

BAB IV

HASIL DAN PEMBAHASAN

4.1. Hasil Penelitian

Pada bagian ini, akan dijelaskan hasil penelitian yang mencakup spesifikasi perangkat keras (hardware) dan perangkat lunak (software), tampilan antarmuka hasil deploy aplikasi. Bagian ini bertujuan untuk memberikan gambaran mendetail tentang teknologi yang digunakan, implementasi, dan hasil yang diperoleh dari aplikasi yang telah dikembangkan.

4.1.1. Spesifikasi Hardware dan Software yang Digunakan

Dalam pengembangan sistem Smart Access berbasis IoT, digunakan beberapa perangkat keras dan lunak yang saling terintegrasi untuk mendukung fungsi utama seperti autentikasi pengguna, monitoring status pintu, notifikasi real-time, dan kontrol aktuator. Rangkaian pada perangkat keras dibagi menjadi 2 yaitu, Rangkaian Akses Masuk dan Rangkaian akses Keluar.

Pada Tabel 4.1. merupakan daftar komponen utama yang digunakan dalam pembangunan Sistem *Smart Access* berbasis IoT. Tabel ini mencakup nama perangkat keras (*Hardware*), jumlah unit yang digunakan, serta fungsi masing – masing komponen keseluruhan sistem, baik sistem akses masuk ataupun akses keluar, termasuk dengan perangkat keras pendukung seperti, sensor, aktuator, dan sumber daya .

Tabel 4. 1. Tabel Perangkat Keras (Hardware)

| No. | Alat | Qty | Keterangan |
|-----|-----------------------|-------|---|
| 1 | ESP32 DevKit V1 | 1 pcs | Mikrokontroler utama sebagai pusat kontrol sistem, bertugas mengelola autentikasi pengguna akses masuk. Dilengkapi konektivitas WiFi dan banyak GPIO. |
| 2 | NodeMCU ESP8266 | 1 pcs | Mikrokontroler yang ditugaskan untuk mengontrol akses keluar serta mengelola koneksi internet untuk notifikasi dan pencatatan data ke Aplikasi Web. |
| 3 | Sensor RFID | 2 pcs | Digunakan untuk membaca kartu RFID pengguna sebagai metode autentikasi. Dua sensor digunakan untuk akses yaitu akses masuk dan akses keluar. |
| 4 | Sensor PIR | 1 pcs | Sensor ini digunakan untuk mendeteksi keberadaan manusia di depan pintu. Jika terdeteksi, mikrokontroler akan memicu proses autentikasi melalui pembacaan RFID. |
| 5 | Sensor MC38 | 1 pcs | Sensor switch magnetik yang dipasang pada pintu untuk mendeteksi apakah pintu dalam keadaan terbuka atau tertutup, membantu dalam mendeteksi terjadinya trespassing/pembobolan. |
| 6 | Layar OLED 128x64 I2C | 1 pcs | Layar tampilan untuk menampilkan informasi seperti status pintu, status akses ("Akses Diterima"/"Access Ditolak"), dan notifikasi lainnya. |
| 7 | Buzzer | 2 pcs | Pada akses masuk digunakan untuk memberikan umpan balik suara tambahan saat proses booting, ditolak, atau terjadi pelanggaran keamanan atau trespassing. Pada akses keluar digunakan untuk memberikan umpan balik saat rfid masuk atau ditolak. |
| 8 | Relay | 2 pcs | Mengendalikan aliran listrik ke solenoid pintu sesuai perintah dari mikrokontroler setelah autentikasi berhasil. |
| 9 | Solenoid | 1 pcs | Aktuator elektromagnetik yang fungsinya hanya membuka/mengunci pintu secara otomatis berdasarkan hasil autentikasi. |

| | | | |
|----|---------------------|------------|---|
| 10 | Speaker | 1 pcs | Digunakan untuk output suara seperti ucapan selamat datang, pintu terbuka, pintu tertutup atau peringatan suara ketika terjadi pelanggaran atau pembobolan. |
| 11 | Kabel Jumper | secukupnya | Digunakan untuk menghubungkan antara komponen satu dengan lainnya pada breadboard atau rangkaian. |
| 12 | BreadBoard | 1 pcs | Digunakan sebagai tempat untuk merangkai rangkaian elektronik akses keluar tanpa melakukan penyolderan. |
| 13 | DF Player Mini | 1 pcs | Modul pemutar audio yang dikendalikan oleh mikrokontroler ESP32 untuk akses masuk yang dapat digunakan untuk memainkan file suara (.mp3) melalui kartuSD |
| 14 | Step Down LM2596 | 3 pcs | Regulator tegangan untuk menurunkan tegangan input menjadi tegangan yang sesuai untuk komponen elektronik, agar bekerja stabil dan aman. Terdapat 3 pcs yaitu, untuk supply relay (5V), untuk supply solenoid (12V), dan untuk supply kedua mikrokontroler. |
| 15 | Power Supply 19v 3A | 1 pcs | Sumber daya utama untuk menjalankan seluruh sistem, termasuk kunci pintu solenoid. |

Perangkat Lunak (Software)

Pada Tabel 4.2. merupakan daftar perangkat lunak (software) yang digunakan dalam proses pengembangan dan implementasi sistem *Smart Access* berbasis IoT. Tabel ini mencakup kategori perangkat lunak yang meliputi backend, frontend, *tools* pengembangan, serta bahasa pemrograman yang digunakan untuk membangun dan menjalankan sistem secara keseluruhan.

Tabel 4. 2. Tabel Perangkat Lunak (Software)

| No | Kategori | Perangkat Lunak | Keterangan |
|----|--------------------|-------------------------------|--|
| 1 | Backend | HonoJS | Framework backend ringan untuk menangani logika bisnis dan menyediakan API endpoint. |
| | | Wrangler (Cloudflare Workers) | Alat untuk mendeploy sistem aplikasi menggunakan arsitektur <i>serverless</i> . |
| | | Neon PostgreSQL | Layanan database relasional <i>serverless</i> berbasis cloud <i>Database as a Service (DBaaS)</i> untuk menyimpan data pengguna dan log akses. |
| 2 | FrontEnd | HTML, CSS, JS | Bahasa dasar untuk membangun antarmuka pengguna web aplikasi. |
| | | Bootstrap 5 | Framework CSS modern dan responsif untuk mempercepat proses desain UI/UX. |
| 3 | Tools Pengembangan | Visual Studio Code | Editor kode utama untuk pengembangan frontend dan backend. |
| | | Arduino IDE | Digunakan untuk mengupload firmware ke mikrokontroler ESP32 dan NodeMCU ESP8266. |
| | | Discord | Platform komunikasi real-time untuk menerima log perubahan pengguna dan aktivitas akses. |
| | | Webhook Discord | Digunakan untuk mengirimkan notifikasi otomatis dari sistem ke saluran Discord menggunakan HTTP request POST. |
| 5 | Bahasa Pemrograman | JavaScript | Digunakan untuk pengembangan logika backend dan frontend web aplikasi. |
| | | C++ | Digunakan untuk pemrograman mikrokontroler melalui Arduino IDE. |

4.1.1.1. Proses Perakitan Hardware (Sistem Access Masuk)

Proses perakitan hardware pada sistem access masuk dilakukan dengan menghubungkan berbagai komponen elektronik ke mikrokontroller ESP32 sesuai dengan fungsi dan kebutuhan dari sistem smart access. Berikut ini tahapan perakitannya:

1. Pemasangan Sensor RFID RC522



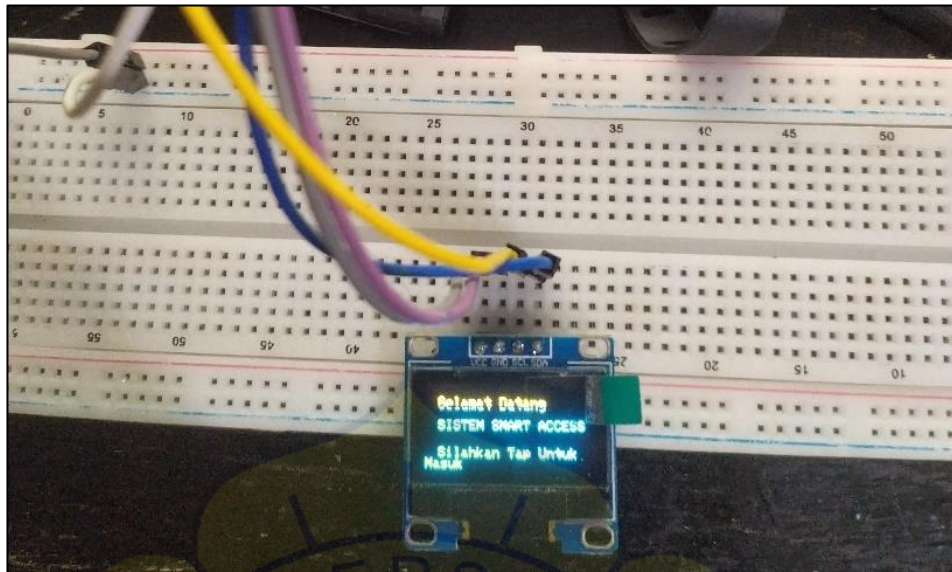
Gambar 4. 1. Sensor RFID RC522 (Akses Masuk)

Pada gambar 4.3. merupakan sensor RFID (*Radio Frequency Identification*). Sensor ini berfungsi sebagai alat pembaca kartu RFID saat pengguna meletakkan kartu ke sistem akses masuk. Sensor RFID terhubung ke ESP32 melalui komunikasi SPI (*Serial Peripheral Interface*). Pada Tabel 4.3. merupakan tabel perkabelan dari sensor RFID akses masuk sebagai berikut :

Tabel 4. 3. Tabel Perkabelan Sensor RFID RC522 dengan ESP32

| Pin ESP32 | Fungsi | Pin RFID RC522 |
|-----------|--------------------|----------------|
| 3.3v | Catu daya | 3.3v |
| GND | Ground | GND |
| GPIO4 | Reset (RST) | RST |
| GPIO19 | MISO (Master In) | MISO |
| GPIO23 | MOSI (Master Out) | MOSI |
| GPIO18 | Serial Clock (SCK) | SCK |
| GPIO5 | SDA/SS | SDA |

2. Pemasangan OLED 128x64 I2C



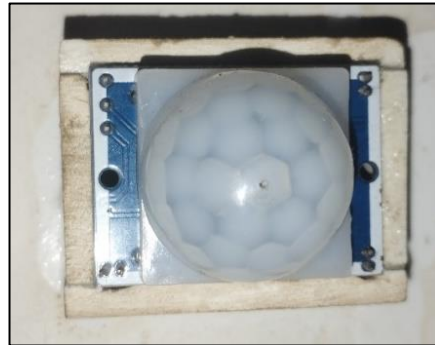
Gambar 4. 2. Layar OLED 128x64 I2C

Pada gambar 4.2. merupakan Modul layar OLED 128x64 I2C digunakan untuk menampilkan objek dalam bentuk visual. Modul ini menggunakan komunikasi I2C (*Inter-Integrated Circuit*), yang hanya memerlukan dua jalur daya, yaitu SDA (*Serial Data Line*) dan SCL (*Serial Clock Line*). Pin VCC dari modul OLED terhubung ke pin 3.3V dari ESP32 sebagai catu daya, dan pin GND dari modul OLED terhubung ke ground. Pada Tabel 4.4. merupakan tabel perkabelan dari modul OLED:

Tabel 4. 4. Tabel Perkabelan Modul OLED

| Pin ESP32 | Fungsi | Pin Layar OLED |
|-----------|------------|----------------|
| 3.3v | Catu daya | 3.3v |
| GND | Ground | GND |
| GPIO21 | SDA (Data) | SDA |
| GPIO19 | SCL (Clock | SCL |

3. Pemasangan Sensor PIR



Gambar 4. 3. Sensor PIR

Pada gambar 4.3. merupakan Sensor PIR digunakan untuk mendeteksi keberadaan atau pergerakan manusia berdasarkan perubahan sinyal inframerah. Dalam implementasi, pin VCC sensor dihubungkan ke 3.3v ESP32 sebagai catu daya, pin Sinyal (output digital) dihubungkan ke GPIO27, dan GND ke ground. Pada Tabel 4.5. merupakan tabel perkabelan dari sensor PIR:

Tabel 4. 5. Tabel Perkabelan Sensor PIR

| Pin ESP32 | Fungsi | Pin Layar OLED |
|-----------|---------------|----------------|
| 3.3v | Catu daya | VCC |
| GPIO27 | Input Digital | Signal |
| GND | Ground | GND |

4. Pemasangan Sensor Magnetik MC-38



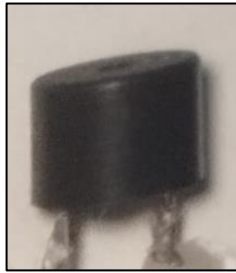
Gambar 4. 4. Sensor *Magnetic* MC-38

Pada gambar 4.4. merupakan Sensor *Magnetic* MC38 atau bisa dibilang juga dengan *reed switch* digunakan untuk mendeteksi status pintu (terbuka atau tertutup) Sensor ini bekerja berdasarkan prinsip medan magnet yaitu ketika magnet berada dekat dengan sensor (pintu tertutup), sirkuit berada dalam kondisi tertutup. Sedangkan saat magnet menjauh (pintu dibuka), sirkuit terbuka dan menghasilkan perubahan logika pada pin input mikrokontroler. Dalam implementasi, sinyal keluaran GPIO25 pada ESP32 diberi resistor 4.7k ohm sebagai resistor pull-up dan digunakan untuk menjaga kestabilan logika input. Sementara itu, pin (-) dari sensor dihubungkan ke GND pada ESP32 sebagai ground. Tabel 4.6. berikut menunjukkan konfigurasi perkabelan sensor MC-38.

Tabel 4. 6. Tabel Perkabelan Sensor MC-38

| Pin ESP32 | Fungsi | Pin Sensor MC-38 |
|-----------|------------|------------------|
| GPIO25 | + (Sinyal) | VCC |
| GND | - (Ground) | GND |

5. Pemasangan Buzzer Aktif



Gambar 4. 5. Buzzer Aktif

Pada gambar 4.5. merupakan buzzer aktif yang digunakan untuk memberikan fungsi tambahan output berupa suara dalam sistem. Buzzer ini digunakan untuk memberikan *feedback* dalam bentuk audio, seperti sistem berhasil melakukan *booting* atau proses pertama kali dalam menyalakan sistem, memberikan peringatan kepada sistem saat akses ditolak, dan memberikan kondisi jika terjadi *trespassing* atau pembobolan. Dalam implementasi, pin sinyal dari buzzer dihubungkan ke GPIO12 pada ESP32, sedangkan pin GND dihubungkan ke ground sebagai referensi tegangan. Pada Tabel 4.7. menunjukkan konfigurasi perkabelan buzzer dengan ESP32.

