|  |  |
| --- | --- |
|  | UNIVERSITAS DIPONEGORO – FAKULTAS TEKNIK  DEPARTEMEN TEKNIK ELEKTRO  *Jl. Prof. H. Soedarto, SH, Tembalang, Semarang 50275*  *Telp/Faks. (024)-7460057 e-mail: departemen@elektro.undip.ac.id* |

**Dokumen Pengembangan Produk**

Lembar Sampul Dokumen

|  |  |  |
| --- | --- | --- |
| Judul Dokumen | **TUGAS AKHIR:**  **Rancang Bangun Sistem Keamanan Kunci Pintu GedungBerbasis *Internet of Things*** | |
|  |  | |
| Jenis Dokumen | **IMPLEMENTASI** | |
|  | Catatan: Dokumen ini dikendalikan penyebarannya oleh Dept. Teknik Elektro Undip | |
| Nomor Dokumen | **B500-01-TA2223.2.19012** | |
|  |  | |
| Nomor Revisi | **01** | |
|  |  | |
| Nama File | **B500-2-TA2223** | |
|  |  | |
| Tanggal Penerbitan | **8 September 2023** | |
|  |  | |
| Unit Penerbit | **Departemen Teknik Elektro Undip** | |
|  |  | |
| Jumlah Halaman | **15** | (termasuk lembar sampul ini) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Pengusul | | | | |
| Pengusul | Nama  NIM | **Henric Dhiki Wicaksono**  21060119120011 | Jabatan  Tanda Tangan | Anggota |
|  | Nama  NIM | **Novi Dianasari**  21060119120039 | Jabatan  Tanda Tangan | Anggota |
|  | Nama  NIM | **Muhammad Khoiril Wafi**  21060119140133 | Jabatan  Tanda Tangan | Anggota |
| Pembimbing | Nama | **M. Arfan, S.Kom., M.Eng.** | Tanda Tangan |  |
| Utama | NIP | 198408172015041002 |  |  |
| Pendamping | Nama | **Imam Santoso, S.T., M.T.** | Tanda Tangan |  |
|  | NIP | 197012031997021001 |  |  |

**DAFTAR ISI**

[Catatan Sejarah Perbaikan Dokumen 3](#_Toc143868352)

[1. PENDAHULUAN 4](#_Toc143868353)

[**1.1** **Ringkasan Isi Dokumen** 4](#_Toc143868354)

[**1.2** **Aplikasi Dokumen** 4](#_Toc143868355)

[1.3 **Referensi** 4](#_Toc143868356)

[**1.4** **Daftar Singkatan** 5](#_Toc143868357)

[2. PENGUJIAN 6](#_Toc143868358)

[2.1 Pengujian Fungsionalitas Komunikasi Bluetooth 6](#_Toc143868359)

[2.2 Pengujian Performa Komunikasi Bluetooth 7](#_Toc143868360)

[2.3 Pengujian Fungsionalitas Komunikasi WiFi 11](#_Toc143868361)

[2.4 Pengujian Performa Komunikasi WiFi 12](#_Toc143868362)

[2.5 Pengujin Fungsionalitas Penguncian 18](#_Toc143868363)

[2.6 Pengujian Fungsionalitas API 23](#_Toc143868364)

[2.7 Pengujian Performa API 35](#_Toc143868365)

[2.8 Pengujian Fungsionalitas Website 38](#_Toc143868366)

[2.9 Pengujian Performa Website 38](#_Toc143868367)

[2.10 Pengujian Fungsionalitas Aplikasi Mobile 38](#_Toc143868368)

[2.11 Pengujian Performa Aplikasi Mobile 38](#_Toc143868369)

[3. PENUTUP 39](#_Toc143868370)

# Catatan Sejarah Perbaikan Dokumen

|  |  |
| --- | --- |
| **VERSI, TGL, OLEH** | **PERBAIKAN** |
| 01,  8 September 2023,  oleh Henric Dhiki Wicaksono, Novi Dianasari  dan Muhammad Khoiril Wafi. | *Draft* Dokumen B200 |

1. PENDAHULUAN

## **Ringkasan Isi Dokumen**

Dokumen ini berisikan proses pengujian yang dilakukan pada proses pengembangan “Sistem Keamanan Kunci Pintu Gedung Berbasis *Internet of Things*” proses pengujian yang dilakukan meliputi pengujian secara fungsional dan pengujian untuk mengukur performa sistem. Dokumen ini menjelaskan metode serta alat yang digunakan dalam proses pengujian. Proses pengujian dilakukan untuk melakukan verifikasi dari hasil proses implementasi dengan spesifikasi dan standarisasi yang telah ditentukan pada proses desain.

## **Aplikasi Dokumen**

Dokumen ini digunakan dalam proses pengembangan “Rancang Bangun Sistem Keamanan Kunci Pintu Gedung Berbasis *Internet of Things*” untuk:

1. Sebagai penjelasan proses pengujian yang dilakukan.
2. Sebagai penjelasan mengenai metode dan alat yang digunakan dalam proses pengujian.
3. Sebagai acuan keberhasilan sistem sesuai dengan spesifikasi dan standarisasi yang telah ditentukan.
4. Sebagai dokumentasi dan pencatatan perubahan.

Dokumen B400 ini diajukan kepada dosen pembimbing tugas akhir dan tim tugas akhir Program Studi Sarjana Teknik Elektro Undip sebagai bahan penilaian tugas akhir.

## **Referensi**

[1] E. Novriza Alam and F. Dewi, “Performance Testing Analysis of Bandungtanginas Application With Jmeter,” *Int. J. Innov. Enterp. Syst.*, vol. 6, no. 01, pp. 85–94, 2022.

## **Daftar Singkatan**

**Tabel 1.1** Daftar Singkatan

| Singkatan | Arti |
| --- | --- |
| IoT | *Internet of Things* |
| WiFi | *Wireless Fidelity* |
| JSON | *Javascript Object Notation* |
| PC | *Personal Computer* |
| UI | *User Interface* |
| QR-*Code* | *Quick Response Code* |

1. PENGUJIAN
2. Pengujian Fungsionalitas Komunikasi *Bluetooth*

Pengujian komunikasi *bluetooth* bertujuan untuk melakukan pemeriksaan dan verifikasi fungsionalitas dari sistem komunikasi bluetooth yang sudah diimplementasikan pada perangkat kunci pintu. Pengujian dilakukan dengan menggunakan aplikasi *serial bluetooth terminal*. Aplikasi *serial bluetooth terminal* merupakan sebuah aplikasi yang menyediakan sebuah terminal yang dapat digunakan untuk mengirimkan dan menerima data dengan menggunakan komunikasi *bluetooth*. Proses pengujian fungsionalitas *bluetooth* dilakukan dalam 3 bagian yaitu :

1. Pengujian koneksi

Pengujian koneksi dilakukan dengan melakukan *pairing* pada aplikasi *serial bluetooth terminal*. Hasil proses *pairing* dapat dilihat pada gambar 2.1 dibawah.

