

SISTEM CERDAS PENGENDALI *SMARTLOCK* BERBASIS

SIDIK JARI UNTUK KEAMANAN KAMAR HOTEL

PROYEK AKHIR

Laporan akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat kelulusan
Sarjana Terapan/Diploma IV Politeknik Manufaktur Negeri Bangka Belitung



Diusulkan oleh:

Nabila Oktariana NIM 1052151

Adysha Tristha NIM 1062132

POLITEKNIK MANUFAKTUR NEGERI

BANGKA BELITUNG

TAHUN 2024

LEMBAR PENGESAHAN

SISTEM CERDAS PENGENDALI *SMARTLOCK* BERBASIS SIDIK JARI UNTUK KEAMANAN KAMAR HOTEL

Oleh:

Nabila Oktariana / 1052151

Adysa Tristha / 1062132

Laporan akhir ini telah disetujui dan disahkan sebagai salah satu syarat kelulusan
Program Sarjana Terapan Politeknik Manufaktur Negeri Bangka Belitung

Menyetujui,

Pembimbing 1

Eko Sulisty, M.T

Pembimbing 2

Irwan, M.Sc., Ph.D.

Penguji 1

Aan Febriansyah, M.T

Penguji 2

Ahmat Josi, M.Kom

PERNYATAAN BUKAN PLAGIAT

Yang menandatangani di bawah ini:

Nama Mahasiswa : Nabila Oktariana NIM : 1052151

Nama Mahasiswa : Adyssha Tristha NIM : 1062132

Dengan Judul : Sistem Cerdas Pengendali *Smartlock* Berbasis Sidik Jari Untuk Keamanan Kamar Hotel

Kami menyatakan bahwa laporan akhir ini adalah hasil karya kami sendiri dan bukan merupakan tindakan plagiat. Pernyataan ini kami buat dengan sejujurnya, dan jika pada suatu waktu terbukti melanggar pernyataan ini, kami siap menerima sanksi yang diberlakukan.

Nama Mahasiswa

Sungailiat, 07 Agustus 2024

Tanda tangan

1. Nabila Oktariana

.....


2. Adyssha Tristha

.....


ABSTRAK

Hotel menyediakan penginapan serta layanan makanan dan minuman, di mana pencatatan administrasi sangat penting untuk menjaga keamanan data. Kamar hotel sebagai ruang pribadi tamu memerlukan sistem keamanan yang andal. Saat ini, teknologi RFID digunakan untuk mengunci pintu kamar, namun memiliki kekurangan seperti akses dengan satu kartu dan risiko kehilangan kartu. Teknologi kunci pintu berbasis sidik jari lebih direkomendasikan karena menghilangkan kebutuhan akan kartu atau kunci fisik dan mengurangi risiko kehilangan. Teknologi sidik jari yang sudah digunakan di kantor dan rumah sakit menawarkan keamanan tinggi dan memungkinkan pintu dibuka otomatis serta menyalakan lampu dan AC saat pintu dibuka. Metodologi penelitian mencakup beberapa tahap: studi literatur untuk memahami teori yang ada, perancangan sistem termasuk *hardware* dan *software*, pembuatan dan perakitan sistem, serta pengujian dan penyusunan laporan akhir. Studi literatur membantu dalam merumuskan dasar teori. Perancangan sistem meliputi pembuatan diagram dan desain komponen. Pembuatan *hardware* melibatkan aspek mekanik dan elektrik seperti rangkaian elektrik dan perakitan komponen. Pembuatan *software* dilakukan dengan Arduino IDE untuk mengontrol sistem. Pengujian bertujuan memastikan sistem berfungsi dengan baik dalam berbagai kondisi. Laporan akhir merangkum seluruh proses dan hasil penelitian, memberikan kontribusi pada pengembangan sistem keamanan kamar hotel dengan teknologi sidik jari.

Kata Kunci : Pengembangan Sistem Keamanan, Teknologi Sidik Jari, Sistem Otomatis.

ABSTRACT

Hotels provide lodging as well as food and beverage services, where administrative records are essential to maintain data security. Hotel rooms as guests' private space require a reliable security system. Currently, RFID technology is used to lock room doors, but it has drawbacks such as access with a single card and the risk of losing the card. Fingerprint-based door lock technology is more recommended as it eliminates the need for physical cards or keys and reduces the risk of loss. Fingerprint technology already used in offices and hospitals offers high security and allows doors to be opened automatically as well as turning on lights and air conditioning when the door is opened. The research methodology includes several stages: literature study to understand the existing theory, system design including hardware and software, system manufacturing and assembly, and testing and preparation of the final report. The literature study helped in formulating the theoretical basis. System design includes diagramming and component design. Hardware development involves mechanical and electrical aspects such as electronic circuits and component assembly. Software development is done with Arduino IDE to control the system. Testing aims to ensure the system functions properly under various conditions. The final report summarizes the entire research process and results, contributing to the development of a hotel room security system with fingerprint technology.

Keyword : Security System Develoment, Fingerprint Technologyy, Automatic System.

KATA PENGANTAR

Assalamu'alaikum warahmatullahi wabarakatuh,

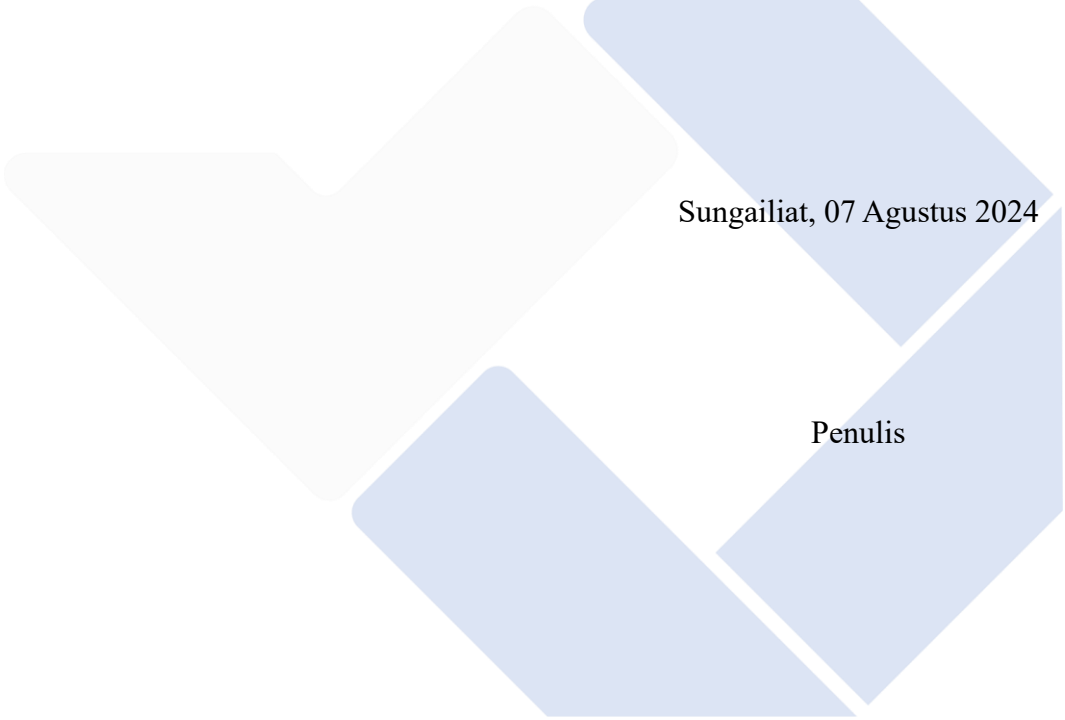
Puji syukur kami panjatkan kepada Allah SWT Yang Maha Pengasih lagi Maha Penyayang atas rahmat serta karuniaNya, Nabi Muhammad SAW sebagai pedoman pemberi petunjuk sehingga kami dapat menyelesaikan penelitian ini dengan judul “Sistem Cerdas Pengendali *Smartlock* Berbasis Sidik Jari Untuk Keamanan Kamar Hotel”. Penelitian ini dibuat dalam rangka sebagai salah satu persyaratan dalam kelulusan pada program studi D4 Teknik Elektronika dan D4 Teknologi Rekayasa Perangkat Lunak pada Politeknik Manufaktur Negeri Bangka Belitung. Penulis menyadari bahwa dalam proses penyusunan penelitian serta pembuatan sistem yang penulis lakukan tidak akan lepas dari pengetahuan, doa, dukungan, serta bantuan dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan rasa terima kasih kepada:

1. Allah SWT atas segala keberkahan serta nikmat yang telah diberikan.
2. Orang Tua, Saudara serta Keluarga penulis yang telah memberikan motivasi serta perhatian kepada penulis.
3. Bapak I Made Andik Setiawan, M.Eng, Ph.D selaku Direktur Politeknik Manufaktur Negeri Bangka Belitung.
4. Bapak Irwan, M.Sc, Ph.D selaku Wakil Direktur I Politeknik Manufaktur Negeri Bangka Belitung.
5. Bapak Muhammad Subhan, M.T selaku Wakil Direktur II Politeknik Manufaktur Negeri Bangka Belitung.
6. Bapak Eko Sulisty, M.T selaku Wakil Direktur III Politeknik Manufaktur Negeri Bangka Belitung dan Dosen Pembimbing I serta Bapak Irwan, M.Sc, Ph.D sebagai pembimbing II, atas dukungan, kritik dan saran dalam membangun sistem..
7. Bapak Zanu Saputra S.ST., M.Tr.T selaku Ka. Jurusan Teknik Elektro & Informatika Politeknik Manufaktur Negeri Bangka Belitung.

8. Teman-teman penulis yang mungkin tidak dapat penulis sebutkan satu-satu, terutama teman seperjuangan pada D4 Teknologi Rekayasa Perangkat Lunak dan D4 Teknik Elektronika baik dalam suka dan duka, selalu mendukung serta memberikan arahan kepada penulis.

Demikian laporan yang penulis buat, penulis mengucapkan terima kasih dan berharap semoga penelitian yang penulis buat dapat menambah wawasan untuk kita semua.

Wassalamu'alaikum Warahmatullahi Wabarakatuh



Sungailiat, 07 Agustus 2024

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN	Error! Bookmark not defined.
PERNYATAAN BUKAN PLAGIAT	Error! Bookmark not defined.
ABSTRAK	iv
KATA PENGANTAR.....	vi
DAFTAR ISI	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN	xiv
BAB I	2
PENDAHULUAN	2
1.1 Latar Belakang	2
1.2 Tujuan.....	3
1.3 Perumusan Masalah.....	3
1.4 Batasan Masalah.....	3
BAB II.....	5
DASAR TEORI	5
2.1 Smart door lock	5
2.2 Internet of things (IoT)	6
2.3 Sidik Jari.....	8
2.4 Microcontroller.....	10
2.5 NodeMCU ESP32	11
2.6 Sensor Fingerprint	12
2.7 Website	12
2.8 Framework Codeigniter.....	12
2.9 Bootstrap	13
2.10 Xampp	14

2.11	Firestore	14
BAB III		15
METODE PELAKSANAAN.....		15
3.1	Studi Literatur.....	16
3.2	Perancangan Sistem.....	16
3.2.1	Rangkaian Elektrikal	16
3.2.2	Desain Alat	17
3.2.3	Use Case Diagram.....	18
3.2.4	Perancangan <i>Software</i>	20
3.3	Pembuatan dan Perakitan Sistem.....	22
3.4	Pengujian Keseluruhan.....	22
3.5	Pembuatan Laporan Proyek Akhir.....	23
BAB IV		24
PEMBAHASAN		24
4.1	Deskripsi Alat	24
4.2	Pembuatan <i>Hardware</i>	26
4.2.1	Pembuatan <i>Hardware</i> Secara Mekanik	26
4.2.2	Pembuatan <i>Hardware</i> Secara Elektrikal	27
4.3	Perancangan <i>Database</i>	28
4.3.1	Tabel <i>Categories</i>	28
4.3.2	Tabel <i>Email_Check_In</i>	28
4.3.3	Tabel <i>Guests</i>	29
4.3.4	Tabel <i>Invoice</i>	29
4.3.5	Tabel <i>Invoice_Pos_Relation</i>	30
4.3.6	Tabel <i>Point_Of_Sales</i>	30
4.3.7	Tabel <i>Pos_Transactions</i>	30
4.3.8	Tabel <i>Rooms</i>	31
4.3.9	Tabel <i>Room_Service</i>	31
4.3.10	Tabel <i>Room_Type</i>	32
4.3.11	Tabel <i>Users</i>	32
4.4	Tampilan Antarmuka	32

4.4.1	Homepage Login.....	32
4.4.2	<i>Homepage Dashbord Admin</i>	33
4.4.3	Homepage Check-in.....	34
4.4.4	Homepage Check Out	35
4.4.5	Homepage Point Of Sale.....	37
4.4.6	Homepage Housekeeping.....	38
4.4.7	Homepage Rooms di Data Master	39
4.4.8	Homepage Point Of Sale di Data Master	41
4.4.9	Homepage Guests di Data Master.....	42
4.4.10	Homepage Users di Data Master	43
4.4.11	Homepage Check In di Reporting.....	44
4.4.12	Homepage Point Of Sales di Reporting	44
4.5	Pengujian	45
4.5.1	Pengujian Sidik Jari Saat Pendaftaran Kamar.....	45
4.5.2	Pengujian Sensor.....	46
4.5.3	Pengujian Lampu, Kipas dan Selenoid	47
BAB V.....		49
KESIMPULAN DAN SARAN.....		49
5.1	Kesimpulan.....	49
5.2	Saran	49
DAFTAR PUSTAKA		51
LAMPIRAN.....		54

DAFTAR TABEL

Tabel 3. 1 Tabel Aktor	19
Tabel 4. 1 Hasil Pengujian Sidik Jari pada setiap kamar	46
Tabel 4. 2 Hasil pengujian sensor	47
Tabel 4. 3 Hasil Pengujian Lampu, Kipas dan Selenoid	48



DAFTAR GAMBAR

Gambar 2. 1 Penerapan <i>Smartlock Door</i>	5
Gambar 2. 2 Struktur Sistem IoT	7
Gambar 2. 3 Pola Sidik Jari.....	9
Gambar 2. 4 Tipe <i>minutiae</i> sidik jari.....	10
Gambar 2. 5 NodeMCU ESP32	11
Gambar 3. 1 Diagram Pelaksanaan Proyek Akhir.....	15
Gambar 3. 2 Rangkaian Elektrikal	16
Gambar 3. 3 Desain Kontruksi.....	17
Gambar 3. 4 Desain Kontruksi Tampak Depan.....	17
Gambar 3. 5 Use Case Diagram Sistem Reservasi Hotel.....	18
Gambar 3. 6 <i>Homepage Login</i>	20
Gambar 3. 7 <i>Homepage Proses Check-in</i>	20
Gambar 3. 8 <i>Homepage Proses Check-out</i>	21
Gambar 3. 9 <i>Homepage Dashboard</i>	21
Gambar 4. 1 Flowchart Sistem Kerja Alat	25
Gambar 4. 2 Hasil Pembuatan Alat Tampak Depan Secara Mekanik	26
Gambar 4. 3 Hasil Pembuatan Alat Tampak Belakang Secara Mekanik	27
Gambar 4. 4 Pembuatan <i>Hardware</i> Secara Elektrikal.....	27
Gambar 4. 5 Pembuatan <i>Hardware</i> Secara Elektrikal Tampak Keseluruhan.....	28
Gambar 4. 6 Tabel <i>Categories</i>	28
Gambar 4. 7 Tabel <i>Email_Check_In</i>	29
Gambar 4. 8 Tabel <i>Guests</i>	29
Gambar 4. 9 Tabel <i>Invoice</i>	29
Gambar 4. 10 Tabel <i>Invoice_Pos_Relation</i>	30
Gambar 4. 11 Tabel <i>Point_Of_Sales</i>	30
Gambar 4. 12 Tabel <i>Pos_Transactions</i>	31
Gambar 4. 13 Tabel <i>Rooms</i>	31
Gambar 4. 14 Tabel <i>Room_Service</i>	31
Gambar 4. 15 Tabel <i>Room_Type</i>	32

Gambar 4. 16 Tabel <i>Users</i>	32
Gambar 4. 17 <i>Homepage Login</i>	33
Gambar 4. 18 <i>Homepage Dashbord Admin</i>	33
Gambar 4. 19 <i>Homepage Check-in</i>	34
Gambar 4. 20 <i>Homepage Check-in Process</i>	34
Gambar 4. 21 <i>Homepage Check-Out</i>	35
Gambar 4. 22 <i>Homepage Check-Out Process</i>	35
Gambar 4. 23 <i>Homepage Check-Out Sidik Jari</i>	36
Gambar 4. 24 <i>Homepage Check Out Pin</i>	36
Gambar 4. 25 <i>Homepage Point Of Sales</i>	37
Gambar 4. 26 <i>Homepage Point Of Sales Process</i>	37
Gambar 4. 27 <i>Homepage Housekeeping</i>	38
Gambar 4. 28 <i>Homepage Housekeeping Process\</i>	38
Gambar 4. 29 <i>Homepage Data Master Rooms</i>	39
Gambar 4. 30 <i>Homepage Data Master Add Rooms</i>	39
Gambar 4. 31 <i>Homepage Data Master Edit Rooms</i>	40
Gambar 4. 32 <i>Homepage Data Master Point Of Sales</i>	41
Gambar 4. 33 <i>Homepage Data Master Add Point Of Sales</i>	41
Gambar 4. 34 <i>Homepage Data Master Edit Point Of Sales</i>	42
Gambar 4. 35 <i>Homepage Data Master Guest</i>	42
Gambar 4. 36 <i>Homepage Data Master Add Guests</i>	43
Gambar 4. 37 <i>Homepage Data Master Users</i>	43
Gambar 4. 38 <i>Homepage Data Master Add Users</i>	44
Gambar 4. 39 <i>Homepage Reporting Check-in</i>	44
Gambar 4. 40 <i>Homepage Reporting Point Of Sales</i>	45

DAFTAR LAMPIRAN

Lampiran 1 Daftar Riwayat Hidup Penulis.....	Error! Bookmark not defined.
Lampiran 2 Kode Program Arduino.....	56
Lampiran 3 Hasil Dokumentasi Pengujian Sistem Cerdas Pengendali Smartlock Berbasis Sidik Jari Untuk Keamanan Kamar Hotel.....	84



BAB I

PENDAHULUAN

1.1 Latar Belakang

Hotel adalah sebuah bisnis komersial atau tempat yang ditujukan untuk siapa saja yang ingin menikmati fasilitas penginapan dan layanan, termasuk penyediaan makanan dan minuman. Dalam industri perhotelan, proses pencatatan administrasi sangatlah penting. Oleh karena itu, pihak hotel harus memastikan mekanisme pencatatan yang akurat dan terorganisir dengan baik agar data terlindungi dan keamanannya terjamin.[1]. Kamar hotel adalah ruangan yang sangat pribadi bagi setiap tamu yang menyewanya. Keamanan kamar hotel menjadi sangat penting.

Saat ini, sistem pengunci pintu kamar hotel telah menggunakan teknologi RFID. Namun, penggunaannya masih kurang optimal karena setiap kamar hanya dapat diakses dengan satu kartu. Jika kartu tersebut hilang, maka perlu dilakukan penginputan ulang ID kartu. Selain itu, siapa pun yang memegang kartu akses dapat mengakses kamar hotel tersebut.[2]. Oleh karena itu, teknologi kunci pintu otomatis berbasis sidik jari diperlukan untuk mempermudah tamu dalam keluar masuk kamar tanpa harus membawa kunci atau kartu khusus, serta mengurangi risiko kehilangan kunci atau kartu kamar.

Pola sidik jari memiliki tingkat keamanan yang tinggi dan sudah banyak digunakan di perkantoran, perusahaan, sekolah, pemerintahan, dan rumah sakit. Selain itu, sistem ini mudah digunakan. Penggunaan sistem biometrik untuk keamanan pintu masih jarang ditemukan. Kebanyakan di Indonesia masih menggunakan kunci manual, yang memerlukan waktu untuk membuka jika hilang. Dengan menggunakan sidik jari, salah satu jari kita menjadi kunci, sehingga tidak ada risiko terlupakan atau tertinggal di suatu tempat.[3].

Sistem ini tidak hanya berfungsi untuk membuka dan menutup pintu kamar hotel secara otomatis menggunakan sidik jari, tetapi juga memiliki keunggulan lain, yaitu dapat secara otomatis menyalakan lampu dan AC saat pintu kamar terbuka.

Meskipun demikian, lampu dan AC tetap dapat dikontrol secara manual menggunakan saklar lampu seperti biasa.

1.2 Tujuan

Berdasarkan latar belakang yang telah dijelaskan, tujuan dari penelitian ini adalah:

1. Memberikan tingkat keamanan yang lebih tinggi dengan menghilangkan risiko pencurian kunci atau kartu akses, serta menghindari masuknya orang yang tidak diinginkan ke dalam kamar.
2. Memberikan kenyamanan kepada tamu hotel dengan memungkinkan mereka untuk masuk ke dalam kamar tanpa harus membawa kunci atau kartu akses, cukup dengan menggunakan sidik jari mereka.
3. Penggunaan teknologi *Smartlock* dapat membantu meningkatkan efisiensi operasional hotel dengan meminimalkan waktu yang diperlukan untuk penanganan masalah kunci yang hilang atau rusak.

1.3 Perumusan Masalah

Permasalahan yang dapat diambil dalam proyek akhir adalah:

1. Bagaimana pengembangan sistem kunci pintu otomatis berbasis sidik jari untuk meningkatkan keamanan dan kenyamanan tamu di hotel.
2. Integrasi sistem kunci pintu otomatis berbasis sidik jari dengan pengaturan otomatis lampu dan AC dalam kamar hotel untuk meningkatkan efisiensi energi dan kenyamanan tamu.

1.4 Batasan Masalah

Pada penelitian ini, penulis membatasi masalah utama sebagai berikut:

1. Penggunaan kartu akses pada kunci pintu kamar hotel yang hanya dapat diakses oleh pemilik kartu, namun memungkinkan akses yang tidak diinginkan jika kartu hilang.

2. Sistem ini hanya dirancang untuk digunakan oleh tamu hotel dan staf yang memiliki otorisasi, tidak mencakup penggunaan oleh pihak ketiga.
3. Proyek ini hanya mencakup implementasi dalam skala terbatas pada beberapa kamar hotel untuk pengujian dan evaluasi, bukan untuk implementasi skala penuh di seluruh hotel.

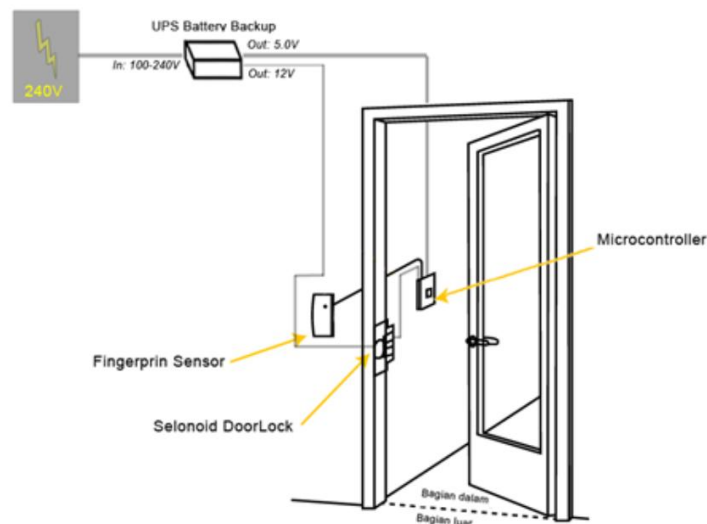


BAB II

DASAR TEORI

2.1 *Smart door lock*

Smart door lock adalah perangkat pengunci pintu yang mengimplementasikan metode operasional yang tidak konvensional. Sistem ini memungkinkan pengoperasian melalui pemindai sidik jari, yang berfungsi utama untuk membatasi akses dan memastikan hanya individu yang memiliki hak yang dapat membuka pintu tersebut [4]. Gambar 2.1 memperlihatkan desain dan penempatan perangkat keras, serta jalur sumber energi listrik yang diperlukan. Namun, terdapat kelemahan dalam sistem ini, yaitu ketergantungan pada sumber listrik utama hotel; jika terjadi pemutusan aliran listrik, sistem penguncian pintu tidak dapat berfungsi. Untuk mengatasi masalah tersebut, digunakan sumber arus cadangan berupa *UPS Battery Backup* [5]. UPS ini menyediakan dua saluran *output*, yaitu 12V dan 5V, yang langsung terhubung ke solenoid *door lock* serta mikrokontroler, sehingga memperkuat sistem dalam mencegah kemungkinan kegagalan penguncian pintu kamar hotel [6].



Gambar 2. 1 Penerapan *Smartlock Door*

Solenoid *door lock* adalah sebuah perangkat elektromekanik yang berfungsi sebagai sistem pengunci pintu otomatis. Dalam kondisi standar, solenoid *door lock* berada dalam keadaan terkunci. Ketika diberi arus listrik, solenoid *door lock* akan membuka [7]. Perangkat ini mengoperasikan dua sistem, yaitu *Normally Closed* (NC) dan *Normally Open* (NO). Perbedaan utama di antara keduanya terletak pada respons terhadap tegangan: pada sistem NC, pemberian tegangan akan menyebabkan tuas solenoid memanjang dan tetap dalam keadaan tertutup, sedangkan pada sistem NO, sebaliknya, pemberian tegangan akan menyebabkan solenoid membuka. Umumnya, solenoid *door lock* beroperasi dengan *input* tegangan sebesar 12V DC. Namun, terdapat juga model yang dapat berfungsi dengan tegangan lebih rendah, seperti 5V DC, sehingga kompatibel dengan *output* dari pin IC digital [8]. Untuk solenoid *door lock* yang memerlukan tegangan *input* lebih dari 5V DC, penggunaan rangkaian tambahan berupa relay sering kali diperlukan untuk memastikan operasi yang tepat [9].

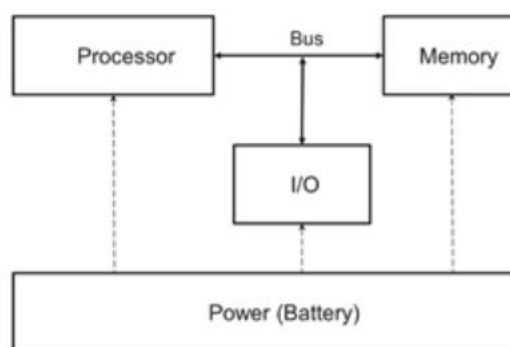
2.2 *Internet of things (IoT)*

Dengan pesatnya perkembangan teknologi saat ini, sistem penguncian otomatis berbasis solenoid dapat diintegrasikan dengan teknologi *Internet of Things* (IoT). IoT mengacu pada sebuah sistem di mana baik objek maupun individu diberikan identitas yang unik dan kemampuan untuk mentransfer data melalui jaringan tanpa memerlukan interaksi langsung antara manusia, atau antara manusia dan komputer. Prinsip dasar kerja IoT melibatkan pemberian alamat *Internet Protocol* (IP) pada setiap objek, yang berfungsi sebagai identitas dalam jaringan dan memungkinkan objek tersebut untuk diakses atau dikendalikan oleh entitas lain dalam jaringan yang sama. Selanjutnya, alamat IP ini akan terhubung dengan jaringan internet, memungkinkan komunikasi dan kontrol yang efisien antara berbagai perangkat dalam ekosistem IoT [11].

Pada masa kini, akses ke jaringan internet telah menjadi sangat mudah diakses. Hal ini memungkinkan pengguna untuk memantau dan mengendalikan perangkat secara jarak jauh melalui internet. Ketika sebuah perangkat memiliki alamat IP dan

terhubung ke jaringan internet, perangkat tersebut dapat dipasang sensor yang memungkinkan pengumpulan data yang diperlukan. Data yang dikumpulkan kemudian dapat diproses oleh perangkat itu sendiri dan, dalam beberapa kasus, perangkat tersebut dapat berkomunikasi dengan perangkat lain yang juga memiliki alamat IP dan terhubung ke internet. Pertukaran data antara perangkat-perangkat tersebut memfasilitasi operasi otomatis atau instruksi kepada perangkat lain untuk melakukan tugas tertentu. Fenomena ini merupakan salah satu keunggulan utama dari konsep *Internet of Things* (IoT) [12].

Sistem *Internet of Things* (IoT) berfungsi dengan mekanisme komunikasi ujung-ke-ujung atau multi-hop, menggambarkan hubungan yang rumit antara sumber data dan penerima data. Secara umum, IoT beroperasi dengan mengumpulkan data melalui perangkat sensor, memproses informasi tersebut, dan kemudian melakukan berbagai tindakan, mulai dari mengirim pemberitahuan dan meningkatkan alarm hingga mengendalikan aktuator dalam sistem fisik. Selain itu, IoT merupakan bentuk sistem komputasi tertanam yang memanfaatkan arsitektur polisilikon untuk beragam aplikasi. Struktur tipikal dari sistem IoT, yang digambarkan pada Gambar 2.1, mencakup empat subsistem utama, yaitu: (i) pemrosesan, (ii) memori, (iii) *input/output*, dan (iv) daya [6]. Sistem ini mengintegrasikan berbagai elemen untuk mencapai efisiensi operasional dan responsif terhadap kondisi yang terdeteksi.



Gambar 2. 2 Struktur Sistem IoT

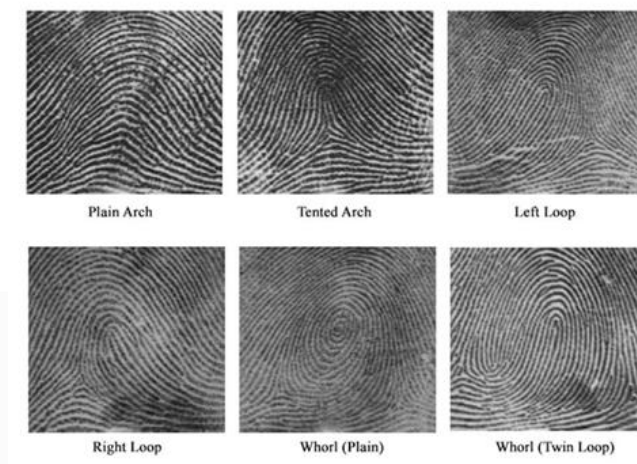
2.3 *Sidik Jari*

Pada tahun 1893, Kementerian Inggris mengeluarkan pernyataan bahwa setiap individu memiliki sidik jari yang unik, sehingga tidak mungkin ada dua orang dengan pola sidik jari yang identik. Penemuan ini membuka jalan bagi berbagai instansi penegak hukum untuk mengeksplorasi potensi identifikasi menggunakan sidik jari. Konsep biometrik, atau sering dikenal sebagai *biometric*, merujuk pada penerapan karakteristik anatomi dan perilaku khas (seperti sidik jari, wajah, suara, atau geometri tangan) untuk pengenalan individu secara otomatis. Teknologi biometrik menghadirkan peningkatan signifikan dalam hal keamanan, efisiensi, dan kenyamanan bagi pengguna, dengan memungkinkan identifikasi yang lebih akurat dan proses yang lebih cepat dibandingkan dengan metode tradisional.

Salah satu studi tentang penerapan biometrik sidik jari melibatkan pengembangan prototipe sistem keamanan pintu berskala kecil. Sistem ini memanfaatkan identifikasi sidik jari sebagai metode pengamanan pintu, beroperasi dengan menggunakan sensor sidik jari yang mampu mengidentifikasi dan memverifikasi sidik jari yang telah direkam dalam memori sistem. Kontrol dan pengelolaan seluruh fungsi sistem ini dilakukan oleh mikrokontroler Atmega 8 dari Atmel, yang menyimpan data pemrograman esensial untuk operasional sistem keamanan pintu. Mikrokontroler ini bertanggung jawab atas koordinasi dan pengendalian semua proses yang diperlukan untuk menjalankan sistem dengan efektif. Penggunaan kunci konvensional sering kali menghadapi masalah, seperti kemungkinan kehilangan atau kelalaian kunci, serta kerusakan kunci akibat penggunaan yang terburu-buru. Ketidakadaan kunci cadangan dapat menyebabkan ketidakmampuan membuka pintu. Dalam konteks ini, penerapan teknologi elektronik pada sistem keamanan pintu menawarkan solusi yang lebih praktis dan dapat meningkatkan kemudahan serta keamanan bagi pengguna dalam melindungi rumah mereka [13].

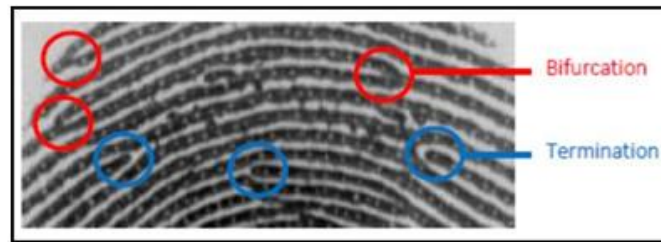
Sidik jari terbentuk dari pola-pola yang meliputi elemen-elemen gunung (bagian yang menonjol ke atas) dan lembah (bagian yang cekung). Pada tahap awal proses identifikasi sidik jari, biasanya dilakukan pengelompokan sidik jari ke dalam

salah satu dari lima kategori dasar yang dikenal sebagai sistem klasifikasi Henry. Kategori-kategori ini mencakup *Plain Arch*, *Tented Arch*, *Left Loop*, *Right Loop*, serta *Whorl*, yang selanjutnya dibagi lagi menjadi dua subkategori, yaitu lingkaran polos dan lingkaran kembar (D. Maltoni, D. Maio, A. Jain, & S. Prabhakar, 2009a). Sistem klasifikasi ini memungkinkan untuk identifikasi dan analisis sidik jari yang lebih sistematis dan terstruktur.



Gambar 2. 3 Pola Sidik Jari

Langkah kedua dalam proses identifikasi sidik jari melibatkan analisis pada tingkat lokal, yang mencakup pemeriksaan elemen-elemen kecil pada garis punggung yang dikenal sebagai *minutiae*. *Minutiae* ini mencakup berbagai fitur penting yang dapat digunakan untuk membedakan sidik jari individu. Dua jenis *minutiae* yang paling sering ditemukan adalah bifurkasi dan penghentian *ridge*. Bifurkasi terjadi ketika garis punggung bercabang menjadi dua segmen terpisah, sementara penghentian *ridge* merujuk pada ujung garis punggung yang berakhir secara tiba-tiba. Sidik jari yang biasanya ditemukan memiliki sekitar 80 *minutiae*; meskipun demikian, jumlah *minutiae* ini sering kali lebih sedikit pada gambar yang diperoleh melalui pemindai standar dalam sistem biometrik, dikarenakan keterbatasan resolusi pemindaian.



Gambar 2. 4 Tipe *minutiae* sidik jari

Sensor sidik jari beroperasi dengan mendeteksi titik-titik *minutiae* pada sidik jari dan membentuk pola dengan menghubungkan titik-titik tersebut. Pola yang dihasilkan dari proses penghubungan ini kemudian digunakan untuk proses pencocokan saat sidik jari dipindai. Dengan kata lain, sistem identifikasi sidik jari tidak mencocokkan gambar secara langsung, melainkan membandingkan pola yang diperoleh dari *minutiae*. Sensor sidik jari berfungsi dengan mengambil gambar sidik jari dan menganalisis setiap pola atau alur yang terdapat di dalamnya. Berbagai teknik dapat digunakan untuk pemotretan sidik jari, namun metode yang paling umum diterapkan meliputi dua pendekatan utama: sensor optik dan sensor kapasitansi. Sensor optik mengandalkan pencitraan berbasis cahaya untuk memperoleh gambar sidik jari, sedangkan sensor kapasitansi memanfaatkan perubahan kapasitansi untuk mendeteksi pola *minutiae* [14].

2.4 *Microcontroller*

Mikrokontroler ini telah dilengkapi dengan modul internet bawaan, sehingga tidak memerlukan perangkat tambahan untuk terhubung ke jaringan internet. NodeMCU merupakan sebuah platform IoT berbasis *open-source* yang menggabungkan perangkat keras berupa *System On Chip* (SoC) ESP8266 dan ESP32 dari *Espressif Systems* dengan firmware yang memanfaatkan bahasa pemrograman C++. Dengan integrasi komponen-komponen ini, NodeMCU memungkinkan implementasi aplikasi IoT dengan efisiensi tinggi dan kemudahan pengembangan, tanpa ketergantungan pada modul eksternal.



Gambar 2. 5 NodeMCU ESP32

2.5 *NodeMCU ESP32*

ESP32 merupakan hasil pengembangan dari *Espressif Systems*, sebuah perusahaan yang berlokasi di Shanghai, Tiongkok. Salah satu fitur utama dari ESP32 adalah kemampuannya untuk mengintegrasikan modul WiFi dan Bluetooth secara bersamaan. Fitur ini memberikan kemudahan signifikan dalam perancangan sistem *Internet of Things* (IoT) yang memerlukan konektivitas nirkabel. Integrasi ini memungkinkan implementasi yang lebih efisien dan fleksibel dalam berbagai aplikasi IoT, meningkatkan kinerja dan efisiensi sistem yang mengandalkan komunikasi tanpa kabel [15]. Modul ini memiliki beragam aplikasi potensial, termasuk dalam pengendalian sistem, pemantauan, dan berbagai fungsi lainnya. ESP32 dirancang dengan fitur *deep sleep* yang memungkinkan penghematan energi dengan menonaktifkan modul saat tidak aktif. Tampilan fisik dari NodeMCU ESP32 dapat dilihat pada ilustrasi yang disediakan. Mikrokontroler ESP32 menawarkan berbagai keunggulan, termasuk biaya yang terjangkau, konsumsi daya yang efisien, serta integrasi modul WiFi langsung pada chip mikrokontroler. Lebih lanjut, ESP32 juga dilengkapi dengan kemampuan Bluetooth dalam mode ganda dan teknologi hemat energi, yang menambah fleksibilitas penggunaannya. Selain itu, modul NodeMCU ESP32 berfungsi sebagai papan prototipe yang kompak dan dapat diprogram dengan mudah menggunakan Arduino IDE maupun Python [16].

2.6 *Sensor Fingerprint*

Perangkat sensor sidik jari merupakan alat elektronik yang memanfaatkan teknologi pemindaian untuk tujuan identifikasi individu melalui analisis sidik jari. Pada tahap awal, sensor ini melakukan perekaman data sidik jari sebagai referensi dasar. Data yang direkam kemudian disimpan dalam sebuah basis data yang terstruktur. Ketika seseorang berusaha mengakses perangkat menggunakan sensor sidik jari, proses pemindaian ulang dilakukan untuk membandingkan sidik jari yang baru dengan data yang telah tersimpan dalam basis data tersebut. Sistem ini berfungsi untuk memastikan kesesuaian dan keakuratan identifikasi yang dilakukan [17].

2.7 *Website*

Website merupakan kumpulan halaman yang dirancang untuk menyajikan berbagai tipe informasi, termasuk teks, gambar statis maupun dinamis, animasi, suara, serta kombinasi dari elemen-elemen tersebut. Halaman-halaman ini dapat bersifat statis atau dinamis, disusun dengan cermat untuk membentuk sebuah entitas yang saling berhubungan. Setiap halaman di dalam website terintegrasi melalui jaringan tautan yang menciptakan sebuah struktur, memungkinkan pengguna untuk menjelajahi dan memperoleh akses ke informasi yang tersedia secara efisien. Struktur ini memfasilitasi navigasi yang mulus dan integrasi antara berbagai konten yang disajikan di dalam *website*. [18].

2.8 *Framework Codeigniter*

CodeIgniter merupakan sebuah kerangka kerja PHP berbasis *open source* yang dirancang untuk mempercepat proses pengembangan aplikasi web. *Framework* ini menawarkan beragam pustaka dan alat bantu yang berguna, yang pada gilirannya berkontribusi pada pengurangan waktu dan tenaga yang dibutuhkan dalam proses pembangunan aplikasi. Dengan serangkaian fitur yang komprehensif, *CodeIgniter*

memfasilitasi para pengembang untuk menyelesaikan proyek dengan tingkat efisiensi dan efektivitas yang lebih tinggi. Penerapan *CodeIgniter* memungkinkan pengelolaan kode yang lebih terstruktur dan responsif terhadap kebutuhan pengembangan aplikasi yang terus berkembang, sehingga memberikan dukungan yang signifikan terhadap upaya pengoptimalan dalam penyelesaian proyek perangkat lunak [19]. *CodeIgniter* mengadopsi pendekatan desain berbasis Model, *View*, *Controller* (MVC), suatu metode yang membagi aplikasi menjadi tiga komponen utama. Pendekatan ini bertujuan untuk menyediakan berbagai alat bantu, seperti *helpers* dan *libraries*, yang diperlukan untuk menyelesaikan tugas-tugas umum dalam pengembangan aplikasi. Dengan penerapan konsep ini, pengembangan proyek menjadi lebih terstruktur dan efisien, mengurangi kebutuhan untuk penulisan kode dari awal pada setiap pembuatan aplikasi baru. Hal ini tidak hanya mempercepat proses pengembangan tetapi juga meningkatkan konsistensi dan pemeliharaan aplikasi melalui penggunaan kembali kode dan struktur yang telah ditetapkan [20]

2.9 *Bootstrap*

Bootstrap adalah sebuah framework front-end yang unggul dan berfokus pada optimalisasi tampilan untuk perangkat mobile seperti ponsel dan *smartphone*. *Framework* ini dikembangkan untuk mempercepat serta menyederhanakan proses pembuatan situs web. *Bootstrap* menyediakan berbagai komponen yang meliputi HTML, CSS, dan JavaScript yang siap pakai dan dapat disesuaikan dengan mudah. Dengan menggunakan *Bootstrap*, pengembang dapat menciptakan desain web yang bersifat responsif, yang artinya antarmuka situs web akan otomatis menyesuaikan dengan ukuran layar perangkat yang digunakan, baik itu desktop, tablet, atau perangkat *mobile*. *Fleksibilitas* fitur responsif ini memungkinkan pengaturan, sehingga pengembang dapat memilih untuk membuat situs web yang hanya dioptimalkan untuk tampilan desktop. Dalam kasus seperti itu, tampilan situs web tidak akan menyesuaikan dengan ukuran layar perangkat *mobile* saat diakses [21]

2.10 *Xampp*

XAMPP merupakan perangkat lunak berbasis sumber terbuka yang mendukung berbagai sistem operasi dan terdiri dari beberapa aplikasi terintegrasi. Perangkat lunak ini berfungsi sebagai server lokal (localhost), mencakup sejumlah komponen esensial seperti Apache HTTP Server, sistem manajemen basis data MySQL, serta penerjemah bahasa pemrograman PHP dan Perl. Kombinasi dari berbagai aplikasi ini memungkinkan *XAMPP* untuk menyediakan lingkungan pengembangan yang komprehensif dan *user-friendly*, khususnya untuk pengembangan web secara lokal. Dengan fitur-fitur tersebut, *XAMPP* mempermudah proses pengembangan, pengujian, dan implementasi aplikasi web dalam setting yang terisolasi sebelum diterapkan di server produksi [22]

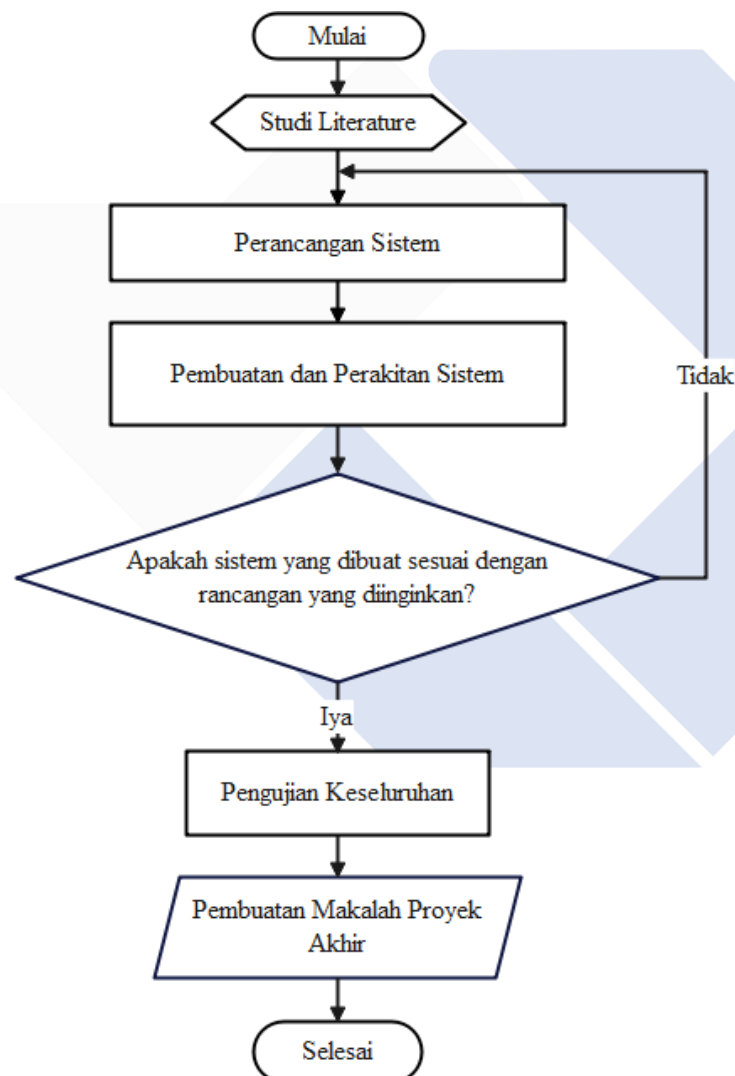
2.11 *Firebase*

Firebase merupakan penyedia layanan berbasis *cloud* yang menawarkan infrastruktur *back-end* sebagai layanan dan berlokasi di San Francisco, California. Platform ini menyediakan serangkaian produk dan solusi yang dikembangkan untuk mendukung proses pengembangan aplikasi pada platform *mobile* dan web. Berbagai fitur yang ditawarkan oleh *Firebase* mencakup database *real-time*, autentikasi pengguna, dan analitik yang memungkinkan pengembang untuk membangun dan mengelola aplikasi dengan lebih efisien [23]. *Firebase* memerlukan akses internet untuk mengoperasikan aplikasi karena data yang digunakan disimpan di platform penyimpanan berbasis awan. Implementasi basis data *online* ini memungkinkan perangkat-perangkat yang berbeda untuk saling berinteraksi dan bertukar informasi dengan tingkat efisiensi yang tinggi. Sebagai penyedia layanan cloud dan platform *backend* yang dimiliki oleh Google, *Firebase* menawarkan berbagai kemudahan dalam pengembangan aplikasi *mobile* dan web. Platform ini menghadirkan solusi *Realtime Database* yang memungkinkan sinkronisasi data secara langsung, mendukung keterhubungan yang dinamis dan pembaruan informasi yang instan [24].

BAB III

METODE PELAKSANAAN

Dalam pelaksanaan proyek akhir ini, penulis melakukan tahapan-tahapan yang bertujuan untuk menyusun secara sistematis yang membuat proyek akhir ini lebih terarah agar memudahkan penulis dalam pembuatan proyek akhir ini. Berikut merupakan tahapan yang disajikan dalam bentuk *flowchart* seperti dibawah ini:



Gambar 3. 1 Diagram Pelaksanaan Proyek Akhir

3.1 Studi Literatur

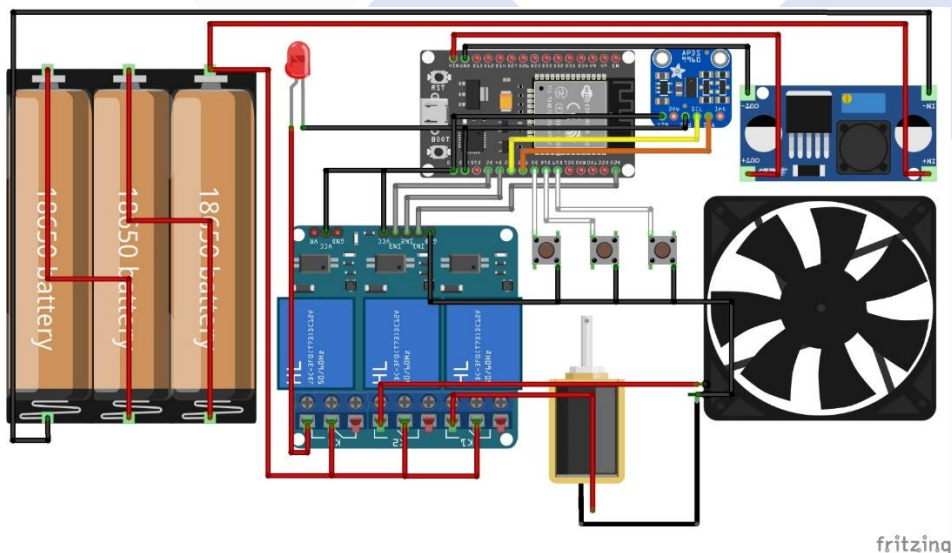
Studi literatur dilakukan untuk mengumpulkan dan menganalisis jurnal serta penelitian terdahulu yang relevan dan valid guna menyusun dasar teori yang kokoh bagi penelitian ini. Langkah-langkah ini mencakup pencarian, pembacaan, dan penyimpulan literatur yang relevan untuk memahami dan mengatasi berbagai permasalahan yang dihadapi.

3.2 Perancangan Sistem

Perancangan *hardware* dan *software* dilakukan dengan merancang rangkaian elektrikal, desain alat, dan use case diagram pada tahap selanjutnya dilakukan pembuatan alat.

3.2.1 Rangkaian Elektrikal

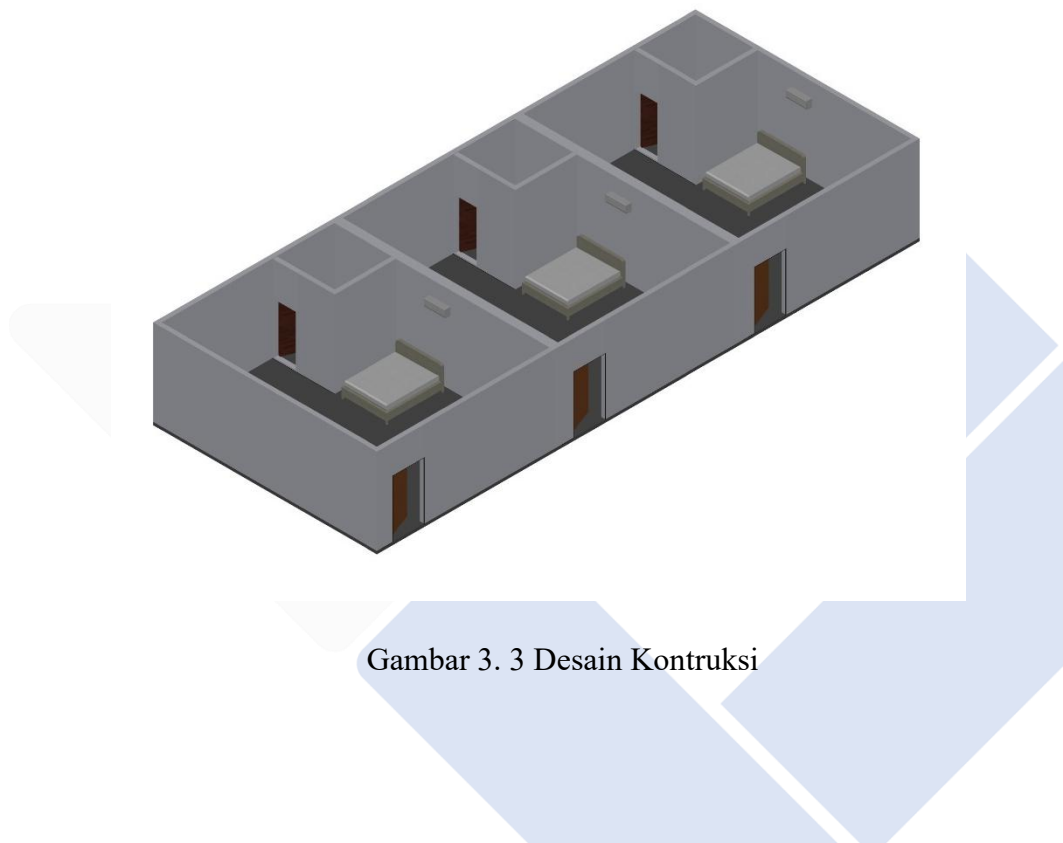
Rangkaian elektrikal dirancang dengan menghubungkan input dan *output* antara komponen yang digunakan, serta disusun menggunakan *software* Fritzing.



Gambar 3. 2 Rangkaian Elektrikal

3.2.2 Desain Alat

Pada tahap perancangan desain mekanikal *hardware*, akan dilakukan pembuatan desain Sistem Cerdas Pengendali *Smartlock* Berbasis Sidik Jari Untuk Keamanan Kamar Hotel. Desain kontruksi alat akan dibuat dengan merancang desain box sistem menggunakan aplikasi triplek kayu untuk meletakkan komponen-komponen yang digunakan.



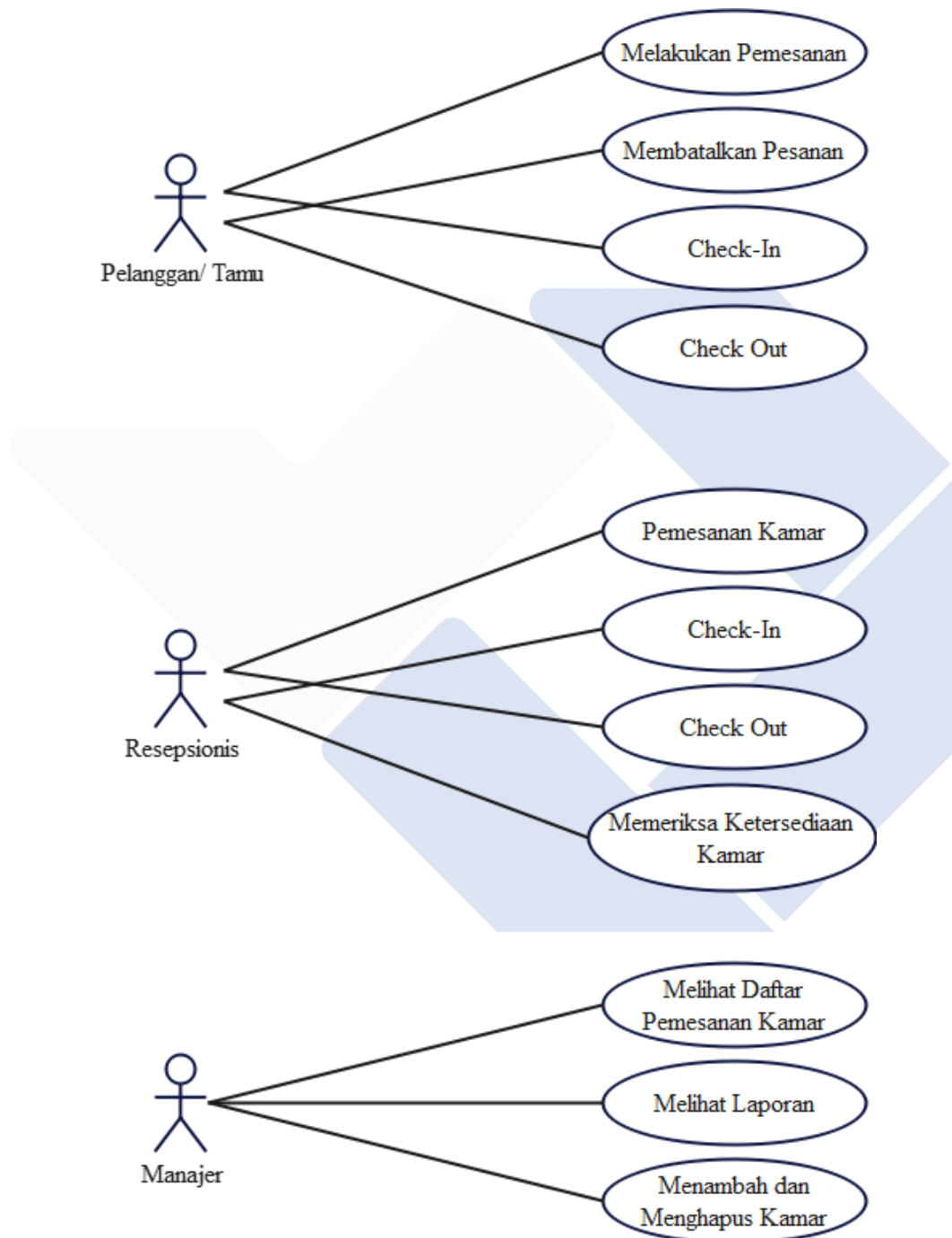
Gambar 3. 3 Desain Kontruksi



Gambar 3. 4 Desain Kontruksi Tampak Depan

3.2.3 Use Case Diagram

Desain Use Case Diagram untuk “Sitem Cerdas Pengendali *Smartlock* Berbasis Sidik Jari Untuk Keamanan Kamar Hotel” dapat dilihat dibawah ini:



Gambar 3. 5 Use Case Diagram Sistem Reservasi Hotel

Diagram use case di atas menunjukkan bagaimana tindakan yang dilakukan oleh aktor sistem:

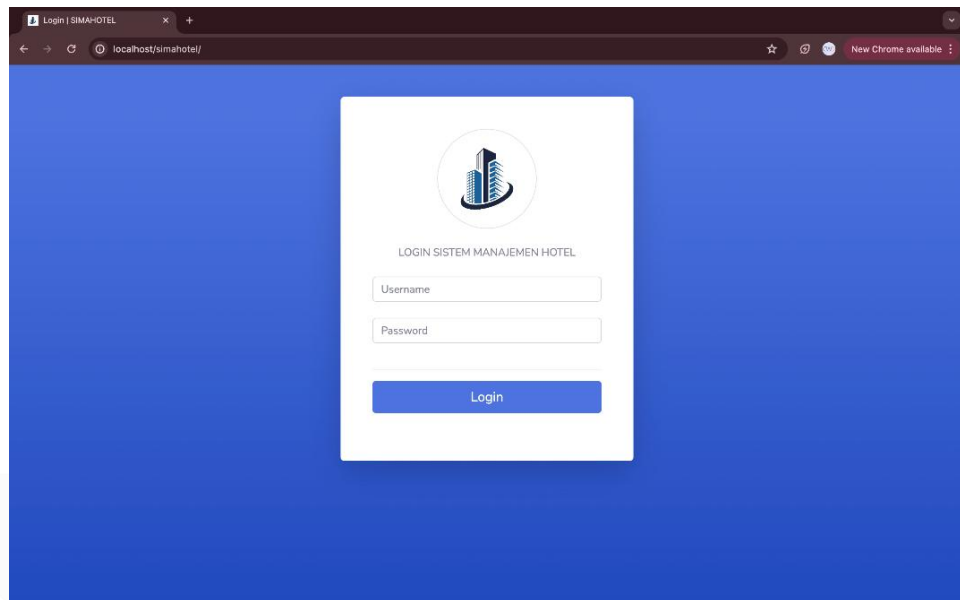
Tabel 3. 1 Tabel Aktor

No.	Aktor	Penjelasan
1.	Pelanggan	Individu yang menggunakan sistem untuk memesan kamar hotel. Pelanggan menggunakan sistem untuk melakukan pemesanan, membatalkan pesanan, <i>check-in</i> , atau <i>check-out</i> jika diperlukan.
2.	Resepsionis	Pegawai hotel yang membantu pelanggan dalam berbagai proses pemesanan dan manajemen kamar. memberikan layanan kepada pelanggan dengan membantu mereka melakukan pemesanan, menangani proses <i>check-in</i> dan <i>check-out</i> , serta memeriksa ketersediaan kamar sesuai permintaan pelanggan.
3.	Manajer	Orang yang bertanggung jawab atas pengelolaan dan pengawasan operasional hotel. Manajer mengelola operasi hotel dengan melihat laporan keterisian kamar, memantau daftar pemesanan, serta menambah atau menghapus kamar dari sistem untuk memastikan operasional hotel berjalan dengan lancar.

3.2.4 Perancangan *Software*

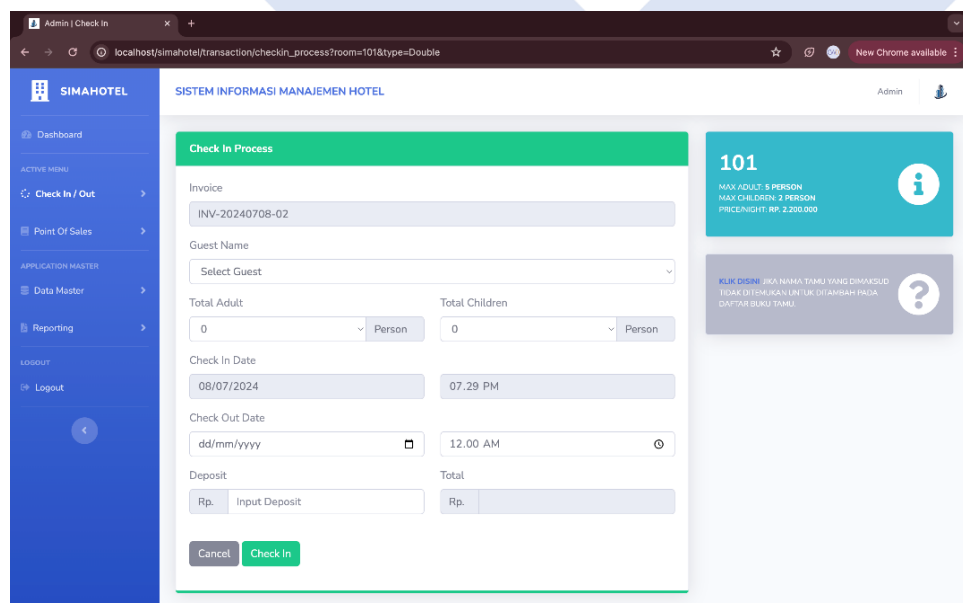
Berikut ini adalah tampilan dari desain perancangan *software* yang sudah dibuat:

1. *Homepage Login*



Gambar 3. 6 *Homepage Login*

2. *Homepage Proses Check-in*



Gambar 3. 7 *Homepage Proses Check-in*

3. Homepage Proses Check-out

Check Out Process

Invoice: INV-20240708-01 Check In Date: 08 July 2024 14:47

Guest Name: Mrs. Jane Smith Check Out Date: 09 July 2024 00:00

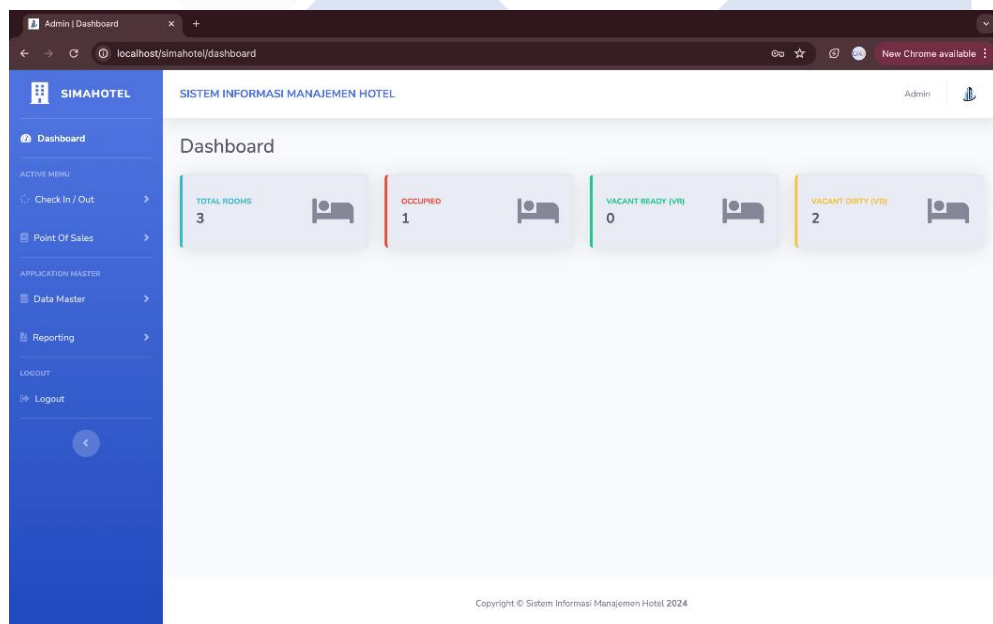
Email: nabilah@gmail.com Phone Number: 0987654321

Invoice Details

Product	Price	Quantity	Total
ROOM RESERVED TYPE : Suite	Rp. 3.000.000	2 Night	Rp. 6.000.000
	Sub-Total		Rp. 6.000.000
	PPN 10%		Rp. 600.000
	Deposit		Rp. 600.000
	Grand Total		Rp. 6.000.000

Gambar 3. 8 Homepage Proses Check-out

4. Homepage Dashboard



Gambar 3. 9 Homepage Dashboard

3.3 Pembuatan dan Perakitan Sistem

Pada tahap ini melibatkan beberapa tahapan penting dalam pembuatannya. Secara umum, tahapan ini dibagi menjadi 2 bagian yaitu pembuatan *hardware* dan pembuatan *software*. Pertama, dalam pembuatan *hardware*, dilakukan pembuatan wadah sistem yang menampung komponen elektronik dan mekanis *Smartlock* serta perakitan komponen seperti sensor sidik jari, motor pengunci, dan modul komunikasi. Selanjutnya, dalam pembuatan *software*, menggunakan Arduino IDE untuk memprogram mikrokontroler agar dapat mengontrol fungsi-fungsi *Smartlock*, seperti pembacaan sidik jari dan pengendalian motor pengunci. Menghubungkan sistem *Smartlock* dengan platform web untuk pengelolaan akses dan monitoring dari jarak jauh, termasuk manajemen pengguna dan pemantauan status kunci. Integrasi antara *hardware* dan *software* bertujuan meningkatkan keamanan dan kenyamanan pengguna hotel melalui teknologi biometrik dan kontrol akses yang cerdas.

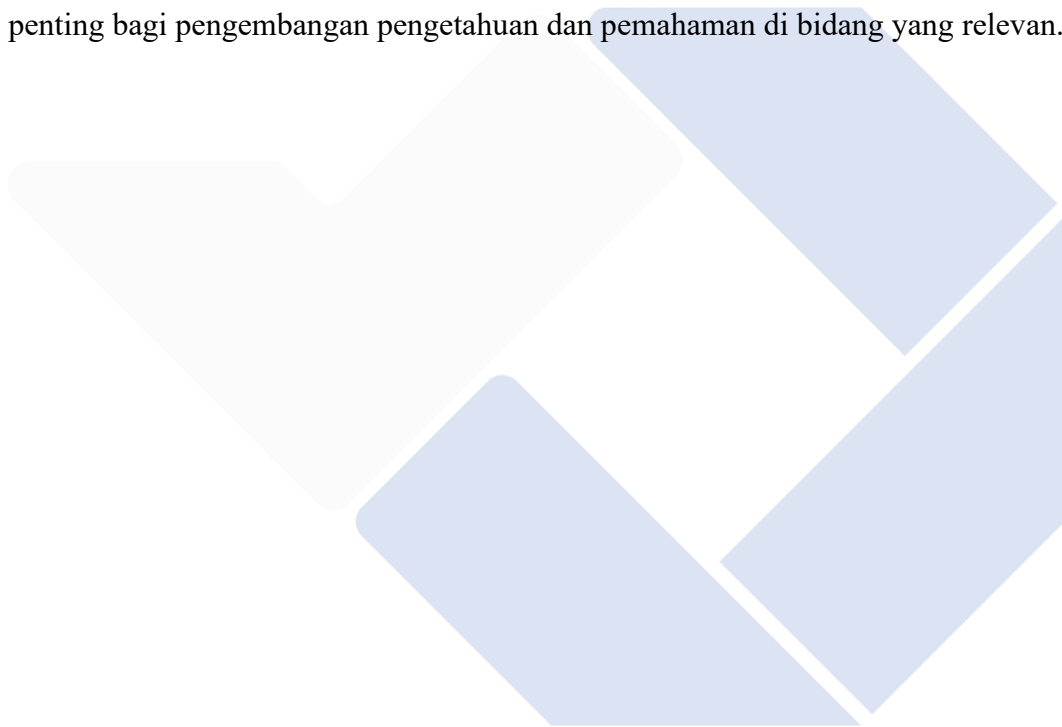
3.4 Pengujian Keseluruhan

Pengujian keseluruhan "Sistem Cerdas Pengendali *Smartlock* Berbasis Sidik Jari Untuk Keamanan Kamar Hotel" bertujuan untuk mengevaluasi performa dan keandalan sistem dalam kondisi operasional yang representatif. Adapun tahapan pengujian yang akan dilakukan yaitu sebagai berikut:

1. Pengujian terhadap fungsionalitas utama sistem, seperti proses registrasi sidik jari, verifikasi sidik jari, pembukaan pintu secara otomatis, dan pengaturan ulang sistem.
2. Pengujian performa untuk menilai respons sistem terhadap berbagai kondisi input yang mungkin terjadi, seperti waktu respons, kecepatan pembukaan pintu, dan keandalan operasional dalam jangka waktu yang panjang.
3. Pengujian untuk memastikan integrasi yang baik antara komponen perangkat keras (seperti sensor sidik jari, LCD) dan perangkat lunak.

3.5 Pembuatan Laporan Proyek Akhir

Tahap penutup dalam penyelesaian proyek akhir ini adalah menyusun laporan akhir. Laporan ini bertujuan untuk merangkum semua aspek yang telah dilakukan selama proses penyelesaian tugas akhir, mencakup latar belakang, batasan masalah, tujuan, dasar teori, metode pelaksanaan, pembahasan, hasil, kesimpulan, dan saran. Selain itu, laporan akhir ini berfungsi sebagai sumber informasi yang bermanfaat bagi para pembaca, diharapkan dapat menjadi referensi berharga untuk studi-studi terkait di masa depan. Dengan demikian, laporan akhir ini tidak hanya mendokumentasikan proses dan hasil proyek, tetapi juga memberikan kontribusi penting bagi pengembangan pengetahuan dan pemahaman di bidang yang relevan.



BAB IV

PEMBAHASAN

Pada bab ini, akan dijelaskan secara rinci mengenai proses pembuatan proyek akhir, mulai dari konstruksi alat, pembuatan sistem kontrol, hingga pengujian yang berkaitan dengan pembuatan sistem. Bab ini juga akan membahas masalah-masalah yang muncul selama pengujian dan pembuatan alat. Setelah pengujian selesai, data yang diperoleh akan dianalisis dan dievaluasi.

4.1 Deskripsi Alat

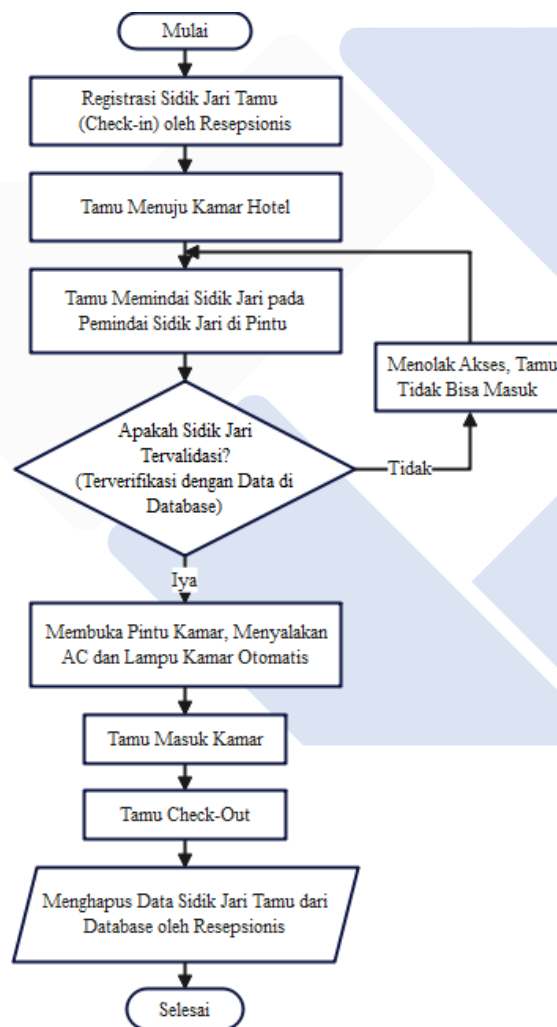
"Sistem Cerdas Pengendali *Smartlock* Berbasis Sidik Jari Untuk Keamanan Kamar Hotel" adalah sistem keamanan yang memanfaatkan teknologi sidik jari untuk mengunci dan membuka pintu kamar hotel. Sistem ini menggantikan kunci fisik atau kartu dengan pemindai sidik jari, yang hanya dapat diakses oleh tamu yang terdaftar. Saat tamu *check-in*, sidik jari mereka direkam dan digunakan untuk mengakses kamar. Sistem ini menawarkan tingkat keamanan yang lebih tinggi dan kenyamanan, karena tamu tidak perlu membawa kunci fisik.

Sistem ini terdiri dari beberapa komponen utama, seperti pemindai sidik jari, modul kontrol elektronik, dan mekanisme kunci elektronik. Pemindai sidik jari membaca dan mengenali sidik jari tamu, modul kontrol elektronik mengolah data yang diterima, dan mekanisme kunci elektronik membuka atau mengunci pintu kamar. Manajer hotel dapat mengelola data sidik jari melalui antarmuka pengguna yang memudahkan registrasi dan penghapusan data sidik jari tamu. Sistem ini juga memungkinkan manajer untuk melakukan audit dan memantau riwayat akses untuk memastikan keamanan.

Selain itu, sistem ini juga dilengkapi dengan fitur otomatis yang dapat menghidupkan AC dan lampu kamar ketika tamu sudah *check-in* dan bisa mengakses kamar. Ketika sidik jari tamu berhasil diverifikasi dan pintu kamar terbuka, sistem akan secara otomatis menyalakan AC dan lampu, memastikan

kamar berada dalam kondisi nyaman saat tamu masuk. Fitur ini tidak hanya meningkatkan kenyamanan tamu, tetapi juga membantu dalam penghematan energi dengan memastikan bahwa perangkat listrik hanya aktif saat kamar sedang digunakan.

Dengan menggunakan sistem ini, proses *check-in* dan *check-out* menjadi lebih efisien, mengurangi risiko kehilangan kunci, dan memberikan pengalaman yang lebih aman dan nyaman bagi tamu hotel. *Flowchart* sistem secara keseluruhan dapat dilihat pada Gambar 4.1.



Gambar 4. 1 *Flowchart* Sistem Kerja Alat

4.2 Pembuatan *Hardware*

Dalam tahap pembuatan *hardware*, alat dibuat berdasarkan desain yang telah disusun pada tahap sebelumnya. Pembuatan *hardware* ini terbagi menjadi 2 bagian, yaitu pembuatan *hardware* secara mekanik dan secara elektrik.

4.2.1 Pembuatan *Hardware* Secara Mekanik

Proses pembuatan *hardware* secara mekanik adalah pembuatan model fisik sistem yang sesuai dengan desain yang telah dirancang sebelumnya. Tahap awalnya melibatkan pemilihan bahan tripleks yang sesuai dengan kebutuhan struktural dan estetika. Setelah itu, dilakukan pemotongan dan perakitan tripleks sesuai dengan desain yang telah direncanakan sebelumnya. Pada setiap pintu kamar, terintegrasi sistem *fingerprint* dan LCD yang dipasang secara presisi untuk memastikan keamanan dan kemudahan akses bagi penghuni kamar. Pada setiap kamar juga dapat secara otomatis menghidupkan AC dan lampu pada saat pintu kamar terbuka. Proses ini melibatkan penggunaan alat-alat pemotongan dan pemasangan yang tepat, serta penyesuaian dimensi untuk menciptakan kerangka yang akurat dan fungsional sesuai dengan spesifikasi yang ditetapkan.

Hasil pembuatan protipe sistem dapat dilihat pada Gambar 4.2 dan 4.3.



Gambar 4. 2 Hasil Pembuatan Alat Tampak Depan Secara Mekanik



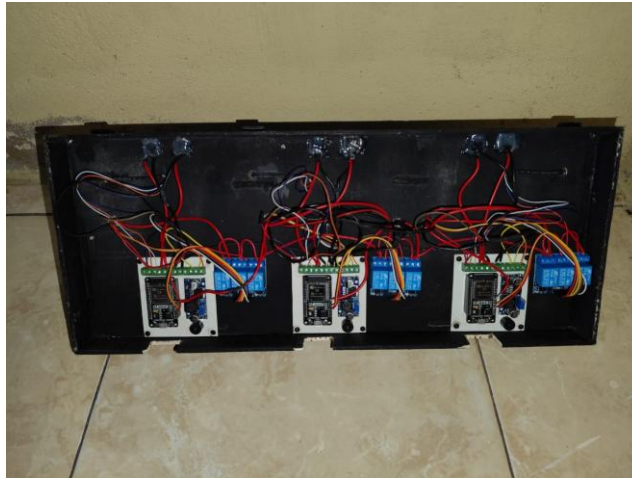
Gambar 4. 3 Hasil Pembuatan Alat Tampak Belakang Secara Mekanik

4.2.2 Pembuatan *Hardware* Secara Elektrikal

Pada tahap ini, skema rangkaian elektrikal disusun sesuai dengan desain dirancang sebelumnya. Skema rangkaian tersebut dibuat menggunakan *software* Fritzing dan KiCad, lalu desainnya dicetak pada papan sirkuit PCB. Setiap komponen yang digunakan dihubungkan secara sistematis sesuai jalur dan koneksi yang telah dirancang dalam desain. Komponen-komponen yang disambungkan pada papan PCB yaitu NodeMCU ESP 32, relay 3 *channel* dan modul *step down*. Berikut adalah gambar hasil pembuatan *hardware* secara elektrikal dapat dilihat pada Gambar 4.3.



Gambar 4. 4 Pembuatan *Hardware* Secara Elektrikal



Gambar 4. 5 Pembuatan *Hardware* Secara Elektrikal Tampak Keseluruhan

4.3 Perancangan *Database*

Pada tahapan perancangan “Sistem Cerdas Pengendali *Smartlock* Berbasis Sidik Jari Untuk Keamanan Kamar Hotel” dengan menggunakan *Structure Query Language*, kita dapat mengelola dan mengakses *database*. Adapun tabel yang digunakan yaitu sebagai berikut :

4.3.1 Tabel *Categories*

Tabel data *categories* dalam database digunakan untuk menyimpan berbagai kategori makanan. Berikut ini adalah struktur tabel kategori:

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
☐ 1	category_id	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
☐ 2	category_name	varchar(50)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
☐ 3	description	text	utf8mb4_general_ci		Ya	NULL			Ubah Hapus Lainnya

Gambar 4. 6 Tabel *Categories*

4.3.2 Tabel *Email_Check_In*

Tabel data *check in* pada database digunakan untuk menyimpan data email tamu yang masuk. Adapun tabel dibawah ini adalah struktur tabel email *check in*:

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
☐ 1	id	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
☐ 2	check_in_id	int(11)			Tidak	Tidak ada			Ubah Hapus Lainnya
☐ 3	email	varchar(20)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
☐ 4	email_status	enum('Sent', 'Failed')	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
☐ 5	pin	varchar(6)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
☐ 6	created_at	timestamp			Tidak	current_timestamp()			Ubah Hapus Lainnya

Gambar 4. 7 Tabel *Email_Check_In*

4.3.3 Tabel *Guests*

Tabel data *guests* pada database digunakan untuk menyimpan informasi mengenai tamu yang ada dihotel. Berikut adalah struktur tabel *guests* yang ditampilkan dalam tabel dibawah ini:

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
☐ 1	guest_id	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
☐ 2	name	varchar(100)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
☐ 3	id_type	varchar(50)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
☐ 4	id_number	varchar(50)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
☐ 5	country	varchar(50)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
☐ 6	address	text	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
☐ 7	phone	varchar(20)	utf8mb4_general_ci	Ya		NULL			Ubah Hapus Lainnya
☐ 8	email	varchar(100)	utf8mb4_general_ci	Ya		NULL			Ubah Hapus Lainnya
☐ 9	fingerprint	varchar(20)	utf8mb4_general_ci	Ya		NULL			Ubah Hapus Lainnya

Gambar 4. 8 Tabel *Guests*

4.3.4 Tabel *Invoice*

Tabel data *invoice* pada database digunakan untuk menyimpan data pembayaran tamu yang masuk. Berikut adalah struktur tabel *invoice* pada tabel dibawah ini:

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
☐ 1	invoice_id	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
☐ 2	invoice_number	varchar(30)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
☐ 3	check_in_id	int(11)			Tidak	Tidak ada			Ubah Hapus Lainnya
☐ 4	status	enum('Pending', 'Rejected', 'Completed')	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya

Gambar 4. 9 Tabel *Invoice*

4.3.5 Tabel Invoice_Pos_Relation

Tabel *invoice_pos_relation* pada database digunakan untuk mengerelasikan tabel *pos_transactions* dan tabel *invoice*. Adapun tabel dibawah ini adalah struktur tabel *invoice_pos_relation*:

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
☐ 1	id 🔑	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
☐ 2	invoice_id 🔑	int(11)			Ya	NULL			Ubah Hapus Lainnya
☐ 3	transaction_id 🔑	int(11)			Ya	NULL			Ubah Hapus Lainnya

Gambar 4. 10 Tabel *Invoice_Pos_Relation*

4.3.6 Tabel Point_Of_Sales

Tabel data *point_of_sales* pada database digunakan untuk menyimpan data penjualan makanan dan minuman yang dijual di dalam hotel. Tabel dibawah ini adalah struktur dari tabel *point_of_sales*:

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
☐ 1	pos_id 🔑	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
☐ 2	point_of_sales	varchar(100)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
☐ 3	category_id 🔑	int(11)			Tidak	Tidak ada			Ubah Hapus Lainnya
☐ 4	price	decimal(10,0)			Tidak	Tidak ada			Ubah Hapus Lainnya
☐ 5	unit	varchar(50)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya

Gambar 4. 11 Tabel *Point_Of_Sales*

4.3.7 Tabel Pos_Transactions

Tabel data *pos_transactions* pada database digunakan untuk menyimpan data transaksi yang masuk dan total transaksi. Berikut ini adalah struktur dari tabel *pos_transactions*:

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
☐ 1	transaction_id	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
☐ 2	guest_id	int(11)			Tidak	Tidak ada			Ubah Hapus Lainnya
☐ 3	pos_id	int(11)			Tidak	Tidak ada			Ubah Hapus Lainnya
☐ 4	quantity	int(11)			Tidak	Tidak ada			Ubah Hapus Lainnya
☐ 5	transaction_date	datetime			Tidak	Tidak ada			Ubah Hapus Lainnya
☐ 6	total_amount	decimal(10,0)			Tidak	Tidak ada			Ubah Hapus Lainnya

Gambar 4. 12 Tabel *Pos_Transactions*

4.3.8 Tabel *Rooms*

Tabel *rooms* pada database digunakan untuk menyimpan dan menampilkan data mengenai kamar yg telah dipesan oleh tamu. Adapun tabel dibawah ini adalah struktur dari tabel *rooms*:

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
☐ 1	room_id	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
☐ 2	room_number	varchar(10)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
☐ 3	room_type_id	int(11)			Tidak	Tidak ada			Ubah Hapus Lainnya
☐ 4	max_adult	int(11)			Tidak	Tidak ada			Ubah Hapus Lainnya
☐ 5	max_children	int(11)			Tidak	Tidak ada			Ubah Hapus Lainnya
☐ 6	room_status	enum('Available', 'Occupied', 'Maintenance')	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
☐ 7	price_per_night	decimal(10,0)			Tidak	Tidak ada			Ubah Hapus Lainnya

Gambar 4. 13 Tabel *Rooms*

4.3.9 Tabel *Room_Service*

Tabel *room_service* pada database digunakan untuk menyimpan dan menampilkan data mengenai pelayanan untuk kamar yg dipesan. Dibawah ini adalah struktur dari tabel *room_service*:

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
☐ 1	id	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
☐ 2	room_id	int(11)			Tidak	Tidak ada			Ubah Hapus Lainnya
☐ 3	user_id	int(11)			Tidak	Tidak ada			Ubah Hapus Lainnya
☐ 4	status	enum('on progress', 'completed')	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
☐ 5	created_at	datetime			Tidak	current_timestamp()			Ubah Hapus Lainnya

Gambar 4. 14 Tabel *Room_Service*

4.3.10 Tabel Room_Type

Tabel data *room_type* pada database digunakan untuk menyimpan dan menampilkan data mengenai tipe kamar yang ada pada hotel. Adapun tabel dibawah ini adalah struktur tabel *room_type*:

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
☐	1 room_type_id 🔑	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
☐	2 room_type_name 🗣️	varchar(50)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
☐	3 price_per_night	decimal(10,2)			Tidak	Tidak ada			Ubah Hapus Lainnya
☐	4 price_per_person	decimal(10,2)		Ya		NULL			Ubah Hapus Lainnya
☐	5 description	text	utf8mb4_general_ci	Ya		NULL			Ubah Hapus Lainnya

Gambar 4. 15 Tabel *Room_Type*

4.3.11 Tabel Users

Tabel data users pada database digunakan untuk menyimpan data user dan admin. Adapun tabel dibawah ini adalah struktur tabel *users*:

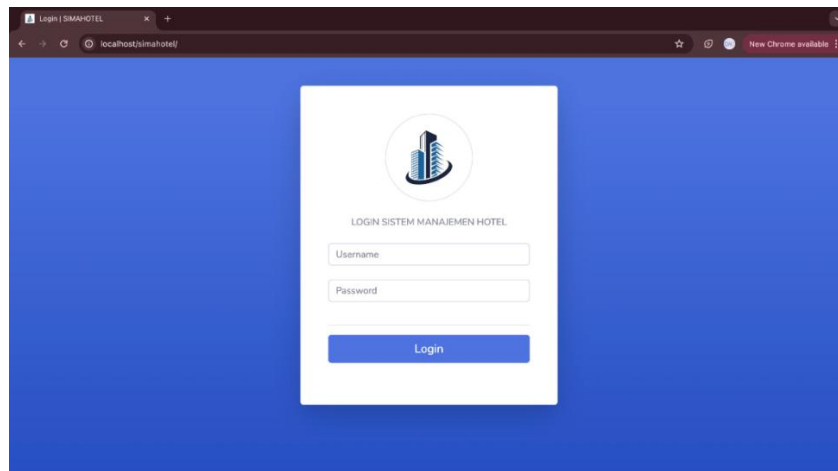
#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
☐	1 user_id 🔑	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
☐	2 name	varchar(100)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
☐	3 username 🗣️	varchar(50)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
☐	4 password	varchar(255)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
☐	5 role	varchar(50)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
☐	6 status	enum("active", "inactive")	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
☐	7 phone	varchar(20)	utf8mb4_general_ci	Ya		NULL			Ubah Hapus Lainnya
☐	8 email 📧	varchar(100)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
☐	9 session_token	varchar(255)	utf8mb4_general_ci	Ya		NULL			Ubah Hapus Lainnya

Gambar 4. 16 Tabel *Users*

4.4 Tampilan Antarmuka

4.4.1 Homepage Login

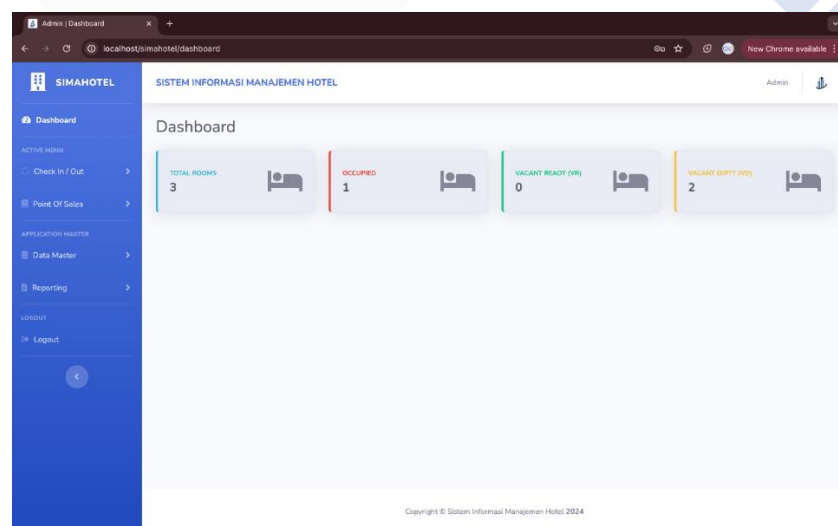
Berikut ini merupakan tampilan homepage login Sistem Manajemen Hotel, halaman *login* tersebut berfungsi untuk mengakses ke menu admin dengan memasukkan *username* dan *password*.



Gambar 4. 17 *Homepage Login*

4.4.2 *Homepage Dashbord Admin*

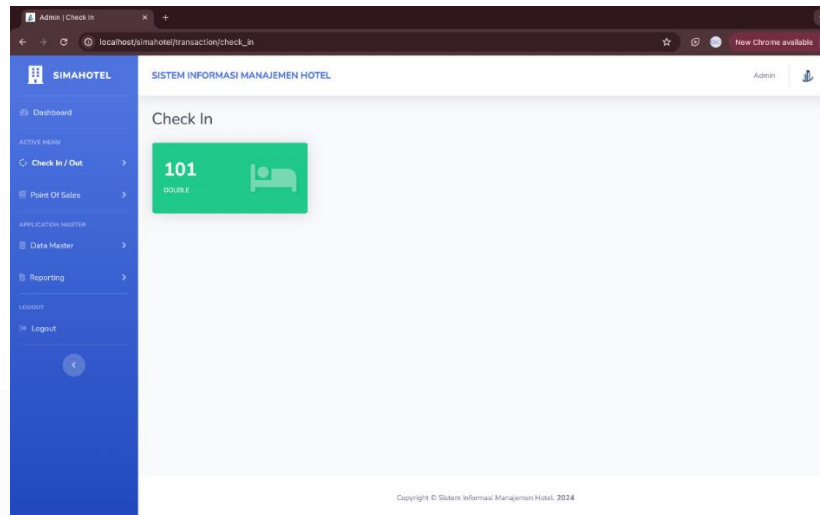
Gambar dibawah ini merupakan bentuk tampilan *homepage dashboard* admin, pada halaman ini dapat melihat



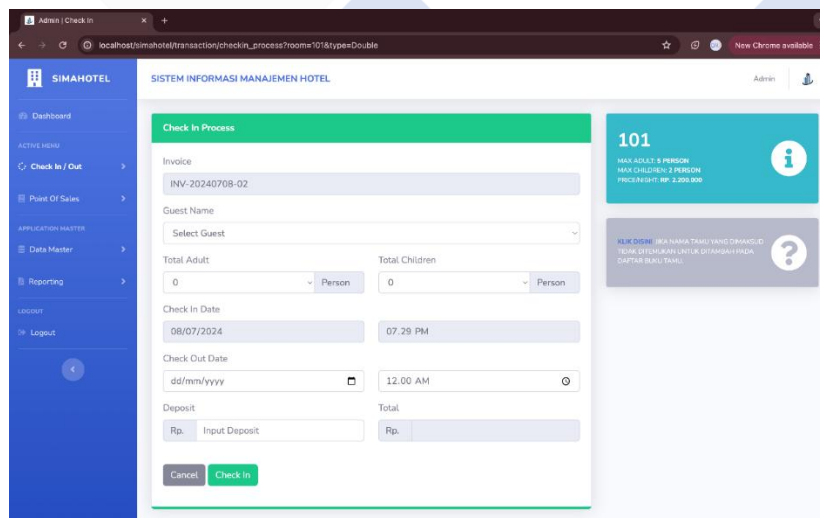
Gambar 4. 18 *Homepage Dashbord Admin*

4.4.3 Homepage Check-in

Gambar dibawah ini merupakan bentuk tampilan dari *homepage check in*. Pada halaman ini berfungsi untuk mengisi data diri calon pelanggan hotel.



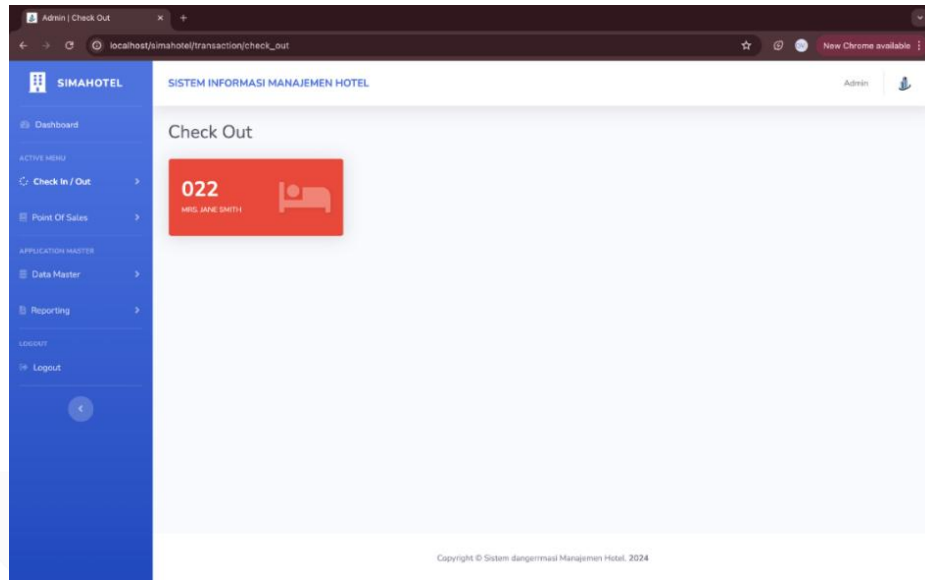
Gambar 4. 19 Homepage Check-in



Gambar 4. 20 Homepage Check-in Process

4.4.4 Homepage Check Out

Gambar dibawah ini menunjukkan *homepage check out* yang berfungsi untuk pengisian data pelanggan pada saat *check out* dan pengecekan detail tagihan hotel.



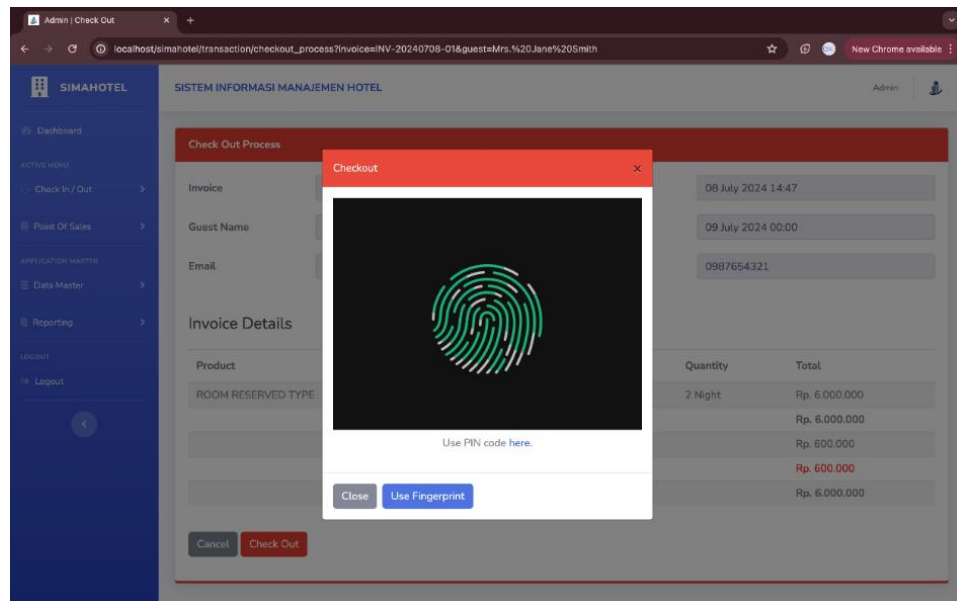
Gambar 4. 21 *Homepage Check-Out*

A screenshot of the 'Check Out Process' page in the SIMAHOTEL application. The browser address bar shows a URL with invoice and guest details. The page features a red header 'Check Out Process' and a form with the following fields: Invoice (INV-20240708-01), Check In Date (08 July 2024 14:47), Guest Name (Mrs. Jane Smith), Check Out Date (09 July 2024 00:00), Email (nabilah@gmail.com), and Phone Number (0987654321). Below the form is an 'Invoice Details' table.

Product	Price	Quantity	Total
ROOM RESERVED TYPE : Suite	Rp. 3.000.000	2 Night	Rp. 6.000.000
	Sub-Total		Rp. 6.000.000
	PPn 10%		Rp. 600.000
	Deposit		Rp. 600.000
	Grand Total		Rp. 6.000.000

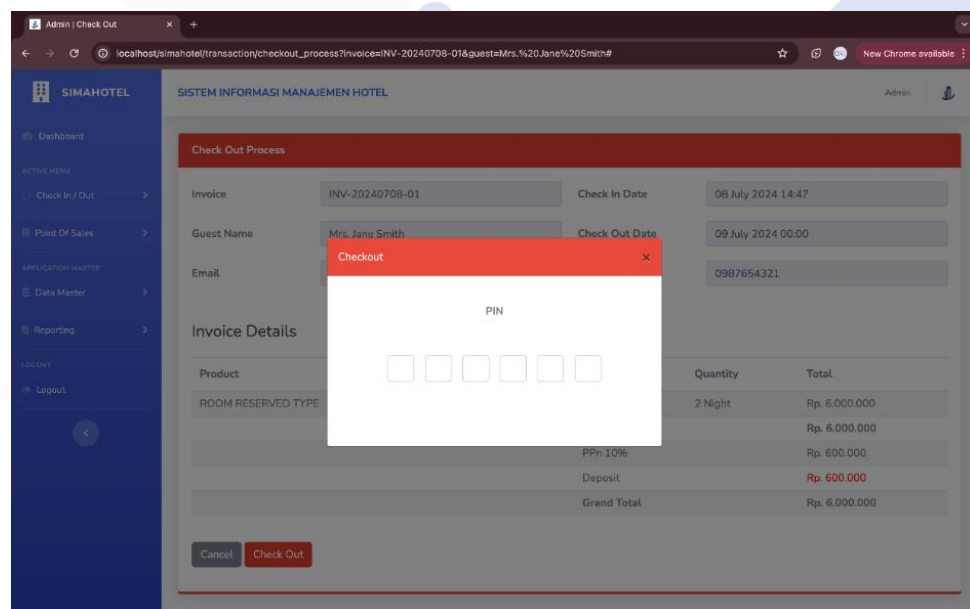
At the bottom of the form, there are two buttons: 'Cancel' and 'Check Out'.

Gambar 4. 22 *Homepage Check-Out Process*



Gambar 4. 23 *Homepage Check-Out Sidik Jari*

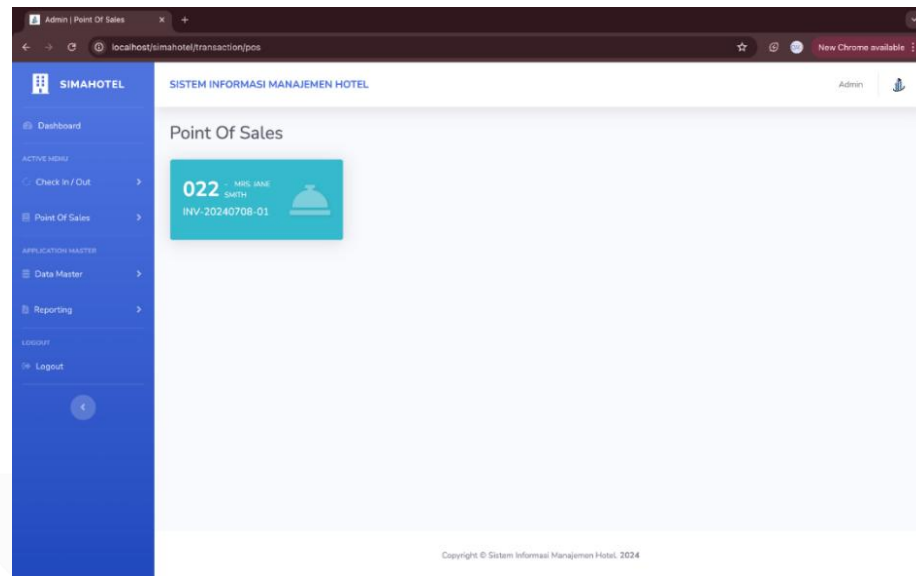
Jika *fingerprint* tidak bisa digunakan atau *eror* bisa menggunakan pin yang sudah dikirim ke email pelanggan.



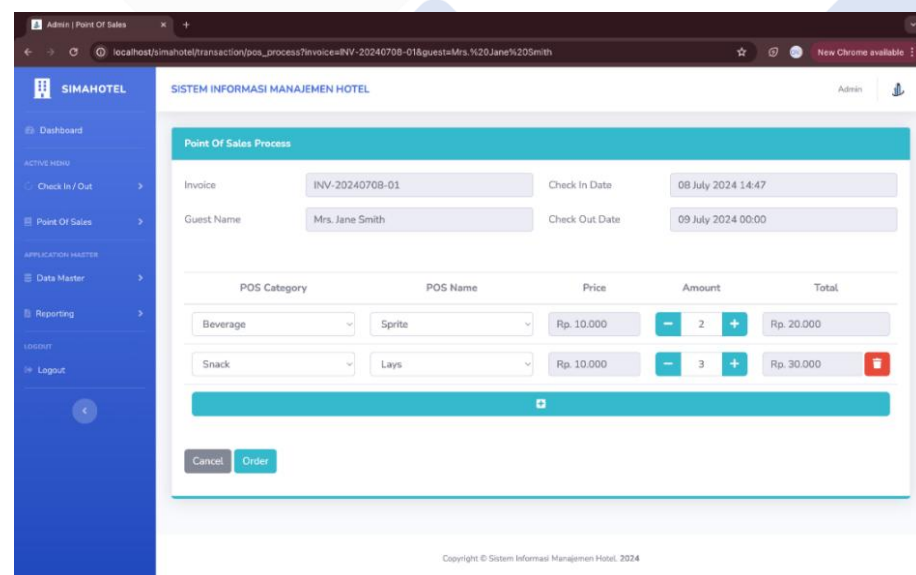
Gambar 4. 24 *Homepage Check Out Pin*

4.4.5 Homepage Point Of Sale

Berikut adalah *homepage* untuk memesan makanan didalam hotel. pada halaman ini pelanggan dapat memesan makanan dan minuman yang tersedia di dalam hotel.



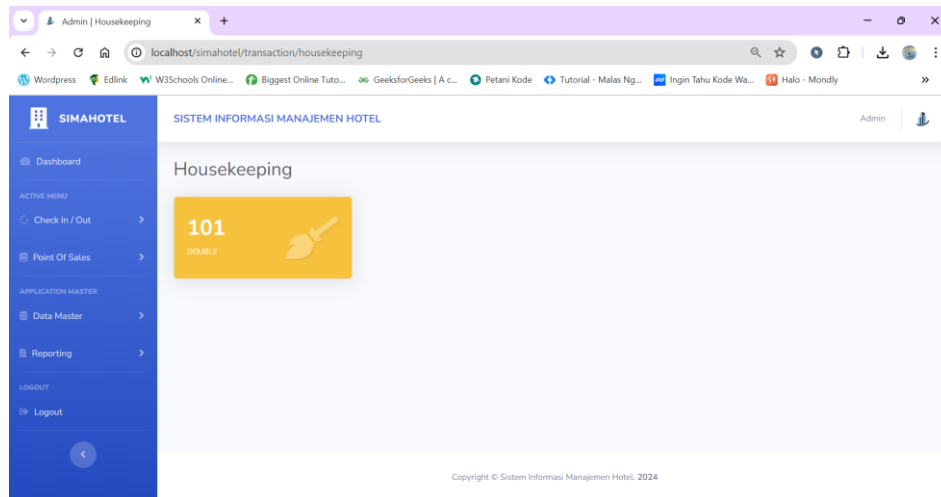
Gambar 4. 25 *Homepage Point Of Sales*



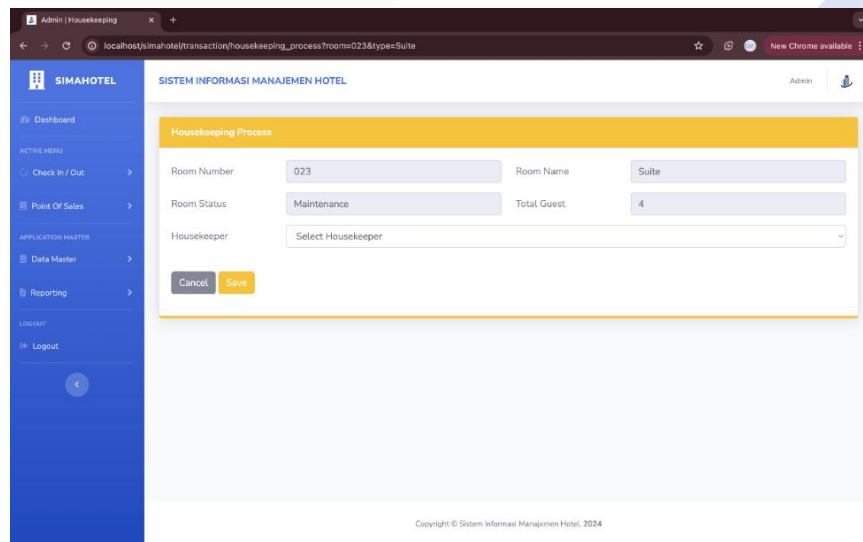
Gambar 4. 26 *Homepage Point Of Sales Process*

4.4.6 Homepage Housekeeping

Berikut adalah *homepage housekeeping* digunakan ketika ada kamar yang sudah dipakai pelanggan dan pelanggan tersebut sudah *check out* lalu *housekeeper* atau staf hotel akan membersihkan kamar yang sudah diapakai.



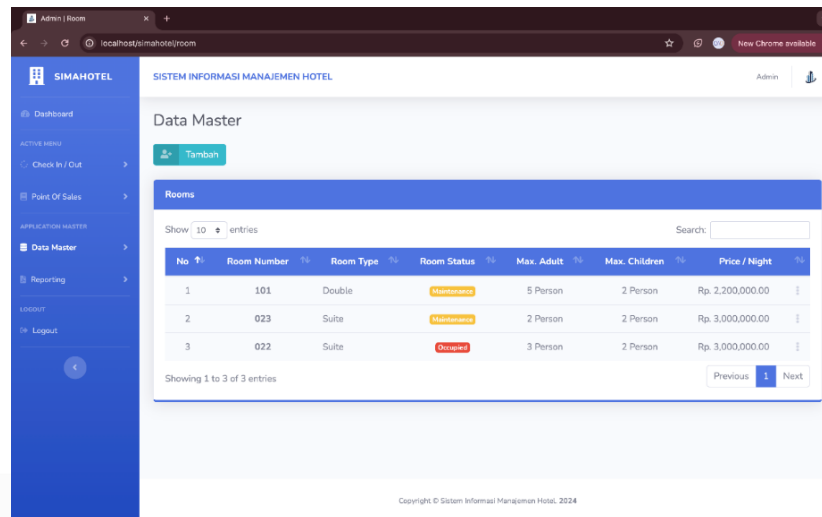
Gambar 4. 27 *Homepage Housekeeping*



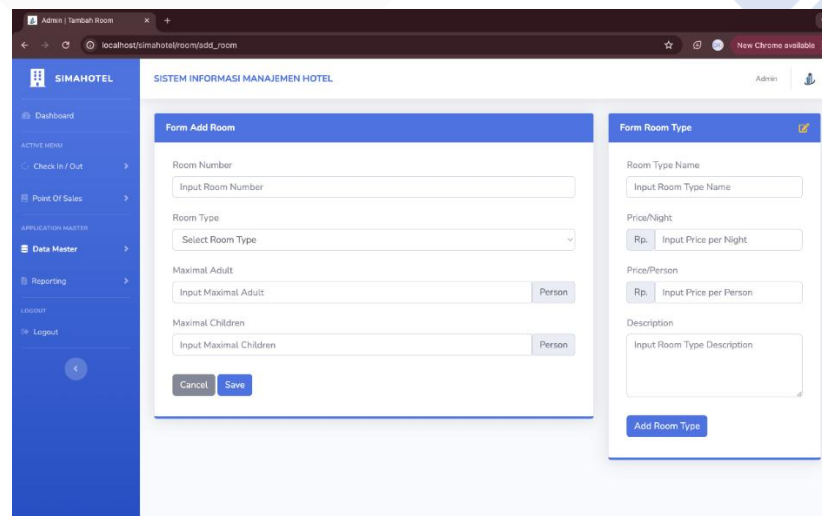
Gambar 4. 28 *Homepage Housekeeping Process*

4.4.7 Homepage Rooms di Data Master

Dibawah ini adalah *homepage* data master *rooms* untuk menambahkan kamar di dalam hotel dan mengedit data kamarnya.



Gambar 4. 29 *Homepage Data Master Rooms*



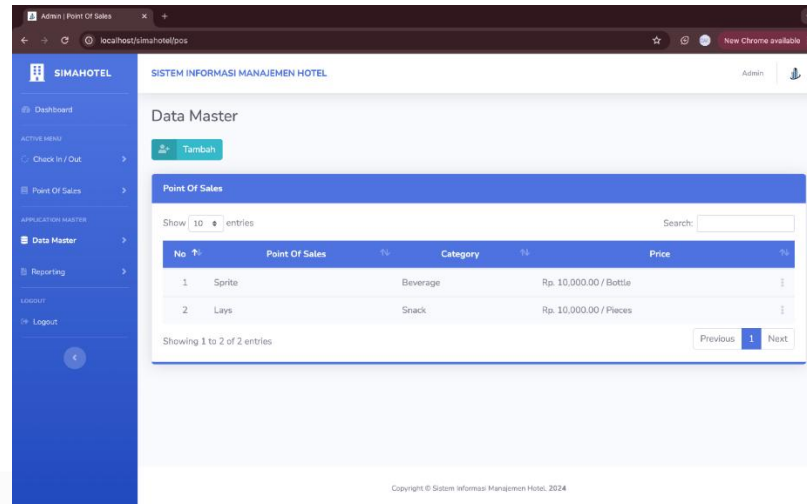
Gambar 4. 30 *Homepage Data Master Add Rooms*

The screenshot displays the SIMAHOTEL web application interface. On the left is a blue sidebar menu with the following items: Dashboard, Active Menu (with sub-items: Check In / Out, Point Of Sales), Application Master (with sub-items: Data Master, Reporting), and User (with sub-item: Logout). The main content area is titled 'SISTEM INFORMASI MANAJEMEN HOTEL' and contains two forms. The 'Form Add Room' form on the left includes fields for 'Room Number' (text input), 'Room Type' (dropdown menu), 'Maximal Adult' (text input with a 'Person' button), and 'Maximal Children' (text input with a 'Person' button). It has 'Cancel' and 'Save' buttons at the bottom. The 'Form Edit Room Type' form on the right includes a 'Select Room Type' dropdown, 'Room Type Name' (text input), 'Price/Night' (text input with a 'Rp.' prefix and a 'Person' button), 'Price/Person' (text input with a 'Rp.' prefix and a 'Person' button), and a 'Description' (text area). It has an 'Edit Room Type' button at the bottom.

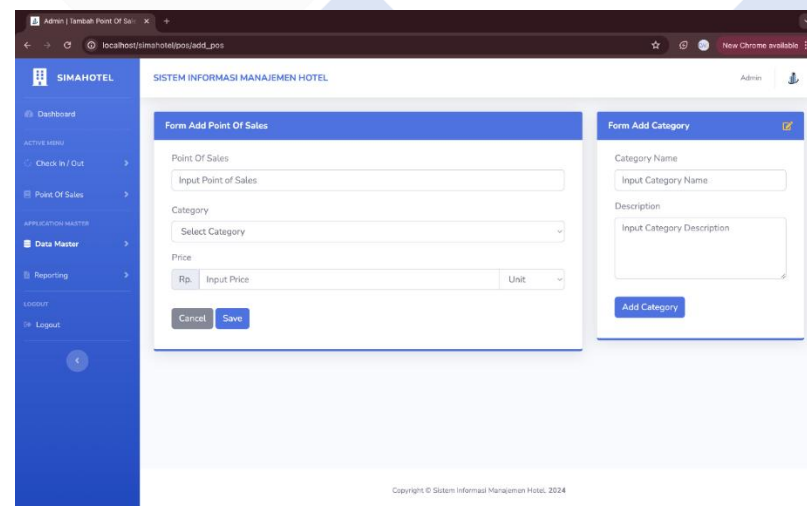
Gambar 4. 31 *Homepage Data Master Edit Rooms*

4.4.8 Homepage Point Of Sale di Data Master

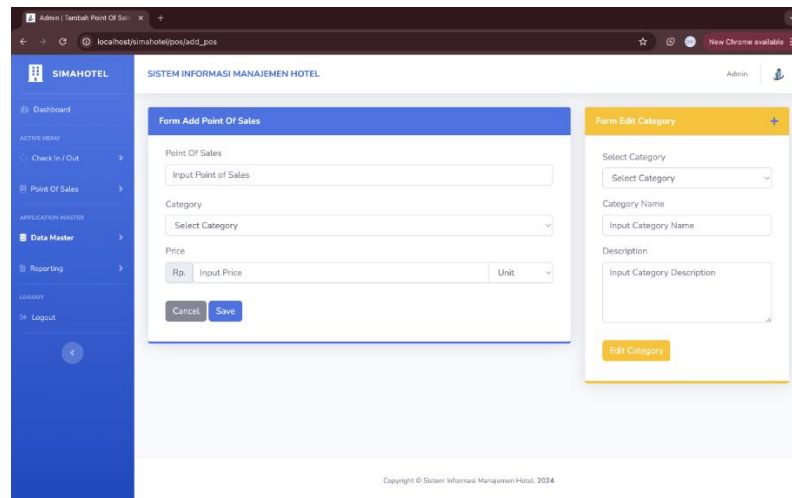
Berikut adalah *homepage* data master *point of sales* untuk menambahkan data makanan dan minuman untuk dijual di dalam hotel.



Gambar 4. 32 *Homepage Data Master Point Of Sales*



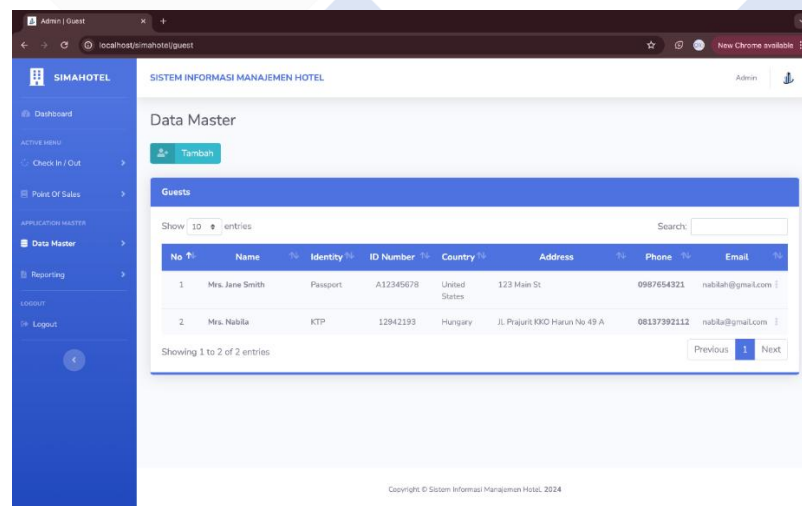
Gambar 4. 33 *Homepage Data Master Add Point Of Sales*



Gambar 4. 34 *Homepage Data Master Edit Point Of Sales*

4.4.9 Homepage Guests di Data Master

Berikut adalah *homepage* data master *guests* untuk menambahkan data pelanggan yang akan *check in* di hotel.



Gambar 4. 35 *Homepage Data Master Guest*

Form Add Guest

Guest Name

Identity

Address

Country

Phone

Email

Gambar 4. 36 Homepage Data Master Add Guests

4.4.10 Homepage Users di Data Master

Berikut adalah gambar *homepage* data master *users* untuk menambahkan data karyawan yang bekerja di hotel.

Data Master

[Tambah](#)

Users

Show 10 entries

No	Name	Username	Role	Status	Phone	Email
1	Admin	admin	Admin	Active	1234567890	admin@admin.com
2	Nabilla	nabila	Room Service	Active	08137392112	nabila@gmail.com
3	Nabil	nabil	Receptionist	Inactive	08137392112	nabil@gmail.com

Showing 1 to 3 of 3 entries

Previous 1 Next

Copyright © Sistem Informasi Manajemen Hotel, 2024

Gambar 4. 37 Homepage Data Master Users

Gambar 4. 38 Homepage Data Master Add Users

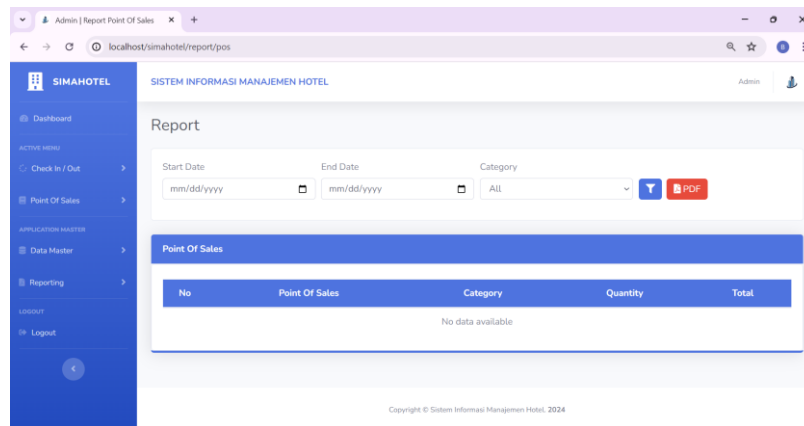
4.4.11 Homepage Check In di Reporting

Berikut adalah *homepage reporting check in* untuk melihat dan merekap seluruh data-data pelanggan yang *check in* ke hotel.

Gambar 4. 39 Homepage Reporting Check-in

4.4.12 Homepage Point Of Sales di Reporting

Berikut adalah *homepage reporting point of sales* untuk melihat dan merekap seluruh data-data pelanggan yang membeli makanan dan minuman di hotel.



Gambar 4. 40 *Homepage Reporting Point Of Sales*

4.5 Pengujian

Pada tahap ini peneliti melakukan pengujian yang bertujuan untuk menganalisa apakah sensor sidik jari pada alat yang diteliti untuk membuka pintu, menyalakan kipas dan lampu sudah berfungsi secara optimal dan sesuai dengan standar yang ditetapkan. Pengujian ini dilakukan pada 3 kamar *prototype* pengendali pintu kamar hotel. Peneliti melakukan pengujian dengan cara menguji sensor *fingerprint* ketika sidik jari yang telah didaftar di salah satu kamar dan dapat membuka kamar serta menyalakan kipas dan lampu. Pada pengujian ini terdapat batas mendaftarkan sidik jari yaitu hanya 5 sidik jari yang bisa terdaftar di setiap kamarnya. Ketika sudah 5 sidik jari yang terdaftar maka sensor tidak dapat menambahkan atau mendaftarkan lagi sidik jari selanjutnya otomatis pintu tidak akan terbuka karena sidik jari belum terdaftar. Oleh karena itu peneliti mempunyai prosedur pengujian yang sama maka data hasil pengujian dari 3 kamar akan ditampilkan sebagai berikut.

4.5.1 Pengujian Sidik Jari Saat Pendaftaran Kamar

Pengujian ini bertujuan untuk menganalisa sensor sidik jari jika didaftarkan sidik jari dan tidak didaftarkan sidik jari. Prosedur pengujian dilakukan dengan mendaftarkan 2 sidik jari dan 1 sidik jari yang tidak didaftarkan untuk kamar 1 dan 3 dan untuk kamar 2 didaftarkan semua. Sensor *fingerprint* yang digunakan disetiap

kamarnya hanya 1 sensor. Oleh karena itu, pada pengujian ini ada 3 sidik jari dan 2 didaftarkan disetiap kamar hotel dan 1 sidik jari lagi tidak didaftarkan untuk mendapatkan variasi data pembacaan sidik jari oleh *fingerprint*. Kemudian dilakukan pengamatan terhadap pembacaan data sensor *fingerprint* aktif atau tidak. Hasil pengujian ini kemudian dicatat pada Tabel 4. 1 untuk 3 kamar.

Tabel 4. 1 Hasil Pengujian Sidik Jari pada setiap kamar

Kamar	Users	Status Jari	Respon Sensor	Persentase
1	Iche	Terdaftar	Valid	100%
	Zikri	Terdaftar	Valid	100%
	Melky	Tidak Terdaftar	Tidak Valid	0%
2	Andira	Terdaftar	Valid	100%
	Rahmi	Terdaftar	Valid	100%
	Vadillah	Terdaftar	Valid	100%
3	Zainul	Tidak Terdaftar	Tidak Valid	0%
	Nabila	Terdaftar	Valid	100%
	Adysha	Terdaftar	Valid	100%

Dari hasil pengujian berikut, kami menguji beberapa jari, jari yang sudah terdaftar dalam 1 kamar dengan jari yang belum terdaftar apakah sensor valid atau tidak mendeteksi jari yang terdaftar.

4.5.2 Pengujian Sensor

Pengujian ini bertujuan untuk menganalisa sensor sidik jari dengan membandingkan antara sidik jari yang terdaftar dan tidak daftar. Prosedur pengujian dilakukan dengan mendaftarkan 3 sidik jari di dua kamar, sementara satu kamar tidak didaftarkan sidik jari sama sekali. Setiap kamar menggunakan satu sensor

sisik jari. Dengan demikian, dalam pengujian ini terdapat 3 sidik jari yang terdaftar di dua kamar hotel, dan satu kamar tanpa sidik jari yang terdaftar, guna mendapatkan variasi data pembacaan sidik jari oleh *fingerprint*. Selanjutnya, dilakukan pengamatan terhadap pembacaan data sensor *fingerprint* untuk melihat apakah sensor tersebut aktif atau tidak dan berapa lama waktu terdeteksinya. Hasil pengujian ini kemudian dicatat pada Tabel 4.2 untuk 3 kamar.

Tabel 4. 2 Hasil pengujian sensor

Kamar	Users	Jari	Status	Waktu
1	1	Jari Jempol	Terdeteksi	1,5 Detik
	2	Jari Telunjuk	Terdeteksi	1,5 Detik
	3	Jari Tengah	Terdeteksi	1,5 Detik
2	1	Jari Jempol	Tidak Terdeteksi	-
	2	Jari Telunjuk	Tidak Terdeteksi	-
	3	Jari Tengah	Tidak Terdeteksi	-
3	1	Jari Jempol	Terdeteksi	1,5 Detik
	2	Jari Telunjuk	Terdeteksi	1,5 Detik
	3	Jari Tengah	Terdeteksi	1,5 Detik

Dari hasil pengujian berikut, kami menguji beberapa jari orang yang terdaftar dan dapat mengakses 1 kamar. Dan pengujian ini dilakukan berulang-ulang sampai 3 kali untuk memastikan apakah sensor berjalan sangat baik atau tidak.

4.5.3 Pengujian Lampu, Kipas dan Selenoid

Pengujian ini bertujuan untuk menganalisa kinerja sensor sidik jari dengan membandingkan kondisi saat jari daftar dan tidak daftar. Prosedur pengujian dilakukan dengan mendaftarkan 1 sidik jari di setiap kamarnya. Sensor *fingerprint*

yang digunakan disetiap kamarnya hanya 1 sensor. Oleh karena itu, dalam pengujian ini terdapat 1 sidik jari disetiap kamar untuk mendapatkan variasi data pembacaan sidik jari oleh *fingerprint*. selanjutnya, dilakukan pengamatan terhadap data yang ditampilkan pada LCD serta reaksi *output* selenoid, kipas dan lampu, apakah menjadi aktif atau tidak dan berapala waktu mendeteksinya. Hasil pengujian ini kemudian dicatat pada Tabel 4. 3 untuk 3 kamar.

Tabel 4. 3 Hasil Pengujian Lampu, Kipas dan Selenoid

Kamar	Status Jari	Pintu	Lampu	Kipas	Waktu
		Kebuka/Tertutup			
1	Tidak Terdeteksi	Tertutup	Mati	Mati	-
2	Terdeteksi	Kebuka	Nyala	Nyala	1,5 Detik
3	Terdeteksi	Kebuka	Nyala	Nyala	1,5 Detik

Dari hasil pengujian berikut, peneliti menguji waktu terbukanya pintu, lampu dan kipas dalam kamar yang beroperasi secara otomatis ketika pintu terbuka. Dan pengujian ini menguji 3 kamar sekaligus dalam waktu yang berbeda dan hasilnya dapat dilihat dari tabel 4.3 diatas.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah menyelesaikan banyak tahapan penelitian pembuatan “Sistem Cerdas Pengendali *Smartlock* Berbasis Sidik Jari Untuk Keamanan Kamar Hotel” maka dapat disimpulkan bahwa telah berhasil dikembangkan sebuah sistem cerdas pengendali *Smartlock* berbasis sidik jari yang dirancang khusus secara *prototype* untuk meningkatkan keamanan kamar hotel. Sistem ini menggunakan teknologi sidik jari sebagai metode autentikasi utama, didukung dengan integrasi LCD untuk interaksi pengguna yang intuitif. Berbagai tahap dalam pengembangan sistem, mulai dari perancangan konseptual hingga implementasi *prototype*, telah dilakukan secara sistematis.

Hasil pengujian menunjukkan bahwa sistem mampu beroperasi dengan efisien. Keamanan sistem telah ditingkatkan melalui penggunaan teknologi enkripsi yang memastikan data pengguna tetap aman. Selain itu, desain *hardware* yang menggunakan material yang tepat dan proses manufaktur yang cermat memastikan kehandalan dan daya tahan sistem dalam kondisi operasional.

5.2 Saran

Selama proses pembuatan proyek akhir ini, penulis menyadari bahwa masih ada banyak kesalahan yang perlu diperbaiki. Oleh karena itu, penulis memberikan beberapa terkait perbaikan dan pengembangan pada proyek akhir ini. Berikut adalah beberapa saran tersebut:

1. Lakukan evaluasi mendalam terhadap performa sistem *Smartlock* dalam kondisi operasional yang berbeda-beda, termasuk tingkat akurasi deteksi sidik jari dan respons sistem terhadap akses yang valid atau tidak valid.
2. Tinjau aspek biaya pengembangan, implementasi, dan pemeliharaan sistem *Smartlock* dalam jangka panjang.

3. Diskusikan tentang penerapan praktis dari hasil penelitian ini dalam skala yang lebih luas, misalnya implementasi di hotel-hotel lain.



DAFTAR PUSTAKA

- [1] B. Putu, W. Nirmala, P. Agung, and P. Sari, “Perancangan Sistem Informasi Reservasi Berbasis Website Pada Hotel di Nusa Penida,” *J. Teknol. Inf. dan Komput.*, vol. 9, no. 3, pp. 356–365, 2023.
- [2] Ikramullah, Sunardi, and S. Syamsuddin, “Perancangan Prototipe Pintu Otomatis dengan Face Detection Menggunakan Mikrokontroler,” *J. Inform. dan Tek. Elektro Terap.*, vol. 11, no. 3, pp. 767–774, 2023.
- [3] A. Yudhana, Sunardi, and Priyatno, “Perancangan Pengaman Pintu Rumah Berbasis Sidik Jari Menggunakan Metode Uml,” *J. Teknol.*, vol. 10, no. 2, pp. 131–138, 2018, [Online]. Available: <https://dx.doi.org/10.24853/jurtek.10.2.131-138>
- [4] T. Lonika and S. Hariyanto, “Simulasi *Smart door lock* Berbasis QR Code Menggunakan Arduino Uno Pada Penyewaan Apartemen Online,” *J. ALGOR*, vol. 1, no. 1, pp. 9–15, 2019.
- [5] A. H. Sabry, W. Z. Wan Hasan, Y. Alkubaisi, and M. Z. A. Ab-Kadir, “Battery Backup Power System for Electrical Appliances with Two Options of Primary Power Sources,” *2018 IEEE 5th Int. Conf. Smart Instrumentation, Meas. Appl. ICSIMA 2018*, no. November, pp. 28–30, 2019, doi: 10.1109/ICSIMA.2018.8688757.
- [6] A. Mude and L. B. F. Mando, “Implementasi Keamanan Rumah Cerdas Menggunakan *Internet of things* dan Biometric Sistem,” *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 21, no. 1, pp. 179–188, 2021, doi: 10.30812/matrik.v21i1.1381.
- [7] D. Widcaksono and Masyhadi, “Rancang Bangun Secured Door Automatic System Untuk Keamanan Rumah Menggunakan SMS Berbasis Arduino,” *J. Kaji. Tek. Elektro*, vol. 3, no. 1, pp. 52–66, 2018.
- [8] S. Octaviany, “Rancang Bangun Feeder Machine Berbasis KTP Elektronik,”

Univ. Pembang. Panca Budi, 2020.

- [9] M. S. Asih and A. Z. Hasibuan, "Pemanfaatan Internet Of Thing untuk Pengendalian Solenoid Doorlock dalam Sistem Keamanan Rumah Cerdas," *Jikstra*, vol. 5, no. 01, 2023.
- [10] M. S. Asih and A. Z. Hasibuan, "Pengamanan Kunci Pintu Brankas Menggunakan Kriptografi One Time Pad (OTP) Berbasis Android," *Explorer (Hayward)*., vol. 3, no. 2, pp. 58–68, 2023, [Online]. Available: <https://journal.fkpt.org/index.php/Explorer/article/view/738%0Ahttps://journal.fkpt.org/index.php/Explorer/article/download/738/425>
- [11] Wilianto and A. Kurniawan, "Sejarah , Cara Kerja Dan Manfaat *Internet of things*," *Matrix*, vol. 8, no. 2, pp. 36–41, 2018.
- [12] Barbara Kitchenham, "Procedures for Performing Systematic Reviews," *Keele Univ. Tech. Rep.*, vol. 33, no. 2004, pp. 1–26, 2014, [Online]. Available: <https://www.researchgate.net/publication/228756057>
- [13] M. Adam, "Pemanfaat Mikrokontroller Atmega8 sebagai Pengaman Pintu menggunakan Metode Sidik Jari (Fingerprint)," *Pros. Semin. Nas. Kewirausahaan*, vol. 1, no. 1, pp. 279–289, 2019.
- [14] A. Siswanto, A. Efendi, and A. Yulianti, "Alat Kontrol Akses Pintu Rumah Dengan Teknologi Sidik Jari Di Lingkungan Rumah Pintar Dengan Data Yang Di Enkripsi," *J. Penelit. Pos dan Inform.*, vol. 8, no. 2, p. 97, 2018, doi: 10.17933/jppi.2018.080201.
- [15] A. Sanaris and I. Suharjo, "Prototype Alat Kendali Otomatis Penjemur Pakaian Menggunakan NodeMCU ESP32 Dan Telegram Bot Berbasis *Internet of things* (IOT)," *J. Prodi Sist. Inf.*, no. 84, pp. 17–24, 2020.
- [16] S. A. Arrahma and R. Mukhaiyar, "Pengujian Esp32-Cam Berbasis Mikrokontroler ESP32," *JTEIN J. Tek. Elektro Indones.*, vol. 4, no. 1, pp. 60–66, 2023.

- [17] S. SOTYOHADI, "Rancangan dan Implementasi Sistem Absensi dengan Sensor Fingerprint dan Sensor Suhu Non-Contact Berbasis IoT Menggunakan Google Sheets," *ALINIER J. Artif. Intell. Appl.*, vol. 2, no. 1, pp. 28–35, 2021, doi: 10.36040/alinier.v2i1.3545.
- [18] M. Tabrani, Suhardi, and H. Priyandaru, "Sistem Informasi Manajemen Berbasis Website Pada Unl Studio Dengan Menggunakan Framework Codeigniter," *J. Ilm. M-Progress*, vol. 11, no. 1, pp. 13–21, 2021, doi: 10.35968/m-pu.v11i1.598.
- [19] H. Ridha, "Implementasi Twitter Bootstrap pada CodeIgniter," *IlmuKomputer*, pp. 4–5, 2007.
- [20] W. Krisna, H. J. Muhammad, and N. Ambadar, "Rancang Bangun Sistem Informasi Akademik Menggunakan Framework Codeigniter Pada universitas Muhammadiyah Purworejo," *J. Sist. Cerdas*, vol. 5, no. 2, pp. 107–116, 2022, doi: 10.37396/jsc.v5i2.187.
- [21] B. Suprayogi and A. Rahmanesa, "PENERAPAN FRAMEWORK BOOTSTRAP DALAM SISTEM INFORMASI PENDIDIKAN SMA NEGERI 1 PACET CIANJUR JAWA BARAT," *J. Teknol. Inf. Dan Komun.*, vol. 6, no. 2, pp. 119–127, 2019.
- [22] E. N. Hartiwati, "Aplikasi Inventori Barang Menggunakan Java Dengan Phpmyadmin," *Cross-border*, vol. 5, no. 1, pp. 601–610, 2022.
- [23] Y. Darnita and M. Muntahanah, "Rancang Bangun Aplikasi Mobile Penjadwal Perkuliahan Dengan Firebase Dengan Realtime Notification," *Pseudocode*, vol. 8, no. 1, pp. 58–65, 2021, doi: 10.33369/pseudocode.8.1.58-65.
- [24] R. Andrianto and M. H. Munandar, "Aplikasi E-Commerce Penjualan Pakaian Berbasis Android Menggunakan Firebase Realtime Database," *J. Comput. Sci. Inf. Technol.*, vol. 3, no. 1, pp. 20–29, 2022, [Online]. Available: <https://jurnal.ulb.ac.id/index.php/JCoInT/article/view/2478>

LAMPIRAN

Lampiran 1 Daftar Riwayat Hidup Penulis

Daftar Riwayat Hidup Penulis 1

1. Data Pribadi

Nama Lengkap : Nabila Oktariana
Tempat dan Tanggal Lahir : Sungailiat, 16 Oktober 2003
Alamat Rumah : Jl. Swadaya
Gg. Raya 1
Telp : 0852-6906-3656
Email : nabilaoktariana0310@gmail.com
Jenis Kelamin : Perempuan
Agama : Islam



2. Riwayat Pendidikan

TK PERTIWI SUNGAILIAT
SDN 25 SUNGAILIAT
SMPN 5 SUNGAILIAT
SMKN 1 SUNGAILIAT

Sungailiat, 24 Juli 2024


Nabila Oktariana

Daftar Riwayat Hidup Penulis 2

1. Data Pribadi

Nama Lengkap : Adyssha Tritha
Tempat dan Tanggal Lahir : Depok, 15 Oktober 2002
Alamat Rumah : Cikempong Rt 04/09
Kel. Pakansari, Kec. Cibinong
Telp : 0896-1431-0095
Email : adyshat@gmail.com
Jenis Kelamin : Perempuan
Agama : Islam



2. Riwayat Pendidikan

SD IT AMALIA
SMP IT BINANTARA
SMAN 4 CIBINONG

Sungailiat, 24 Juli 2024



Adyssha Tritha

Lampiran 1 Kode Program Arduino

1. MAIN PROGRAM

```
#include <Adafruit_Fingerprint.h>

#include <Arduino.h>

#include <Wire.h>

#include <LiquidCrystal_I2C.h>

#include <WiFi.h>

#include <Firebase_ESP_Client.h>

#include "addons/TokenHelper.h"

#include "addons/RTDBHelper.h"

#define WIFI_SSID "bilaaa"

#define WIFI_PASSWORD "bila1234"

#define API_KEY "AIzaSyBmC7ArdaK-94J2KrBoN_3MwPzGAu5vfjo"

#define DATABASE_URL "https://tugasakhirnabila-b6d08-default-rtdb.firebaseio.com/"

// #define API_KEY "AIzaSyDAqOStK7Be_j9mUexdKu-BSUWXjn1mte4"
```

```
// #define DATABASE_URL "https://tugasakhirnabila-482f8-default-  
rtdb.firebaseio.com/"
```

```
#define relayLampu 2
```

```
#define relayKipas 4
```

```
#define relaySolenoid 23
```

```
#define buttonRegister 18
```

```
#define buttonDelete 19
```

```
#define buttonOpen 5
```

```
FirebaseData fbdo;
```

```
FirebaseAuth auth;
```

```
FirebaseConfig config;
```

```
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&Serial2);
```

```
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

```
bool signupOK = false;
```

```
bool buttonRegister_state;
```

```
bool buttonDelete_state;
```

```
bool buttonOpen_state;
```

```
byte proses;
```

```
uint8_t id;
```

```
void setup() {
```

```
    Serial.begin(115200);
```

```
    pinMode(relayLampu, OUTPUT);
```

```
    pinMode(relayKipas, OUTPUT);
```

```
    pinMode(relaySolenoid, OUTPUT);
```

```
    pinMode(buttonRegister, INPUT_PULLUP);
```

```
    pinMode(buttonDelete, INPUT_PULLUP);
```

```
    pinMode(buttonOpen, INPUT_PULLUP);
```

```
    Serial.println("READY");delay(2000);
```

```
    lcd.init();
```

```
    lcd.backlight();
```

```
    fingerprintBegin();
```

```
    wifiBegin();
```

```
}
```

```
void loop() {  
  
    readButton();  
  
    getFingerprintID();  
  
    delay(50);  
  
    if(buttonRegister_state == 0) {  
  
        id++;  
  
        getFingerprintEnroll();  
  
        delay(1000);  
    }  
  
    if(buttonDelete_state == 0 && id > 0 ) {  
  
        lcd.clear();  
  
        lcd.setCursor(0, 0);  
  
        lcd.print("Penghapusan jari");  
  
        lcd.setCursor(0, 1);  
  
        lcd.print(" scan jari anda ");  
  
        delay(1000);  
  
        proses = 1;  
  
        while(proses == 1){  
  
            getFingerprintID2();  
  
        }  
    }  
}
```

```
    delay(1000);  
}  
  
if(buttonOpen_state == 0){  
    digitalWrite(relayLampu, LOW);  
    digitalWrite(relayKipas, LOW);  
    digitalWrite(relaySolenoid, HIGH);  
  
    lcd.clear();  
    lcd.setCursor(0, 0);  
    lcd.print(" Pintu Terbuka ");  
    delay(5000);  
    lcd.clear();  
    lcd.print(" Pintu Tertutup ");  
    digitalWrite(relaySolenoid, LOW);  
}  
}
```

```
void readButton(){  
    buttonRegister_state = digitalRead(buttonRegister);  
    buttonDelete_state = digitalRead(buttonDelete);  
    buttonOpen_state = digitalRead(buttonOpen);  
}
```

```
// Serial.println((String) "Button Register = " + buttonRegister_state +  
"\t\t" + "Button Delete = " + buttonDelete_state + "\t\t" + "Button Open =  
" + buttonOpen_state);  
  
}
```



2. SCAN SIDIK JARI

```
uint8_t getFingerprintID() {
    uint8_t p = finger.getImage();
    switch (p) {
        case FINGERPRINT_OK:
            Serial.println("Image taken");
            break;
        case FINGERPRINT_NOFINGER:
            Serial.println("No finger detected");
            lcd.clear();
            lcd.setCursor(0, 0);
            lcd.print("SidikJari Kosong");
            lcd.setCursor(0, 1);
            lcd.print(" Scan/ Register ");
            return p;
        case FINGERPRINT_PACKETRECEIVEERR:
            Serial.println("Communication error");
            lcd.clear();
            lcd.setCursor(0, 0);
            lcd.print("SIDIK JARI EROR");
            return p;
        case FINGERPRINT_IMAGEFAIL:
            Serial.println("Imaging error");
            lcd.clear();
            lcd.setCursor(0, 0);
            lcd.print("SIDIK JARI EROR");
            return p;
        default:
            Serial.println("Unknown error");
            lcd.clear();
            lcd.setCursor(0, 0);
```

```

        lcd.print("SIDIK JARI EROR");
        return p;
    }

    // OK success!

    p = finger.image2Tz();
    switch (p) {
        case FINGERPRINT_OK:
            Serial.println("Image converted");
            break;
        case FINGERPRINT_IMAGEMESS:
            lcd.clear();
            lcd.setCursor(0, 0);
            lcd.print("SIDIK JARI EROR");
            Serial.println("Image too messy");
            return p;
        case FINGERPRINT_PACKETRECEIVEERR:
            lcd.clear();
            lcd.setCursor(0, 0);
            lcd.print("SIDIK JARI EROR");
            Serial.println("Communication error");
            return p;
        case FINGERPRINT_FEATUREFAIL:
            lcd.clear();
            lcd.setCursor(0, 0);
            lcd.print("SIDIK JARI EROR");
            Serial.println("Could not find fingerprint features");
            return p;
        case FINGERPRINT_INVALIDIMAGE:
            lcd.clear();

```

```

    lcd.setCursor(0, 0);
    lcd.print("SIDIK JARI EROR");
    Serial.println("Could not find fingerprint features");
    return p;
default:
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("SIDIK JARI EROR");
    Serial.println("Unknown error");
    return p;
}

// OK converted!
p = finger.fingerSearch();
if (p == FINGERPRINT_OK) { //jari terdaftar
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print(" Pintu Terbuka ");

    digitalWrite(relayLampu, HIGH);
    digitalWrite(relayKipas, HIGH);
    digitalWrite(relaySolenoid, HIGH);
    delay(5000);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print(" Pintu Tertutup ");
    digitalWrite(relaySolenoid, LOW);
    delay(1000);

    Serial.println("Found a print match!");
}

```

```

    } else if (p == FINGERPRINT_PACKETRECEIVED) {
        Serial.println("Communication error");
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("  SIDIK JARI ");
        lcd.setCursor(0, 1);
        lcd.print("Belum terdaftar");
        return p;
    } else if (p == FINGERPRINT_NOTFOUND) {
        Serial.println("Did not find a match");
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print(" Tidak deteksi ");
        lcd.setCursor(0, 1);
        lcd.print(" Sidik jari ");
        return p;
    } else {
        Serial.println("Unknown error");
        return p;
    }

    // found a match!
    Serial.print("Found ID #"); Serial.print(finger.fingerID);
    Serial.print(" with confidence of "); Serial.println(finger.confidence);

    return finger.fingerID;
}

// returns -1 if failed, otherwise returns ID #
int getFingerprintIDez() {
    uint8_t p = finger.getImage();

```

```
if (p != FINGERPRINT_OK) return -1;
```

```
p = finger.image2Tz();
```

```
if (p != FINGERPRINT_OK) return -1;
```

```
p = finger.fingerFastSearch();
```

```
if (p != FINGERPRINT_OK) return -1;
```

```
// found a match!
```

```
Serial.print("Found ID #"); Serial.print(finger.fingerID);
```

```
Serial.print(" with confidence of "); Serial.println(finger.confidence);
```

```
return finger.fingerID;
```

```
}
```

3. DAFTAR SIDIK JARI

```
uint8_t getFingerprintEnroll() {
    int p = -1;
    // digitalWrite(buzzer,HIGH); //SUARA BEEP CEPAT
    // delay(100);
    // digitalWrite(buzzer,LOW);
    // delay(100);
    Serial.print("Waiting for valid finger to enroll as #"); Serial.println(id);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Menunggu daftar");
    lcd.setCursor(0, 1);
    lcd.print(" Letakkan Jari ");

    while (p != FINGERPRINT_OK) {
        p = finger.getImage();
        switch (p) {
            case FINGERPRINT_OK:
                Serial.println("Image taken");
                break;
            case FINGERPRINT_NOFINGER:
                Serial.println(".");
                break;
            case FINGERPRINT_PACKETRECEIVEERR:
                Serial.println("Communication error");
                lcd.clear();
                lcd.setCursor(0, 0);
                lcd.print("SIDIK JARI EROR");
                break;
            case FINGERPRINT_IMAGEFAIL:
                Serial.println("Imaging error");
```

```

        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("SIDIK JARI EROR");
        break;
    default:
        Serial.println("Unknown error");
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("SIDIK JARI EROR");
        break;
    }
}

// OK success!

p = finger.image2Tz(1);
switch (p) {
    case FINGERPRINT_OK:
        Serial.println("Image converted");
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Scan sidik jari ");
        lcd.setCursor(0, 1);
        lcd.print("Lepas sidik jari");
        break;
    case FINGERPRINT_IMAGEMESS:
        Serial.println("Image too messy");
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("SIDIK JARI EROR");
        return p;
}

```

```

case FINGERPRINT_PACKETRECEIVEERR:
    Serial.println("Communication error");
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("SIDIK JARI EROR");
    return p;
case FINGERPRINT_FEATUREFAIL:
    Serial.println("Could not find fingerprint features");
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("SIDIK JARI EROR");
    return p;
case FINGERPRINT_INVALIDIMAGE:
    Serial.println("Could not find fingerprint features");
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("SIDIK JARI EROR");
    return p;
default:
    Serial.println("Unknown error");
    return p;
}

```

```

Serial.println("Remove finger");
delay(2000);
p = 0;
while (p != FINGERPRINT_NOFINGER) {
    p = finger.getImage();
}
Serial.print("ID "); Serial.println(id);
p = -1;

```

```
// digitalWrite(buzzer,HIGH); //SUARA BEEP CEPAT
// delay(100);
// digitalWrite(buzzer,LOW);
// delay(100);
```

```
Serial.println("Place same finger again");
lcd.clear();
lcd.setCursor(0, 0);
lcd.print(" Letakkan Jari ");
lcd.setCursor(0, 1);
lcd.print(" yang sama lagi ");
while (p != FINGERPRINT_OK) {
    p = finger.getImage();
    switch (p) {
        case FINGERPRINT_OK:
            Serial.println("Image taken");
            break;
        case FINGERPRINT_NOFINGER:
            Serial.print(".");
            break;
        case FINGERPRINT_PACKETRECEIVEERR:
            Serial.println("Communication error");
            lcd.clear();
            lcd.setCursor(0, 0);
            lcd.print("SIDIK JARI EROR");
            break;
        case FINGERPRINT_IMAGEFAIL:
            Serial.println("Imaging error");
            break;
        default:
            Serial.println("Unknown error");
```

```

        break;
    }
}

// OK success!

p = finger.image2Tz(2);
switch (p) {
    case FINGERPRINT_OK:
        Serial.println("Image converted");
        break;
    case FINGERPRINT_IMAGEMESS:
        Serial.println("Image too messy");
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("SIDIK JARI EROR");
        return p;
    case FINGERPRINT_PACKETRECEIVEERR:
        Serial.println("Communication error");
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("SIDIK JARI EROR");
        return p;
    case FINGERPRINT_FEATUREFAIL:
        Serial.println("Could not find fingerprint features");
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("SIDIK JARI EROR");
        return p;
    case FINGERPRINT_INVALIDIMAGE:
        Serial.println("Could not find fingerprint features");

```

```

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("SIDIK JARI EROR");
    return p;
default:
    Serial.println("Unknown error");
    return p;
}

```

```

// OK converted!
Serial.print("Creating model for #"); Serial.println(id);

p = finger.createModel();
if (p == FINGERPRINT_OK) {
    Serial.println("Prints matched!");
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
    Serial.println("Communication error");
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("SIDIK JARI EROR");
    return p;
} else if (p == FINGERPRINT_ENROLLMISMATCH) {
    Serial.println("Fingerprints did not match");
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("SIDIK JARI EROR");
    return p;
} else {
    Serial.println("Unknown error");
    lcd.clear();
    lcd.setCursor(0, 0);

```

```

    lcd.print("SIDIK JARI EROR");
    return p;
}

Serial.print("ID "); Serial.println(id);
p = finger.storeModel(id);
if (p == FINGERPRINT_OK) {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print(" Jari Terdaftar ");
    lcd.setCursor(0, 1);
    lcd.print("ID = ");
    lcd.setCursor(6, 1);
    lcd.print(id);
    delay(2000);
    if(id == 1){
        Firebase.RTDB.setInt(&fbdo, "Kamar3/id1", 1);
        Serial.println("SIDIK JARI 1 TERDAFTAR");
        delay(1000);
    }
    if(id == 2){
        Firebase.RTDB.setInt(&fbdo, "Kamar3/id2", 1);
        Serial.println("SIDIK JARI 2 TERDAFTAR");
        delay(1000);
    }
    if(id == 3){
        Firebase.RTDB.setInt(&fbdo, "Kamar3/id3", 1);
        Serial.println("SIDIK JARI 3 TERDAFTAR");
        delay(1000);
    }
    if(id == 4){

```

```

        Firebase.RTDB.setInt(&fbdo, "Kamar3/id4", 1);
        Serial.println("SIDIK JARI 4 TERDAFTAR");
        delay(1000);
    }
    if(id == 5){
        Firebase.RTDB.setInt(&fbdo, "Kamar3/id5", 1);
        Serial.println("SIDIK JARI 5 TERDAFTAR");
        delay(1000);
    }
    if(id > 5){
        id = id - 1;
    }

} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
    Serial.println("Communication error");
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("SIDIK JARI EROR");
    return p;
} else if (p == FINGERPRINT_BADLOCATION) {
    Serial.println("Could not store in that location");
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("SIDIK JARI EROR");
    return p;
} else if (p == FINGERPRINT_FLASHERR) {
    Serial.println("Error writing to flash");
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("SIDIK JARI EROR");
    return p;
}

```

```

    } else {
        Serial.println("Unknown error");
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("SIDIK JARI EROR");
        return p;
    }

    return true;
}

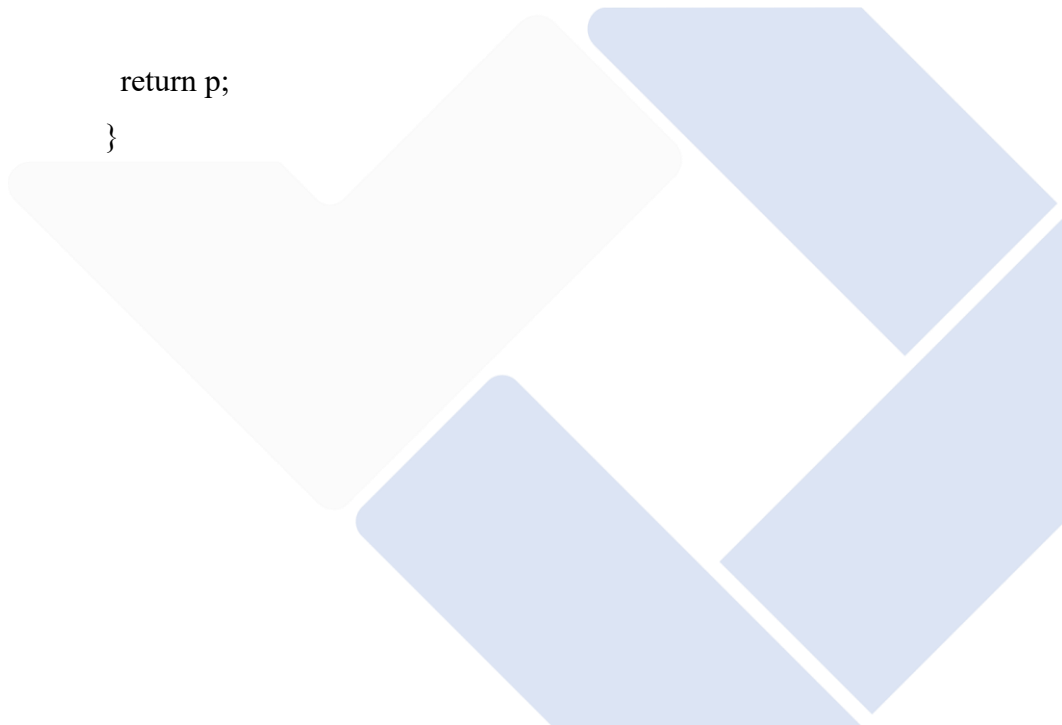
Hapus sidik jari
uint8_t deleteFingerprint(uint8_t id) {
    uint8_t p = -1;

    p = finger.deleteModel(id);

    if (p == FINGERPRINT_OK) {
        Serial.println("Akses Diterima");
        Serial.println("Deleted!");
    } else if (p == FINGERPRINT_PACKETRECEIVEERR) {
        Serial.println("Communication error");
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("SIDIK JARI EROR");
    } else if (p == FINGERPRINT_BADLOCATION) {
        Serial.println("Could not delete in that location");
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("SIDIK JARI EROR");
    } else if (p == FINGERPRINT_FLASHERR) {
        Serial.println("Error writing to flash");
    }
}

```

```
    lcd.clear();  
    lcd.setCursor(0, 0);  
    lcd.print("SIDIK JARI EROR");  
} else {  
    Serial.print("Unknown error: 0x"); Serial.println(p, HEX);  
    lcd.clear();  
    lcd.setCursor(0, 0);  
    lcd.print("SIDIK JARI EROR");  
}  
  
return p;  
}
```



4. FUNGSI

```
void wifiBegin(){
    WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
    Serial.print("Connecting to Wi-Fi");
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print(" Connecting to ");
    lcd.setCursor(0, 1);
    lcd.print("   Wi-Fi   ");
    while (WiFi.status() != WL_CONNECTED){
        Serial.print(".");
        delay(300);
    }
    lcd.clear();
    Serial.println();
    Serial.print("Connected with IP: ");
    Serial.println(WiFi.localIP());
    Serial.println();
    lcd.setCursor(0, 0);
    lcd.print("Connect with IP:");
    lcd.setCursor(0, 1);
    lcd.print(WiFi.localIP());
    config.api_key = API_KEY;
    config.database_url = DATABASE_URL;
    if(Firebase.signUp(&config, &auth, "", "")){
        Serial.println("ok");
        signupOK = true;
    }else{
        Serial.printf("%s\n", config.signer.signupError.message.c_str());
    }
}
```

```

    config.token_status_callback    =    tokenStatusCallback;    //see
addons/TokenHelper.h
    Firebase.begin(&config, &auth);
    Firebase.reconnectWiFi(true);
}

void fingerprintBegin(){
    finger.begin(57600);
    delay(5);

    if (finger.verifyPassword()) {
        lcd.setCursor(0, 0);
        lcd.print("FINGERPRINT SENS");
        lcd.setCursor(0, 1);
        lcd.print("  DETECTED  ");
    } else {
        lcd.setCursor(0, 0);
        lcd.print("FINGERPRINT SENS");
        lcd.setCursor(0, 1);
        lcd.print("  UNDETECTED  ");
        while (1) { delay(1); }
    }
    Serial.println(F("Reading sensor parameters"));
    finger.getParameters();
    Serial.print(F("Status: 0x")); Serial.println(finger.status_reg, HEX);
    Serial.print(F("Sys ID: 0x")); Serial.println(finger.system_id, HEX);
    Serial.print(F("Capacity: ")); Serial.println(finger.capacity);
    Serial.print(F("Security level: ")); Serial.println(finger.security_level);
    Serial.print(F("Device address: ")); Serial.println(finger.device_addr,
    HEX);
    Serial.print(F("Packet len: ")); Serial.println(finger.packet_len);

```

```

Serial.print(F("Baud rate: ")); Serial.println(finger.baud_rate);

finger.getTemplateCount();

if (finger.templateCount == 0) {
    Serial.print("Sensor doesn't contain any fingerprint data. Please run the
'enroll' example.");
}
else {
    Serial.println("Waiting for valid finger...");
    Serial.print("Sensor contains "); Serial.print(finger.templateCount);
Serial.println(" templates");
}
    lcd.clear();
}

uint8_t getFingerprintID2() {
    uint8_t p = finger.getImage();
    delay(2000);
    switch (p) {
        case FINGERPRINT_OK:
            Serial.println("Image taken");
            break;
        case FINGERPRINT_NOFINGER:
            Serial.println("No finger detected");
            lcd.clear();
            lcd.setCursor(0, 0);
            lcd.print(" LETAKKAN JARI ");
            return p;
        case FINGERPRINT_PACKETRECEIVEERR:
            Serial.println("Communication error");

```

```

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("SIDIK JARI EROR");
    return p;
case FINGERPRINT_IMAGEFAIL:
    Serial.println("Imaging error");
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("SIDIK JARI EROR");
    return p;
default:
    Serial.println("Unknown error");
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("SIDIK JARI EROR");
    return p;
}

// OK success!

p = finger.image2Tz();
switch (p) {
case FINGERPRINT_OK:
    Serial.println("Image converted");
    break;
case FINGERPRINT_IMAGEMESS:
    Serial.println("Image too messy");
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("SIDIK JARI EROR");
    return p;

```

```

case FINGERPRINT_PACKETRECEIVEERR:
    Serial.println("Communication error");
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("SIDIK JARI EROR");
    return p;
case FINGERPRINT_FEATUREFAIL:
    Serial.println("Could not find fingerprint features");
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("SIDIK JARI EROR");
    return p;
case FINGERPRINT_INVALIDIMAGE:
    Serial.println("Could not find fingerprint features");
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("SIDIK JARI EROR");
    return p;
default:
    Serial.println("Unknown error");
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("SIDIK JARI EROR");
    return p;
}

// OK converted!
p = finger.fingerSearch();
if (p == FINGERPRINT_OK) { //jari terdaftar
    lcd.clear();
    lcd.setCursor(0, 0);

```

```

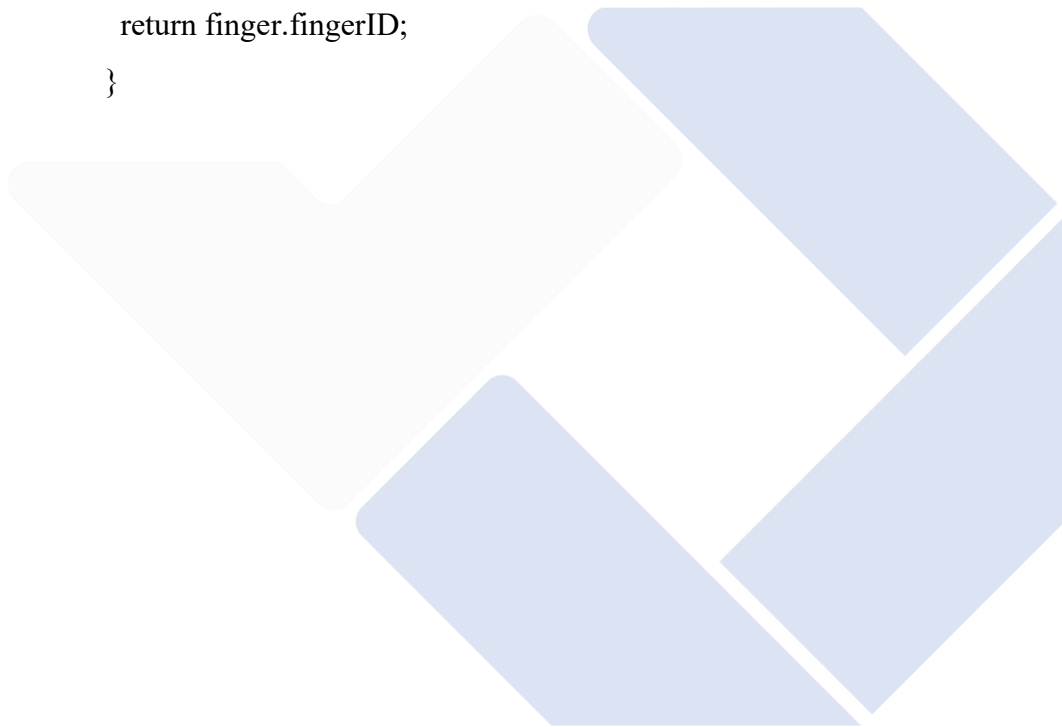
lcd.print(" Hapus berhasil ");

for(int i=1; i<=id; i++){
    deleteFingerprint(i);
    Serial.println((String) "Delete Sidik jari ke = " + i);
}
Firebase.RTDB.setInt(&fbdo, "Kamar3/id1", 0);
Firebase.RTDB.setInt(&fbdo, "Kamar3/id2", 0);
Firebase.RTDB.setInt(&fbdo, "Kamar3/id3", 0);
Firebase.RTDB.setInt(&fbdo, "Kamar3/id4", 0);
Firebase.RTDB.setInt(&fbdo, "Kamar3/id5", 0);
id=0;
delay(2000);
proses = 0;

Serial.println("Found a print match!");
} else if (p == FINGERPRINT_PACKETRECEIVED) {
    Serial.println("Communication error");
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("SIDIK JARI EROR");
    return p;
} else if (p == FINGERPRINT_NOTFOUND) { //tidak terdaftar
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("SIDIK JARI EROR");
    Serial.println("Did not find a match");
    return p;
} else {
    Serial.println("Unknown error");
    lcd.clear();

```

```
    lcd.setCursor(0, 0);  
    lcd.print("SIDIK JARI EROR");  
    return p;  
}  
  
// found a match!  
Serial.print("Found ID #"); Serial.print(finger.fingerID);  
Serial.print(" with confidence of "); Serial.println(finger.confidence);  
  
return finger.fingerID;  
}
```



**Lampiran 2 Hasil Dokumentasi Pengujian Sistem Cerdas Pengendali
Smartlock Berbasis Sidik Jari Untuk Keamanan Kamar Hotel**

