplikasi *backend server* dan keperluan pemrograman umum.
- Flutter SDK v3.10+ / Android Studio Electric Eel: Framework dan IDE untuk pengembangan aplikasi mobile lintas platform (fokus pada Android dalam penelitian ini).
- Python 3.9+ dengan library Scikit-learn v1.2+: Bahasa pemrograman dan library untuk pengembangan, training, dan evaluasi model *Machine Learning*.

- Firebase Realtime Database / MySQL v8.0: Sistem manajemen basis data untuk menyimpan data pengguna (akun mahasiswa), log penggunaan loker, dan status real-time ketersediaan loker.
- Postman v10+: Tool untuk melakukan testing dan debugging REST API yang digunakan untuk komunikasi antara aplikasi mobile, server, dan ESP32.
- Node.js v18+ / Python Flask v2.3+: Platform runtime atau framework untuk membangun *backend server* yang menangani autentikasi, API endpoints, dan komunikasi dengan database.

3.3 Metode Pengumpulan Data

Penelitian ini menggunakan empat metode pengumpulan data untuk mendukung berbagai tahapan penelitian:

1. Studi Literatur

Mengkaji paper ilmiah, artikel jurnal, dan prosiding konferensi terkait teknologi *smart locker* berbasis IoT, mekanisme otentikasi keamanan, serta algoritma *Machine Learning* untuk prediksi ketersediaan fasilitas. Literatur yang dikaji mencakup penelitian terbaru (2020-2025) dari database seperti IEEE Xplore, ACM Digital Library, dan Google Scholar.

2. Survei Kebutuhan Pengguna

Menyebarkan kuesioner kepada minimal 30 mahasiswa pengguna sepeda motor di kampus untuk menganalisis kebutuhan fungsional, preferensi fitur, dan ekspektasi terhadap sistem loker helm pintar. Kuesioner mencakup pertanyaan mengenai:

- Pengalaman masalah penyimpanan helm saat ini
- Fitur yang diinginkan dalam sistem loker pintar
- Tingkat kesediaan menggunakan sistem otomatis
- Preferensi metode otentikasi (aplikasi mobile, RFID, QR Code, dll.)

3. Data Eksperimen untuk *Machine Learning* (Data Primer)

Data log penggunaan loker dikumpulkan secara otomatis oleh sistem IoT selama periode minimal 4 minggu (28 hari) pengoperasian prototipe di lokasi strategis kampus. Data yang dicatat oleh sistem meliputi:

- ID Loker: Identifikasi unik untuk setiap unit loker (1-4)
- User ID: Identifikasi pengguna yang telah dianonim untuk privasi
- Timestamp Check-in: Waktu tepat saat mahasiswa menitipkan helm (format: YYYY-MM-DD HH:MM:SS)
- Timestamp Check-out: Waktu tepat saat mahasiswa mengambil helm
- Durasi Penggunaan: Lama waktu helm disimpan dalam loker (dalam menit)
- Hari dalam Seminggu: Senin (1) hingga Minggu (7)
- Jam: Waktu dalam format 24 jam (0-23)
- Status Ketersediaan Loker: Jumlah loker yang terisi pada setiap timestamp (0-4)

Data ini disimpan secara otomatis dalam database MySQL/Firebase setiap kali terjadi transaksi (check-in atau check-out). Untuk keperluan training model ML, data akan diekspor dalam format CSV dan diproses menggunakan Python Pandas untuk pembersihan data (*data cleaning*) dan transformasi fitur.

Target jumlah data: minimal 200 transaksi selama periode 4 minggu untuk memastikan dataset yang cukup representatif mencakup berbagai pola penggunaan (hari kerja vs akhir pekan, jam sibuk vs sepi).

4. Data Pengujian *Usability* (Data Primer)

Kuesioner *System Usability Scale* (SUS) akan diisi oleh minimal 10-15 responden mahasiswa setelah mereka menggunakan prototipe sistem dalam skenario penggunaan nyata. Responden diminta untuk menilai kemudahan penggunaan aplikasi mobile dan sistem secara keseluruhan berdasarkan 10 pertanyaan standar SUS dengan skala Likert 1-5.

3.4 Perancangan Sistem

3.4.1 Arsitektur IoT

Arsitektur sistem loker helm pintar dirancang menggunakan pendekatan berlapis (*layered architecture*) yang terdiri dari tiga layer utama:

3.4.1.1 A. Layer Perangkat Keras (*Hardware Layer*)

Layer ini merupakan fondasi fisik sistem yang terdiri dari:

- ESP32 Mikrokontroler: Bertindak sebagai *edge device* yang dilengkapi dengan modul Wi-Fi 802.11 b/g/n untuk koneksi internet. ESP32 menjalankan firmware yang ditulis dalam Arduino C/C++ untuk membaca sensor, mengontrol aktuator, dan berkomunikasi dengan server.
- Sensor Inframerah (IR): Terpasang di bagian dalam setiap pintu loker untuk mendeteksi keberadaan objek (helm). Sensor IR akan mengirimkan sinyal digital HIGH (1) ke ESP32 jika helm terdeteksi, dan LOW (0) jika loker kosong.
- Solenoid Door Lock 12V: Pengunci pintu elektronik yang dikontrol melalui Relay Module 4-Channel. Setiap solenoid lock terhubung ke satu channel relay yang diaktifkan oleh pin GPIO ESP32 (misal: GPIO 16, 17, 18, 19).
- Relay Module 4-Channel: Berfungsi sebagai saklar elektronik yang mengisolasi tegangan tinggi (12V) dari ESP32 yang beroperasi pada level logika 3.3V. Relay diaktifkan dengan sinyal LOW dari ESP32 untuk membuka kunci (solenoid aktif) dan dinonaktifkan dengan sinyal HIGH untuk menutup kunci.
- Power Supply 12V 5A: Menyuplai daya untuk solenoid lock (memerlukan arus 500-800 mA per solenoid saat aktivasi). ESP32 dapat disuplai dari regulator 3.3V internal atau menggunakan USB power bank untuk fleksibilitas.

3.4.1.2 B. Layer Komunikasi (*Communication Layer*)

ESP32 berkomunikasi dengan *backend server* melalui protokol HTTP/HTTPS menggunakan REST API. Skema komunikasi dirancang sebagai berikut:

- Request dari ESP32 ke Server (Status Update):

ESP32 mengirimkan status sensor secara periodik (setiap 5 detik) atau ketika terjadi perubahan status (dari kosong menjadi terisi atau sebaliknya). Data dikirim dalam format JSON melalui HTTP POST request:

POST /api/locker/status

Content-Type: application/json


```
{
  "device_id": "ESP32_001",
  "locker_id": 1,
  "status": "occupied",
  "timestamp": "2025-12-01 14:30:15"
}
```

- Response dari Server ke ESP32 (Control Command):

Server mengirimkan perintah pembukaan kunci (unlock) jika ada request dari aplikasi mobile yang telah terautentikasi. ESP32 secara kontinu melakukan *polling* atau menggunakan mekanisme *long polling*/WebSocket untuk menerima perintah:

Response 200 OK

Content-Type: application/json

```
{
  "command": "unlock",
  "locker_id": 1,
  "duration": 5
}
```

ESP32 akan mengaktifkan relay yang sesuai selama durasi tertentu (default: 5 detik) untuk membuka kunci, kemudian otomatis menutup kembali.

- Keamanan Komunikasi:

Untuk memastikan keamanan, setiap request dari ESP32 menyertakan API key atau token autentikasi di header. Dalam implementasi produksi, komunikasi menggunakan HTTPS dengan sertifikat SSL/TLS untuk enkripsi data.

3.4.1.3 C. Layer Backend (*Server Layer*)

Backend server dibangun menggunakan Node.js (dengan framework Express.js) atau Python Flask, yang bertugas:

- Autentikasi Pengguna: Mengelola registrasi dan login mahasiswa menggunakan email institusi (.ac.id) dengan password yang di-hash menggunakan algoritma bcrypt atau Argon2.
- Manajemen Sesi: Menggunakan JSON Web Token (JWT) untuk mengelola sesi pengguna setelah login berhasil. Token disimpan di sisi aplikasi mobile dan dikirimkan pada setiap request yang memerlukan autentikasi.
- API Endpoints: Menyediakan REST API untuk operasi CRUD (Create, Read, Update, Delete) terhadap data pengguna, status loker, dan log transaksi.
- Database Management: Berkomunikasi dengan database MySQL/Firebase untuk menyimpan dan mengambil data secara real-time.
- Integration dengan Model ML: Meload model prediksi (file .pkl) dan melakukan inference ketika aplikasi mobile meminta prediksi ketersediaan loker.
- Message Broker: Menjembatani komunikasi antara aplikasi mobile dan ESP32, memastikan perintah unlock diteruskan ke device yang tepat.

Diagram arsitektur lengkap dapat dilihat pada Gambar 3.2.

Gambar 3.2 Arsitektur Sistem IoT Loker Helm Pintar

3.4.2 Aplikasi Mobile

Aplikasi mobile dikembangkan menggunakan framework Flutter untuk platform Android, dengan desain antarmuka yang mengutamakan kesederhanaan dan kemudahan penggunaan. Fitur-fitur utama aplikasi meliputi:

1. Halaman Registrasi dan Login

Pengguna mendaftar menggunakan email institusi, nomor induk mahasiswa (NIM), dan password. Sistem melakukan validasi email domain (.ac.id) untuk memastikan hanya civitas akademika yang dapat menggunakan layanan. Setelah login, JWT token disimpan secara lokal menggunakan

SharedPreferences untuk autentikasi request berikutnya.

2. Dashboard Utama

Menampilkan status ketersediaan loker secara real-time (berapa loker yang tersedia dari total 4 loker). Status diperbarui secara otomatis setiap beberapa detik dengan melakukan API call ke server. Dashboard juga menampilkan riwayat penggunaan terakhir pengguna.

3. Fitur Prediksi Ketersediaan

Pengguna dapat melihat prediksi tingkat ketersediaan loker pada waktu-waktu tertentu dalam bentuk grafik atau tabel. Misalnya, aplikasi dapat menampilkan: "Pada pukul 13:00 nanti, diperkirakan 2 dari 4 loker akan tersedia." Fitur ini membantu mahasiswa merencanakan waktu penitipan helm agar tidak datang saat loker penuh.

4. Tombol "Pesan Loker"

Jika ada loker tersedia, pengguna dapat menekan tombol untuk memesan loker kosong. Sistem akan mengalokasikan loker tertentu (misal: Loker 3) kepada pengguna dan menampilkan nomor loker beserta tombol "Buka Kunci".

5. Tombol "Buka Kunci"

Setelah pengguna tiba di depan loker fisik, mereka menekan tombol "Buka Kunci" di aplikasi. Aplikasi mengirim API request yang telah diautentikasi dengan JWT token ke server, dan server meneruskan perintah unlock ke ESP32 yang sesuai. Pengguna dapat memasukkan helm dan menutup pintu loker secara manual. Sistem otomatis mendeteksi helm masuk melalui sensor IR dan mengunci kembali solenoid.

6. Tombol "Ambil Helm"

Ketika pengguna ingin mengambil helm, mereka menekan tombol "Ambil Helm" di aplikasi. Proses serupa terjadi: unlock command dikirim, loker terbuka, pengguna mengambil helm, dan sistem mendeteksi loker kosong melalui sensor IR. Transaksi dicatat dengan timestamp check-out dan durasi penggunaan dihitung otomatis.

7. Notifikasi Push

Aplikasi dapat mengirimkan notifikasi push menggunakan Firebase Cloud

Messaging (FCM) untuk mengingatkan pengguna jika helm sudah terlalu lama disimpan (misal: > 6 jam) atau jika terjadi anomali (helm diambil tanpa request dari aplikasi pengguna).

Desain antarmuka mengikuti prinsip *Material Design* dengan palet warna yang konsisten dan navigasi intuitif. Mockup dan wireframe aplikasi dapat dilihat pada Lampiran.

3.4.3 Model *Machine Learning*

Model prediksi ketersediaan loker dikembangkan menggunakan algoritma Support Vector Regression (SVR) dengan kernel RBF (Radial Basis Function), yang telah terbukti efektif untuk memodelkan data time-series dengan pola non-linear seperti fluktuasi penggunaan fasilitas sepanjang hari [10].

3.4.3.1 A. *Feature Engineering*

Dari data log mentah yang dikumpulkan, dilakukan ekstraksi dan transformasi fitur untuk dijadikan input model. Fitur-fitur yang digunakan adalah:

- Hari dalam Seminggu (day_of_week): Dikodekan sebagai variabel numerik 0-6, di mana 0 = Senin, 6 = Minggu. Pola penggunaan loker cenderung berbeda antara hari kerja dan akhir pekan.
- Jam (hour): Waktu dalam format 24 jam (0-23). Ini adalah fitur paling penting karena penggunaan loker sangat bergantung pada jadwal kuliah (jam sibuk: 07:00-09:00 dan 12:00-14:00).
- Apakah Akhir Pekan (is_weekend): Variabel boolean (0 atau 1) yang menunjukkan apakah hari tersebut adalah Sabtu atau Minggu. Biasanya kampus lebih sepi di akhir pekan.
- Apakah Hari Libur (is_holiday): Variabel boolean untuk menandai hari libur nasional atau libur akademik, yang dapat diisi manual atau diintegrasikan dengan API kalender.
- Rata-rata Durasi Penggunaan Sebelumnya (avg_duration_prev): Rata-rata durasi penggunaan loker pada jam yang sama di hari-hari sebelumnya. Fitur ini memberikan informasi temporal tambahan.

- Jumlah Loker Terisi Jam Sebelumnya (*occupancy_prev_hour*): Jumlah loker yang terisi pada 1 jam sebelum waktu prediksi. Ini membantu model memahami tren jangka pendek.

Fitur input (X) berbentuk vektor: $\mathbf{X} = [day, hour, is_weekend, is_holiday, avg_duration, occupancy_prev]$

Target output (y):

- Jumlah loker yang terisi pada waktu tertentu (0, 1, 2, 3, atau 4)
- Atau dapat diubah menjadi persentase ketersediaan: $availability_percent = (4 - occupancy)/4 \times 100\%$

3.4.3.2 B. Training dan Validasi

Proses training model dilakukan dengan langkah-langkah berikut:

1. Pembagian Dataset:

Data dibagi menjadi 80% training set dan 20% test set menggunakan fungsi `train_test_split` dari Scikit-learn dengan stratifikasi berdasarkan jam untuk memastikan distribusi data yang seimbang.

2. Normalisasi Fitur:

Fitur numerik dinormalisasi menggunakan `StandardScaler` untuk memiliki mean = 0 dan standard deviation = 1. Normalisasi penting untuk SVR agar semua fitur memiliki skala yang sebanding.

3. Hyperparameter Tuning:

Parameter optimal SVR (C, epsilon, gamma) dicari menggunakan Grid Search Cross-Validation dengan 5-fold CV. Parameter yang diuji:

- C (parameter regularisasi): [0.1, 1, 10, 100]
- epsilon (toleransi error): [0.01, 0.1, 0.5]
- gamma (koefisien kernel RBF): ['scale', 'auto', 0.01, 0.1]

Kombinasi parameter terbaik dipilih berdasarkan skor R-squared tertinggi pada validation set.

4. Training Model:

Model SVR dilatih menggunakan library Scikit-learn (versi 1.2+) dengan kode Python:

```

from sklearn.svm import SVR
from sklearn.model_selection import GridSearchCV

param_grid = {
    'C': [0.1, 1, 10, 100],
    'epsilon': [0.01, 0.1, 0.5],
    'gamma': ['scale', 'auto', 0.01, 0.1]
}

svr = SVR(kernel='rbf')
grid_search = GridSearchCV(svr, param_grid, cv=5,
                           scoring='r2', n_jobs=-1)
grid_search.fit(X_train, y_train)

best_model = grid_search.best_estimator_

```

5. Evaluasi pada Test Set:

Model terbaik dievaluasi pada test set yang belum pernah dilihat selama training untuk mengukur kemampuan generalisasi.

3.4.3.3 C. Deployment

Setelah training selesai, model yang telah dilatih disimpan dalam format file .pkl (pickle) menggunakan library joblib:

```

import joblib
joblib.dump(best_model, 'model_prediksi_loker.pkl')
joblib.dump(scaler, 'scaler.pkl')

```

File model kemudian diunggah ke server backend. Ketika aplikasi mobile meminta prediksi ketersediaan loker, server akan:

1. Menerima input waktu yang diinginkan (misal: "Besok jam 13:00")
2. Mengekstrak fitur dari input (hari, jam, is_weekend, dll.)
3. Meload model dan scaler dari file .pkl

4. Melakukan normalisasi fitur menggunakan scaler
5. Menjalankan `model.predict(X)` untuk mendapatkan prediksi
6. Mengirim hasil prediksi kembali ke aplikasi mobile dalam format JSON

Proses inference ini dilakukan secara real-time dengan latensi rendah (< 100 ms) karena model SVR memiliki kompleksitas komputasi yang ringan setelah training.

3.5 Rencana Pengujian

Sistem akan diuji menggunakan empat jenis pengujian yang mencakup aspek fungsional, kinerja, akurasi model ML, dan pengalaman pengguna. Setiap jenis pengujian memiliki metrik evaluasi yang jelas dan target yang harus dicapai.

3.5.1 Pengujian Fungsional

Pengujian fungsional dilakukan menggunakan metode *Black Box Testing* untuk memastikan semua fitur sistem berjalan sesuai dengan spesifikasi kebutuhan yang telah didefinisikan. Pengujian mencakup skenario-skenario berikut:

Kriteria Keberhasilan: Semua skenario harus memiliki status “Pass” (minimal 11 dari 12 skenario berhasil, atau 91.67%).

3.5.2 Pengujian Kinerja (Performance)

Pengujian kinerja dilakukan untuk mengukur responsivitas sistem dan memastikan pengalaman pengguna yang lancar. Pengujian mencakup tiga aspek utama:

3.5.2.1 A. Response Time (Latency)

Mengukur selisih waktu antara penekanan tombol “Buka Kunci” pada aplikasi mobile (t_{request}) hingga solenoid lock secara fisik terbuka (t_{open}):

$$\text{Target: Response time} < 3 \text{ detik (idealnya 1-2 detik).} \quad \text{Latency} = t_{\text{open}} - t_{\text{request}} \quad (\text{Rumus 5.1})$$

Prosedur Pengujian:

1. Melakukan 30 kali percobaan unlock dari aplikasi mobile.
2. Mencatat waktu response untuk setiap percobaan.
3. Menghitung rata-rata, median, dan standar deviasi.

4. Mengidentifikasi outlier.

3.5.2.2 B. System Reliability

Pengujian konsistensi sistem dengan 50 percobaan berurutan (Buka Kunci → Tutup Kunci → ulangi).

Target: *Success rate* minimal 95%.

3.5.2.3 C. Concurrent User Handling

Menguji 3–5 pengguna melakukan request secara simultan.

Target: Semua request diproses < 5 detik tanpa konflik.

3.5.3 Evaluasi Akurasi Machine Learning

Model prediksi dievaluasi dengan MAE dan RMSE.

3.5.3.1 A. Mean Absolute Error (MAE)

Target: MAE < 1 loker. $MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$ (Rumus 5.2)

3.5.3.2 B. Root Mean Squared Error (RMSE)

Target: RMSE < 1.5 loker. $RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$ (Rumus 5.3)

3.5.3.3 C. Baseline Comparison

Baseline:

- Naive: rata-rata historis jam yang sama.
- Last Value: nilai pada jam yang sama hari sebelumnya.

3.5.3.4 D. Interpretasi Hasil

- MAE < 0.5: Excellent
- $0.5 \leq MAE < 1$: Good
- $1 \leq MAE < 1.5$: Acceptable
- MAE ≥ 1.5: Poor

3.5.4 Pengujian Usability

Menggunakan metode *System Usability Scale* (SUS).

3.5.4.1 A. Responden

- 10–15 mahasiswa pengguna motor.

3.5.4.2 B. Prosedur Pengujian

Berisi:

1. Briefing
2. Sesi penggunaan sistem
3. Pengisian SUS
4. Interview singkat (opsional)

3.5.4.3 C. Perhitungan Skor SUS

$$Skor_{SUS} = \left[\sum (R_{ganjil} - 1) + \sum (5 - R_{genap}) \right] \times 2.5 \quad (\text{Rumus 5.4})$$

Contoh Perhitungan: Skor akhir = 85 (Excellent)

3.5.4.4 D. Kriteria Keberhasilan

- < 50: buruk
- 50–68: marginal
- 68–80: baik
- > 80: sangat baik

Target: rata-rata ≥ 68 .

3.5.5 Dokumentasi Hasil Pengujian

- Tabel hasil pengujian fungsional
- Grafik response time
- Scatter plot actual vs predicted
- Tabel skor SUS
- Screenshot aplikasi dan foto prototipe
- Video demonstrasi

Tabel 3.1 Skenario Pengujian Fungsional

No	Skenario Pengujian	Hasil yang Diharapkan	Status
1	Registrasi dengan email valid (.ac.id)	Akun berhasil dibuat, konfirmasi dikirim	Pass/Fail
2	Registrasi dengan email non-institusi	Sistem menolak dan menampilkan error	Pass/Fail
3	Login dengan kredensial benar	Pengguna masuk ke dashboard	Pass/Fail
4	Login dengan password salah	Sistem menampilkan pesan error	Pass/Fail
5	Melihat status ketersediaan loker real-time	Status loker ditampilkan dengan akurat	Pass/Fail
6	Pesan loker ketika ada loker tersedia	Sistem mengalokasikan loker dan menampilkan nomor loker	Pass/Fail
7	Pesan loker ketika semua loker penuh	Sistem menampilkan notifikasi “Loker penuh”	Pass/Fail
8	Tekan tombol “Buka Kunci” di aplikasi	Solenoid lock terbuka dalam < 3 detik	Pass/Fail
9	Masukkan helm dan tutup pintu	Sensor IR mendeteksi helm, sistem update status menjadi “terisi”	Pass/Fail
10	Tekan tombol “Ambil Helm”	Solenoid lock terbuka, sistem mencatat check-out	Pass/Fail
11	Ambil helm dan tutup pintu	Sensor IR tidak mendeteksi helm, status menjadi “kosong”	Pass/Fail
12	Akses fitur prediksi ketersediaan	Aplikasi menampilkan grafik/tabel prediksi untuk beberapa jam ke depan	Pass/Fail

DAFTAR PUSTAKA

- [1] Shital Pawar et al. “OTP based Smart Locker for Time Flexibility and Safe Delivery”. *2025 International Conference on Inventive Computation Technologies (ICICT)*. 2025, pp. 1126–1133.
- [2] Yuting Wu et al. “A design of smart locker system based on IoT for campus”. *2023 IEEE 13th International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*. 2023, pp. 1287–1292.
- [3] Hanan F. Alqahtani et al. “Automated Smart Locker for College”. *2020 3rd International Conference on Computer Applications & Information Security (ICCAIS)*. 2020.
- [4] Kristianto Wijaya et al. “Smart RFID System for Locker Cabinet Security Using Android App”. *2022 4th International Conference on Cybernetics and Intelligent System (ICORIS)*. 2022.
- [5] Andi Pramono et al. “Manufacturing a Smart Locker Security System for Public Spaces using E-KTP as a Primary Access”. *2022 International Conference on ICT for Smart Society (ICISS)*. 2022, pp. 1–6.
- [6] Ika Karlina Laila Nur Suciningtyas et al. “IoT-Based Helmet Storage Cabinet with Solar Panel”. *EAI International Conference on Advanced Engineering*. 2024.
- [7] Aditya Candra Permana, Afu Ichsan Pradana, and Dwi Hartanti. “SMART LOKER BERBASIS IOT DENGAN AUTENTIKASI QR CODE TERINTEGRASI DENGAN WEB”. *JITET (Jurnal Informatika dan Teknik Elektro Terapan)* 13.3 (2025), pp. 1716–1724.
- [8] Mohammed Balfaqih. “Enhancing Security and Flexibility in Smart Locker Systems: A Multi-Authentication Approach with IoT Integration”. *2024 21st Learning and Technology Conference (L&T)*. 2024, pp. 325–329.

- [9] S. Chandrappa et al. “Smart Locker 2.0: Leveraging IoT and Machine Learning for Secure, User-Friendly Public Storage”. *2025 International Conference on Artificial Intelligence and Data Engineering (AIDE)*. 2025, pp. 696–700.
- [10] G. Anitha et al. “Parkeezy: An IoT and Machine Learning-Based Parking System”. *2025 IEEE International Students’ Conference on Electrical, Electronics and Computer Science (SCEECS)*. 2025.

LAMPIRAN

A Worksheet Proses Review Proposal Awal