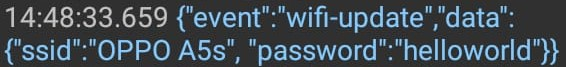


Gambar 2. Status *smartphone* terhubung ke HC-05

Berdasarkan Gambar 2.1 diatas, proses *pairing* berhasil dilakukan dengan status koneksi pada aplikasi *serial bluetooth terminal* yaitu “*Connected*”.

1. Pengujian pengiriman data

Pengujian pengiriman data dilakukan dengan mengirimkan data JSON yang disajikan dalam sebuah *string*, data yang dikirimkan sesuai dengan format data pada perangkat kunci pintu. Hasil dari proses pengujian pengiriman data dapat dilihat pada Gambar 2.2 dibawah.

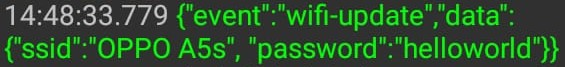


Gambar 2. Pengujian pengiriman data dari smartphone ke modul *bluetooth*

Pada Gambar 2.2 diatas, data {**"event":"wifi-update","data":{"ssid":"OPPO A5s", "password":"helloworld"}** berhasil terkirim ke perangkat kunci pintu ditandai dengan pesan yang berwarna biru.

1. Pengujian balasan data

Pengujian balasan data dilakukan dengan menunggu balasan dari setiap perintah yang dikirimkan dari *smartphone* ke perangkat kunci pintu. Hasil dari pengujian balasan data dapat dilihat pada Gambar 2.3 dibawah.



Gambar 2. Pengujian balasan data dari modul *bluetooth* ke *smartphone*

Pada Gambar 2.3 diatas, terlihat bahwa data {**"event":"wifi-update","data":{"ssid":"OPPO A5s", "password":"helloworld"}** berhasil diterima oleh aplikasi dan sesuai dengan balasan yang diharapkan.

Hasil pengujian di atas membuktikan bahwa komunikasi data dua arah melalui *bluetooth* yang diuji coba menggunakan aplikasi *serial bluetooth terminal* berjalan dengan baik. Perangkat *bluetooth* mampu menerima data dari *smartphone* dan mengirimkan balasan kembali ke *smartphone* melalui koneksi *bluetooth*. Hasil pengujian menunjukkan bahwa protokol komunikasi dan konfigurasi *serial* telah berhasil diatur dengan benar. Selama pengujian tersebut, tidak ada gangguan atau kesalahan dalam komunikasi yang terdeteksi. Pesan dapat dikirim dengan sukses dan balasan diterima dengan benar. Hal tersebut menunjukkan keandalan komunikasi antara perangkat *bluetooth* dan *smartphone*.

1. Pengujian Performa Komunikasi *Bluetooth*

Pada pengujian kinerja komunikasi data dua arah melalui *bluetooth* ini dilakukan dengan memperhatikan parameter QoS yang didefinisikan oleh TIPHON. Pengujian ini bertujuan untuk mengukur dan menganalisis karakteristik kinerja komunikasi *bluetooth*, termasuk *delay*, *throughput*, dan *packet loss*. Pengujin yang dilakukan yaitu :

1. Pengujian *delay*

Pengujian *delay* dilakukan dengan cara mengirimkan paket data dari *smartphone* ke modul *bluetooth* melalui koneksi *Bluetooth*. Kemudian, mencatat waktu yang dibutuhkan untuk paket data mencapai modul *bluetooth* setelah dikirim dari *smartphone*. Pengujian ini dilakukan beberapa kali dan rerata *delay* dihitung. Hasil pengujian *delay* komunikasi *bluetooth* ditunjukkan pada Tabel 2.1 di bawah.

Tabel 2. Hasil pengujian *delay* komunikasi *bluetooth*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Jarak** | **Waktu A** | **Waktu B** | **RTT** | ***Delay*** | **Keterangan** |
| 1 m | 15:24:43.762 | 15:24:44.304 | 542 ms | 271 ms | Bagus |
| 15:24:56.354 | 15:24:56.708 | 354 ms | 177 ms |
| 15:25:00.749 | 15:25:00.858 | 109 ms | 54,5 ms |
| Rata-rata *delay* (ms) | | | | **167,5** |
| 5 m | 15:29:20.865 | 15:29:20.994 | 129 ms | 64,5 ms | Sangat Bagus |
| 15:29:24.604 | 15:29:24.722 | 118 ms | 59 ms |
| 15:29:27.581 | 15:29:27.742 | 161 ms | 80,5 ms |
| Rata-rata *delay* (ms) | | | | **68 ms** |
| 10 m | 15:32:05.030 | 15:32:05.174 | 144 ms | 72 ms | Sangat Bagus |
| 15:32:08.974 | 15:32:09.095 | 121 ms | 60,5 ms |
| 15:32:11.982 | 15:32:12.109 | 127 ms | 63,5 ms |
| Rata-rata *delay* (ms) | | | | **65,3 ms** |

Keterangan : Waktu A = waktu ketika data dikirim

Waktu B = waktu ketika data diterima kembali oleh pengirim

RTT = *Round-Trip Time*

Dari Tabel 2.1 hasil pengujian *delay* komunikasi *bluetooth* antara *smartphone* ke modul *bluetooth* terjadi penurunan rata-rata *delay* seiring dengan meningkatnya jarak. Pada jarak 1 m rata-rata hasil dari pengujian *delay* komunikasi *bluetooth* untuk tiga kali percobaan adalah sebesar 167,5 ms sehingga tergolong bagus. Pada jarak 5 m rata-rata hasil dari pengujian *delay* komunikasi *bluetooth* untuk tiga kali percobaan adalah sebesar 68 ms sehingga tergolong sangat bagus. Pada jarak 10 m rata-rata hasil dari pengujian *delay* komunikasi *bluetooth* untuk tiga kali percobaan adalah sebesar 65,3 ms sehingga tergolong sangat bagus. Dengan demikian, *delay* tertinggi ditunjukkan pada saat pengujian pada jarak 1 m dan terendah pada jarak 10 m. Hal tersebut bisa saja terjadi karena faktor sinyal yang tidak stabil ataupun adanya proses pada perangkat sehingga terjadi waktu tunda pada sistem.

1. Pengujian *throughput*

Pengujian *throughput* dilakukan untuk mengetahui kecepatan rata-rata transfer data. Untuk mengukur *throughput*, dilakukan dengan cara mengirimkan sejumlah besar data dari *smartphone* ke modul *bluetooth* dalam satu sesi komunikasi *bluetooth*. Kemudian, mencatat jumlah data yang berhasil dikirimkan dan waktu yang diperlukan untuk mengirimkan data tersebut. Dari informasi tersebut, *throughput* (jumlah data per satuan waktu) dihitung. Hasil pengujian *throughput* komunikasi *bluetooth* ditunjukkan pada Tabel 2.2 di bawah.

Tabel 2. Hasil Pengujian *throughput* komunikasi *bluetooth*

|  |  |  |  |
| --- | --- | --- | --- |
| **Jarak (m)** | **Total Paket (bit)** | **Waktu Pengukuran (s)** | ***Throughput* (Kbps)** |
| 1 m | 592 | 0,271 | 2,184 |
| 5 m | 592 | 0,0645 | 9,178 |
| 10 m | 592 | 0,072 | 8,222 |

Pada Tabel 2.2 hasil pengujian *throughput* komunikasi *bluetooth* dapat dilihat bahwa hasil *throughput* paling tinggi pada jarak 5 m dan paling rendah pada jarak 1 m. Pada jarak 1 m hasil dari pengujian *throughput* komunikasi *bluetooth* sebesar 2,184 Kbps. Pada jarak 5 m hasil dari pengujian *throughput* komunikasi *bluetooth* sebesar 9,178 Kbps. Pada jarak 10 m hasil dari pengujian *throughput* komunikasi *bluetooth* sebesar 8,222 Kbps.

1. Pengujian *packet loss*

Pengujian *packet loss* dilakukan untuk mengetahui apakah terjadi *loss* pada sistem yang dibuat ataukah tidak. Untuk mengukur *packet loss*, dilakukan dengan cara mengirimkan sejumlah paket data dari *smartphone* ke modul *bluetooth*. Kemudian, mencatat jumlah paket yang berhasil dikirimkan dan jumlah paket yang hilang atau tidak diterima oleh modul *bluetooth*. Dari informasi tersebut, *packet loss rate* (persentase paket yang hilang) dihitung. Hasil pengujian *packet loss* komunikasi *bluetooth* dapat dilihat pada Tabel 2.3 di bawah.

Tabel 2. Hasil pengujian *packet* *loss* komunikasi *bluetooth*

|  |  |  |  |
| --- | --- | --- | --- |
| **Jarak** | **Data Terkirim** | **Data Diterima** | **Persentase *Loss*** |
| 1 m | 3 | 3 | 0% |
| 5 m | 3 | 3 | 0% |
| 10 m | 3 | 3 | 0% |

Dari Tabel 2.3 hasil pengujian *packet loss* komunikasi *bluetooth* terlihat bahwa pada jarak pengujian sampai 10 m tidak terjadi *packet loss,* semua data pada pengujian yang dikirim sebanyak 3 kali pada setiap pengujian dapat diterima dengan baik yaitu 3 data, sehingga perangkat dapat berkomunikasi dengan baik sampai pada jarak 10 meter.

Berdasarkan pengujian yang telah dilakukan maka selanjutnya dilakukan analisa QoS dengan cara menghitung nilai *delay, throughput,* dan *packet loss* terhadap perubahan jarak agar didapatkan nilai QoS seperti yang ditunjukkan pada Tabel 2.4 di bawah.

Tabel 2. *Indeks* parameter QoS komunikasi *bluetooth*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Jarak** | ***Delay*** | ***Throughput*** | ***Packet Loss*** | **Nilai QoS** | **Indeks** |
| 1 m | 3 | 1 | 4 | 2,67 | Kurang Memuaskan |
| 5 m | 4 | 1 | 4 | 3 | Memuaskan |
| 10 m | 4 | 1 | 4 | 3 | Memuaskan |

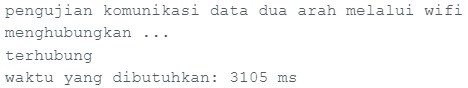
Gambar 2. Grafik hasil perhitungan nilai QoS komunikasi *bluetooth*

Berdasarkan Tabel 2.4 dan Gambar 2.4 dapat dilihat bahwa hasil dari pengujian parameter QoS komunikasi *bluetooth* seperti *delay, throughput,* dan *packet loss* dihitung dan dianalisis sesuai dengan nilai pada indeks parameter QoS. Hasil yang didapat bahwa pada jarak 1 m memiliki *indeks* kurang memuaskan dengan nilai QoS sebesar 2,67, pada jarak 5 m memiliki indeks memuaskan dengan nilai QoS sebesar 3, dan pada jarak 10 m memiliki indeks memuaskan dengan nilai QoS sebesar 3.

1. Pengujian Fungsionalitas Komunikasi WiFi

Pada pengujian ini, digunakan aplikasi *serial monitor* dalam Arduino IDE untuk menguji fungsionalitas komunikasi data dua arah melalui WiFi. Tujuan utamanya adalah untuk mengirimkan data dari ESP32 ke *server* dan menerima balasan dari *server* kembali ke ESP32 melalui koneksi WiFi. Proses pengujian fungsionalitas WiFi dilakukan dalam 2 bagian yaitu :

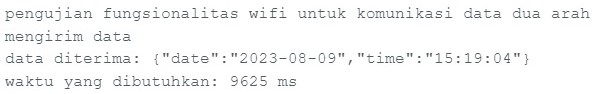
1. Pengujian koneksi ke *router*



Gambar 2. Status perangkat WiFi terhubung ke jaringan WiFi

Berdasarkan pada Gambar 2.5, proses pengujian koneksi dilakukan dengan menghubungkan perangkatWiFi (ESP32)ke jaringan WiFi yang tersedia. Kemudian, status koneksi WiFi“Terhubung” ditampilkan pada aplikasi *serial monitor* dalam Arduino IDE*.* Adapun waktu yang dibutuhkan ESP32 untuk terhubung ke jaringan WiFi adalah 3105 ms.

1. Pengujian pengiriman dan penerimaan data



Gambar 2. Pengujian pengiriman dan penerimaan data

Berdasarkan pada Gambar 2.6, hasil pengujian pengiriman dan penerimaan data menggunakan aplikasi *serial monitor* dalam Arduino IDE memperlihatkan bahwa ESP32 mengirim data ke *server.* Kemudian, data diterima kembali oleh ESP32 pada **{“date”:”2023-08-09”,”time”:”15:19:04”}.** Adapunwaktu yang dibutuhkan dalam proses pengiriman dan penerimaan data tersebut adalah 9625 ms.

Hasil pengujian tersebut berhasil membuktikan bahwa komunikasi data dua arah melalui WiFi yang diuji menggunakan aplikasi *serial monitor* dalam Arduino IDE berjalan dengan baik. *Server* mampu menerima data dari ESP32 melalui koneksi WiFi dan mengirimkan balasan kembali ke ESP32 melalui jalur yang sama. Pesan berhasil dikirim dan diterima, serta balasan berhasil diterima dan ditampilkan di *Serial Monitor* komputer.

1. Pengujian Performa Komunikasi WiFi

Pada pengujian performa kinerja komunikasi data dua arah melalui WiFi ini dengan memperhatikan parameter QoS yang didefinisikan oleh TIPHON. Pengujian ini bertujuan untuk mengukur dan menganalisis karakteristik kinerja komunikasi WiFi, termasuk *delay*, *throughput*, dan *packet loss*. Pengujian yang dilakukan yaitu :

1. Pengujian *delay*

Pengujian *delay* dilakukan dengan cara mengirimkan paket data dari ESP32 ke *server* melalui koneksi WiFi. Kemudian, mencatat waktu yang dibutuhkan untuk paket data mencapai *server* setelah dikirim dari ESP32. Pengukuran ini dilakukan beberapa kali dan rerata *delay* dihitung. Hasil pengujian *delay* komunikasi WiFi ditunjukkan pada Tabel 4.5 di bawah.

Tabel 2. Hasil Pengujian *delay* komunikasi WiFi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Kecepatan Internet (Mbps)** | **Waktu A** | **Waktu B** | **RTT (ms)** | ***Delay***  **(ms)** | **Ket.** |
| U = 11.90  D = 16.63 | 17:39:22.808 | 17:39:24.885 | 2077 | 1038,5 | Buruk |
| 17:41:36.089 | 17:41:37.756 | 1667 | 833,5 |
| 17:43:02.389 | 17:43:04.408 | 2019 | 1009,5 |
| Rata-rata *delay* (ms) | | | | **960,5** |
| U = 41.97  D = 62.93 | 07:17:04.068 | 07:17:05.303 | 1235 | 617,5 | Buruk |
| 07:24:38.336 | 07:24:39.568 | 1232 | 616 |
| 07:28:05.561 | 07:28:06.815 | 1254 | 627 |
| Rata-rata *delay* (ms) | | | | **620,167** |
| U = 2.82  D = 8.30 | 09:35:04.975 | 09:3507.876 | 2901 | 1450,5 | Buruk |
| 09:36:32.482 | 09:36:35.992 | 3510 | 1755 |
| 09:37:57.434 | 09:37:59.472 | 2038 | 1019 |
| Rata-rata *delay* (ms) | | | | **1408,167** |
| U = 17.52  D = 15.14 | 10:30:19.570 | 10:30:21.237 | 1667 | 833,5 | Buruk |
| 10:33:17.431 | 10:33:18.943 | 1512 | 756 |
| 10:34:28.657 | 10:34:30.289 | 1632 | 816 |
| Rata-rata *delay* (ms) | | | | **801,83** |

**Tabel 2.5** (lanjutan)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Kecepatan Internet (Mbps)** | **Waktu A** | **Waktu B** | **RTT (ms)** | ***Delay***  **(ms)** | **Ket.** |
| U = 13.86  D = 9.99 | 10:56:44.493 | 10:56:46.075 | 1582 | 791 | Buruk |
| 10:58:27.993 | 10:58:29.344 | 1351 | 675,5 |
| 10:59:54.064 | 10:59:56.081 | 2017 | 1008,5 |
| Rata-rata *delay* (ms) | | | | **825** |

Ket.: Waktu A = Waktu ketika data dikirim

Waktu B = Waktu ketika data diterima kembali oleh pengirim

RTT = *Round-Trip Time*

U = *Upload*

D = *Download*

Dari Tabel 2.5 hasil pengujian *delay* komunikasi WiFiantara ESP32 ke *server* terjadi kenaikan atau penurunan rata-rata *delay* seiring dengan meningkat atau menurunnya kecepatan internet. Pada kecepatan internet sebesar (U = 11.90 Mbps dan D = 16.63 Mbps), rata-rata hasil dari pengujian *delay* komunikasi WiFi untuk tiga kali percobaan adalah sebesar 960,5 ms sehingga tergolong buruk. Pada kecepatan internet sebesar (U = 41.97 Mbps dan D = 62.93 Mbps), rata-rata hasil dari pengujian *delay* komunikasi WiFi untuk tiga kali percobaan adalah sebesar 620,167 ms sehingga tergolong buruk. Pada kecepatan internet sebesar (U = 2.82 Mbps dan D = 8.30 Mbps), rata-rata hasil dari pengujian *delay* komunikasi WiFi untuk tiga kali percobaan adalah sebesar 1408,167 ms sehingga tergolong buruk. Pada kecepatan internet sebesar (U = 17.52 Mbps dan D = 15.14 Mbps), rata-rata hasil dari pengujian *delay* komunikasi WiFi untuk tiga kali percobaan adalah sebesar 801,83 ms sehingga tergolong buruk. Pada kecepatan internet sebesar (U = 13.86 Mbps dan D = 9.99 Mbps), rata-rata hasil dari pengujian *delay* komunikasi WiFi untuk tiga kali percobaan adalah sebesar 825 ms sehingga tergolong buruk. Dengan demikian dapat disimpulkan bahwa dengan adanya kecepatan internet yang semakin meningkat, maka *delay* semakin menurun.

1. Pengujian *throughput*

Pengujian *throughput* dilakukan untuk mengetahui kecepatan rata-rata transfer data. Untuk mengukur *throughput*, dilakukan dengan cara mengirimkan sejumlah besar data dari ESP32 ke *server* dalam satu sesi komunikasi WiFi. Kemudian, mencatat jumlah data yang berhasil dikirimkan dan waktu yang diperlukan untuk mengirimkan data tersebut. Dari informasi tersebut, *throughput* (jumlah data per satuan waktu) dihitung. Hasil pengujian *throughput* komunikasi WiFi ditunjukkan pada Tabel 2.6 di bawah.

Tabel 2. Hasil pengujian *throughput* komunikasi WiFi

|  |  |  |  |
| --- | --- | --- | --- |
| **Kecepatan Internet (Mbps)** | **Total Paket (*Bytes*)** | **Waktu Pengukuran (s)** | ***Throughput* (Mbps)** |
| U = 11.90  D = 16.63 | 882523 | 1,0385 | 6,79836 |
| U = 41.97  D = 62.93 | 882523 | 0,6175 | 11,43349 |
| U = 2.82  D = 8.30 | 882523 | 1,4505 | 4,86741 |
| U = 17.52  D = 15.14 | 882523 | 0,8335 | 8,47052 |
| U = 13.86  D = 9.99 | 882523 | 0,791 | 8,92564 |

Pada Tabel 2.6 hasil pengujian *throughput* komunikasi WiFidapat dilihat bahwa terjadi kenaikan atau penurunan *throughput* seiring dengan meningkat atau menurunnya kecepatan internet. Pada kecepatan internet sebesar (U = 11.90 Mbps dan D = 16.63 Mbps), hasil dari pengujian *throughput* komunikasi WiFi adalah sebesar 6,79836 Mbps. Pada kecepatan internet sebesar (U = 41.97 Mbps dan D = 62.93 Mbps), hasil dari pengujian *throughput* komunikasi WiFi adalah sebesar 11,43349 Mbps. Pada kecepatan internet sebesar (U = 2.82 Mbps dan D = 8.30 Mbps), hasil dari pengujian *throughput* komunikasi WiFi adalah sebesar 4,86741 Mbps. Pada kecepatan internet sebesar (U = 17.52 Mbps dan D = 15.14 Mbps), hasil dari pengujian *throughput* komunikasi WiFi adalah sebesar 8,47052 Mbps. Pada kecepatan internet sebesar (U = 13.86 Mbps dan D = 9.99 Mbps), hasil dari pengujian *throughput* komunikasi WiFi adalah sebesar 8,92564 Mbps. Dengan demikian dapat disimpulkan bahwa dengan adanya kecepatan internet yang semakin meningkat, maka *throughput* juga semakin meningkat.

1. Pengujian *packet loss*

Pengujian *packet loss* dilakukan untuk mengetahui apakah terjadi *loss* pada sistem yang dibuat ataukah tidak. Untuk mengukur *packet loss*, dilakukan dengan cara mengirimkan sejumlah paket data dari ESP32 ke *server*. Kemudian, mencatat jumlah paket yang berhasil dikirimkan dan jumlah paket yang hilang atau tidak diterima oleh *server*. Dari informasi tersebut, *packet loss rate* (persentase paket yang hilang) dihitung. Hasil pengujian *packet loss* komunikasi WiFi dapat dilihat pada Tabel 2.7 di bawah.

Tabel 2. Hasil pengujian *packet* *loss* komunikasi WiFi

|  |  |  |  |
| --- | --- | --- | --- |
| **Kecepatan Internet (Mbps)** | **Data Terkirim** | **Data Diterima** | **Persentase *Loss*** |
| U = 11.90  D = 16.63 | 3 | 3 | 0% |
| U = 41.97  D = 62.93 | 3 | 3 | 0% |
| U = 2.82  D = 8.30 | 3 | 3 | 0% |
| U = 17.52  D = 15.14 | 3 | 3 | 0% |

**Tabel 2.7** (lanjutan)

|  |  |  |  |
| --- | --- | --- | --- |
| **Kecepatan Internet (Mbps)** | **Data Terkirim** | **Data Diterima** | **Persentase *Loss*** |
| U = 13.86  D = 9.99 | 3 | 3 | 0% |

Dari Tabel 2.7 hasil pengujian *packet loss* komunikasi WiFi terlihat bahwa semua data pada pengujian yang dikirim sebanyak 3 kali pada setiap pengujian dapat diterima dengan baik yaitu 3 data untuk beberapa variasi kecepatan internet. Dengan demikian dapat disimpulkan bahwa perangkat dapat berkomunikasi dengan baik.

Berdasarkan pengujian yang telah dilakukan maka selanjutnya dilakukan analisa QoS dengan menghitung nilai *delay, throughput,* dan *packet loss* terhadap perubahan kecepatan internet agar didapatkan nilai QoS seperti yang ditunjukkan pada Tabel 2.8 di bawah.

Tabel 2. *Indeks* parameter QoS komunikasi WiFi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Kecepatan Internet (Mbps)** | ***Delay*** | ***Throughput*** | ***Packet Loss*** | **Nilai QoS** | ***Indeks*** |
| U = 11.90  D = 16.63 | 1 | 4 | 4 | 3 | Memuaskan |
| U = 41.97  D = 62.93 | 1 | 4 | 4 | 3 | Memuaskan |
| U = 2.82  D = 8.30 | 1 | 4 | 4 | 3 | Memuaskan |
| U = 17.52  D = 15.14 | 1 | 4 | 4 | 3 | Memuaskan |
| U = 13.86  D = 9.99 | 1 | 4 | 4 | 3 | Memuaskan |

Gambar 2. Grafik hasil pengujian nilai QoS komunikasi WiFi

Berdasarkan Tabel 2.8 dan Gambar 2.7 dapat dilihat bahwa hasil dari pengujian parameter QoS komunikasi WiFi seperti *delay, throughput,* dan *packet loss* dihitung dan dianalisis sesuai dengan nilai pada indeks parameter QoS. Dapat disimpulkan bahwa semua percobaan dengan kecepatan internet yang berbeda memiliki hasil nilai QoS sebesar 3 yang berarti memuaskan.

1. Pengujin Fungsionalitas Penguncian

Pengujian fungsional perangkat penguncian ini dengan menggunakan metode *blackbox* untuk membantu mengidentifikasi bagaimana perangkat penguncian berfungsi dari perspektif pengguna atau pemakai, tanpa perlu mengetahui detail teknis internal perangkat. Hasil pengujian ini memberikan pemahaman tentang sejauh mana perangkat mendukung kebutuhan dan fungsionalitas yang telah ditetapkan.

Pengujian melibatkan penggunaan antarmuka yang tersedia (seperti aplikasi *mobile* atau antarmuka *website*) untuk mengendalikan perangkat penguncian. Perintah-perintah yang berkaitan dengan perangkat penguncian diuji untuk memastikan perangkat berfungsi sesuai harapan. Tabel 2.9 berikut ini adalah hasil pengujian yang dilakukan.

Tabel 2. Hasil pengujian fungsional perangkat penguncian

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Komponen** | **Skenario** | **Hasil yang Diharapkan** | **Keterangan** |
| 1. | Solenoid | Memberikan perintah kunci pintu dari *website* | **Mengunci Pintu**: Ketika perangkat mendapatkan perintah untuk mengunci pintu, solenoid akan di-*set* ke LOW. Ini akan menghasilkan gaya magnetik yang akan mengunci pintu secara fisik. | Berhasil |
| 2. | Solenoid | Memberikan perintah membuka pintu melalui penggunaan aplikasi *mobile, website, push button,* dan *touch sensor.* | **Membuka Pintu**: Ketika perangkat mendapatkan perintah untuk membuka pintu, solenoid akan diaktifkan (HIGH). Hal ini akan menghilangkan gaya magnetik dan memungkinkan pintu untuk dibuka. | Berhasil |
| 3. | Solenoid | Mencoba membuka kunci pintu secara paksa | **Pengaman Pintu**: Solenoid juga digunakan untuk memberikan pengamanan tambahan. Jika pintu terdeteksi terbuka tanpa adanya autentikasi yang sah (seperti perintah dari *server* atau tindakan tombol), solenoid akan diaktifkan untuk mengunci pintu kembali. | Berhasil |

**Tabel 2.9** (lanjutan)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Komponen** | **Skenario** | **Hasil yang Diharapkan** | **Keterangan** |
| 4. | Solenoid | Memberikan perintah penjadwalan buka kunci pintu melalui *website* | **Pengendalian Jadwal**: Solenoid juga digunakan untuk mengendalikan pintu berdasarkan jadwal. Jika jadwal kunci pintu berakhir, solenoid akan diaktifkan untuk mengunci pintu sesuai dengan pengaturan jadwal. | Berhasil |
| 5. | *Push Button* | Menekan tombol pada perangkat penguncian | **Membuka Pintu Manual**: Ketika tombol ditekan, program akan mengenali aksi tersebut dan membuka pintu secara manual (*unlock*) jika kondisi memungkinkan. Ini memungkinkan pengguna untuk membuka pintu secara langsung tanpa harus menggunakan autentikasi tambahan. | Berhasil |
| 6. | Saklar Magnetik | Melakukan aksi buka tutup pintu | **Deteksi Status Pintu**: Sensor magnetik akan mendeteksi apakah pintu dalam keadaan terbuka atau tertutup berdasarkan perubahan medan magnet. Jika pintu terbuka, sensor akan memberikan sinyal LOW (0), dan jika pintu tertutup, sensor akan memberikan sinyal HIGH (1). | Berhasil |

**Tabel 2.9** (lanjutan)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Komponen** | **Skenario** | **Hasil yang Diharapkan** | **Keterangan** |
| 7. | Saklar Magnetik | Melakukan aksi buka tutup pintu dan melihat status kondisi pintu melalui aplikasi *mobile* atau *website.* | ***Update* Status Pintu**: Program akan membaca nilai dari pin yang terhubung dengan sensor magnetik secara berkala. Jika sensor memberikan sinyal LOW, program akan menganggap pintu dalam keadaan terbuka, dan jika sensor memberikan sinyal HIGH, program akan menganggap pintu dalam keadaan tertutup. | Berhasil |
| 8. | *Buzzer* | Menekan tombol dengan cepat, membuka pintu secara paksa, membuka pintu terlalu lama dalam mode penguncian. | **Tanda Bunyi Notifikasi**: *Buzzer* akan memberikan tanda bunyi sebagai respons terhadap berbagai peristiwa. | Berhasil |
| 9. | *Buzzer* | Melakukan buka tutup kunci pintu melalui aplikasi *mobile* atau *website.* | **Indikator Status**: *Buzzer* dapat digunakan sebagai indikator status untuk memberi tahu pengguna tentang kondisi sistem. | Berhasil |
| 10. | *Buzzer* | Membuka pintu dalam waktu lama. | **Tanda Bunyi Pengingat**: *Buzzer* dapat digunakan sebagai pengingat. | Berhasil |

**Tabel 2.9** (lanjutan)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Komponen** | **Skenario** | **Hasil yang Diharapkan** | **Keterangan** |
| 11. | LED | Menghubungkan perangkat penguncian ke jaringan WiFi. | **Indikasi Status WiFi**: LED dapat berkedip dengan pola tertentu untuk menunjukkan status koneksi WiFi. Kedipan LED menunjukkan apakah perangkat terhubung ke jaringan WiFi atau tidak. | Berhasil |
| 12. | LED | Memberikan perintah melalui aplikasi *mobile* atau *website* agar perangkat penguncian berkomunikasi dengan *server.* | **Indikasi Status Data**: LED dapat berkedip dengan pola tertentu untuk menunjukkan aktivitas data atau komunikasi dengan *server*. | Berhasil |
| 13. | LED | Memberikan perintah buka tutup kunci pintu melalui aplikasi *mobile* atau *website.* | **Indikasi Status *Lock***: LED dapat berkedip dengan pola tertentu untuk menunjukkan status penguncian pintu atau apakah solenoid sedang aktif atau tidak. | Berhasil |
| 14. | Sensor Sentuh | Melakukan aksi memegang gagang pintu saat adanya penjadwalan pintu. | **Deteksi Sentuhan**: Komponen *touch sensor* akan mendeteksi apakah ada sentuhan atau sentuhan jari pada area yang diindikasikan. Jika ada sentuhan, nilai sensor akan bervariasi sesuai dengan intensitas sentuhan. | Berhasil |

**Tabel 2.9** (lanjutan)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Komponen** | **Skenario** | **Hasil yang Diharapkan** | **Keterangan** |
| 15. | Sensor Sentuh | Melakukan aksi memegang gagang pintu saat adanya penjadwalan pintu. | **Pengendalian Tindakan**: Nilai sensor sentuh akan dipantau dan dihitung dalam kode program. Jika nilai sensor mencapai ambang tertentu yang mengindikasikan sentuhan, program dapat mengambil tindakan tertentu, seperti mengaktifkan solenoid untuk membuka pintu. | Berhasil |

1. Pengujian Fungsionalitas API

Pada pengujian fungsional API berfokus pada fungsi dari fitur-fitur yang ada pada sistem API yang telah dibangun. Pengujian fungsional dilakukan menggunakan metode *blackbox* dengan bantuan *postman* sebagai alat pengujian. Pengujian *blackbox* merupakan sebuah metode pengujian perangkat lunak yang dilakukan tanpa memperhatikan struktuk kode program didalamnya. Tabel 2.10 menjelaskan beberapa fitur hasil implementasi dari kebutuhan fungsional pada sistem keamanan kunci pintu gedung.

Tabel 2. Hasil pengujian fungsional

|  |  |  |
| --- | --- | --- |
| **No** | **Kebutuhan** | **Hasil** |
| 1 | Autentikasi *login* dan *logout* | Tersedia |
| 2 | Ganti *password* | Tersedia |
| 3 | Reset *password* | Tersedia |
| 4 | Verifikasi *email* | Tersedia |
| 5 | Ganti profil | Tersedia |
| 6 | Lihat daftar akses | Tersedia |
| 7 | Verifikasi akses | Tersedia |
| 8 | Lihat riwayat aktifitas | Tersedia |
| 9 | Lihat daftar pintu | Tersedia |

**Tabel 2.10** (lanjutan)

|  |  |  |
| --- | --- | --- |
| **No** | **Kebutuhan** | **Hasil** |
| 10 | Membuka pintu jarak jauh | Tersedia |
| 11 | Register pintu baru | Tersedia |
| 12 | Koneksi *websocket* untuk pintu | Tersedia |
| 13 | Mendapatkan *signature* | Tersedia |
| 14 | *Update* status pintu | Tersedia |
| 15 | Peringatan pintu | Tersedia |

Terlihat pada Tabel 2.10 diatas, beberapa fitur yang telah diimplementasikan pada sistem API. Pengujian fungsional dari masing-masing API adalah sebagai berikut :

1. *Login*

Pengujian pada fungsi *login* dilaksanakan untuk memeriksa kinerja dari fungsi *login*, *login* dikatakan berhasil jika client mendapatan respon *success* dari server disertai dengan dikirimkannya data *client* dan token akses. Hasil dari pengujian fungsi login dapat dilihat pada Tabel 2.11.

Tabel 2. Hasil pengujian fungsi *login*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama** | **Bentuk Pengujian** | **Respon** | **Hasil** |
| 1 | *Login* dengan data benar | Melakukan *login* menggunakan *email* dan *password* yang sesuai | *success* | Berhasil |
| 2 | *Login* dengan data salah | Melakukan *login* dengan menggunakan *email* atau *password* yang salah | *failed* | Gagal |
| 3 | *Login* dengan data kurang | Melakukan *login* dengan menggunakan *email* saja atau *password* saja | *missing*\_*parameter* | Gagal |
| 4 | *Login* data tidak terdaftar | Melakukan *login* dengan menggunakan *email* yang belum terdaftar | *missing*\_*parameter* | Gagal |
| 5 | *Login* dengan format tidak sesuai | Melakukan *login* dengan menggunakan *username* bukan *email* | *missing*\_*parameter* | Gagal |

**Tabel 2.11** (lanjutan)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama** | **Bentuk Pengujian** | **Respon** | **Hasil** |
| 6 | *Login* dengan *email* belum terverifikasi | Melakukan *login* dengan menggunakan *email* yang berlum terverifikasi | *email*\_*unverified* | Gagal |

Dapat dilihat pada Tabel 2.11 proses *login* akan berhasil jika menggunakan email dan *password* yang sesuai, proses *login* juga memastikan semua parameter yang digunakan pada autentikasi tersedia dan juga sesuai. Pada proses pengujian menggunakan email yang belum terverifikasi *login* akan tertahan dengan status *email*\_*unverified* dan menunggu *client* untuk melakukan verifikasi *email*.

1. *Logout*

Pengujian pada fungsi *logout* dilaksanakan untuk memeriksa kinerja dari fungsi tersebut. *Logout* dikatakan berhasil jika *client* menerima respon *succsess* dari *server*. Hasil dari pengujian fungsi *logout* dapat dilihat pada Tabel 2.12.

Tabel 2. Hasil Pengujian fungsi logout

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama** | **Bentuk Pengujian** | **Respon** | **Hasil** |
| 1 | *Logout* dengan token | Melakukan *logout* menggunakan token yang sesuai | *success* | Berhasil |
| 2 | *Logout*  tanpa token | Melakukan *logout* tanpa menggunakan token | *Unauthenticated* | Gagal |
| 3 | *Login* dengan token salah | Melakukan *logout* dengan menggunakan token yang sudah terhapus | *Unauthenticated* | Gagal |

Dapat dilihat pada Tabel 2.12 proses *logout* hanya berhasil jika menggunakan token yang sesuai, jika *client* melakukan *logout* tanpa menggunakan token atau menggunakan token yang sudah terhapus maka *logout* akan gagal dengan respon *unathenticated* atau tidak terautentikasi.

1. Verifikasi *email*

Pengujian pada fungsi verifikasi *email* dilaksanakan untuk memeriksa kinerja dari fungsi tersebut. Verifikasi email dikatakan berhasil jika *client* mendapatkan respon *success* dari *server*. Hasil dari pengujian fungsi verifikasi *email* dapat dilihat pada Tabel 2.13.

Tabel 2. Hasil pengujian fungsi verifikasi *email*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama** | **Bentuk Pengujian** | **Respon** | **Hasil** |
| 1 | Verifikasi *email* tanpa token | Melakukan verifikasi *email* tanpa menggunakan token | *Unauthenticated* | Gagal |
| 2 | Verifikasi *email* tidak sesuai | Melakukan verifikasi *email* menggunakan kode OTP yang salah | *otp\_not\_match* | Gagal |
| 3 | Verifikasi *email* kadaluarsa | Melakukan verifikasi *email* menggunakan kode OTP yang sudah kadaluarsa | *otp\_expired* | Gagal |
| 4 | Verifikasi *email* sesuai | Melakukan verifikasi *email* menggunakan kode OTP yang sesuai | *success* | Berhasil |
| 5 | Verifikasi *email* token salah | Melakukan verifikasi *email* menggunakan token yang sudah terhapus | *Unauthenticated* | Gagal |

Dapat dilihat pada Tabel 2.13 proses verifikasi *email* hanya berhasil jika *client* mengirimkan kode OTP yang sesuai disertai dengan token yang sesuai. Jika proses verifikasi *email* menggunakan kode OTP yang salah atau sudah kadaluarsa maka proses verifikasi *email* akan gagal.

1. Ganti *password*

Pengujian pada fungsi ganti *password* dilaksanakan untuk memeriksa kinerja dari fungsi tersebut. Proses ganti *password* dikatakan berhasil jika *client* mendapatkan respon *success* dari *server*. hasil dari pengujian ganti *password* dapat dilihat pada Tabel 2.14.

Tabel 2. Hasil pengujian fungsi ganti *password*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama** | **Bentuk Pengujian** | **Respon** | **Hasil** |
| 1 | Ganti *password* tanpa token | Melakukan penggantian *password* tanpa menggunakan token | *Unauthenticated* | Gagal |
| 2 | Ganti *password* dengan token salah | Melakukan penggantian *password* menggunakan token yang sudah terhapus | *Unauthenticated* | Gagal |
| 3 | Ganti *password* sesuai | Melakukan penggantian *password* sesuai | *success* | Berhasil |
| 4 | Ganti *password* minimal | Melakukan penggantian *password* dengan 3 karakter | *missing\_parameter* | Gagal |
| 5 | Ganti *password* maksimal | Melakukan penggantian *password* dengan 100 karakter | *missing\_parameter* | Gagal |
| 6 | Ganti *password* konfirmasi salah | Melakukan penggantian *password* dengan konfirmasi *password* berbeda | *missing\_parameter* | Gagal |

Pada Tabel 2.14 terlihat proses ganti *password* berhasil jika *client* mengirimkan *password* dan konfirmasi *password* sesuai, *client* juga harus mengirimkan token yang sesuai juga. Jika salah satu parameter tidak terpenuhi maka proses ganti *password* akan gagal.

1. *Reset* *password*

Pengujian pada fungsi *reset* *password* dilaksanakan untuk memeriksa kinerja dari fungsi tersebut. Proses *reset* *password* dikatakan berhasil jika *client* mendapatkan respon *success* dari *server*. hasil dari pengujian *reset* *password* dapat dilihat pada Tabel 2.15.

Tabel 2. Hasil pengujian fungsi *reset* *password*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama** | **Bentuk Pengujian** | **Respon** | **Hasil** |
| 1 | *Reset* *password* *email* sesuai | Melakukan *reset* *password* menggunakan *email* dengan format sesuai dan terdaftar | *success* | Berhasil |
| 2 | *Reset* *password* *email* tidak sesuai | Melakukan *reset* *password* menggunakan *email* yang belum terdaftar | *failed* | Gagal |
| 3 | *Reset* *password* format *email* salah | Melakukan *reset* *password* menggunakan *email* dengan format salah | *missing\_parameter* | Gagal |

Pada Tabel 2.15 terlihat proses *reset* *password* berhasil jika *client* mengirimkan alamat *email* yang sesuai, Jika *email* tidak sesuai atau belum terdaftar maka *reset* *password* akan gagal.

1. *Update* profil

Pengujian pada fungsi *update* profil dilaksanakan untuk memeriksa kinerja dari fungsi tersebut. Proses *update* profil dikatakan berhasil jika *client* mendapatkan respon *success* dari *server*. hasil dari pengujian *update* profil dapat dilihat pada Tabel 2.16.

Tabel 2. Hasil pengujian fungsi *update* profil

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama** | **Bentuk Pengujian** | **Respon** | **Hasil** |
| 1 | Update nama minimal | Melakukan update profil menggunakan nama 3 karakter | *missing\_parameter* | Gagal |
| 2 | Update nama sesuai | Melakukan update profil menggunakan nama sesuai | *success* | Berhasil |
| 3 | Update email sesuai | Melakukan update profil menggunakan email yang sesuai | success | Berhasil |
| 4 | Update email format salah | Melakukan update profil menggunakan email dengan format tidak sesuai | *missing*\_*parameter* | Gagal |
| 5 | Update jenis kelamin sesuai | Melakukan update jenis kelamin dengan format sesuai | *success* | Berhasil |
| 6 | Update nomor hp sesuai | Melakukan update nomor hp dengan format sesuai | *success* | Berhasil |
| 7 | Update nomor hp tidak sesuai | Melakukan update nomor hp dengan format salah (digit angka kurang/lebih) | *missing*\_*parameter* | Gagal |

Pada Tabel 2.16 terlihat proses *update* profil akan berhasil jika semua data yang dikirimkan sesuai dengan format sehingga mendapatkan respon *success*, jika ada salah satu data yang tidak sesuai format misalnya format *email* tidak sesuai atau nomor hp kurang maka *update* profil akan gagal.

1. *Update* avatar

Pengujian pada fungsi *update* avatar dilaksanakan untuk memeriksa kinerja dari fungsi tersebut. Proses *update* avatar dikatakan berhasil jika *client* mendapatkan respon *success* dari *server*. hasil dari pengujian *reset* *password* dapat dilihat pada Tabel 2.17.

Tabel 2. Hasil pengujian fungsi *update* avatar

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama** | **Bentuk Pengujian** | **Respon** | **Hasil** |
| 1 | *Update* avatar format sesuai | Melakukan *update* gambar avatar sesuai dengan format | *success* | Berhasil |
| 2 | *Update* avatar file besar | Melakukan *update* gambar menggunakan gambar dengan ukuran lebih dari 1 MB | *request entity too large* | Gagal |

**Tabel 2.17** (lanjutan)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama** | **Bentuk Pengujian** | **Respon** | **Hasil** |
| 1 | *Update* avatar format sesuai | Melakukan *update* gambar avatar sesuai dengan format | *success* | Berhasil |
| 2 | *Update* avatar file besar | Melakukan *update* gambar menggunakan gambar dengan ukuran lebih dari 1 MB | *request entity too large* | Gagal |
| 3 | *Update* avatar bukan gambar | Melakukan *update* avatar menggunakan file selain gambar | *missing*\_*parameter* | Gagal |

Pada Tabel 2.17 terlihat proses *update* avatar hanya berhasil jika file yang dikirim adalah gambar dengan ukuran kurang dari 1 MB, jika data yang dikirim bukan merupakan gambar atau ukuran gambar lebih dari 1 MB maka proses *update* avatar akan gagal.

1. