

RANCANG BANGUN *AUTOMATIC SMART LOCKER* DENGAN SISTEM IOT MENGUNAKAN *DATABASE*

Iftitah Walinda Fujiarti¹, Satriyo², Muhammad Ridhwan Sufandi³, Agus Riyanto⁴, Eko Mardianto⁵

^{1,2,3,4,5}Program Studi Teknologi Rekayasa Sistem Elektronika, Politeknik Negeri Pontianak

Jl. Jendral Ahmad Yani - Pontianak 78124, Kalimantan Barat

¹Iftitah2105@gmail.com, ²satrio@polnep.ac.id, ³mr.sufandi86@gmail.com, ⁴Ariyanto228@gmail.com,
⁵emardianto74@gmail.com

Abstract

Door security is an important aspect that must be considered. One way to enhance security is by implementing an electronic system. A common problem occurs when users forget to bring their keys or accidentally lose them while away from home. Conventional key-based systems, which are still widely used, are vulnerable to break-ins, making them less practical and more susceptible to theft. Smart lockers are generally still equipped with traditional locks. However, with technological advancements, conventional locks are increasingly being replaced by electronic locking systems, which are more practical and reduce the risks of lost or duplicated keys. Therefore, a locker security system that can be remotely monitored via the Internet of Things (IoT) is required. The hardware used in this research includes the Arduino Mega 2560 with built-in Wi-Fi, which communicates with a Nextion HMI LCD as a self-service user interface and uses serial communication with the ESP8266 module to connect with a native PHP-based system, with data stored in a Laragon database. The LOKERPINTARPOLNEP application is designed to store user data, assist administrators in controlling and monitoring locker usage, and send notifications via the Telegram application. The results of this study indicate that the developed smart locker system enhances security and improves locker management efficiency through an integrated and real-time approach.

Keywords: LOKERPINTARPOLNEP, Arduino Mega 2560 built-in Wi-Fi, LCD Nextion HMI, ESP8266, Telegram Application.

Abstrak

Keamanan pintu merupakan aspek yang sangat penting untuk diperhatikan. Salah satu upaya yang dapat dilakukan dengan menambah sistem keamanan. Permasalahan yang sering terjadi adalah *user* lupa membawa kunci saat bepergian atau secara tidak sengaja kehilangan kunci ketika berada di luar rumah. Sistem pengamanan menggunakan kunci konvensional yang umum digunakan oleh masyarakat cenderung mudah dibobol oleh pelaku tindak kejahatan. Sehingga dianggap kurang praktis dan rentan terhadap tindakan pencurian. *Smart locker* biasanya dilengkapi pengaman berupa kunci konvensional. seiring perkembangan teknologi, penggunaan kunci konvensional tergantikan dengan kunci elektronik karena lebih praktis dan dapat mengurangi resiko kehilangan kunci serta penggantian anak kunci yang disalahgunakan. Oleh karena itu, diperlukan sistem keamanan *locker* yang dapat dipantau dari jarak jauh dengan metode *Internet of Things*. Perangkat keras yang digunakan dalam penelitian adalah Arduino Mega 2560 *built in wifi* yang berkomunikasi dengan LCD Nextion HMI untuk *user interface* sebagai *self-service* serta menggunakan komunikasi serial dengan modul ESP8266 untuk menghubungkan sistem PHP *web native* yang tersimpan ke *database* Laragon. Aplikasi LOKERPINTARPOLNEP berfungsi untuk menyimpan data, memudahkan *admin* dalam mengontrol dan *monitoring locker* yang digunakan oleh *user* serta mengirimkan notifikasi ke aplikasi Telegram. Hasil dari penelitian ini menunjukkan bahwa sistem *smart locker* yang dikembangkan mampu meningkatkan keamanan dan efisiensi pengelolaan locker secara terintegrasi dan *real-time*.

Kata kunci: LOKERPINTARPOLNEP, Arduino Mega 2560 Built in Wifi, LCD Nextion HMI, ESP8266 dan Aplikasi Telegram.

1. PENDAHULUAN

Locker adalah fasilitas penyimpanan barang yang umum digunakan di berbagai tempat publik, seperti institusi pendidikan, pusat kebugaran, perkantoran, terminal, dan bandara

dengan fungsi utama menjaga keamanan barang pribadi milik *user*. Namun, sistem pengamanan yang digunakan pada umumnya masih mengandalkan kunci konvensional, yang dinilai kurang efisien serta memiliki risiko tinggi terhadap kehilangan atau penyalahgunaan oleh *user* maupun pihak yang tidak berwenang [1]. Kunci fisik mudah untuk diduplikasi maupun dibobol, sehingga rawan terhadap tindak pencurian [2]. Seiring dengan perkembangan teknologi informasi dan komunikasi, sistem pengamanan konvensional mulai digantikan dengan sistem berbasis elektronik yang menawarkan kepraktisan dan keamanan lebih tinggi [3].

Perkembangan *Internet of Things* (IoT) sebagai teknologi yang menghubungkan berbagai perangkat fisik ke jaringan *internet* membuka peluang besar dalam menciptakan sistem *smart locker* yang dapat dikendalikan dan dipantau dari jarak jauh. IoT memungkinkan perangkat untuk saling terhubung dan bertukar data secara otomatis tanpa campur tangan manusia secara langsung, sehingga sangat cocok diterapkan pada sistem pengamanan *locker* [4]. Dengan pemanfaatan teknologi IoT, *smart locker* dapat dilengkapi dengan fitur *monitoring real-time*, kendali akses digital, notifikasi otomatis, serta pendataan aktivitas *user* ke dalam *database* [5].

Berbagai penelitian telah menunjukkan keberhasilan penerapan IoT dalam pengembangan sistem cerdas. Misalnya, penerapan IoT pada sistem *monitoring* tanaman hidroponik [6], sistem perpustakaan digital berbasis *web* [7], serta *smart locker* berbasis aplikasi Android dengan penggunaan QR-Code untuk membuka dan menutup *locker* secara otomatis [8] selain itu, *smart greenhouse* berbasis IoT untuk mengoptimalkan pertumbuhan sayuran dengan menjaga stabilitas suhu dan kelembaban, mengurangi konsumsi air, energi, serta meningkatkan efisiensi dan hasil panen [9]. Penelitian lainnya juga menunjukkan bahwa sistem deteksi kualitas udara berbasis IoT mampu memberikan peringatan dini melalui aplikasi Telegram saat kadar polusi seperti karbon monoksida (CO) melebihi ambang batas, sehingga membantu masyarakat mengambil

langkah preventif terhadap bahaya kesehatan akibat polusi udara [10].

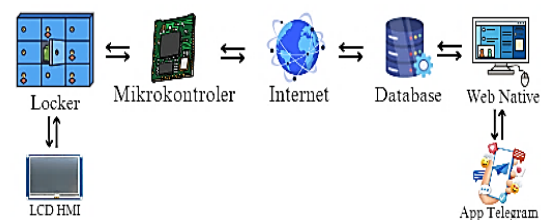
Namun, integrasi teknologi *Internet of Things* (IoT) dengan sistem *self-service* menggunakan LCD Nextion HMI serta sistem kontrol berbasis *web native* yang dilengkapi notifikasi secara *real-time* melalui aplikasi Telegram masih terbatas.

Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk merancang dan membangun sistem *automatic smart locker* berbasis *Internet of Things* (IoT) dengan menggunakan mikrokontroler Arduino Mega 2560 *built-in* Wi-Fi dan modul ESP8266 yang terintegrasi dengan aplikasi *web native* bernama LOKERPINTARPOLNEP. *Smart locker* ini memanfaatkan LCD Nextion HMI sebagai *user interface* dan mengirim data ke dalam *database* Laragon. Selain itu, *smart locker* dilengkapi fitur notifikasi otomatis melalui aplikasi Telegram, sehingga memudahkan *admin* dalam memantau dan mengontrol penggunaan *locker* secara efisien dari jarak jauh.

2. METODOLOGI PENELITIAN

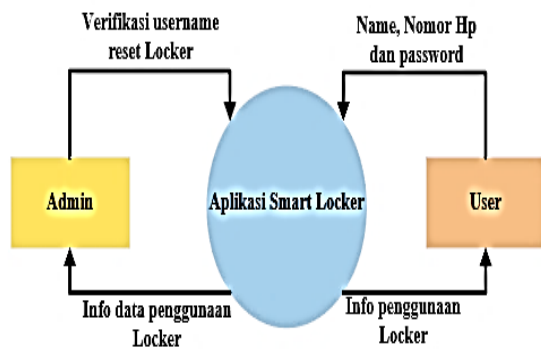
2.1 Rancangan Penelitian

Rancangan arsitektur sistem merupakan rancangan yang diimplementasikan pada aplikasi *web native* bernama LOKERPINTARPOLNEP.



Gambar 1. Rancangan Arsitektur Sistem

Pada Gambar 1 menunjukkan arsitektur sistem *automatic smart locker* berbasis *Internet of Things* (IoT) yang terdiri dari beberapa komponen utama sebagai media penyimpanan dengan mekanisme buka-tutup otomatis, mikrokontroler sebagai pusat kendali yang terhubung ke *server* melalui *internet*, serta LCD Nextion HMI sebagai antarmuka *user*. Data *user* dan status *locker* disimpan dalam *database*, sedangkan pemantauan dan pengendalian sistem melalui antarmuka *web* yang hanya diakses oleh *admin*. Selain itu, aplikasi Telegram digunakan untuk mengirim notifikasi otomatis. Seluruh komponen terintegrasi guna mendukung pengoperasian sistem secara *real-time* dan efisien dari jarak jauh. Berikut merupakan alur *diagram context* yang diterapkan dalam penelitian.



Gambar 2. Alur Diagram Context

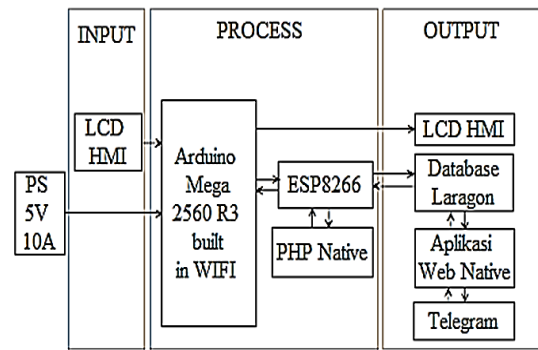
Pada Gambar 2 menunjukkan alur penelitian *diagram context* yang menggambarkan interaksi antara *admin*, *user*, dan sistem *Smart Locker*. Aplikasi *smart locker* berfungsi sebagai pusat pengelolaan dan penghubung antara *admin* dan *user* dalam sistem penyimpanan otomatis berbasis IoT. *admin* memverifikasi identitas *user*, mereset locker jika terjadi kendala, serta menerima data penggunaan *locker* yang ditampilkan melalui antarmuka *web native*.

User berinteraksi dengan sistem dengan memasukkan identitas berupa nama, nomor *handphone* yang terhubung ke Telegram, *password*, serta nomor *locker* yang digunakan. Informasi seperti status *locker* dikirim melalui Telegram. Riwayat penggunaan secara menyeluruh hanya dapat diakses oleh *admin*.

Analisis kebutuhan sistem dilakukan untuk mengidentifikasi spesifikasi perangkat keras dan perangkat lunak yang diperlukan. Perangkat keras mencakup prosesor Intel i5 dengan RAM 8 GB untuk mendukung kinerja optimal. Antarmuka visual menggunakan LCD Nextion HMI tipe Nx4832T035. Pengendalian dan pemrosesan data dilakukan oleh mikrokontroler Arduino Mega 2560 dengan modul WiFi ESP8266 sebagai konektivitas nirkabel.

Perangkat untuk *admin* mencakup smartphone iPhone X dan laptop ThinkPad guna mendukung fleksibilitas dalam pengoperasian. Dari sisi perangkat lunak, sistem operasi yang digunakan adalah Windows 10 64-bit untuk stabilitas dan kompatibilitas. Akses web dilakukan melalui browser Google Chrome, sedangkan *server* aplikasi menggunakan Apache versi 2.4.54. Sistem basis data dikelola dengan MySQL versi 8.0.30 sebagai DBMS. Untuk kebutuhan pelaporan, digunakan Microsoft Office 2010.

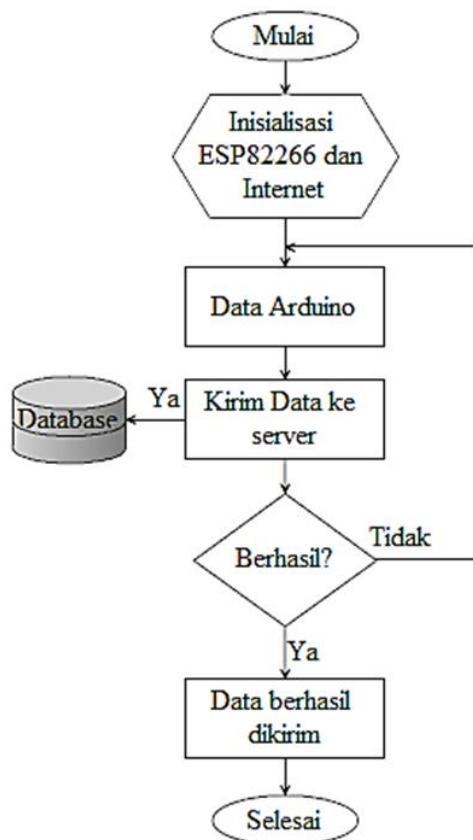
Perancangan sistem merupakan tahapan dari proses pengembangan, mulai dari konseptualisasi hingga implementasi.



Gambar 3. Block Diagram Smart Locker

Pada Gambar 3 menunjukkan *block diagram* yang merepresentasikan alur kerja sistem *smart locker*, yang terdiri atas komponen *input*, proses, dan *output* yang saling terhubung. Komponen *input* terdiri dari LCD Nextion HMI sebagai *user interface* untuk memasukkan perintah, serta *power supply* 5V 10A sebagai catu daya sistem. Pada bagian proses, Arduino Mega 2560 R3 dengan *built-in* Wi-Fi berfungsi sebagai mikrokontroler utama untuk memproses data dari LCD Nextion HMI dan mengendalikan perangkat keras. Serta didukung oleh modul ESP8266 untuk konektivitas jaringan dan komunikasi dengan *server*. Sistem perangkat lunak, menggunakan bahasa pemrograman PHP *native* untuk mengelola data dan menjalankan aplikasi web LOKERPINTARPOLNEP. Komponen *output* mencakup LCD Nextion HMI sebagai media informasi kepada *user*, database Laragon berbasis MySQL untuk menyimpan data *user*, aplikasi *web native* sebagai antarmuka *admin* untuk pemantauan dan pengendalian oleh *admin*, serta aplikasi Telegram untuk notifikasi otomatis sebagai pemantauan jarak jauh.

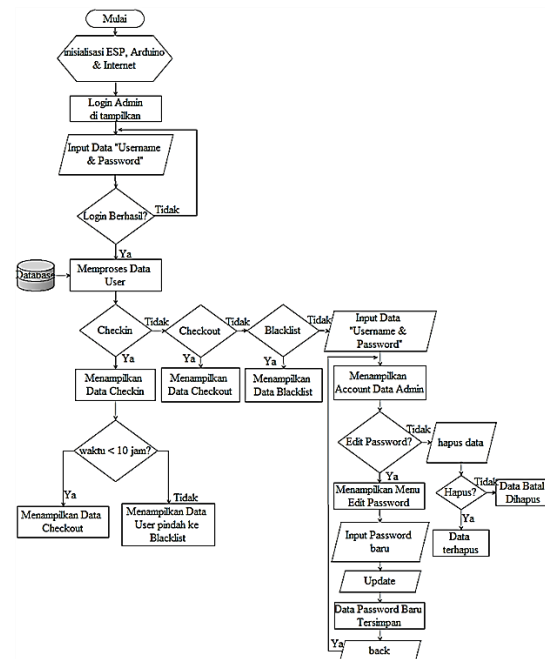
Penjelasan di atas dapat dilihat secara rinci melalui *flowchart* sistem yang menggambarkan alur logika program dalam perancangan.



Gambar 5. Flowchart Sistem ESP8266

Pada Gambar 5 Menunjukkan alur kerja pada modul ESP8266 yang bertugas untuk mengambil data dari Arduino dan mengirimkannya ke server agar tersimpan dalam database. Proses dimulai dengan inisialisasi ESP8266 dan koneksi internet, kemudian sistem membaca data dari Arduino. Data tersebut dikirim ke server melalui jaringan. Setelah proses pengiriman dilakukan, sistem akan melakukan verifikasi keberhasilan.

Jika pengiriman data berhasil, sistem akan memberikan konfirmasi bahwa data telah tersimpan di database. Namun, jika pengiriman gagal (karena gangguan jaringan atau koneksi server), maka sistem akan mengulang proses pengiriman hingga berhasil. Mekanisme ini memastikan bahwa setiap data yang dikirim benar-benar diterima dengan baik oleh server sebelum proses dinyatakan selesai.



Gambar 6. Flowchart Website Native.

Pada Gambar 6 menunjukkan alur kerja sistem smart locker yang mengirim data ke aplikasi web native, yang dimulai dari proses inisialisasi perangkat ESP, Arduino, dan koneksi internet. Setelah inisialisasi, sistem menampilkan halaman login untuk admin. Apabila autentikasi berhasil, sistem memproses data user dengan beberapa opsi yaitu, check-in, check-out, dan blacklist.

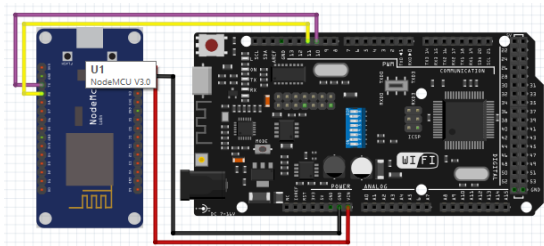
Pada proses check-in, sistem mencatat waktu penggunaan dan memverifikasi apakah durasi pemakaian kurang dari 10 jam. Jika melebihi batas waktu tersebut, user secara otomatis dipindahkan ke daftar blacklist. Proses check-out memastikan bahwa user keluar sesuai prosedur, serta memperbarui status locker.

Selain mengelola data user, admin memiliki akses untuk mengatur akun dan keamanan sistem. Fitur ini mencakup pengubahan kata sandi melalui input data yang diperbarui dalam sistem, serta penghapusan akun user dengan konfirmasi terlebih dahulu.

Dengan alur tersebut, sistem smart locker dirancang untuk memastikan validitas penggunaan loker secara efisien, serta memberikan kontrol penuh kepada admin dalam menjaga keamanan dan integritas data user.

Untuk mengintegrasikan komunikasi antara Arduino Mega 2560 dan ESP8266 dalam sistem smart locker, diperlukan perancangan wiring diagram. Perancangan ini bertujuan untuk memastikan pertukaran data antarperangkat dapat berlangsung secara stabil melalui komunikasi serial. berikut rancangan wiring

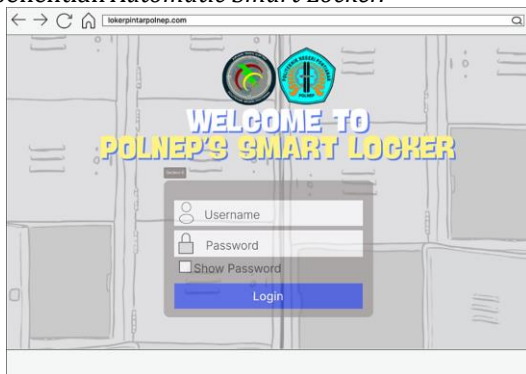
diagram dari sistem utama yang menunjukkan koneksi pin arduino antara Arduino Mega 2560 dan modul ESP8266



Gambar 7. Wiring Komunikasi Serial.

Pada Gambar 7 Merupakan rangkaian yang menghubungkan modul ESP8266 dengan Arduino Mega 2560, di mana ESP8266 berfungsi sebagai mikrokontroler dengan modul Wi-Fi untuk mendukung komunikasi nirkabel. Rangkaian dirancang guna memberikan kemampuan konektivitas berbasis *Internet of Things* (IoT), seperti pengendalian jarak jauh, pemantauan, serta pengiriman data ke *server*. Arduino Mega bertugas untuk mengelola logika dan proses kontrol sistem. Konfigurasi pin pada rangkaian meliputi pin TX pada modul ESP8266 yang dihubungkan ke pin 10 pada Arduino Mega untuk transmisi data, pin RX ke pin 11 untuk menerima data, pin Vin ke Vin sebagai suplai daya, serta pin GND ke GND untuk menyamakan referensi tegangan atau sistem *grounding*.

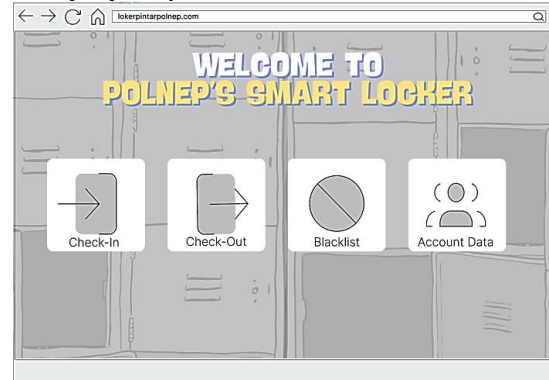
Perancangan antarmuka aplikasi *web native* bernama LOKERPINTARPOLNEP merupakan tampilan aplikasi yang akan digunakan dalam penelitian *Automatic Smart Locker*.



Gambar 8. Halaman Login Admin.

Pada Gambar 8 Halaman *login* bertujuan untuk memberikan akses kepada *admin*. *form login* terdiri dari dua kolom utama, yaitu, kolom *username* untuk memasukkan nama *user* dan kolom *password* untuk memasukkan kata sandi. Fungsi utama halaman *login* untuk membatasi akses hanya kepada *user* yang memiliki kredensial valid, seperti *admin*, dengan cara memverifikasi *username* dan *password*. Setelah *login* berhasil, *admin* dapat mengakses *dashboard* yang memungkinkan pengelolaan

data sistem, termasuk penambahan, pengeditan, dan penghapusan akun *user locker*, serta pemantauan riwayat penggunaan *locker*. Sistem keamanan juga dilengkapi dengan mekanisme enkripsi pada *password*.



Gambar 9. Halaman Menu.

Pada Gambar 9 Merupakan halaman menu utama sistem *smart locker* dengan empat fitur yang dirancang untuk mendukung pengelolaan *locker* secara efisien dan aman, Menu *check-in* berfungsi mencatat data *user* seperti nama, nomor *handphone*, nomor *locker*, dan waktu peminjaman secara *real-time* untuk memudahkan pemantauan ketersediaan *locker*. Menu *check-out* berfungsi untuk mencatat pengambilan barang dari *locker* dengan menyimpan data identitas *user* dan waktu pengambilan, sehingga memastikan hanya *user* yang berwenang dapat mengakses *locker*. Menu *blacklist* digunakan *admin* memblokir akses *user* yang melanggar kebijakan sistem, seperti melebihi batas waktu peminjaman atau melakukan penyalahgunaan. Sementara itu, menu *account data* menyimpan informasi akun *admin* yang memiliki hak akses terhadap aplikasi LOKERPINTARPOLNEP, mencakup data *login* seperti *username*, *password*, serta kewenangan untuk mengelola sistem dan *user*. sistem diuji pada 24 unit loker dengan durasi penggunaan maksimal 10 jam.

2.2 Sumber Data

Sumber data utama dalam penelitian ini diperoleh melalui:

1. Data primer merupakan hasil observasi langsung terhadap sistem *automatic smart locker* yang dirancang dan diuji oleh peneliti selama penelitian untuk memperoleh informasi faktual dan kontekstual dari lapangan [11].
2. Data sekunder merupakan data yang membantu memperkuat dasar teori dalam penelitian [12]. Data sekunder biasanya diperoleh dari literatur terdahulu, publikasi

ilmiah, laporan resmi, atau sumber terpercaya lainnya yang relevan dengan topik penelitian.

2.3 Pengumpulan Data

Data pada *automatic smart locker* menggunakan metode:

1. Observasi dengan mengamati langsung proses kerja sistem *Automatic Smart Locker* selama implementasi dan pengujian, mencakup *respons* LCD Nextion HMI saat *check-in* dan *check-out* serta data yang dikirim dan diterima oleh aplikasi *web native*. membantu peneliti memahami interaksi nyata antara *user*, sistem, dan perangkat keras yang digunakan [13].
2. Studi Dokumentasi digunakan untuk memperoleh informasi seperti *datasheet* modul Arduino Mega 2560 dan modul ESP8266, struktur dan konfigurasi *database* Laragon, serta kode program yang diimplementasikan untuk data primer dan memperkuat validitas penelitian dengan data dari sumber terpercaya [14].
3. Wawancara Informal dilakukan kepada dosen pembimbing, *admin* jurusan elektro, serta rekan perkuliahan untuk mendapatkan masukan dan saran dalam pengembangan sistem antarmuka, konektivitas, dan fungsionalitas sistem. Wawancara informal berguna dalam konteks penelitian pengembangan untuk pengumpulan opini atau pengalaman dari *user* dan ahli secara fleksibel [15].

2.4 Analisis Data

Analisis data dalam penelitian menggunakan pendekatan deskriptif kualitatif, yaitu dengan mendeskripsikan performa sistem *Automatic Smart Locker* berdasarkan hasil observasi langsung dan pengujian fungsional selama proses implementasi. Adapun analisis mencakup tiga aspek utama, yaitu:

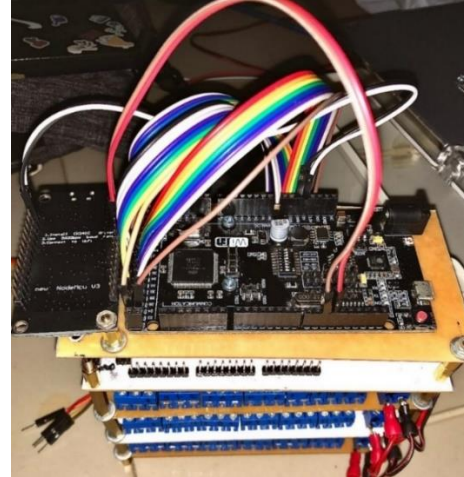
1. Kinerja perangkat keras seperti, *locker*, mikrokontroler, LCD Nextion HMI dalam merespons perintah sistem, termasuk buka-tutup *locker*, stabilitas komunikasi serial antara mikrokontroler, aplikasi *web native* dan LCD Nextion HMI, serta daya tahan perangkat selama pengujian berulang.
2. Fungsionalitas perangkat lunak *web server*, *database*, bot Telegram mengirimkan notifikasi. Evaluasi dilakukan berdasarkan akses antarmuka oleh *admin*, waktu

respon aplikasi saat proses *check-in* dan *check-out*, serta validitas data yang disimpan dan ditampilkan.

3. Integrasi sistem secara keseluruhan dalam skenario penggunaan actual. Pengujian pengiriman notifikasi otomatis ke *admin* melalui aplikasi Telegram.

3. HASIL DAN PEMBAHASAN

Berikut ini adalah implementasi perangkat keras pada *Automatic Smart Locker*.



Gambar 10. Implementasi Rangkaian Mikrokontroler

Pada Gambar 10 Merupakan implementasi perangkat keras dari sistem *Automatic Smart Locker* yang mengintegrasikan mikrokontroler ESP8266 dengan Arduino Mega 2560. Arduino Mega sebagai kendali utama yang mengontrol modul eksternal seperti modul relay yang berfungsi untuk mengontrol mekanisme kunci elektronik pada sistem *smart locker*. Modul ESP8266 bertugas sebagai modul komunikasi WiFi. Pada sistem pin TX ESP8266 terhubung ke pin 10 Arduino Mega dan pin RX ESP8266 ke pin 11 Arduino Mega untuk komunikasi dua arah.

Rangkaian bekerja dari instruksi yang diterima ESP8266 mengirimkan ke Arduino Mega melalui komunikasi serial, yang memproses untuk mengaktifkan atau menonaktifkan relay sesuai kebutuhan. Sebaliknya, jika Arduino Mega mengirimkan data status ke *server*, akan mengirimkan informasi tersebut ke ESP8266 yang kemudian meneruskannya melalui koneksi Wi-Fi. Dengan konfigurasi ini, sistem *smart locker* dapat dimonitor secara *real-time* dari lokasi mana pun tanpa perlu terhubung langsung ke perangkat secara fisik.

Berikut merupakan hasil serial monitor pada Arduino Mega 2560

```
Data check-in yang diterima: aaaaa,0855555,12346,1,
Nama: aaaaa
WhatsApp: 08555555
Password: 12346
Nomor Loker: 1
Check-in berhasil:
Nama: aaaaa
Nomor loker: 1
Mengirim data ke ESP: aaaaa,0855555,12346,1
Loker 1 aktif! (selama 15 detik)
Loker 1 mati!
Pesan diterima dari HMI: Checkout,aaaaa,12346,
Nama: aaaaa
Password: 12346
Loker 1 siap digunakan kembali.
Loker 1 aktif! (selama 15 detik)
```

Gambar 11. Hasil Serial Monitor pada Arduino Mega 2560.

Pada Gambar 11 menunjukkan komunikasi antara sistem *smart locker* dengan antarmuka HMI (Human Machine Interface). sistem siap menerima data dari HMI, kemudian menerima perintah "*check-out*" dengan data nama "aaaa" dan *password* "12346", tetapi gagal karena *user* belum melakukan *check-in* sebelumnya. Setelah itu, sistem menerima perintah "*check-in*" dengan data nama "aaaa", nomor WhatsApp "0855555", *password* "12346", dan loker nomor 1, yang berhasil diproses. Sistem mengirimkan data *check-in* ke ESP, dan loker 1 aktif selama 15 detik sebelum mati. Selanjutnya, sistem kembali menerima perintah "*check-out*" dari HMI dengan data yang sama, dan kali ini berhasil karena *user* terdaftar dalam sistem. Loker 1 dinyatakan siap digunakan kembali, kemudian aktif lagi 15 detik sebelum mati.

Berikut tampilan hasil serial monitor pada ESP8266 dan *server* dalam sistem automatic *smart locker*

```
Sending data to server: data=
Data sent successfully. HTTP Response code: 500
Server response:
Received data from Arduino: sss,088888,321,1
Sending data to server: data=sss,088888,321,1
Data sent successfully. HTTP Response code: 500
Server response:
Received data from Arduino:
Sending data to server: data=
Data sent successfully. HTTP Response code: 500
Server response:
Received data from Arduino:
Sending data to server: data=
Data sent successfully. HTTP Response code: 500
Server response:
```

Gambar 12. Hasil Serial Monitor pada ESP8266.

Pada Gambar 12 menunjukkan komunikasi antara sistem *smart locker* dengan *server* melalui HTTP. Data dikirim dari Arduino ke *server* dalam *format* tertentu, yang berisi informasi *user* atau status *locker*. Meskipun data dikirim dengan sukses, *server* memberikan HTTP *Response Code* 500, yang menunjukkan adanya kesalahan internal pada *server*, seperti kesalahan

skrip atau ketidakmampuan menangani permintaan dengan benar. Siklus ini terus berulang, dengan Arduino mengirimkan data, menerima respons gagal dari *server*, dan mengulangi prosesnya. Kesalahan dapat disebabkan oleh konfigurasi *server* yang salah, endpoint yang tidak tersedia, atau format data tidak sesuai dengan yang diharapkan oleh *server*.

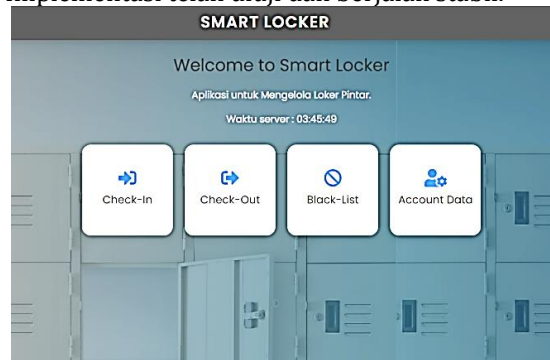
Implementasi *Software* untuk rancangan antarmuka pada aplikasi yang dibuat peneliti, dibangun menggunakan PHP Laragon dan MySQL Aplikasi LOKERPINTARPOLNEP merupakan sistem berbasis *web*. Aplikasi ini dirancang khusus untuk keperluan pengelolaan *smart locker* oleh *admin* secara efisien dan aman.



Gambar 13. Halaman Login

Pada Gambar 13 menampilkan halaman awal antarmuka aplikasi, yaitu Halaman Login. Menyediakan dua kolom yaitu *username* dan *password*, yang digunakan untuk memverifikasi identitas *admin* sebelum memberikan akses ke sistem. Terdapat fitur "*Show password*" yang memudahkan *admin* dalam memastikan kebenaran karakter sandi, serta tombol "*Login*" akan memproses data yang dimasukkan dan mencocokkannya dengan informasi yang tersimpan di *database* MySQL.

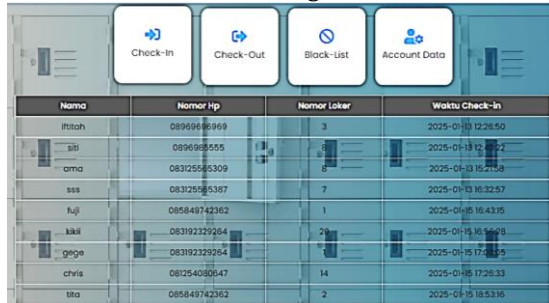
Tampilan halaman didesain secara sederhana dan minimalis agar fokus *admin* tetap terjaga selama proses otentikasi, tanpa distraksi dari elemen visual yang tidak diperlukan. Implementasi telah diuji dan berjalan stabil.



Gambar 14. Halaman Menu

Pada Gambar 14 menampilkan halaman menu pada sistem *Smart Locker* yang hanya diakses *admin*. Tampilan ada 4 fitur utama *check-*

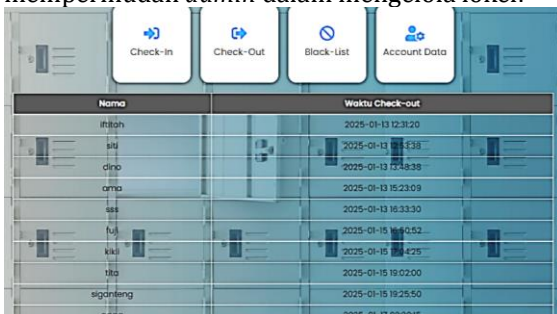
in, check-out, blacklist, dan account data. Setiap fitur dilengkapi ikon yang intuitif dan informasi waktu *server real-time* dalam pendataan secara akurat. Desain latar belakang gambar *locker* menciptakan kesan indah serta memudahkan *admin* memantau dan mengontrol secara efisien.



Nama	Nomor Hp	Nomor Locker	Waktu Check-in
ititah	08969889969	3	2025-01-13 12:28:50
iti	08969889555	6	2025-01-13 12:28:52
ama	08325565309	8	2025-01-13 15:23:58
ssa	08325565387	7	2025-01-13 16:32:57
faji	085849742362	1	2025-01-15 16:43:15
kiki	083192329264	2	2025-01-15 16:50:08
gege	083192329264	9	2025-01-15 17:00:05
chris	08254080647	14	2025-01-16 17:26:33
titia	085849742362	2	2025-01-16 18:53:16

Gambar 15. Halaman *Check-In*

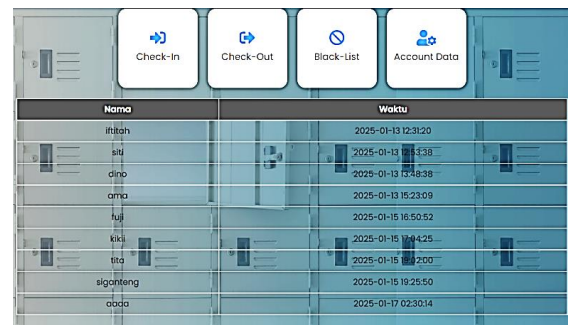
Pada Gambar 15 menampilkan Halaman *check-in* pada *smart locker* untuk mendata *user*. Menampilkan tabel dengan empat kolom utama yaitu, Nama, Nomor *handphone*, Nomor *Locker*, dan Waktu *check-in*. Sehingga dapat membantu *admin* dalam *monitoring* aktivitas penyimpanan secara akurat dan terorganisir. Hasil pengujian menunjukkan bahwa sistem berhasil mencatat data *check-in user* secara *real-time* dan menampilkannya secara terstruktur. Aplikasi *user interface* yang dilengkapi menu *check-in*, *check-out*, *blacklist* dan *account data* mempermudah *admin* dalam mengelola *locker*.



Nama	Waktu Check-out
ititah	2025-01-13 12:31:20
iti	2025-01-13 12:43:38
dino	2025-01-13 12:48:38
ama	2025-01-13 15:23:09
ssa	2025-01-13 16:33:30
faji	2025-01-15 16:40:52
kiki	2025-01-15 16:42:25
titia	2025-01-15 19:02:00
siganteng	2025-01-15 19:25:50
adida	2025-01-17 02:20:15

Gambar 16. Halaman *Check-Out*

Pada Gambar 16 menampilkan halaman *check-out* pada sistem *smart locker* yang berfungsi untuk mencatat aktivitas *check-out* dari *locker* oleh *user*. Dengan adanya pendataan, *admin* dengan mudah melakukan *monitoring* terhadap setiap transaksi yang terjadi pada sistem *smart locker*. Selain itu, pendataan juga berguna dalam hal pelacakan apabila terjadi permasalahan terkait penggunaan *locker*, seperti barang yang belum diambil dalam jangka waktu tertentu atau adanya laporan kehilangan.



Nama	Waktu
ititah	2025-01-13 12:31:20
iti	2025-01-13 12:43:38
dino	2025-01-13 12:48:38
ama	2025-01-13 15:23:09
faji	2025-01-15 16:50:52
kiki	2025-01-15 17:04:25
titia	2025-01-15 17:02:00
siganteng	2025-01-15 19:25:50
adida	2025-01-17 02:20:14

Gambar 17. Halaman *Blacklist*

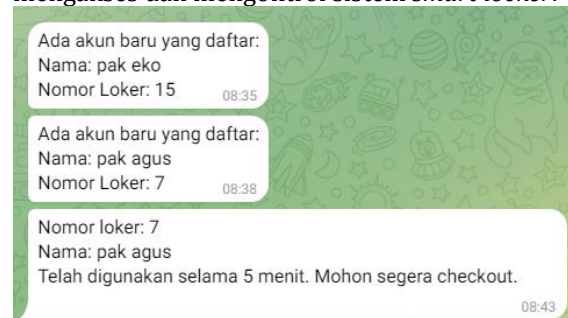
Pada Gambar 17 menampilkan Halaman *blackList* berfungsi untuk mencatat dan mengelola daftar *user* yang diblokir dari akses penggunaan *locker*. Fitur ini dirancang untuk membantu *admin* mencegah penyalahgunaan sistem oleh *user* yang melanggar aturan, seperti melebihi batas waktu penggunaan. Pada halaman menampilkan informasi seperti Nama, tanggal dan waktu pemblokiran, *user* yang dimasukkan ke dalam daftar tersebut.



Nama	Username	Aksi
admin	admin	Edit Password Hapus
ititah Wulinda Fujiarti	ititah	Edit Password Hapus
Devil Aulia	devil	Edit Password Hapus

Gambar 18. Halaman *Account Data*

Pada Gambar 18 menampilkan Halaman *account data* pada sistem *smart locker* dirancang untuk mengelola akun *admin* yang memiliki akses terhadap aplikasi. Fitur dapat menambah, menghapus, atau memperbarui data akun *admin* lainnya. Informasi yang ditampilkan pada halaman meliputi Nama *admin*, *username* dan aksi yang berfungsi sebagai akun yang berwenang menggunakan aplikasi. Dengan sistem ini, hanya akun yang terdaftar yang dapat mengakses dan mengontrol sistem *smart locker*.



Message
Ada akun baru yang daftar: Nama: pak eko Nomor Loker: 15 08:35
Ada akun baru yang daftar: Nama: pak agus Nomor Loker: 7 08:38
Nomor loker: 7 Nama: pak agus Telah digunakan selama 5 menit. Mohon segera checkout. 08:43

Gambar 19. Halaman *Login*

Pada Gambar 19 menampilkan pesan yang memberikan informasi terkait pendaftaran akun

baru serta penggunaan *locker* dalam sistem *smart locker*. Pada percobaan 1 pukul 08:35, sistem mencatat bahwa seorang *user* bernama Pak Eko telah mendaftar dan diberikan akses ke *locker* nomor 15. Selanjutnya, pada percobaan 2 pukul 08:38, *user* bernama Pak Agus juga mendaftar dan mendapatkan *locker* nomor 7. Kemudian, pada pukul 08:43, sistem mengirimkan pemberitahuan bahwa loker nomor 7 yang digunakan oleh Pak Agus telah dipakai selama lima menit dan meminta *user* untuk segera melakukan check-out. Sebelum melakukan percobaan, kode program pada *smart locker* diganti dari 10 jam menjadi 5 menit bertujuan untuk memaksimalkan waktu pada saat sidang. Notifikasi berfungsi sebagai pemantauan dan pengingat bagi *user* serta *admin* dalam mengelola penggunaan loker secara efisien.

Pengujian sistem *smart locker* bertujuan untuk memastikan fungsi sistem beroperasi sesuai yang telah direncanakan. Salah satu metode yang digunakan adalah *User Acceptance Testing* (UAT), yang bertujuan untuk memastikan sistem tidak hanya sesuai dengan spesifikasi, tetapi mampu memenuhi kebutuhan *user* secara nyata. Pengujian UAT merupakan pengujian *black-box* untuk menemukan fungsi yang tidak benar, kesalahan antarmuka, kesalahan struktur data, kesalahan performansi, kesalahan inisialisasi dan terminasi [16].

No.	Skenario Pengujian	Pengujian (Username & Password)	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	Username & password kosong kemudian klik tombol masuk	(kosong) & (kosong)	Tetap di halaman login dan menampilkan pesan wajib diisi	✓	Sesuai
2.	Mengisi username tetapi password kosong kemudian klik tombol masuk	username : ifitah password: (kosong)	Tetap di halaman login dan menampilkan pesan wajib diisi pada password	✓	Sesuai
3.	Mengosongkan username dan mengisi password 123456 kemudian klik tombol masuk	username: (kosong) password: 123456	Tetap di halaman login dan menampilkan pesan wajib diisi pada username	✓	Sesuai
4.	Mengisi username dan password yang sesuai kemudian klik tombol masuk	username: ifitah password: 123456	Sistem akan mengizinkan akses login kemudian masuk ke halaman dashboard	✓	Sesuai
5.	Mengisi username dan password yang tidak sesuai dengan data kemudian klik tombol masuk	username: ifitah21 Password: 12345678	Sistem akan menampilkan pesan error dan tetap berada di halaman login	✓	Sesuai

Gambar 20. Pengujian Halaman Login

Pada Gambar 20 Pengujian pada fitur *login* untuk memastikan validasi *input* dan autentikasi berjalan sesuai spesifikasi tanpa melihat struktur internal sistem. Hasil pengujian menunjukkan bahwa sistem mampu menolak *input* kosong atau

tidak lengkap, serta memberikan umpan balik berupa pesan kesalahan yang sesuai. Ketika kredensial yang dimasukkan benar, sistem berhasil mengarahkan *user* ke halaman *dashboard*, menunjukkan bahwa autentikasi berjalan dengan benar. Sebaliknya, jika hanya salah satu dari *username* atau *password* yang benar, sistem tetap menolak akses, menandakan mekanisme keamanan berfungsi baik. Secara keseluruhan, fitur *login* telah memenuhi standar fungsional dalam pengujian *black-box*.

No.	Role	Skenario Pengujian	Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	A D M I N	Pada grafik klik tabel checkin	Export to database	Menampilkan data pada tabel checkin	✓	Sesuai
2.		Pada grafik klik tabel checkout		Menampilkan data pada tabel checkout	✓	Sesuai
3.		Pada grafik klik data blacklist		Menampilkan data pada tabel blacklist	✓	Sesuai
4.		Pada grafik klik tabel account data		Menampilkan data pada tabel account data	✓	Sesuai

Gambar 21. Pengujian Halaman Menu

Pada Gambar 21 menunjukkan hasil pengujian fitur menu ke *database*, yaitu tabel *check-in*, *check-out*, *blacklist*, dan *account data*. Setiap skenario pengujian dilakukan dengan mengklik *icon*, dan hasil yang diharapkan adalah sistem mampu menampilkan data dengan benar. Berdasarkan hasil pengujian, semua fitur berhasil menampilkan data sesuai harapan, menunjukkan bahwa sistem berfungsi dalam pengambilan dan penyajian data. Dengan demikian, pengujian membuktikan bahwa seluruh fitur ekspor data berjalan dengan stabil sesuai spesifikasi.

No.	Role	Skenario Pengujian	Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	A D M I N	Idlik tabel checkin	Export to database	Menampilkan data pada tabel checkin seperti Nama, Nomor HP, Nomor Locker dan Waktu server ketika loker digunakan oleh pengguna	✓	Sesuai

Gambar 22. Pengujian Halaman Check-In

Pada Gambar 22 menunjukkan hasil pengujian fitur ekspor data pada tabel *check-in*. Pengujian dilakukan dengan mengklik tabel *check-in* untuk memastikan sistem dapat menampilkan data *user* seperti nama, nomor *handphone*, nomor *locker*, dan waktu *server* saat *locker* digunakan. Hasil pengujian menunjukkan seluruh data berhasil ditampilkan dengan benar sesuai format yang diharapkan. Menegaskan bahwa fitur ini berfungsi dengan baik. Dengan

demikian, sistem dinyatakan memenuhi standar fungsional untuk proses *check-in*.

No.	Role	Skenario Pengujian	Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	ADMIN	Klik tabel checkout	Export to database	Menampilkan data pada tabel checkout seperti Nama dan Waktu server ketika locker digunakan	✓	Sesuai

Gambar 23. Pengujian Halaman *Check-Out*

Pada Gambar 23 menunjukkan hasil pengujian pada fitur ekspor data dari tabel *check-out*. Pengujian dilakukan dengan mengklik tabel *check-out* dan memverifikasi apakah sistem mampu menampilkan data seperti nama *user* dan waktu *server* ketika *locker* digunakan. Hasil pengujian menunjukkan bahwa data ditampilkan dengan benar sesuai harapan. Menunjukkan bahwa fitur berfungsi dengan baik. Dengan demikian, sistem berhasil memenuhi kebutuhan fungsional pada bagian *check-out*.

No	Role	Skenario Pengujian	Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	ADMIN	Klik tabel blacklist	Export to database	Menampilkan data pada tabel blacklist seperti Nama dan Waktu server ketika locker digunakan	✓	Sesuai

Gambar 24. Pengujian Halaman *Blacklist*

Pada Gambar 24 menunjukkan hasil pengujian fitur ekspor data pada tabel *blacklist*. Pengujian dilakukan dengan mengklik tabel *blacklist* untuk memverifikasi apakah sistem dapat menampilkan data seperti nama pengguna dan waktu *server* saat locker digunakan melebihi batas waktu. Hasil pengujian menunjukkan bahwa data berhasil ditampilkan sesuai harapan. Menandakan bahwa sistem telah berfungsi secara optimal dalam menangani data *user* yang masuk ke daftar *blacklist*. Dengan demikian, fitur ini dinyatakan lulus uji fungsional.

No.	Role	Skenario Pengujian	Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	ADMIN	Klik tabel blacklist	Export to database	Menampilkan data pada tabel account data seperti Nama dan Waktu server pengguna	✓	Sesuai
2		Klik edit password	Ke halaman ubah password	Menampilkan halaman untuk ubah password	✓	Sesuai
3		Klik hapus	Tetap berada dihalaman	Menghapus account data admin	✓	Sesuai

Gambar 25. Pengujian Halaman *Account Data*

Pada Gambar 25 menampilkan hasil pengujian terhadap fitur manajemen *account*

data admin pada sistem. Pengujian pertama dilakukan dengan mengklik tabel *blacklist* untuk memverifikasi data seperti nama dan waktu *ode user* yang ditampilkan dengan benar, dan hasilnya sesuai. Pengujian kedua menguji fungsi edit *password*, di mana sistem berhasil menampilkan halaman untuk mengubah kata sandi. Pengujian ketiga dilakukan dengan mengklik tombol hapus, dan sistem tetap berada di halaman yang sama sambil berhasil menghapus data akun *admin*. Seluruh skenario menunjukkan hasil sesuai harapan sehingga dapat disimpulkan bahwa fitur-fitur manajemen akun telah berfungsi dengan baik.

No.	role	Skenario Pengujian	Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	Admin	Waktu ketika pengguna checkin	Fitur notifikasi Telegram	Notifikasi masuk dan mengirimkan pesan sesuai dengan format	✓	Sesuai
2		Waktu ketika pengguna menggunakan locker lebih dari 10 jam	Fitur notifikasi Telegram	Notifikasi masuk mengirimkan pesan sesuai dengan format	✓	Sesuai

Gambar 26. Pengujian Notifikasi Telegram

Gambar tersebut menampilkan hasil pengujian fitur notifikasi Telegram oleh *admin* dalam dua skenario berbeda. Pada skenario pertama, sistem diuji untuk memastikan bahwa saat *user* melakukan *check-in*, notifikasi berhasil masuk dan mengirimkan pesan sesuai format yang telah ditentukan. Hasil pengujian menunjukkan fungsi berjalan dengan baik. Skenario kedua menguji kondisi ketika *user* menggunakan *locker* lebih dari 10 jam, dan sistem berhasil mengirimkan notifikasi otomatis kepada *admin*, juga sesuai dengan format pesan yang ditetapkan. Kedua skenario memperoleh hasil yang sesuai. Menunjukkan bahwa fitur notifikasi Telegram bekerja secara optimal dalam berbagai kondisi penggunaan sistem.

Metode pengujian teknik *System Usability Scale* (SUS) bertujuan untuk mengevaluasi tingkat kemudahan penggunaan sistem. Pengujian dengan metode kuesioner *System Usability Scale* (SUS) memiliki lima pilihan jawaban. Responden dapat memilih dari "Sangat Tidak Setuju" yang bernilai 1 poin, "Tidak Setuju" bernilai 2 poin, "Netral/Ragu-ragu" bernilai 3 poin, "Setuju" bernilai 4 poin, hingga "Sangat Setuju" dengan nilai 5 poin. Aturan dalam metode SUS adalah untuk pertanyaan bernomor ganjil, skornya dihitung dengan rumus $(\text{Skor jawaban responden} - 1)$, sedangkan untuk pertanyaan

bernomor genap, dihitung menggunakan rumus (5 – Skor jawaban responden).

No.	Pertanyaan (Question)	Sangat Tidak Setuju [1]	Tidak Setuju [2]	Ragu-ragu [3]	Setuju [4]	Sangat Setuju [5]
1.	Saya merasa aplikasi LOKERPINTARPOLNEP mudah digunakan.					
2.	Saya merasa ada banyak aspek dari aplikasi LOKERPINTARPOLNEP yang kurang dipahami.					
3.	Saya merasa mudah menggunakan aplikasi LOKERPINTARPOLNEP untuk mengelola data pengguna locker.					
4.	Saya memerlukan bantuan orang lain untuk bisa menggunakan aplikasi LOKERPINTARPOLNEP dengan baik.					
5.	Fitur-fitur di aplikasi LOKERPINTARPOLNEP terintegrasi dengan baik.					
6.	Fitur-fitur di aplikasi LOKERPINTARPOLNEP tidak sesuai.					
7.	saya merasa aplikasi LOKERPINTARPOLNEP mudah dipelajari dan digunakan orang lain.					
8.	saya merasa aplikasi LOKERPINTARPOLNEP membingungkan dan harus mempelajari banyak hal sebelum menggunakannya dengan efektif.					
9.	saya merasa aplikasi LOKERPINTARPOLNEP sangat berguna dalam memonitoring dan mengontrol Smart Locker secara real-time.					
10.	saya merasa ada banyak hal yang tidak konsisten saat menggunakan aplikasi LOKERPINTARPOLNEP.					

Gambar 27. Pengujian SUS (System Usability Scale)

Pada Gambar 27 merupakan hasil pengujian SUS (System Usability Scale) untuk menilai aspek *usability* secara keseluruhan, mencakup efektivitas, efisiensi, serta tingkat kepuasan subjektif yang dirasakan oleh *user*. Pada penelitian ini, pengukuran SUS mengacu pada pengalaman *user* aplikasi. Data diperoleh melalui survei daring menggunakan *Google Forms*, dengan responden berjumlah 10 orang yang terdiri dari *user* aplikasi, *admin* dan *user*.

Proses analisis dilakukan berdasarkan hasil kuisioner yang diolah menggunakan langkah-langkah perhitungan sebagai berikut:

1. Untuk pertanyaan dengan nomor ganjil (1, 3, 5, 7, dan 9), skor yang diperoleh dari jawaban dikurangi 1.
2. Untuk pertanyaan dengan nomor genap (2, 4, 6, 8, dan 10), skor dihitung dengan mengurangkan nilai jawaban dari 5.
3. Hasil dari setiap pertanyaan kemudian dikalikan 2,5 untuk mendapatkan total skor dalam rentang 10-100.

4. Setelah skor individu masing-masing responden diperoleh, langkah berikutnya adalah menghitung rata-rata skor dengan menjumlahkan seluruh skor responden, lalu membaginya dengan jumlah responden.

Data penilaian dari responden terhadap aplikasi ini telah dihitung dan hasilnya disajikan Pada Gambar 28 terkait Hasil Kuisioner SUS.

No.	Responden	Skor Hasil Hitung										jumlah	Nilai (jumlahx2.5)
		Q 1	Q 2	Q 3	Q 4	Q 5	Q 6	Q 7	Q 8	Q 9	Q 10		
1	Respon 1	5	2	5	2	4	2	5	2	5	3	35	87,5
2	Respon 2	4	2	4	2	3	2	5	2	4	2	30	75
3	Respon 3	4	3	4	3	4	3	2	4	3	1	31	77,5
4	Respon 4	5	4	5	4	5	1	4	2	5	4	39	97,5
5	Respon 5	4	3	4	5	4	3	3	5	3	3	37	92,5
6	Respon 6	4	3	4	3	4	3	4	3	4	3	35	87,5
7	Respon 7	5	1	5	1	5	1	5	1	5	1	31	77,5
8	Respon 8	5	3	4	4	2	5	4	3	4	3	37	92,5
9	Respon 9	5	3	3	4	4	3	3	4	4	5	38	95
10	Respon 10	5	3	4	2	5	2	4	3	5	2	35	87,5
Skor Rata-rata													94,75

Gambar 28. Hasil Kuesioner SUS

Hasil penilaian dari 10 responden, yang terdiri atas 7 *user* dan 3 *admin* yang berpartisipasi dalam penelitian, menunjukkan rata-rata skor SUS sebesar 94,75. Nilai diperoleh dari total skor keseluruhan sebesar 947,5 kemudian dibagi dengan jumlah responden, yaitu 10 orang. Skor rata-rata dianalisis menggunakan *acceptability ranges*, *grade scale*, dan *adjective ratings*.

Berdasarkan analisis, skor SUS sebesar 94,75 dapat disimpulkan sebagai berikut:

1. Dalam kategori *acceptability ranges* (tingkat penerimaan), skor ini berada pada tingkat *marginal high*.
2. Dalam *grade scale*, skor ini termasuk dalam kategori B.
3. Dalam *adjective ratings*, skor ini tergolong *Good*.

Interpretasi skor SUS mengacu pada Jeff Sauro dengan menggunakan grade dari A hingga F serta peringkat persentil untuk menunjukkan tingkat *usability* [17].

Ketentuan untuk *percentile rank* dan *letter grades* adalah sebagai berikut:

1. Grade A: nilai ≥ 80.3 , *percentile* $\geq 90\%$
2. Grade B: $74 \leq$ nilai < 80.3 , $70\% \leq$ *percentile* $< 80\%$
3. Grade C: $68 \leq$ nilai < 74 , $40\% \leq$ *percentile* $< 70\%$
4. Grade D: $51 \leq$ nilai < 68 , $20\% \leq$ *percentile* $< 40\%$
5. Grade F: nilai < 51 , *percentile* $< 20\%$

Berdasarkan skor SUS sebesar 94,75 aplikasi LOKERPINTARPOLNEP berada pada kategori *Grade A* dengan persentase *percentile*

90%. Hal ini menunjukkan bahwa aplikasi tersebut dapat diterima oleh *admin* elektro. Selain itu, LOKERPINTARPOLNEP juga berpotensi menjadi alat pendukung dalam memantau dan mengelola Locker, sehingga proses penggunaan locker menjadi lebih efektif dan efisien.

4. KESIMPULAN DAN SARAN

4.1 KESIMPULAN

Berdasarkan hasil analisis yang telah dipaparkan pada bab-bab sebelumnya, kesimpulan yang dapat diambil antara lain sebagai berikut:

1. Sistem Monitoring dan Kontrol berbasis *Internet of Things* pada Aplikasi Web yang diberi nama LOKERPINTARPOLNEP telah dikembangkan dan memiliki tampilan yang responsif. Sistem ini menggunakan PHP *web native* dan *JavaScript* pada platform Laragon dan database MySQL, serta dilengkapi fitur notifikasi Telegram untuk memenuhi kebutuhan *user* sebagai *admin*.
2. Aplikasi LOKERPINTARPOLNEP telah meningkatkan efisiensi dan efektivitas para *user* dalam menggunakan loker serta *admin* yang dapat memonitor dan mengontrol penggunaan loker secara *real-time*. Selain itu, proses monitoring dan kontrol dapat dilakukan dari mana saja melalui aplikasi berbasis web yang memanfaatkan teknologi *Internet of Things*. Hal ini dibuktikan melalui pengukuran menggunakan metode System Usability Scale (SUS) dan uji coba oleh *admin* elektro.
3. Meskipun sistem telah berjalan dengan baik, masih terdapat beberapa keterbatasan yang perlu diperhatikan, seperti belum tersedianya fitur sinkronisasi waktu otomatis antara antarmuka *web* dan perangkat keras, serta keterbatasan pada *user interface*. Selain itu, aplikasi berbasis *web* saat ini hanya dapat diakses oleh *admin*, sehingga belum mendukung fleksibilitas optimal pada perangkat *mobile*. Keterbatasan ini menjadi dasar pertimbangan dalam pengembangan sistem yang lebih adaptif di masa mendatang.

4.2 SARAN

Adapun beberapa saran yang dapat membantu dalam mengembangkan sistem

LOKERPINTARPOLNEP antara lain sebagai berikut:

1. Sistem LOKERPINTARPOLNEP masih perlu dioptimalkan dari segi waktu dengan menambahkan sensor RTC untuk menyinkronkan waktu antara aplikasi *web* dan *hardware*. Hal ini bertujuan untuk mendukung pencatatan waktu penggunaan loker yang lebih akurat dan optimal.
2. Tampilan aplikasi masih sederhana dan fitur yang terbatas perlu ditingkatkan. Fitur tambahan yang dapat dipertimbangkan adalah fitur penghapusan atau pembaruan status *blacklist*, serta fitur pemantauan barang yang disimpan dalam *locker*.
3. Berdasarkan hasil survei, aplikasi dinilai cukup menarik, mudah digunakan dan dipahami oleh *user*, khususnya *admin*.
4. Pengembangan aplikasi dalam versi *mobile* dilengkapi dengan notifikasi *real-time* sangat disarankan. Fitur ini akan memudahkan *user* menerima informasi dan peringatan langsung melalui perangkat *mobile*, sehingga lebih responsif.

5. UCAPAN TERIMA KASIH

Puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat dan hidayahnya sehingga penelitian ini dapat terselesaikan. Untuk itu, diucapkan terimakasih sebesar-besarnya kepada :

1. Kedua orang tua dan saudara kandung penulis yang senantiasa memberikan doa, dukungan moral, dan bantuan finansial.
2. Dosen Pembimbing pertama dari Program Studi Teknologi Rekayasa Sistem Elektronika, Jurusan Teknik Elektro, Politeknik Negeri Pontianak
3. Dosen Pembimbing kedua dari Program Studi Teknologi Rekayasa Sistem Elektronika, Jurusan Teknik Elektro, Politeknik Negeri Pontianak
4. Dosen Penguji pertama Pendamping dari Program Studi Teknologi Rekayasa Sistem Elektronika, Jurusan Teknik Elektro, Politeknik Negeri Pontianak
5. Dosen Penguji kedua Sistem Elektronika, Jurusan Teknik Elektro, Politeknik Negeri Pontianak

DAFTAR PUSTAKA:

- [1] A. Pradana, "Perancangan Kunci Elektronik Menggunakan RFID," *Jurnal*

- Teknologi dan Rekayasa, vol. 5, no. 2, pp. 34–38, 2020.
- [2] R. Ahmad, A. Wagya, dan P. D. Junianto, "Perancangan dan Implementasi Sistem Smart Locker dengan Face Recognition Berbasis Internet of Things: Penerapan Teknologi Pengenal Wajah dalam Sistem Smart Locker," *Spektral*, vol. 5, no. 1, pp. 239–245, 2024.
- [3] D. Sasmoko, R. Veliyanti, dan S. A. Wijayanto, "Sistem Keamanan Kotak Paket Cerdas Berbasis IoT dengan Teknologi MQTT," *LogicLink*, vol. 1, no. 2, pp. 166–179, 2024.
- [4] M. Nahdi, A. Huda, dan T. Rachmawati, "Internet of Things: Konsep dan Implementasi," *Jurnal Teknologi Informasi*, vol. 3, no. 1, pp. 21–28, 2021.
- [5] CH. M. Shruthi, S. K. Bandari, C. K. Reddy Ala, dan M. Reddy, "Locker Security System using Internet of Things", *E3S Web of Conferences*, vol. 391, Art. No. 01153, 2023.
- [6] S. Fuada *et al.*, "Pemanfaatan Internet of Things untuk Monitoring Persemaian Tanaman Hidroponik," *Info Tech Journal*, vol. 9, pp. 40–45, 2023.
- [7] F. Feta *et al.*, "Pembangunan Aplikasi Digital Library Berbasis Web Menggunakan PHP Native," *Journal of Social Science Research*, vol. 4, no. 4, pp. 101–108, 2024.
- [8] R. Ananda *et al.*, "Perancangan Smart Locker dengan Implementasi Sistem IoT dan Aplikasi Mobile Android," *Jurnal Rekayasa dan Teknologi Elektro*, vol. 17, no. 1, pp. 55–61, 2023.
- [9] A. N. Iman dan S. Widiono, "Perancangan Aplikasi Smart Greenhouse Berbasis IoT untuk Optimalisasi Pertumbuhan Sayuran," *JIRE (Jurnal Informatika & Rekayasa Elektronika)*, vol. 7, no. 2, pp. 389–396, Nov. 2024.
- [10] A. N. A. Muhid, O. Rahmanto, A. Noor, and Fathurrahmani, "Sistem Early Warning Deteksi Kualitas Udara Berbasis IoT," *JIRE (Jurnal Informatika & Rekayasa Elektronika)*, vol. 7, no. 2, pp. 402–412, Nov. 2024.
- [11] U. Sulung dan M. Muspawi, "Memahami Sumber Data Penelitian: Primer, Sekunder, dan Tersier," **EDU RESEARCH**, vol. 5, no. 3, hlm. 110–116, 2024.
- [12] Faradiba Jabnabillah, Aswin Aswin, dan Mahfudz Reza Fahlevi, "Efektivitas Situs Web Pemerintah Sebagai Sumber Data Sekunder Bahan Ajar Perkuliahan Statistika," *Jurnal Sustainable*, vol. 6, no. 1, hlm. 59–70, 2023.
- [13] Fajar B. M., Lestary F. D., Hidayat A., Fadhilatunisa D., & Eka A., "Prototype Sistem Monitoring Pendeteksi dan Penyaringan Udara pada Ruangan Berbasis Internet of Things (IoT)," *Jurnal MediaTIK*, vol. 6, no. 2, pp. 1–8, 2024.
- [14] Y. S. Parihar, "Internet of Things and NodeMCU: A review of use of NodeMCU ESP8266 in IoT products," *Journal of Emerging Technologies & Innovative Research (JETIR)*, vol. 6, no. 6, pp. 1085–1093, 2020.
- [15] J. M. Swain dan Z. D. Spire, "The Role of Informal Conversations in Generating Data, and the Ethical and Methodological Issues," **Forum Qualitative Sozialforschung / Forum: Qualitative Social Research**, vol. 21, no. 1, Art. 3344, 2020.
- [16] D. Y. Wijaya dan W. M. Astuti, "Pengujian Black-box Sistem Informasi Penilaian Kinerja Karyawan PT INKA (Persero) Berbasis Equivalence Partitions," *Jurnal Digital Teknologi Informasi*, vol. 4, no. 1, 2021.
- [17] R. Y. Manurung, D. Krisbiantoro, dan D. A. B. Utami, "Usability Evaluation of Tokopedia Application Version 3.242 Using System Usability Scale (SUS) Method," *Sinkron: Jurnal dan Penelitian Teknik Informatika*, vol. 9, no. 1, pp. 366–374, 2024.