Tabel 4. 7. Tabel Perkabelan Buzzer Aktif

| Pin ESP32 | Fungsi | Pin Buzzer Aktif |
|-----------|------------|------------------|
| GPIO12 | + (Sinyal) | Sinyal |
| GND | - (Ground) | Ground |

6. Pemasangan Lampu Led Indikator Akses



Gambar 4. 6. Led Indikator (Akses Masuk)

Pada gambar 4.6. ditampilkan dua buah lampu indikator yang digunakan untuk memberikan umpan balik visual terhadap akses masuk. lampu berwarna merah menunjukkan akses ditolak, sedangkan lampu berwarna hijau menandakan akses diterima. Masing masing Led dihubungkan ke GPIO13 dan Led hijau ke GPIO14, keduanya melalui resistor pembatas arus sebesar 330ohm untuk melindungi komponen dari arus berlebih. Pada Tabel 4.8. menunjukkan konfigurasi perkabelan Led Indikator dengan ESP32.

Tabel 4. 8. Tabel Perkabelan Led Indikator untuk Akses Masuk

| Pin ESP32 | Warna Led | Pin LED | Keterangan |
|-----------|-----------|------------|---|
| GPIO13 | Merah | Anoda (+) | Terhubung ke GPIO melalui resistor 330ohm |
| GND | Merah | Katoda (-) | Terhubung ke Ground |
| GPIO14 | Hijau | Anoda (+) | Terhubung ke GPIO melalui resistor 330ohm |
| GND | Hijau | Katoda (-) | Terhubung ke Ground |

7. Pemasangan Relay untuk Solenoid Lock



Gambar 4. 7. Relay dan Solenoid

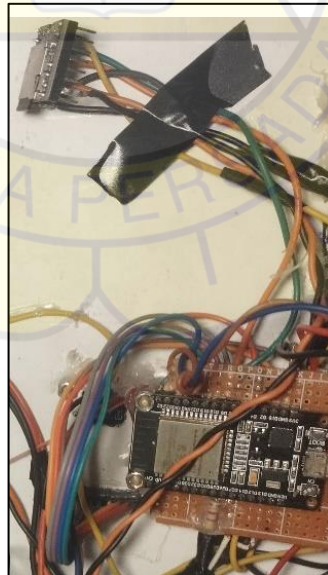
Pada gambar 4.7. merupakan modul relay yang digunakan untuk mengendalikan kunci solenoid sebagai aktuator sistem penguncian. Modul relay ini dikontrol oleh GPIO32 dari ESP32, dengan menggunakan sumber daya eksternal 5V melalui modul *step-down lm2596*. Pin GND ESP32 dan GND Relay dihubungkan bersama dengan GND dari *step-down lm2596* agar referensi tegangan tetap konsisten. Relay dikonfigurasi dalam mode *Normally Open (NO)*, dimana terminal COM (*Common*) relay dihubungkan ke positif dari power supply 12V, dan terminal NO dihubungkan ke pin positif dari solenoid. Ketika relay aktif, arus mengalir ke solenoid sehingga aktuator dapat bekerja. Pada Tabel 4.9. menunjukkan konfigurasi perkabelan Led Indikator dengan ESP32.

Tabel 4. 9. Tabel Perkabelan Modul Relay untuk Solenoid

| Komponen | Pin | Terhubung ke | Keterangan |
|----------|-----|--------------|------------|
|----------|-----|--------------|------------|

| | | | |
|-----------------------|---------------------------|--|---|
| ESP32 | GPIO32 | Input Relay | Pin kontrol untuk mengaktifkan modul relay |
| ESP32 | GND | GND Relay, dan GND <i>Step-down LM2596 5V</i> | Ground disatukan sebagai referensi tegangan |
| Step-down 5V (LM2596) | VCC | VCC Relay | Menyediakan catu daya 5V untuk modul relay |
| Relay | COM | (+) Power Supply 12V melalui stepdown LM2596 (12V) | Terminal COM relay, terhubung ke sumber tegangan |
| Relay | <i>NO (Normally Open)</i> | (+) Solenoid Lock | Terminal Output relay, hanya aktif saat relay menyala |

8. Pemasangan DFPlayer Mini Mp3 dan Speaker



Gambar 4. 8. DF Player Mini

Pada Gambar 4.8. memperlihatkan modul DFPlayer Mini MP3, ESP32. Modul DFPlayer Mini digunakan untuk memberikan fungsi sebagai pemutar audio yang dikendalikan melalui komunikasi UART (*Universal Asynchronous Receiver/Transmitter*). Dalam sistem ini, pin TX pada DFPlayer dihubungkan ke pin GPIO16 (RX2) pada ESP32, dan pin RX ke GPIO17 (TX2). Modul diberi catu daya melalui pin VCC (terhubung ke 3.3V ESP32) dan GND. Sinyal keluaran audio dari DFPlayer Mini dialirkan ke speaker menggunakan SPK_1 sebagai positif dan SPK_2 sebagai negatif.

Tabel 4.10. merupakan perkabelan dari DFPlayerMini

Tabel 4. 10. Tabel Perkabelan DFPlayer Mini MP3

| Komponen | Pin | Fungsi Pin | Keterangan |
|----------|--------------|------------|----------------------------------|
| ESP32 | GPIO16 (RX2) | TX | Komunikasi UART ke DFPlayer Mini |
| ESP32 | GPIO17 (TX2) | RX | Komunikasi UART ke DFPlayer Mini |
| ESP32 | 3.3V | VCC | Catu daya ke DFPlayer Mini |
| ESP32 | GND | GND | Ground DFPlayerMini |
| Speaker | (+)SPK | SPK_1 | Output audio positif |
| Speaker | (-)SPK | SPK_2 | Output audio negatif |

4.1.1.2. Proses Perakitan Hardware (Sistem Access Keluar)

Perakitan hardware untuk sistem akses keluar dilakukan dengan menyusun dan menghubungkan setiap komponen ke NodeMCU ESP8266 sesuai dengan fungsi dan peran masing – masing. Berikut adalah proses perakitannya:

1. Pemasangan Sensor RFID RC522 (Akses Keluar)



Gambar 4. 9. Sensor RFID RC522 (Akses Keluar)

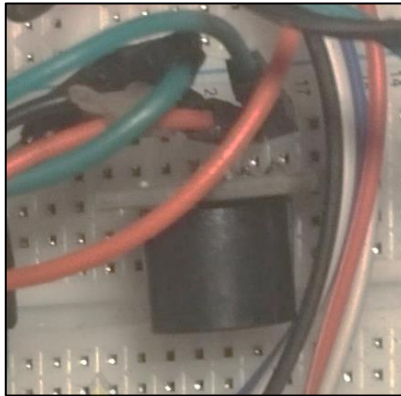
Gambar 4.9. memperlihatkan pemasangan modul RFID RC522 yang berfungsi sebagai alat identifikasi pengguna saat melakukan akses keluar.

Modul ini beroperasi menggunakan komunikasi SPI (*Serial Peripheral Interface*) dan dihubungkan menggunakan NodeMCU ESP8266. Pin SDA/SS terhubung ke D8, SCK ke D0, MOSI ke D7, MISO ke D6, dan RST ke D5. Sementara itu, pin VCC dihubungkan ke 3.3V NodeMCU, dan GND ke GND NodeMCU untuk catu daya dan ground. Pada tabel 4.11. merupakan tabel perkabelan dari rangkaian RFID untuk akses keluar.

Tabel 4. 11. Perkabelan Sensor RFID untuk Akses Keluar

| Pin NodeMCU ESP8266 | Fungsi | Pin RFID RC522 |
|---------------------|--------------------|----------------|
| 3.3v | Catu daya | 3.3v |
| GND | Ground | GND |
| D5 | Reset (RST) | RST |
| D6 | MISO (Master In) | MISO |
| D7 | MOSI (Master Out) | MOSI |
| D0 | Serial Clock (SCK) | SCK |
| D8 | SDA/SS | SDA |

2. Pemasangan Buzzer Pasif



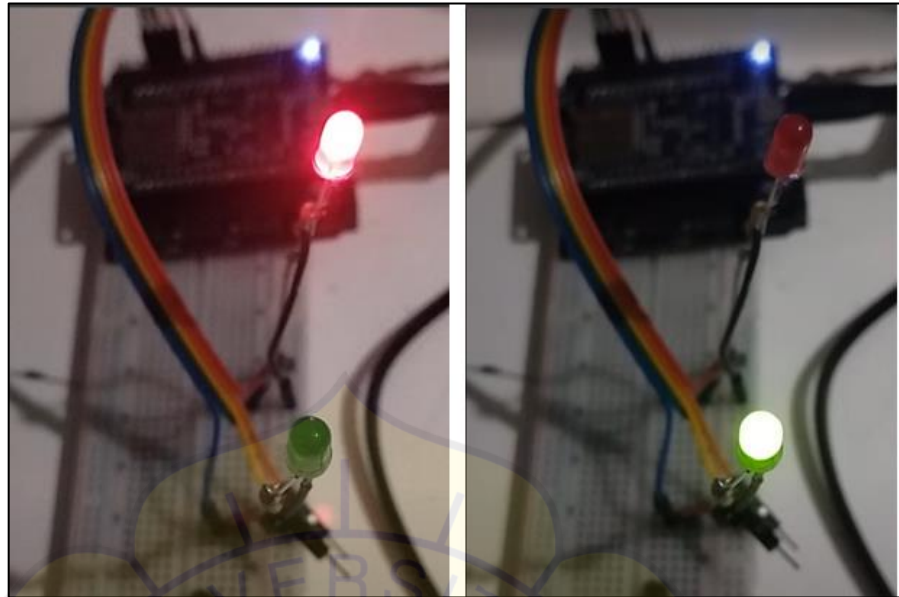
Gambar 4. 10. Buzzer Pasif (Akses Keluar)

Pada gambar 4.10. menunjukkan pemasangan buzzer pasif pada sistem akses keluar digunakan untuk memberikan notifikasi suara saat kartu berhasil di-scan. Buzzer ini dihubungkan ke pin D1 NodeMCU sebagai jalur sinyal, sedangkan pin VCC dihubungkan dengan tegangan 3.3v dan pin GND ke ground dari NodeMCU ESP8266. Pemasangan ini memastikan pemberian notifikasi yang sederhana, namun efektif dalam proses autentikasi keluar. Pada Tabel 4.12. merupakan tabel perkabelan dari buzzer pasif.

Tabel 4. 12. Tabel Perkabelan Buzzer Pasif

| Pin NodeMCU ESP8266 | Fungsi | Pin Buzzer Pasif |
|---------------------|----------------|------------------|
| 3.3V ESP8266 | Catu daya | VCC |
| GND | Ground | GND |
| D1 | Sinyal kontrol | SIGNAL |

3. Pemasangan LED Indikator



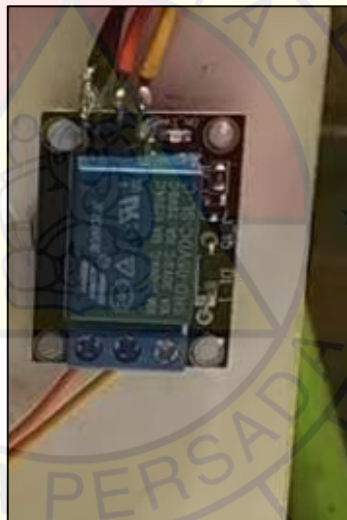
Gambar 4. 11. Led Indikator (Akses Keluar)

Pada gambar 4.11. merupakan pemasangan Led indikator untuk akses keluar. Fungsinya sama seperti sistem akses masuk, led indikator pada sistem akses keluar digunakan untuk sebagai indikator visual. Terdapat dua buah led berwarna merah dan hijau. Lampu berwarna merah menunjukkan akses ditolak, sedangkan lampu berwarna hijau menandakan akses diterima. Led merah terhubung ke pin D3 dan berfungsi sebagai indikator akses ditolak, sedangkan led hijau terhubung ke pin D4 sebagai indikator akses diterima. Masing – masing dilengkapi dengan resistor pembatas arus, yaitu 220ohm untuk led merah dan 330 ohm untuk led hijau. Pada tabel 4.13. merupakan tabel perkabelan dari led indikator.

Tabel 4. 13. Tabel perkabelan Led Indikator untuk Akses Keluar

| Pin NodeMCU ESP8266 | Warna Led | Pin LED | Keterangan |
|---------------------|-----------|------------|---|
| D3 | Merah | Anoda (+) | Terhubung ke GPIO melalui resistor 220ohm |
| GND | Merah | Katoda (-) | Terhubung ke Ground |
| D4 | Hijau | Anoda (+) | Terhubung ke GPIO melalui resistor 330ohm |
| GND | Hijau | Katoda (-) | Terhubung ke Ground |

4. Pemasangan Modul Relay untuk Solenoid



Gambar 4. 12. Relay 2 Solenoid (Akses Keluar)

Pada gambar 4.12. merupakan modul relay pada sistem akses keluar. Modul relay ini berfungsi untuk mengatur aliran listrik untuk menuju ke kunci solenoid. Modul ini dikendalikan oleh pin D2 dari NodeMCU ESP8266 dengan menggunakan catu daya eksternal 5V melalui modul stepdown lm2596, sama seperti pada akses masuk. Ground dari NodeMCU dan GND Relay dihubungkan bersama untuk menyamakan

referensi tegangan. Koneksi dan terminal COM relay menuju ke jalur positif power supply 12v dari stepdown lm2596, sedangkan terminal NO (*Normally Open*) terhubung ke kunci solenoid. Ketika relay aktif, arus mengalir dan mengaktifkan solenoid sebagai aktuator kunci pintu. Pada tabel 4.14. merupakan tabel perkabelan dari relay dan solenoid.

Tabel 4. 14. Tabel Perkabelan Relay dan Solenoid (Akses Keluar)

| Komponen | Pin | Terhubung ke | Keterangan |
|-----------------------|---------------------------|-------------------------------------|---|
| NodeMCU ESP8266 | D2 | Input Relay | Pin kontrol untuk mengaktifkan modul relay |
| NodeMCU ESP8266 | GND | GND Relay, dan GND <i>Step-down</i> | Ground disatukan sebagai referensi tegangan |
| Step-down 5V (LM2596) | VCC | VCC Relay | Menyediakan catu daya 5V untuk modul relay |
| Relay | COM | (+) Power Supply 12V | Terminal COM relay, terhubung ke sumber tegangan |
| Relay | <i>NO (Normally Open)</i> | (+) Solenoid Lock | Terminal Output relay, hanya aktif saat relay menyala |

4.1.1.3. Rangkaian keseluruhan

Pada gambar 4.13. merupakan hasil dari rangkaian alat Smart Access yang sudah terhubung menjadi suatu prototipe alat yang siap diuji coba.



Gambar 4. 13. Rangkaian keseluruhan

4.1.2.2. Halaman Dashboard

Pada gambar 4.15. Halaman Dashboard merupakan halaman setelah admin berhasil melakukan login. Di halaman ini, admin dapat melihat ringkasan log akses, jumlah pengguna yang terdaftar.

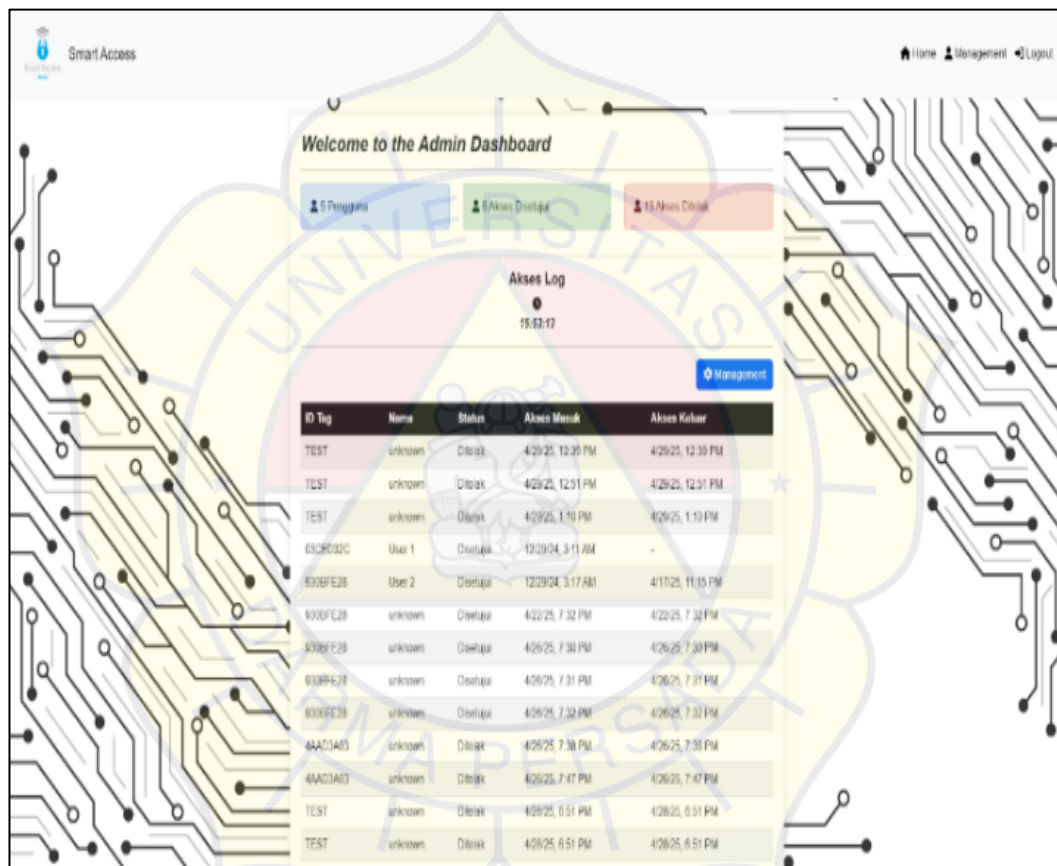
The screenshot displays the 'Smart Access' Admin Dashboard. At the top left, there is a logo and the text 'Smart Access'. At the top right, there are navigation links: 'Home', 'Management', and 'Logout'. The main content area features a 'Welcome to the Admin Dashboard' message. Below this, there are three summary cards: '5 Pengguna' (Users), '2 Akses Disetujui' (Accesses Approved), and '0 Akses Ditolak' (Accesses Denied). A 'Management User' profile is shown with a clock icon and the time '20:30:25'. A green button labeled 'Tambah User' (Add User) is visible. Below the profile is a table with the following data:

| ID | RFID Tag | Nama | Email | Role | Dibuat | Diubah | Aksi |
|----|----------|------------------|-----------------------|-------|-------------------|--------------------|-------------------|
| 2 | 03CED32C | User 1 | user1@gmail.com | User | 12/13/24, 4:30 AM | 12/13/24, 5:42 AM | ✎ |
| 1 | 33DACF29 | Bayu Aji Nuranto | bayunuranto@gmail.com | Admin | 12/7/24, 2:21 PM | 12/12/24, 2:45 AM | ✎ |
| 11 | 63C7D52A | User 4 | user4@gmail.com | User | 12/13/24, 6:49 PM | 12/13/24, 11:49 AM | ✎ |

Gambar 4. 15. Halaman Dashboard

4.1.2.3. Halaman Log Akses

Pada Gambar 4.16. Halaman Log akses menampilkan semua riwayat aktivitas akses, termasuk waktu masuk atau keluar, nama pengguna, ID RFID, dan status akses (diterima/ditolak). Halaman ini memudahkan admin untuk melakukan pemeriksaan terhadap akses log berupa tabel log akses.



Smart Access

Home Management Logout

Welcome to the Admin Dashboard

5 Pengguna 8 Akses Diterima 15 Akses Ditolak

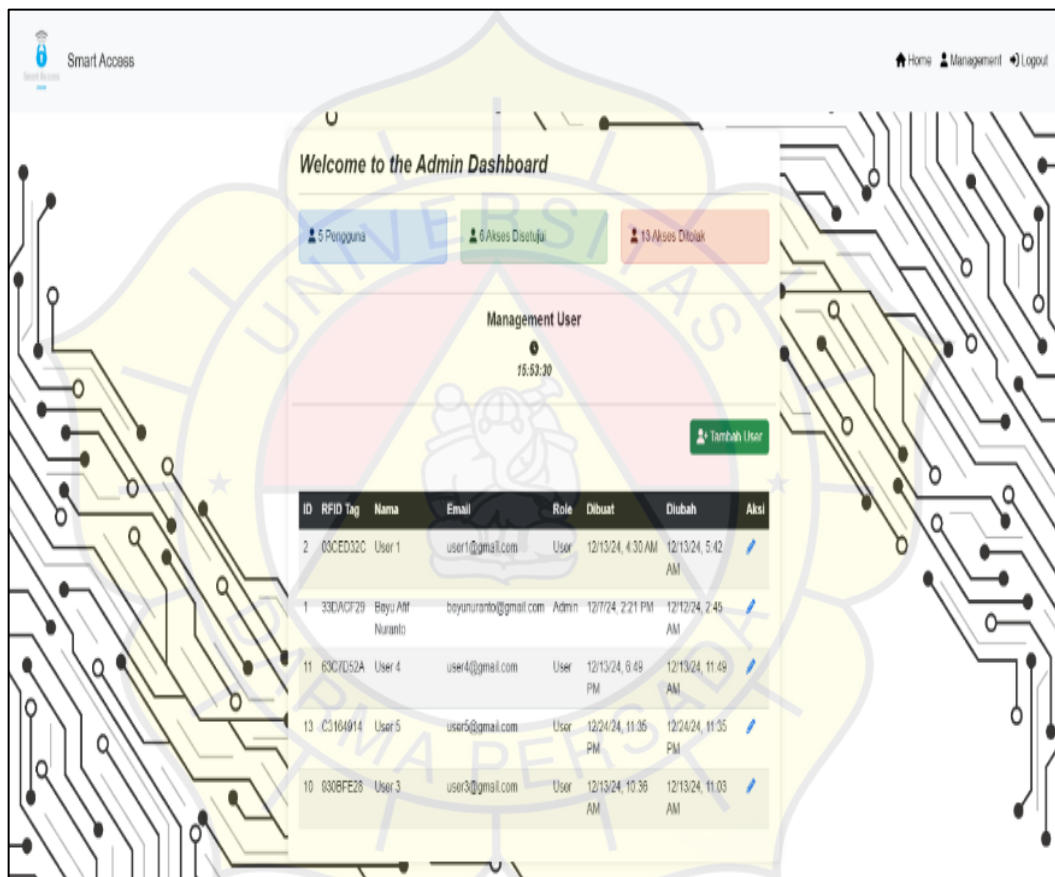
Akses Log 15.07.17 Management

| ID Tag | Nama | Status | Akses Masuk | Akses Keluar |
|----------|---------|----------|-------------------|-------------------|
| TEST | unknown | Ditolak | 4/26/25, 12:30 PM | 4/26/25, 12:30 PM |
| TEST | unknown | Ditolak | 4/26/25, 12:31 PM | 4/26/25, 12:31 PM |
| TEST | unknown | Ditolak | 4/26/25, 1:10 PM | 4/26/25, 1:10 PM |
| 6308030C | User 1 | Diterima | 12/29/24, 3:11 AM | - |
| 6308FE28 | User 2 | Diterima | 12/29/24, 3:17 AM | 4/11/25, 11:15 PM |
| 6308FE28 | unknown | Diterima | 4/26/25, 7:32 PM | 4/26/25, 7:32 PM |
| 6308FE28 | unknown | Diterima | 4/26/25, 7:30 PM | 4/26/25, 7:30 PM |
| 6308FE28 | unknown | Diterima | 4/26/25, 7:31 PM | 4/26/25, 7:31 PM |
| 6308FE28 | unknown | Diterima | 4/26/25, 7:32 PM | 4/26/25, 7:32 PM |
| 4AAC1A03 | unknown | Ditolak | 4/26/25, 7:38 PM | 4/26/25, 7:38 PM |
| 4AAC1A03 | unknown | Ditolak | 4/26/25, 7:47 PM | 4/26/25, 7:47 PM |
| TEST | unknown | Ditolak | 4/26/25, 6:51 PM | 4/26/25, 6:51 PM |
| TEST | unknown | Ditolak | 4/26/25, 6:51 PM | 4/26/25, 6:51 PM |

Gambar 4. 16. Halaman Log Akses

4.1.2.4. Halaman Management Pengguna / Siswa / User

Pada gambar 4.17. halaman Manajemen Pengguna, admin dapat menambahkan, mengedit, atau menghapus data pengguna. Ini sangat berguna untuk mengatur siapa saja yang memiliki otorisasi atau yang memiliki akses untuk masuk ke ruangan.



Gambar 4. 17. Halaman Manajemen Pengguna

4.1.2.5. Halaman Tambah User

Pada gambar 4.18. halaman Tambah User atau pengguna, admin dapat menambahkan pengguna baru dengan memasukkan data seperti nama, email, role (admin/user), dan ID tag rfid.

The screenshot shows a web interface for adding users. The form is titled "Tambah Users" and contains the following fields:

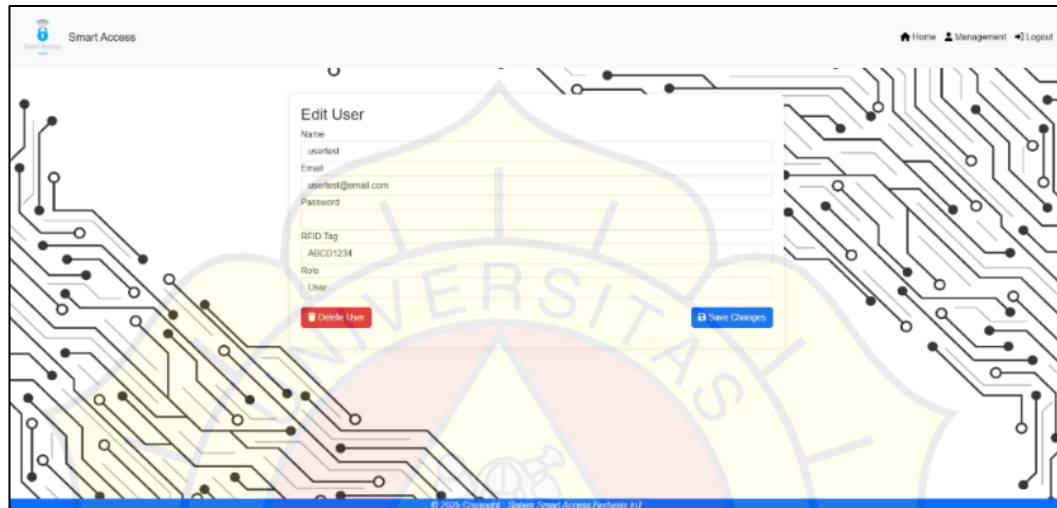
| | |
|----------|--------------------|
| Name | usortest |
| Email | usortest@email.com |
| Password | ***** |
| RFID Tag | ABCD1234 |
| Role | User |

A green button labeled "Tambah User" is located below the form fields. The background of the page features a circuit board pattern and a watermark of Universitas Persada Indonesia. The footer contains the text "© 2025 Copyright. Sistem Smart Access Berbasis IoT".

Gambar 4. 18. Halaman Tambah User

4.1.2.6. Halaman Edit User dan Delete User

Pada gambar 4.19. halaman edit, setiap pengguna yang sudah terdaftar dapat diedit atau dihapus oleh admin melalui tombol aksi yang ada pada halaman manajemen pengguna.



Gambar 4. 19. Halaman Edit pengguna

4.1.2.7. Notifikasi Aplikasi Discord

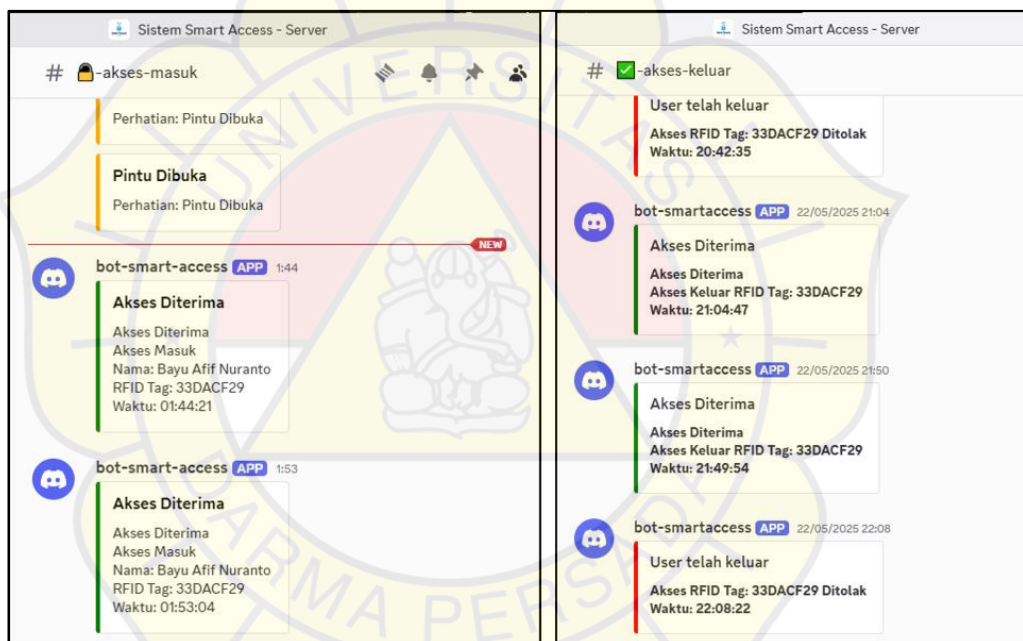


Gambar 4. 20. Notifikasi Discord (Manajemen Pengguna)

Pada gambar 4.20. menampilkan contoh notifikasi aplikasi Discord pada sistem ini yang dikirimkan secara otomatis melalui fitur webhook Discord. Webhook memungkinkan sistem untuk mengirim pesan langsung ke channel Discord setiap kali ada perubahan data user, seperti penambahan, pengeditan, atau penghapusan

pengguna. Saat user berhasil ditambahkan, diubah, atau dihapus, sistem akan melakukan permintaan menggunakan metode HTTP POST ke URL webhook Discord yang sudah dikonfigurasi. Pesan notifikasi kemudian langsung muncul di channel Discord secara real-time, sehingga admin bisa memantau aktivitas tanpa harus membuka aplikasi web.

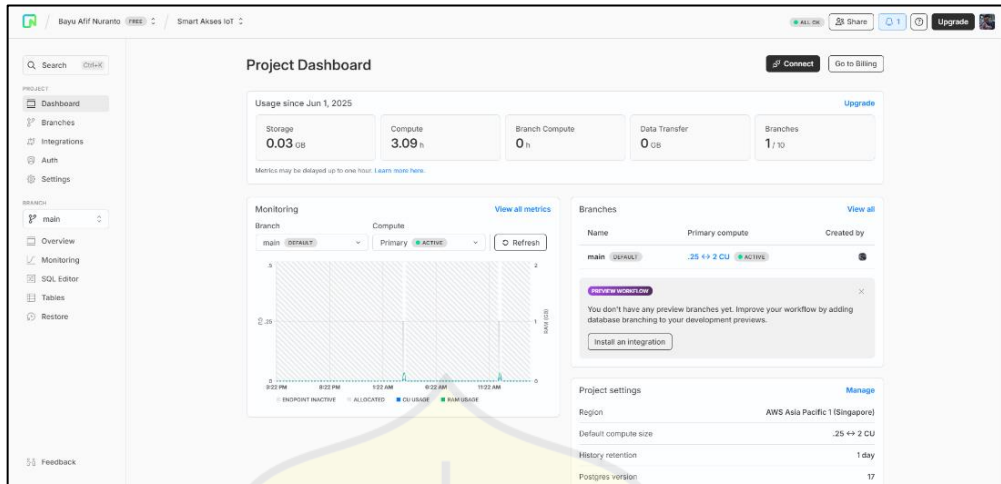
Begitu juga pada gambar 4.21. dapat menampilkan notifikasi untuk log akses masuk dan keluar



Gambar 4. 21. Notifikasi Discord (Akses Masuk dan AksesKeluar)

4.1.3. Struktur Database

Sistem ini menggunakan database PostgreSQL dari sebuah platform *Neon*. yang dihosting di layanan cloud *Neon PostgreSQL* yang terlihat pada gambar 4.22.



Gambar 4. 22. Gambar Platform Neon.tech

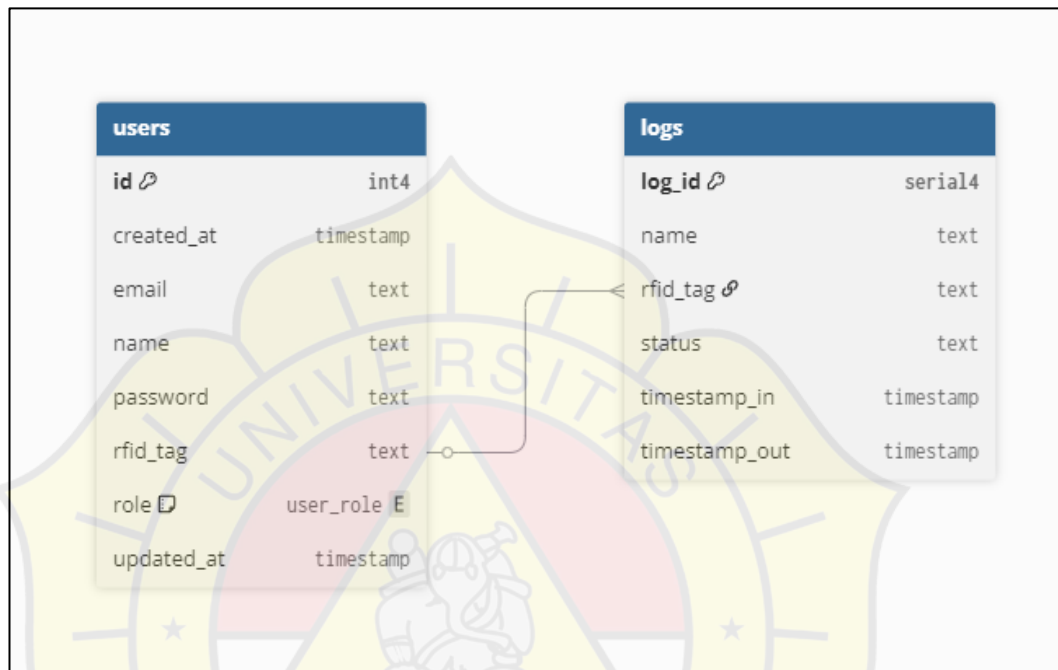
Layanan ini merupakan platform *Database as a Service (DBaaS)* yang menyediakan PostgreSQL dengan arsitektur *serverless* sehingga akses data bisa dilakukan secara online.

| log_id | rfid_tag | name | status | timestamp_in | timestamp_out |
|--------|----------|-------------------|-----------|-------------------------|---------------------------|
| 121 | 9308FE26 | Iqbal | Ditetujui | 2025-05-21 17:03:28.256 | 2025-05-21 17:03:28.27... |
| 122 | 4A03A03 | unknown | Ditolak | 2025-05-21 17:06:02.049 | 2025-05-21 17:06:02.07... |
| 123 | 330ACF29 | Bayu Afif Nuranto | Ditetujui | 2025-05-21 17:06:20.685 | 2025-05-21 17:06:20.71... |
| 124 | 330ACF29 | Bayu Afif Nuranto | Ditetujui | 2025-05-22 05:39:02.894 | 2025-05-22 05:39:02.92... |
| 125 | 9308FE26 | Iqbal | Ditetujui | 2025-05-22 05:39:41.658 | 2025-05-22 05:39:41.68... |
| 126 | 9308FE26 | Iqbal | Ditetujui | 2025-05-22 06:35:02.998 | 2025-05-22 06:35:03.02... |
| 127 | 4A03A03 | unknown | Ditolak | 2025-05-22 06:37:32.091 | 2025-05-22 06:37:32.91... |
| 128 | 9308FE26 | Iqbal | Ditetujui | 2025-05-22 13:07:52.093 | 2025-05-22 13:07:52.12... |
| 129 | 9308FE26 | Iqbal | Ditetujui | 2025-05-22 13:11:26.843 | 2025-05-22 13:11:26.87... |
| 130 | 330ACF29 | Bayu Afif Nuranto | Ditetujui | 2025-05-22 13:28:57.264 | 2025-05-22 13:28:57.28... |
| 131 | 330ACF29 | Bayu Afif Nuranto | Ditetujui | 2025-05-22 14:03:30.822 | 2025-05-22 14:04:46.739 |
| 132 | 330ACF29 | Bayu Afif Nuranto | Ditetujui | 2025-05-22 14:48:04.823 | 2025-05-22 14:48:53.864 |
| 133 | 330ACF29 | Bayu Afif Nuranto | Ditetujui | 2025-05-22 15:50:31.527 | 2025-05-22 15:51:19.142 |
| 134 | 330ACF29 | Bayu Afif Nuranto | Ditetujui | 2025-05-22 15:56:38.781 | 2025-05-22 15:56:59.254 |
| 135 | 330ACF29 | Bayu Afif Nuranto | Ditetujui | 2025-05-22 16:02:08.581 | 2025-05-22 16:03:12.494 |
| 136 | 330ACF29 | Bayu Afif Nuranto | Ditetujui | 2025-05-22 16:21:10.907 | 2025-05-22 16:22:08.224 |
| 137 | 330ACF29 | Bayu Afif Nuranto | Ditetujui | 2025-05-22 16:28:17.134 | 2025-05-22 16:28:30.328 |
| 138 | 330ACF29 | Bayu Afif Nuranto | Ditetujui | 2025-05-22 16:31:02.166 | 2025-05-22 16:31:49.376 |
| 139 | 330ACF29 | Bayu Afif Nuranto | Ditetujui | 2025-05-22 16:34:08.142 | 2025-05-22 16:34:29.065 |
| 140 | 330ACF29 | Bayu Afif Nuranto | Ditetujui | 2025-05-22 16:39:55.066 | 2025-05-22 16:40:18.919 |
| 141 | 330ACF29 | Bayu Afif Nuranto | Ditetujui | 2025-05-22 16:51:00.739 | 2025-05-22 16:51:08.9 |
| 142 | 330ACF29 | Bayu Afif Nuranto | Ditetujui | 2025-05-22 16:56:35.871 | 2025-05-22 17:03:49.423 |
| 143 | 330ACF29 | Bayu Afif Nuranto | Ditetujui | 2025-05-22 17:00:12.762 | 2025-05-22 17:00:26.796 |

Gambar 4. 23. Gambar Tabel pada platform Neon.tech

Pada gambar 4.23. layanan *neon.tech*, terdapat dua tabel, yaitu Tabel *users* dan *logs*. Tabel *users* digunakan untuk menyimpan informasi pengguna yang memiliki hak akses kedalam ruangan, sementara tabel *logs* digunakan untuk mencatat seluruh

aktivitas akses, baik yang disetujui atau diterima maupun ditolak. Struktur tabel dan Entity Relationship Diagram disajikan pada tabel 4.15. dan tabel 4.16, sedangkan gambar entity Relationship Diagram disajikan pada gambar 4.24.



Gambar 4. 24. Gambar Entity Relational Database

Tabel 4. 15. Tabel Users

| No | Nama Kolom | Tipe Data | Length | Null | Primary Key | Keterangan |
|----|------------|-----------|----------------------|------|-------------|----------------------------|
| 1 | id | INT | 4 | No | Yes | ID unik pengguna |
| 2 | name | TEXT | - | No | No | Nama pengguna |
| 3 | email | TEXT | - | No | No | Email pengguna |
| 4 | password | TEXT | - | No | No | Password pengguna |
| 5 | rfid_tag | TEXT | - | No | No | ID tag RFID |
| 6 | role | ENUM | (“Admin”, ”User”) | No | No | Role pengguna (admin/user) |
| 7 | created_at | TIMESTAMP | - | No | No | Waktu pendaftaran pengguna |
| 8 | updated_at | TIMESTAMP | - | Yes | No | Waktu modifikasi pengguna |

Tabel 4. 16. Tabel Logs

| No | Nama Kolom | Tipe Data | Length | Null | Primary Key | Keterangan |
|----|---------------|-----------|--------|------|-------------|--|
| 1 | log_id | SERIAL | 4 | No | Yes | ID unik pengguna |
| 2 | rfid_tag | TEXT | - | No | No | ID tag RFID |
| 3 | name | TEXT | - | No | No | Nama pengguna |
| 4 | Status | TEXT | - | No | No | Status akses, seperti “diterima” atau “ditolak”. |
| 5 | timestamp_in | TIMESTAMP | - | No | No | Waktu pengguna masuk ke ruangan |
| 6 | timestamp_out | TIMESTAMP | - | yes | No | Waktu pengguna keluar dari ruangan |

4.2. Analisa Hasil

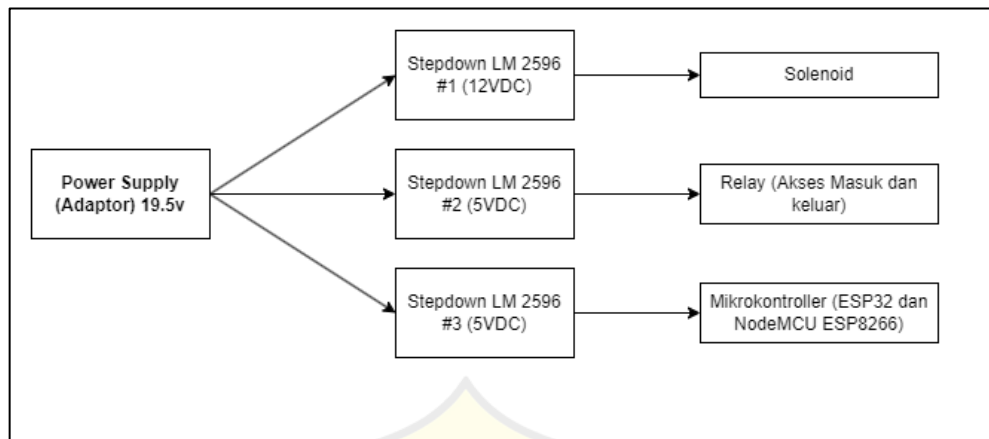
4.2.1. Percobaan Input – Output

Pada tahap percobaan input, sistem diberikan masukan dari berbagai sensor untuk menguji apakah perangkat keras dapat bekerja secara optimal dalam mendukung proses autentikasi akses dan monitoring ruangan. Setiap input dari sensor akan diproses oleh mikrokontroler (ESP32 dan NodeMCU) dan menghasilkan output melalui aktuator seperti solenoid, layar OLED, buzzer, dan LED indikator.

Berikut adalah rincian input, output, serta analisis hasil percobaan:

4.2.1.1. Percobaan Input Power Supply

Pada gambar 4.24 terlihat, sistem dirancang untuk menerima sumber tegangan input dari adaptor 19,5VDC 2.31A, yang kemudian diturunkan menggunakan modul step-down (*buck converter*) menjadi beberapa tegangan yang diperlukan oleh komponen sistem.



Gambar 4. 25. Rangkaian Power Supply

Input yang diberikan :

Terlihat pada gambar 4.24. sumber daya utama berasal dari adaptor Power Supply 19,5V 2,31A. Tegangan ini kemudian diturunkan menggunakan tiga modul step-down LM2596, masing-masing disetel untuk menghasilkan tegangan 12V, 5V untuk relay, dan 5V untuk mikrokontroler (ESP32 dan NodeMCU ESP8266).

Output yang dihasilkan :

1. Power Supply 19,5V: Tegangan tanpa beban berkisar 20,05V – 20,10V, dan saat diberi beban menurun sedikit ke 19,90V – 19,95V. Arus berada di kisaran 1,09A – 1,12A.
2. Stepdown LM2596 (12V): Output untuk solenoid lock menunjukkan tegangan tanpa beban 12,05V – 12,10V, dan dengan beban 11,91V – 11,98V. Arus yang dikonsumsi sekitar 0,84A – 0,86A.
3. Stepdown LM2596 #1 (5V): Untuk relay 5V, tegangan tanpa beban berkisar 5,02V – 5,05V, dan dengan beban 4,96V – 4,98V. Arus sekitar 0,44A – 0,46A.

4. Stepdown LM2596 #2 (5V): Untuk ESP32 dan NodeMCU, tegangan tanpa beban 5,01V – 5,08V, dan saat beban aktif turun menjadi 4,92V – 4,95V. Arus sekitar 0,47A – 0,50A.

Analisa hasil :

Hasil percobaan menunjukkan bahwa seluruh sistem catu daya berfungsi dengan stabil dan efisien. Penurunan tegangan saat beban terjadi dalam batas wajar dan aman untuk semua perangkat. Power supply utama mampu menyuplai arus hingga 1,12A, yang cukup untuk menghidupkan seluruh modul. Ketiga stepdown LM2596 bekerja sesuai konfigurasi, menghasilkan tegangan dan arus yang stabil sesuai kebutuhan masing-masing perangkat seperti solenoid, relay, dan mikrokontroler. Sistem catu daya ini dinyatakan layak dan andal untuk mendukung sistem Smart Access.

4.2.1.2. Percobaan Input Sensor RFID (Akses Masuk)

Tujuan :

Menguji kemampuan sistem dalam memproses akses masuk menggunakan kartu RFID, baik yang mendaftar maupun tidak terdaftar, serta mendeteksi pembukaan pintu secara ilegal (trespassing).

Input dan Skenario yang diberikan:

A. Kartu RFID Terdaftar

Pengguna menempelkan kartu RFID yang telah terdaftar ke pembaca (RFID Reader RC522).

B. Kartu RFID Tidak Terdaftar

Kartu RFID yang tidak terdaftar ditempelkan ke pembaca RFID.

C. Pintu Dibuka Tanpa Akses (Trespassing)

Pintu dibuka secara paksa tanpa melalui proses autentikasi RFID, dideteksi oleh sensor pintu magnetik.

Output yang dihasilkan:

A. Untuk RFID yang Terdaftar

1. Layar OLED menampilkan nama pengguna dan status “Akses Diterima”.
2. Relay aktif selama 5 detik dan solenoid membuka pintu.
3. LED hijau menyala.
4. Suara “Pintu Terbuka” diputar melalui DFPlayer.
5. Notifikasi Discord dikirimkan dengan status akses diterima.
6. Log akses masuk dikirimkan ke server API dan tercatat sebagai “Akses Disetujui”.

B. Untuk RFID Tidak Terdaftar

1. OLED menampilkan pesan “Akses Ditolak” dan “RFID Invalid”.
2. LED merah menyala selama 2 detik.
3. Buzzer berbunyi
4. Suara “Akses Ditolak” diputar.
5. Notifikasi Discord dikirimkan dengan status akses ditolak.
6. Akses Log ditolak terkirim ke Aplikasi Web dengan nama “unknown”.

C. Untuk Akses Paksa (Trespassing)

1. OLED menampilkan pesan “TRESSPASSING”.
2. LED merah menyala.

3. Buzzer menyala selama 1 detik.
4. Suara “Tresspassing” diputar dari DFPLayerMini

Analisa hasil :

Dari ketiga skenario tersebut, sistem mampu memberikan respon berbeda sesuai kondisi. Untuk kartu RFID terdaftar sistem memproses akses secara lengkap mulai dari tampilan visual, pembukaan pintu, hingga pencatatan ke server. Untuk akses RFID yang tidak sah dan upaya pembukaan paksa, sistem memberikan respon penolakan yang jelas disertai peringatan bunyi serta notifikasi keamanan. Hal ini membuktikan bahwa sistem bekerja dengan baik.

4.2.1.3. Percobaan Input Sensor PIR

Tujuan :

Mengetahui apakah sistem dapat mendeteksi keberadaan manusia di depan pintu.

Input yang diberikan :

Pengguna mendekati area pintu, dan sensor PIR mendeteksi gerakan manusia didepan pintu.

Output yang dihasilkan :

- a. RFID Reader diaktifkan dan siap membaca kartu.
- b. Layar OLED aktif dan menampilkan pesan “Gerakan Terdeteksi”.
- c. DFPlayer Mini memutar audio ucapan “Selamat Datang”

Analisa Hasil :

Sensor PIR berhasil mendeteksi keberadaam pengguna di depan pintu dengan responsif. Sensor hanya merespons objek hidup dan tidak bereaksi terhadap benda

mati, menunjukkan sensitivitas yang baik. Ini sangat berguna untuk menghemat daya karena sistem hanya aktif ketika ada pengguna di lokasi.

4.2.1.4. Percobaan Input Sensor MC-38 (Magnetic Switch)

Tujuan :

Pintu dalam kondisi tertutup kemudian dibuka oleh pengguna. Sensor MC-38 akan mendeteksi perubahan posisi antara magnet dan sensor.

Input yang diberikan :

Pintu dalam kondisi tertutup kemudian dibuka oleh pengguna. Sensor MC-38 akan mendeteksi perubahan posisi antara magnet dan sensor.

Output yang dihasilkan:

Ketika pintu dibuka secara sah (melalui autentikasi kartu RFID yang valid), sistem tidak memberikan peringatan.

Jika pintu dibuka tanpa autentikasi atau secara paksa :

- A. Buzzer berbunyi sebagai peringatan.
- B. Notifikasi dikirimkan melalui Discord Webhook yang menginformasikan adanya pembukaan pintu yang mencurigakan.
- C. Data peristiwa dicatat ke dalam sistem aplikasi web.

Analisa Hasil :

Hasil percobaan menunjukkan bahwa sensor MC-38 mampu membedakan status pintu dengan akurat. Sistem mampu membedakan pembukaan pintu yang sah dan tidak sah, serta memberikan respons keamanan yang sesuai. Fitur ini berfungsi

sebagai pengaman tambahan untuk mendeteksi upaya pembobolan atau pelanggaran akses.

4.2.1.5. Percobaan Output Layar OLED Display

Tujuan :

Menguji kemampuan OLED dalam menampilkan informasi sistem *Smart Access* secara dinamis berdasarkan kondisi dan input tertentu, seperti status pintu, deteksi RFID, hingga hasil autentikasi akses (disetujui atau ditolak).

Input yang diberikan :

- A. Sensor magnetik (status pintu: terbuka / tertutup).
- B. Sensor PIR (mendeteksi gerakan).
- C. Kartu RFID (valid atau tidak valid).
- D. Status Sistem (saat booting, idle, timeout).
- E. Dibuka secara paksa (Tresspassing).

Output yang dihasilkan :

- A. Layar menampilkan pesan “Sistem Smart Access” saat sistem aktif.
- B. Layar menampilkan pesan “Silakan tempelkan kartu” saat sensor PIR mendeteksi gerakan.
- C. Layar menampilkan pesan “Akses Diterima” jika kartu valid.
- D. Layar menampilkan pesan “Akses Ditolak” jika kartu tidak valid.
- E. Layar menampilkan pesan “Tresspassing”.

Analisa Hasil :

Hasil percobaan menunjukkan bahwa OLED berhasil menampilkan pesan sesuai kondisi sistem secara real-time.

4.2.1.6. Percobaan Output Buzzer Aktif (Akses Masuk)

Tujuan :

Mengidentifikasi fungsi buzzer sebagai media tambahan umpan balik audio dalam menanggapi kondisi sistem tertentu, seperti sistem sukses melakukan *booting*, tidak terbaca-nya kartu RFID, akses yang tidak sah (pembukaan pintu tanpa otorisasi), atau adanya pelanggaran masuk (*trespassing*)

Input yang diberikan :

- A. Kartu RFID tidak dikenal (akses ditolak).
- B. Pintu dibuka tanpa akses sah (*trespassing*).
- C. Pintu kembali tertutup (buzzer nonaktif).

Output yang dihasilkan :

- A. Buzzer menyala selama 1 detik.
- B. Buzzer menyala terus selama pintu terbuka.
- C. Buzzer dimatikan.

Analisa Hasil :

Dari percobaan yang dilakukan, buzzer berhasil memberikan respon yang sesuai terhadap setiap kondisi sistem.

4.2.1.7. Percobaan Output DFPlayer Mini (Akses Masuk)

Tujuan :

Menguji kemampuan DFPlayer Mini dalam memberikan umpan balik audio yang berbeda sesuai dengan kondisi sistem seperti sensor PIR mendeteksi gerakan, akses berhasil, akses ditolak, atau peringatan keamanan.

Input yang diberikan :

- A. Sistem dihidupkan atau sensor pir berhasil mendeteksi gerakan.
- B. Kartu RFID valid.
- C. Pintu menutup (deteksi dari sensor MC-38).
- D. Pintu dibuka tanpa otorisasi
- E. Kartu rfid tidak terdaftar.

Output yang dihasilkan :

- A. Audio 1 (“Selamat Datang di Smart Access”) diputar saat sistem dihidupkan atau sensor pir berhasil mendeteksi.
- B. Audio 2 (“Pintu terbuka, silakan masuk”) diputar saat kartu valid ditempelkan.
- C. Audio 3 (“Pintu tertutup”) diputar saat pintu tertutup kembali setelah akses.
- D. Audio 4 (Peringatan, Ada pembobolan ...) diputar saat pintu dibuka secara paksa.
- E. Audio 5 (“Akses ditolak”) diputar saat kartu tidak valid ditempelkan.

Analisa Hasil :

DFPlayer Mini berhasil memainkan file audio yang sesuai dengan setiap kondisi sistem. Dengan suara yang jelas dan berbeda untuk setiap status, admin dan

pengguna bisa langsung memahami apa yang sedang terjadi tanpa harus melihat tampilan visual (seperti layar OLED atau LED indikator).

4.2.1.8. Percobaan Output Relay (Akses Masuk)

Tujuan :

Menguji fungsi relay dalam mengaktifkan dan menonaktifkan kunci pintu berdasarkan hasil autentikasi pengguna.

Input yang diberikan :

- A. Kartu RFID valid.
- B. Kartu RFID tidak valid atau tanpa autentikasi.

Output yang dihasilkan :

- A. Relay aktif membuka kunci selama beberapa detik saat kartu valid.
- B. Relay tidak aktif jika kartu tidak valid atau akses ditolak.

Analisa Hasil :

Berdasarkan percobaan yang dilakukan pada output relay, sistem menunjukkan bahwa fungsi relay beroperasi sesuai dengan logika yang telah diprogram. Ketika kartu RFID yang sah ditempelkan pada pembaca, relay bekerja dengan baik dan mengaktifkan kunci pintu selama beberapa detik. Ini menunjukkan bahwa proses autentikasi selesai dan pengguna yang sah dapat mengakses sistem.

Sebaliknya, ketika kartu RFID tidak valid atau proses autentikasi tidak dilakukan, relay tetap tidak aktif, sehingga kunci pintu tidak dapat dibuka. Ini menunjukkan bahwa sistem dapat mencegah pengguna yang tidak berwenang mengaksesnya.

4.2.1.9. Percobaan Output LED Indikator (Akses Masuk)

Tujuan :

Menguji kejelasan sinyal visual dari LED indikator yang menunjukkan status sistem secara cepat kepada pengguna.

Input yang diberikan :

- A. Kartu RFID valid dan tidak valid.
- B. Sistem dalam kondisi standby.
- C. Gerakan terdeteksi oleh sensor PIR.

Output yang dihasilkan :

- A. LED hijau menyala saat akses diterima.
- B. LED merah menyala saat akses ditolak.
- C. Kedua led tidak menyala dalam mode standby atau saat mendeteksi gerakan.

Analisa Hasil :

Percobaan menunjukkan LED indikator berfungsi sesuai harapan. LED hijau menyala saat akses diterima, dan LED merah menyala saat akses ditolak. Dalam kondisi standby atau saat hanya ada gerakan dari sensor PIR, tidak ada LED yang menyala, menunjukkan sistem menunggu autentikasi. Led indikator dibuat agar cepat dikenali, dan mudah dimengerti oleh pengguna, yang pada tujuannya meningkatkan efektivitas sistem dalam menyampaikan informasi secara real-time.

4.2.1.10. Percobaan Input Sensor RFID (Akses Keluar)

Sistem akses keluar tidak menggunakan sensor pintu magnetik ataupun layar OLED, sehingga skenario akses paksa dan tampilan visual tidak diuji pada sisi keluar.

Tujuan :

Menguji kemampuan sistem dalam memproses akses keluar menggunakan kartu RFID, baik yang terdaftar ataupun tidak terdaftar, serta menguji respons sistem terhadap input yang sah dan tidak sah.

Input dan Skenario yang diberikan:

A. Kartu RFID Terdaftar

Pengguna menempelkan kartu RFID yang telah terdaftar ke pembaca (RFID Reader RC522) untuk keluar dari ruangan.

B. Kartu RFID Tidak Terdaftar

Kartu RFID yang tidak terdaftar ditempelkan ke pembaca RFID.

Output yang dihasilkan:

A. Untuk RFID Terdaftar (Akses Keluar):

1. Relay aktif selama 5 detik dan solenoid membuka pintu.
2. LED hijau menyala sebagai indikator sukses.
3. Buzzer berbunyi dua kali sebagai sinyal sukses.
4. Notifikasi Discord dikirimkan dengan status Akses Diterima.
5. Server mencatat log akses keluar sebagai entri berhasil.

B. Untuk RFID yang Tidak Terdaftar (Akses Keluar):

1. Relay tidak aktif, pintu tidak terbuka.

2. Led merah menyala sebagai indikator penolakan.
3. Buzzer berbunyi satu kali panjang diikuti dua bunyi cepat sebagai sinyal penolakan.
4. Notifikasi Discord dikirimkan dengan status “Akses Ditolak”.
5. Server tidak mencatat log akses keluar karena kartu tidak valid.

4.2.1.11. Percobaan Output Buzzer Pasif (Akses Keluar)

Tujuan :

Mengidentifikasi fungsi buzzer sebagai umpan balik audio pada sistem akses keluar, untuk memberi sinyal apakah proses autentikasi pengguna berhasil (akses diterima) atau ditolak (akses ditolak).

Input yang diberikan :

- A. Kartu RFID terdaftar.
- B. Kartu RFID tidak terdaftar.

Output yang dihasilkan :

- A. Buzzer berbunyi dua kali pendek sebagai sinyal bahwa akses keluar berhasil.
- B. Buzzer berbunyi satu kali panjang diikuti dua bunyi cepat sebagai sinyal bahwa akses ditolak.

Analisa Hasil :

Dari percobaan yang dilakukan, buzzer berhasil memberikan respon yang berbeda berdasarkan status autentikasi.

4.2.1.12. Percobaan Output LED Indikator (Akses Keluar)

Tujuan :

Menguji kejelasan sinyal visual yang menunjukkan status hasil autentikasi akses keluar, agar pengguna dapat mengetahui apakah akses diterima atau ditolak.

Input yang diberikan :

- A. Kartu RFID terdaftar.
- B. Kartu RFID tidak terdaftar.
- C. Sistem dalam kondisi standby.

Output yang dihasilkan :

- A. LED hijau menyala saat akses keluar diterima (kartu valid).
- B. LED merah menyala saat akses keluar ditolak.
- C. Kedua led tidak menyala dalam mode standby atau saat mendeteksi gerakan.

Analisa Hasil :

Percobaan menunjukkan LED indikator berfungsi sesuai dengan logika pada sistem akses keluar. LED hijau aktif untuk memberikan sinyal visual bahwa kartu diterima dan pintu terbuka, sedangkan LED merah menyala untuk menandakan akses ditolak. Di luar proses autentikasi, LED tetap mati.

4.2.1.13. Percobaan Output Relay (Akses Keluar)

Tujuan :

Menguji kinerja modul relay sebagai pengendali aktuator solenoid pada akses keluar, berdasarkan hasil autentikasi kartu RFID.

Input yang diberikan :

- A. Kartu RFID valid.
- B. Kartu RFID tidak valid atau tanpa autentikasi.

Output yang dihasilkan :

- A. Relay aktif selama 5 detik, mengalirkan arus ke solenoid untuk membuka pintu.
- B. Relay tidak aktif, pintu tetap terkunci jika kartu tidak valid.

Analisa Hasil :

Hasil pengujian menunjukkan bahwa fungsi relay berjalan sesuai ekspektasi pada sistem akses keluar. Ketika kartu RFID yang telah terdaftar ditempelkan ke pembaca, relay terpicu selama 5 detik untuk membuka pintu. Setelah durasi tersebut, relay kembali nonaktif, memastikan pintu terkunci kembali. Sebaliknya, ketika kartu tidak terdaftar atau akses ditolak, relay tidak aktif sama sekali, mencegah pembukaan pintu. Ini membuktikan bahwa sistem dapat membedakan antara pengguna yang berhak dan tidak berhak, dan menggunakan relay secara efektif sebagai pengendali aktuator pintu.

4.2.1.14. Percobaan Integrasi (Sistem Aplikasi dan Monitoring Discord)

Tujuan :

Menguji keterpaduan antara sistem perangkat keras dan perangkat lunak untuk mengirim log aktivitas secara otomatis ke aplikasi web dan monitoring Discord melalui webhook.

Input yang diberikan :

- A. Data ID kartu, nama, status, waktu akses.

Output yang dihasilkan :

- B. Log dikirim ke server API
- C. Notifikasi otomatis masuk ke channel Discord.
- D. Informasi yang dikirim meliputi Akses Diterima / Ditolak, Nama, RFID Tag, dan Waktu

Analisa Hasil :

Sistem berhasil mengirim log aktivitas ke server API dan selanjutnya diteruskan ke channel atau saluran discord secara otomatis. Informasi yang dikirim meliputi ID kartu, nama, status (akses diterima/ditolak), dan waktu akses. yang seluruhnya muncul secara real-time di Discord. Hal ini menunjukkan integrasi antara perangkat keras (ESP32) dan perangkat lunak (API + Discord Webhook) berjalan sukses dan stabil.

4.2.1.15. Percobaan Pendaftaran Melalui Web Aplikasi

Tujuan :

Menguji fungsionalitas manajemen pengguna pada aplikasi web untuk proses pendaftaran kartu RFID ke dalam sistem agar dapat diotorisasi saat digunakan.

Input yang diberikan :

- A. Admin membuka halaman login dan masuk ke halaman manajemen pengguna.
- B. Admin menambahkan pengguna atau user ke dalam tambah user di halaman manajemen pengguna. Admin mengisi data pengguna seperti :

- C. Nama pengguna,
- D. Email pengguna,
- E. Password untuk login ke aplikasi web (Jika admin mengisi Role sebagai “Admin”).
- F. RFID Tag (diambil dari hasil scan atau label sebelumnya) dan
- G. Role.
- H. Admin menyimpan data ke dalam database melalui form Tambah User

Output yang dihasilkan :

- A. Data pengguna baru muncul di daftar pengguna pada halaman manajemen pengguna.
- B. Data pengguna baru muncul pada notifikasi yang berada di management user pada channel atau saluran discord.
- C. Kartu RFID yang sebelumnya ditolak, kini diakui sebagai valid.
- D. Ketika kartu yang baru didaftarkan digunakan kembali, sistem memberikan respons:
- E. Relay aktif membuka pintu.
- F. OLED menampilkan “Akses Diterima”.
- G. Discord menerima notifikasi akses diterima dengan ID kartu dan nama pengguna yang sebelumnya didaftarkan.
- H. Aplikasi web mencatat log dari data yang baru ditambahkan.

Analisa Hasil :

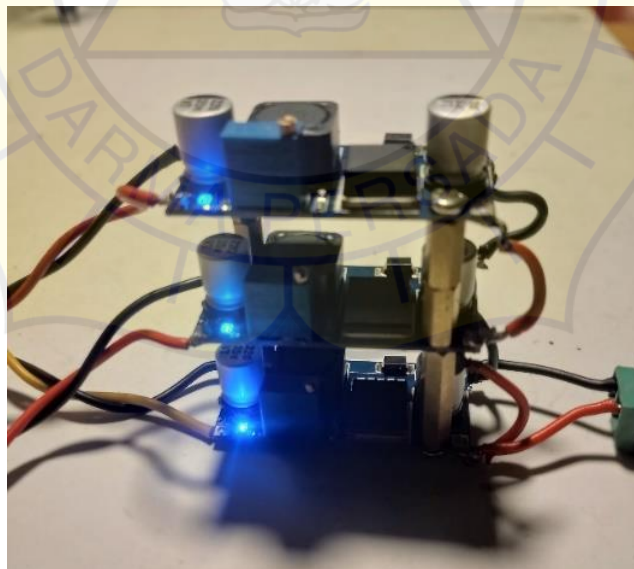
Fitur pendaftaran pengguna melalui web aplikasi berjalan dengan baik. Proses ini memungkinkan admin mengelola otorisasi akses dengan mudah, tanpa perlu

memodifikasi langsung kode pada perangkat ESP32. Data disimpan secara terpusat, sehingga memudahkan untuk pencatatan pengguna. Setelah pendaftaran, integrasi sistem tetap berjalan secara sinkron dengan perangkat *Smart Access* berbasis IoT.

4.2.2. Testing Hasil

Pengujian dilakukan dengan menggunakan testing modul per unit untuk menguji bagaimana sistem dapat merespons terhadap berbagai masukan (input) dari sensor dan interaksi pengguna. Sistem dirancang untuk menerima masukan dari kartu RFID ke sensor RFID, sensor gerak atau PIR, dan sensor magnetik pintu (MC-38), lalu memberikan umpan balik berupa suara (Buzzer dan DFPlayerMini), indikator visual (layar OLED dan LED indikator), aktuator (relay, solenoid, dan speaker), serta pengiriman notifikasi ke sistem aplikasi dan Discord.

4.2.2.1. Pengujian Power Supply



Gambar 4. 26. Power Supply

Pengujian catu daya yang dilakukan pada gambar 4.25. untuk memastikan kinerjanya sesuai desain. Catu daya menggunakan input 19,5V 2.31A yang diatur melalui tiga modul regulator LM2596 , di mana setiap modul menghasilkan tegangan output berbeda sesuai kebutuhan komponen. Masing masing per komponen memiliki Indikator LED biru menunjukkan kondisi catu daya menyala dan stabil. Hasil pengujian membuktikan bahwa sistem mampu menghasilkan tegangan dan arus yang sesuai, memastikan operasional perangkat berjalan lancar.

Tabel 4.17. merupakan hasil dari pengukuran power supply:

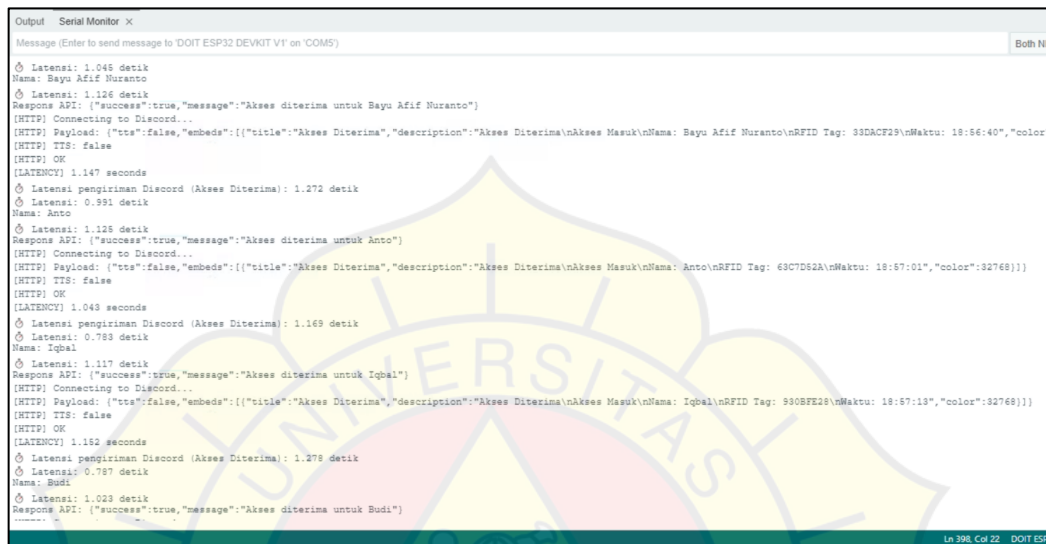
Tabel 4. 17. Tabel pengukuran Power Supply

| Pengukuran | Percobaan ke- | V- Out | Tanpa Beban | Dengan Beban | Arus (A) |
|-------------------------|---------------|--------|-----------------------|--------------|----------|
| | | | <i>Hasil Ukur (V)</i> | | |
| Power Supply 19,5v | 1 | 19,5 | 20,08 | 19,92 | 1,10 |
| | 2 | 19,5 | 20,05 | 19,90 | 1,12 |
| | 3 | 19,5 | 20,10 | 19,95 | 1,09 |
| Stepdown LM2596 (12V) | 1 | 12 | 12,08 | 11,95 | 0,85 |
| | 2 | 12 | 12,10 | 11,92 | 0,84 |
| | 3 | 12 | 12,05 | 11,98 | 0,86 |
| Stepdown LM2596 #1 (5V) | 1 | 5 | 5,04 | 4,98 | 0,45 |
| | 2 | 5 | 5,02 | 4,97 | 0,46 |
| | 3 | 5 | 5,05 | 4,96 | 0,44 |
| Stepdown LM2596 #2 (5V) | 1 | 5 | 5,08 | 4,95 | 0,48 |
| | 2 | 5 | 5,01 | 4,93 | 0,50 |
| | 3 | 5 | 5,07 | 4,92 | 0,47 |

4.2.2.2. Pengujian Waktu Respons Sensor RFID

Pengujian RFID ini akan menganalisis berapa lama waktu yang dibutuhkan oleh sistem untuk mengirim pesan ke web aplikasi ataupun discord. Pengujian dilakukan dengan mencatat waktu mulai saat kartu RFID ditempelkan hingga pesan diterima oleh sistem monitoring. Pada gambar 4.26. menunjukkan proses pengujian waktu respons RFID yang dilakukan secara langsung menggunakan beberapa kartu

berbeda. Tabel 4.18 menyajikan hasil pengujian waktu respons sensor RFID terhadap lima kartu, terdiri dari tiga kartu yang memiliki akses (terdaftar) dan dua kartu yang tidak memiliki akses (tidak terdaftar).



Gambar 4. 27. Pengujian Waktu Respons Sensor RFID

Tabel 4. 18. Tabel pengujian waktu respons Sensor RFID

| <i>Pengujian ke -</i> | <i>Nama Pengguna</i> | <i>Status Akses</i> | <i>Waktu Respon RFID (Detik)</i> |
|------------------------------|-----------------------------|----------------------------|---|
| 1 | Bayu Afif Nuranto | Diterima | 0,991 |
| 2 | Anto | Diterima | 0,783 |
| 3 | Iqbal | Diterima | 0,787 |
| 4 | Null | Ditolak | 0,925 |
| 5 | Null | Ditolak | 0,813 |

Berdasarkan data pada gambar 4.26. dan Tabel 4.18, pengujian dilakukan dengan menggunakan total lima kartu RFID, yang terdiri dari tiga kartu terdaftar (akses diterima) dan dua kartu tidak terdaftar (akses ditolak).

Penghitungan rata-rata waktu respons dilakukan dengan rumus sebagai berikut:

Akses Diterima = 3 data

$$\left(\text{Rata Rata Diterima} = 0,991 + 0,783 + 0,787 = \frac{2,561}{3} = 0,8537 \text{ detik} \right)$$

Akses Ditolak = 2 data

$$\left(\text{Rata Rata Ditolak} = 0,925 + 0,813 = \frac{0,738}{3} = 0,869 \text{ detik} \right)$$

Rata Rata keseluruhan

$$\left(\text{Rata Rata} = 0,991 + 0,783 + 0,787 + 0,925 + 0,813 = \frac{4,299}{5} = 0,8598 \text{ detik} \right)$$

Berdasarkan hasil pengujian ini membutuhkan waktu 0,8598 detik untuk memproses pembacaan kartu RFID hingga mengirimkan notifikasi ke aplikasi web dan Discord. Hasil ini menunjukkan bahwa sistem memiliki waktu respons yang cepat dan efisien.

4.2.2.3. Pengujian Waktu Respons Sensor PIR

Pengujian selanjutnya ini akan menganalisis berapa lama waktu yang dibutuhkan terhadap waktu respons dari gerakan manusia saat tangan didekatkan ke dekat pintu. Pada gambar 4.27. menunjukkan proses pengujian waktu respons sensor PIR secara langsung pada skenario mendekatkan tangan ke pintu. Tabel 4.19 menyajikan hasil pengukuran waktu respons sensor PIR terhadap sepuluh percobaan yang dilakukan secara berulang dengan kondisi gerakan yang seragam.

4.2.2.4. Pengujian Input Sensor MC-38 (Magnetic Switch)

Pengujian terhadap sensor MC-38 sebagai magnetic switch dilakukan untuk mengetahui waktu respons dari sensor ketika mendeteksi perubahan kondisi pintu, yaitu dari tertutup ke terbuka dan sebaliknya. Untuk mengukur waktu respons, logika perubahan status digital dari sensor dicatat menggunakan fungsi millis() pada mikrokontroler. Waktu antara perubahan fisik kondisi pintu dan perubahan sinyal dari sensor dicatat sebagai waktu respons. Pada gambar 4.28. menunjukkan proses pengujian input sensor MC-38 secara langsung.

```

esp32_sensor-mc38.ino
19   currentState = digitalRead(magneticPin);
20
21   if (currentState == lastMagneticState) return; //
22
23   static unsigned long lastChangeTime = 0;
24   float responseTimeSec = (changeTime - lastChangeTime) / 1000.0;
25
26   lastChangeTime = changeTime;
27
28   if (currentState == LOW) {
29     Serial.println("Pintu Tertutup");
30   } else {
31     Serial.println("Pintu Terbuka");
32   }
  
```

Serial Monitor X Output

Message (Enter to send message to 'DOIT ESP32 DEVKIT V1' on 'COM5')

Pintu Terbuka
Waktu Respons Sensor MC-38: 2.618 detik
Pintu Tertutup
Waktu Respons Sensor MC-38: 0.204 detik
Pintu Terbuka
Waktu Respons Sensor MC-38: 0.438 detik
Pintu Tertutup
Waktu Respons Sensor MC-38: 2.503 detik
Pintu Terbuka
Waktu Respons Sensor MC-38: 0.218 detik
Pintu Tertutup
Waktu Respons Sensor MC-38: 0.203 detik
Pintu Terbuka
Waktu Respons Sensor MC-38: 0.116 detik
Pintu Tertutup
Waktu Respons Sensor MC-38: 0.504 detik
Pintu Terbuka
Waktu Respons Sensor MC-38: 0.898 detik
Pintu Tertutup
Waktu Respons Sensor MC-38: 0.503 detik

Gambar 4. 29. Pengujian Input Sensor MC-38 (*Magnetic switch*)

Tabel 4. 20. Tabel pengujian sensor MC-38

| <i>Pengujian ke -</i> | <i>Kondisi Pintu</i> | <i>Waktu Respons Detik</i> |
|-----------------------|----------------------|----------------------------|
| 1 | Terbuka | 2,618 |
| 2 | Tertutup | 0,204 |
| 3 | Terbuka | 0,438 |
| 4 | Tertutup | 2,503 |
| 5 | Terbuka | 0,218 |
| 6 | Tertutup | 0,203 |

| | | |
|-----------|----------------|--------|
| 7 | Terbuka | 0,116 |
| 8 | Tertutup | 0,504 |
| 9 | Terbuka | 0,838 |
| 10 | Tertutup | 0,503 |
| Rata Rata | Pintu terbuka | 0,8456 |
| | Pintu tertutup | 0,7834 |
| | Keseluruhan | 0,8145 |

Pada Tabel 4.20. ditunjukkan hasil pengukuran waktu respons (T) sensor MC-38.

Pengujian dilakukan sebanyak 10 kali dengan menggunakan mikrokontroler ESP32, yang dibagi menjadi dua kondisi utama: pintu terbuka dan tertutup. Untuk kondisi pintu terbuka (pengujian ke 1, 3, 5, 7, dan 9), waktu respons yang dicatat adalah sebagai berikut:

$$(T_1 = 2,618 + T_3 = 0,438 + T_5 = 0,218 + T_7 = 0,116 + T_9 = 0,838)$$

Total waktu respons:

$$(T_{total-terbuka} = 2,618 + 0,438 + 0,218 + 0,116 + 0,838 = 4,228 \text{ detik})$$

Rata – rata waktu respons:

$$\left(T_{rata2-terbuka} = \frac{T_{Total-Terbuka}}{n} = \frac{4,228}{5} = 0,8456 \text{ detik} \right)$$

Untuk kondisi pintu tertutup (pengujian ke 2, 4, 6, 8, dan 10), waktu respons yang dicatat adalah sebagai berikut:

$$(T_2 = 0,204 + T_4 = 2,503 + T_6 = 0,203 + T_8 = 0,504 + T_{10} = 0,503)$$

Total waktu respons:

$$(T_{total-tertutup} = 2,618 + 0,438 + 0,218 + 0,116 + 0,838 = 3,917 \text{ detik})$$

Rata – rata waktu respons:

$$\left(T_{rata2-tertutup} = \frac{T_{Total-Tertutup}}{n} = \frac{3,917}{5} = 0,7834 \text{ detik} \right)$$

Rata – Rata keseluruhan :

$$T_{rata-rata \text{ keseluruhan}} = \frac{T_{Total}}{n_{Total}} = \frac{8,145}{10} = 0,8145 \text{ detik}$$

Hasil ini menunjukkan bahwa sensor MC-38 memiliki performa respons yang baik dalam mendeteksi perubahan status pintu, baik saat terbuka maupun tertutup.

Perbedaan rata-rata antara dua kondisi tersebut tidak signifikan, yang menunjukkan bahwa sensor bekerja secara konsisten dan dapat diandalkan untuk sistem monitoring pintu berbasis IoT.

4.2.2.5. Pengujian Layar OLED Display

Pada tabel 4.21. merupakan pengujian pada layar yang dilakukan untuk memastikan bahwa modul OLED Display mampu menampilkan informasi secara *real-time* dan dinamis berdasarkan kondisi serta input yang diterima dari berbagai komponen Sistem *Smart Access* ini. Informasi yang ditampilkan mencakup status sistem, hasil autentikasi kartu RFID, deteksi gerakan, status pintu, dan indikasi akses paksa (*trespassing*).

Tabel 4. 21. Tabel Pengujian Layar OLED Display

| No | Kondisi Sistem | Input yang Diberikan | Output yang diharapkan |
|----|----------------------------|---|---|
| 1 | Sistem dinyalakan | Sistem dinyalakan | Menampilkan pesan “Sistem Smart Access, Loading...” |
| 2 | Deteksi gerakan | Sensor PIR mendeteksi gerakan | Menampilkan pesan: “Sistem Smart Access, Silakan Tap”. |
| 3 | Autentikasi kartu berhasil | Kartu RFID valid ditempelkan | Menampilkan pesan: “RFID Detected, UID” dan “Akses Diterima, Nama dan Silakan masuk”. |
| 4 | Autentikasi kartu gagal | Kartu RFID tidak valid | Menampilkan “Akses Ditolak”. |
| 5 | Pintu dibuka secara paksa | Sensor magnetik mendeteksi pintu terbuka tanpa izin | Menampilkan pesan: “Trespassing”. |

4.2.2.6. Pengujian Buzzer

Pada tabel 4.22. merupakan tabel pengujian buzzer dilakukan secara langsung pada modul buzzer dengan memberikan input kondisi tertentu seperti RFID tidak dikenali dan pintu dibuka tanpa izin.

Tabel 4. 22. Tabel pengujian Buzzer

| No | Input kondisi | Input | Output yang Diharapkan | Output yang Dihasilkan | Hasil pengujian |
|----|------------------------|---|---|---|-----------------|
| 1 | Akses ditolak | Menempelkan RFID yang tidak terdaftar | Buzzer menyala selama 1 detik | Buzzer menyala selama 1 detik | Sesuai |
| 2 | Tindakan trespassing | Membuka pintu tanpa autentikasi RFID | Buzzer menyala terus selama pintu terbuka | Buzzer menyala terus selama pintu terbuka | Sesuai |
| 3 | Kondisi normal kembali | Pintu ditutup kembali setelah trespassing | Buzzer mati | Buzzer mati | Sesuai |

4.2.2.7. Pengujian DFPlayer Mini

Pada Tabel 4.23 ditampilkan hasil pengujian fungsi DF Player Mini dalam sistem Smart Access. Pengujian dilakukan dengan memberikan berbagai kondisi input, seperti sistem dinyalakan, deteksi gerakan oleh sensor PIR, autentikasi kartu RFID, serta status pintu yang terdeteksi oleh sensor magnetik MC-38.

Tabel 4. 23. Tabel pengujian DF Player Mini

| No | Kondisi Pengujian | Input | Output yang Diharapkan | Hasil pengujian | Keterangan |
|----|--|----------------------------------|---|-----------------|--------------|
| 1 | Sistem dinyalakan atau sensor PIR mendeteksi gerakan | Power ON atau Gerakan terdeteksi | Audio 1: "Selamat Datang di Smart Access" | Sesuai | Audio sesuai |

| | | | | | |
|---|------------------------------|--|---|--------|--------------|
| 2 | Kartu RFID valid ditempelkan | Kartu valid | Audio 2: “Pintu terbuka, silakan masuk” | Sesuai | Audio sesuai |
| 3 | Pintu menutup kembali | Sensor magnetik MC-38 aktif | Audio 3: “Pintu tertutup” | Sesuai | Audio sesuai |
| 4 | Pintu dibuka tanpa otorisasi | Sensor magnetik terputus tanpa deteksi kartu | Audio 4: “Peringatan, Ada pembobolan ...” | Sesuai | Audio sesuai |
| 5 | Kartu RFID tidak valid | Kartu tidak terdaftar | Audio 5: “Akses ditolak” | Sesuai | Audio sesuai |

Dari hasil pengujian, DFPlayer Mini mampu memberikan umpan balik audio secara real-time dan sesuai dengan skenario sistem. Respons yang diberikan oleh perangkat sudah sesuai.

4.2.2.8. Pengujian Relay

Pada Tabel 4.24. merupakan hasil pengujian terhadap fungsi relay dalam sistem Smart Access. Pengujian dilakukan dengan memberikan beberapa kondisi input berupa kartu RFID valid, kartu tidak valid, serta tanpa adanya input RFID. Hasil pengujian menunjukkan bahwa relay berfungsi sesuai harapan: aktif dan membuka kunci pintu ketika kartu valid digunakan, serta tetap nonaktif saat kartu tidak valid atau tidak ada input. Hal ini membuktikan bahwa mekanisme pengendalian kunci melalui relay bekerja dengan baik dan responsif terhadap autentikasi yang diberikan.

Tabel 4. 24. Tabel pengujian Relay

| No | Kondisi Pengujian | Input | Diharapkan | Hasil | Hasil pengujian |
|----|------------------------|----------------------|---|------------------------------|-----------------|
| 1 | Kartu RFID valid | Didekatkan ke reader | Relay aktif, Kunci pintu terbuka selama 3–5 detik | Relay menyala, kunci terbuka | Sesuai |
| 2 | Kartu RFID tidak valid | Didekatkan ke reader | Relay tetap nonaktif | Relay tidak menyala | Sesuai |
| 3 | Tidak ada kartu | Tidak ada input RFID | Relay tetap nonaktif | Relay tidak menyala | Sesuai |

4.2.2.9. Pengujian LED Indikator

Pengujian LED indikator dilakukan untuk mengetahui apakah LED merah dan hijau menyala sesuai dengan kondisi sistem berdasarkan input yang diterima. LED hijau digunakan sebagai indikator autentikasi akses, sedangkan LED merah digunakan sebagai indikator penolakan akses. Hasil pengujian disajikan pada Tabel 4.25.

Tabel 4. 25. Tabel Pengujian Led Indikator

| No | Skenario Pengujian | Input | Kondisi Sistem | Led Merah | Led Hijau | Keterangan |
|----|----------------------------|----------------------------|----------------|-----------|-----------|---|
| 1 | Kartu RFID valid | RFID Terdaftar | Autentikasi | Mati | Menyala | Akses diterima, LED hijau menyala sebagai indikator sukses. |
| 2 | Kartu RFID tidak valid | RFID tidak terdaftar | Autentikasi | Menyala | Mati | Akses ditolak, LED merah menyala sebagai peringatan. |
| 3 | Deteksi gerakan | Gerakan dideteksi oleh PIR | Standby | Mati | Mati | Hanya mendeteksi gerakan, tidak mengubah status LED. |
| 4 | Sistem standby tanpa input | Tidak ada input | Standby | Mati | Mati | Sistem siap menerima input, LED tetap mati |

4.2.2.10. Pengujian Integrasi (Sistem Aplikasi dan Monitoring Discord)

Pengujian ini digunakan untuk menguji keterpaduan antara sistem perangkat keras dengan perangkat lunak (server API dan Discord Webhook), untuk memastikan data log aktivitas dapat dikirim dan ditampilkan secara real-time ke aplikasi web dan saluran Discord. Pada tabel 4.26. merupakan skenario pengujian dan pada tabel 4.27. sebagai hasil dari skenario pengujian.

Tabel 4. 26. Tabel Skenario Pengujian Integrasi Sistem Aplikasi dan Monitoring Discord

| No | Skenario Pengujian | Input | Output yang diharapkan |
|----|--|--|---|
| 1 | Akses diterima oleh sistem. | ID RFID valid, Nama, Status “Akses Diterima”, Waktu | Log masuk API + Notifikasi Discord dengan pesan: Akses Diterima – Nama – ID – Waktu |
| 2 | Akses ditolak oleh sistem. | ID RFID tidak dikenal, Nama "Unknown", Status “Ditolak”, Waktu | Log masuk API + Notifikasi Discord dengan pesan: Akses Ditolak – ID – Waktu |
| 3 | Koneksi terputus saat pengiriman log ke server API | ID RFID valid, namun jaringan mati | Data tidak terkirim, pesan kesalahan pada ESP32, tidak ada notifikasi Discord |
| 4 | Format data tidak sesuai saat dikirim ke API | Format JSON salah / tidak lengkap | API menolak permintaan, error log pada ESP32, tidak ada notifikasi Discord |
| 5 | Sistem idle (tidak ada aktivitas RFID) | Tidak ada aktivitas RFID | Tidak ada log terkirim ke API dan tidak ada notifikasi ke Discord |

Tabel 4. 27. Tabel hasil pengujian integrasi Sistem Aplikasi dan Monitoring

Discord

| No | Skenario Pengujian | Hasil pengujian | Sesuai / Tidak sesuai |
|----|--------------------------|--|-----------------------|
| 1 | Akses diterima | Log berhasil terkirim ke API dan muncul notifikasi di Discord dalam waktu <1 detik | Sesuai |
| 2 | Akses ditolak | Log berhasil terkirim ke API dan muncul notifikasi akses ditolak di Discord | Sesuai |
| 3 | Koneksi terputus | Gagal kirim log, tidak ada notifikasi Discord | Sesuai |
| 4 | Format data tidak sesuai | API menolak request, Discord tidak menerima notifikasi | Sesuai |
| 5 | Sistem Idle | Tidak ada aktivitas dan tidak ada pengiriman log | Sesuai |

Pengujian integrasi dinyatakan berhasil, dengan sistem menunjukkan stabilitas tinggi antara input dari perangkat keras dan output ke sistem monitoring.

4.2.2.11. Pengujian Pendaftaran Melalui Web Aplikasi

Pada Tabel 4.28. Pengujian ini dilakukan untuk menguji fungsionalitas fitur manajemen pengguna pada aplikasi web, khususnya proses pendaftaran pengguna baru beserta RFID-nya ke dalam sistem, untuk memastikan kartu tersebut dapat diotorisasi dan digunakan untuk mengakses sistem *Smart Access*.

Tabel 4. 28. Tabel pengujian Pendaftaran Aplikasi Web

| No | Langkah pengujian | Hasil yang diharapkan | Sesuai / Tidak sesuai |
|----|---------------------------------|-----------------------------|-----------------------|
| 1 | Admin login ke aplikasi web | Berhasil masuk ke dashboard | Sesuai |
| 2 | Buka halaman manajemen pengguna | Halaman daftar user tampil | Sesuai |

| | | | |
|---|---|---|--------|
| 3 | Klik “Tambah User” dan isi data (Nama, Email, RFID Tag, Role) | Form terisi dengan data yang valid | Sesuai |
| 4 | Simpan Data | Data tersimpan dan muncul di daftar pengguna | Sesuai |
| 5 | Sistem kirim notifikasi ke Discord | Discord menerima notifikasi user baru | Sesuai |
| 6 | Gunakan kartu RFID yang baru didaftarkan | Akses diterima, relay aktif, OLED tampil “Akses Diterima” | Sesuai |
| 7 | Periksa log aktivitas di halaman dashboard | Log akses muncul lengkap dengan waktu dan status. | Sesuai |

Pengujian ini berhasil dengan hasil sesuai harapan. Sistem web aplikasi terbukti mampu mendaftarkan pengguna baru dan mengotorisasi akses RFID dengan baik, serta mengintegrasikan log dan notifikasi ke seluruh sistem dengan lancar.

4.2.3. Modifikasi atau Optimalisasi Dari Sistem Terdahulu

Penelitian ini melakukan sejumlah modifikasi dan optimalisasi dibandingkan dengan sistem-sistem terdahulu yang menjadi referensi, baik dari sisi perangkat keras, perangkat lunak, maupun fitur fungsionalitas dan monitoring.:

1. Penambahan Integrasi multi-sensor dan modul ganda

Pada penelitian Peratama & Syazili (2022) hanya menggunakan NodeMCU dan solenoid lock, sistem ini memadukan beberapa sensor seperti Sensor RFID, Sensor PIR, Sensor MC-38 dan kunci solenoid. Hal ini bertujuan untuk meningkatkan akurasi validasi akses serta memperkuat sistem monitoring terhadap kondisi pintu.

2. Notifikasi real-time melalui Discord Webhook.

Pada penelitian Ningrum & Basyir (2022), dan Tamam & Romadhani (2023) mengandalkan Telegram dan WhatsApp sebagai media notifikasi, yang memiliki batasan tertentu dalam hal integrasi. Pada sistem ini, digunakan Discord Webhook sebagai media pengiriman notifikasi real-time, karena lebih cepat untuk pengujian setup awal dan tidak memerlukan nomor telepon.

3. Penggunaan 2 sistem akses (Akses masuk dan Akses keluar).

Berbeda dengan penelitian sebelumnya yang umumnya hanya fokus pada akses masuk, sistem ini mengimplementasikan dua jalur akses terpisah: untuk masuk dan keluar.

4. Peningkatan Antarmuka Aplikasi dan Monitoring.

Sistem ini menyediakan dashboard web yang terintegrasi dengan server untuk monitoring real-time, manajemen pengguna, dan pengamatan log akses. Ini merupakan peningkatan signifikan dibanding penelitian sebelumnya yang hanya menyediakan fungsionalitas dasar atau notifikasi satu arah. Antarmuka ini dibangun menggunakan framework modern seperti Hono.js, serta menyimpan data secara terstruktur di PostgreSQL (NeonDB).

5. Optimalisasi Audio dan Visual Feedback.

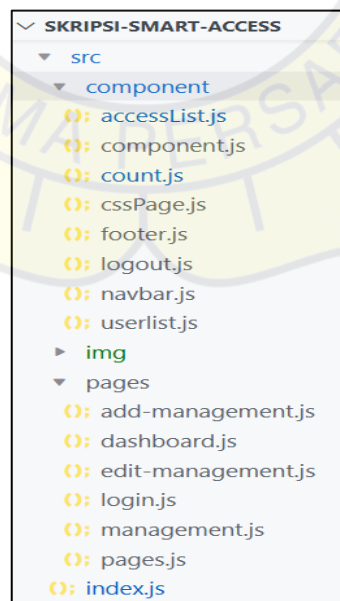
Untuk memberikan umpan balik yang jelas ke pengguna, sistem ini menambahkan Indikator LED (merah dan hijau) dan modul DFPlayer Mini untuk memberikan suara penanda selamat datang, akses diterima, akses ditolak, dan jika terjadi pembobolan.

4.2.4. Proses Deploy Sistem Aplikasi

Proses deploy sistem aplikasi merupakan tahapan akhir dari pembangunan sistem, di mana seluruh komponen *frontend*, *backend*, dan *database* diunggah ke server agar dapat diakses oleh pengguna secara online. Dalam implementasi ini, deployment dilakukan secara manual. Berikut ini merupakan rincian proses deploy yang dilakukan:

4.2.4.1. Struktur Proyek Aplikasi Sistem

Terlihat pada gambar 4.29 merupakan struktur proyek dari sistem ini dibangun dalam satu folder yaitu folder *src* yang terdiri dari:



Gambar 4. 30. Gambar Struktur Proyek dalam pembangunan aplikasi web

Pada gambar 4.29. adalah penjelasan dari folder src :

1. Pada file *index.js* : merupakan isi dari baris kode utama untuk *backend* aplikasi web.
2. Folder *component* : merupakan folder yang berisi file javascript komponen komponen seperti,
 - a. File *accessList.js* : berisi tentang fungsi tabel akses untuk menampilkan daftar tabel log akses.
 - b. File *component.js* : merupakan file yang digunakan untuk melakukan import dan export komponen terhadap file javascript.
 - c. File *count.js* : merupakan file yang digunakan untuk meletakkan fungsi *countUser()*: merupakan fungsi untuk menghitung banyaknya pengguna, *countUserGranted()*: merupakan fungsi untuk menghitung banyaknya pengguna akses yang disetujui dan *countUserdenied()*: merupakan fungsi untuk menghitung banyaknya pengguna akses yang ditolak.
 - d. File *cssPage.js* : digunakan untuk menampilkan style CSS pada halaman web.
 - e. File *Footer.js* : digunakan untuk menampilkan Footer pada halaman aplikasi.
 - f. File *Logout.js* : digunakan untuk fungsi keluar dari aplikasi sistem.
 - g. File *Navbar.js* : digunakan untuk menampilkan Navigasi dari aplikasi sistem.

- h. File *Userlist.js* : digunakan untuk menampilkan fungsi tabel user pengguna dari aplikasi sistem.
3. Folder *img* : digunakan untuk menaruh assets berupa gambar (*logo, favicon*) dari aplikasi sistem.
 4. Folder *pages* : merupakan folder yang berisi halaman halaman dari aplikasi web ini yang terdiri dari :
 - a. File *Add-management.js* : menampilkan halaman tambah pengguna beserta fungsi – fungsi dari backend ke aplikasi.
 - b. File *Dashboard.js* : digunakan untuk menampilkan halaman depan berupa dashboard aplikasi sistem yang terdiri dari (Selamat Datang, menghitung banyaknya pengguna, akses disetujui, dan akses ditolak. Serta menampilkan tabel akses dari database log akses).
 - c. File *Edit-management.js* : digunakan untuk menampilkan dan menyimpan fungsi edit dan delete dari aplikasi.
 - d. File *Login.js* : digunakan untuk halaman login ke aplikasi sistem.
 - e. File *Management.js* : merupakan halaman yang digunakan untuk menampilkan halaman manajemen
 - f. File *Pages.js* : digunakan untuk import dan export function seperti, memanggil halaman dashboard, memanggil halaman management, dan halaman login pada sistem.

4.2.4.2. Deploy Aplikasi ke Cloudflare Workers

Deploy aplikasi dilakukan menggunakan bantuan Wrangler CLI, yaitu *tools* dari Cloudflare untuk mengelola deploy aplikasi ke Workers. Adapun langkah-langkahnya adalah sebagai berikut:

1. Install Wrangler CLI

Pastikan Node.js sudah terpasang, bisa dicek dengan `node -v` atau `npm -v`.

Kemudian jika sudah terinstall, ketikkan pada terminal :

```
npm install -g wrangler (pastikan node.js sudah terinstall)
```

Selanjutnya login ke Cloudflare dengan mengetikkan :

```
wrangler login
```

Perintah ini akan membuka browser dan meminta login ke akun Cloudflare.

2. Setup file konfigurasi wrangler.toml

Buat file `wrangler.toml` di root ('/')

```
name = "sistem-smart-access" # nama proyek
main = "src/index.js" # file utama pada
aplikasi
compatibility_date = "2024-12-05" # tanggal kompatibilitas
worker yang digunakan
compatibility_flags = ["nodejs_compat"] # Mengaktifkan
kompatibilitas fitur Node.js.
```

3. Koneksi ke NeonDB

Karena Cloudflare Workers tidak mendukung koneksi database berbasis TCP Socket, maka diperlukan package `@neondatabase/serverless`.

Install package `neon-serverless`.

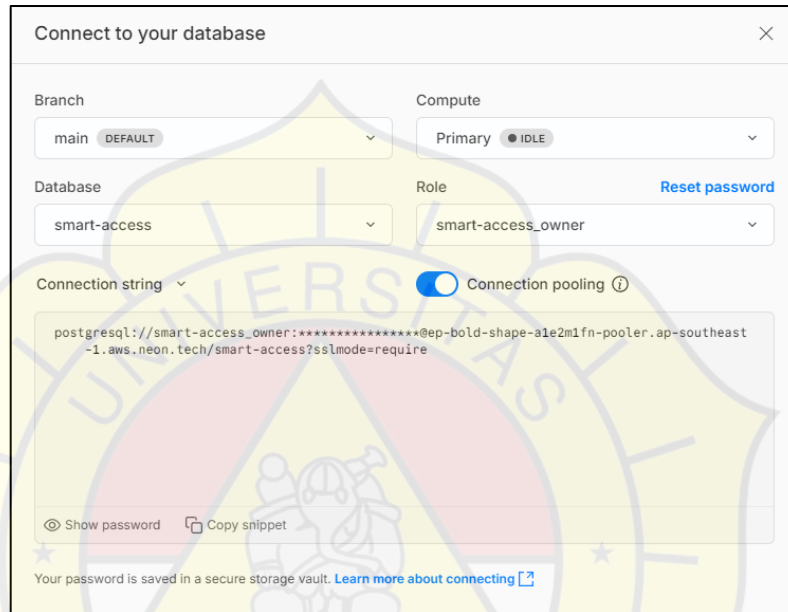
```
npm install @neondatabase/serverless
```

Kemudian import hono pada backend.

```
import { Client } from '@neondatabase/serverless'
```

4. Pasang Environment Variables

Pada Gambar 4.30. Connection String diambil dari platform Neon.Tech, setelah berhasil membuat database, dan klik *Connect to your Database* pada halaman dashboard neon.



Gambar 4. 31. Koneksi Database

Terlihat pada gambar 4.31. Setelah connection string diambil, konfigurasi koneksi database dimasukkan berupa connection string dari platform neon database ke kode project.

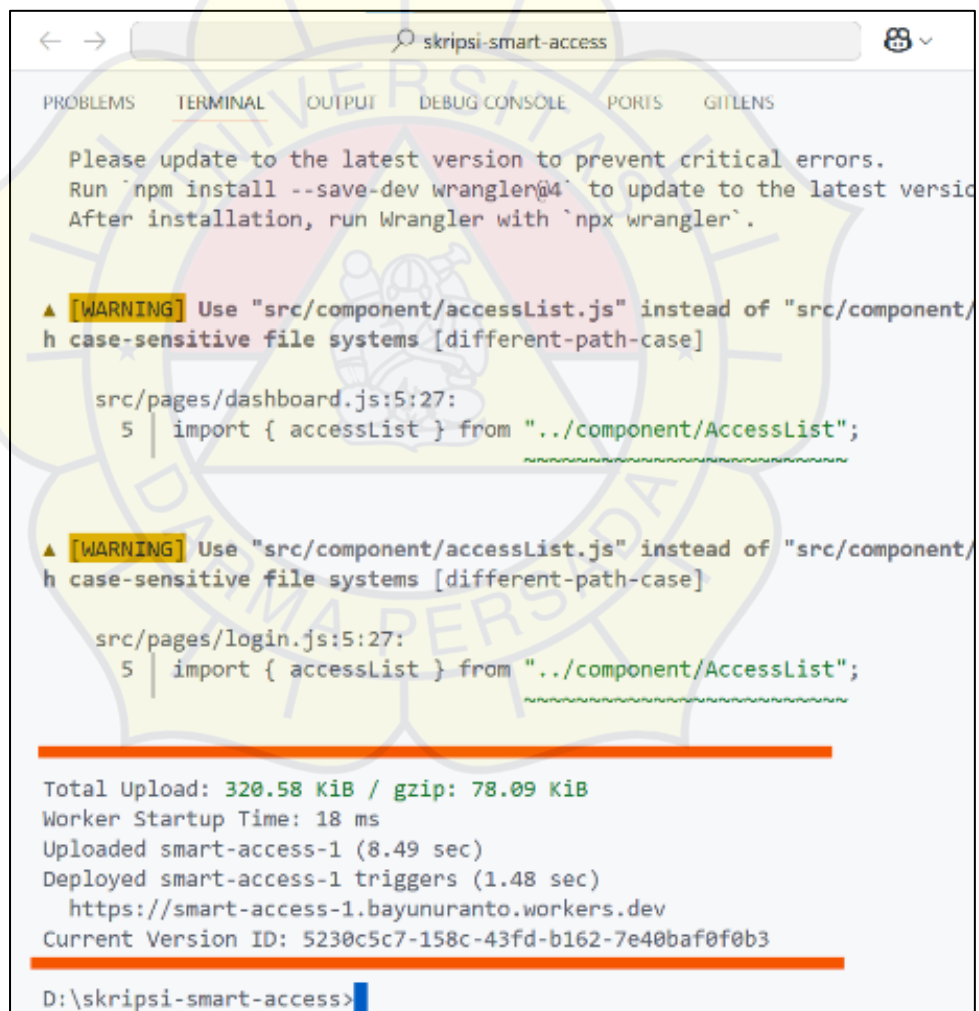
```
src > () index.js > ...
32
33 // Koneksi Database (Middleware) You, 2 months ago • update ...
34 app.use(async (c, next) => {
35   c.env.db = new Client({
36     connectionString: 'postgresql://smart-access_owner:mw3W3L6FjoE@ep-bold-shape-a1e2m1fn-pooler.ap-southeast-1.amazonaws.com/smart-access?sslmode=require'
37   });
38   await c.env.db.connect();
39   c.authToken = getCookie(c.req.header('Cookie'), 'auth_token');
40   await next();
41 });
```

Gambar 4. 32. Koneksi Database

5. Build & Deploy

npx wrangler deploy

Dapat terlihat pada gambar 4.32. merupakan langkah setelah berhasil di deploy. Output pada terminal menunjukkan bahwa proses upload telah selesai dengan total ukuran 320.58KB, dan worker berhasil di-deploy dalam waktu sekitar 8.49 detik. Versi terbaru dari aplikasi kini tersedia dengan ID 5230c5c7-158c-43fd-b162-7e40baf0f0b3, dan dapat diakses melalui <https://smart-access-1.bayunuranto.workers.dev>



```
← → skripsi-smart-access
PROBLEMS  TERMINAL  OUTPUT  DEBUG CONSOLE  PORTS  GITLENS

Please update to the latest version to prevent critical errors.
Run `npm install --save-dev wrangler@4` to update to the latest version.
After installation, run Wrangler with `npx wrangler`.

▲ [WARNING] Use "src/component/accessList.js" instead of "src/component/
h case-sensitive file systems [different-path-case]

src/pages/dashboard.js:5:27:
  5 | import { accesslist } from "../component/AccessList";
    |                                     ^
    |                                     ^
    |                                     ^

▲ [WARNING] Use "src/component/accessList.js" instead of "src/component/
h case-sensitive file systems [different-path-case]

src/pages/login.js:5:27:
  5 | import { accesslist } from "../component/AccessList";
    |                                     ^
    |                                     ^
    |                                     ^

Total Upload: 320.58 KiB / gzip: 78.09 KiB
Worker Startup Time: 18 ms
Uploaded smart-access-1 (8.49 sec)
Deployed smart-access-1 triggers (1.48 sec)
  https://smart-access-1.bayunuranto.workers.dev
Current Version ID: 5230c5c7-158c-43fd-b162-7e40baf0f0b3

D:\skripsi-smart-access>
```

Gambar 4. 33. Hasil Deploy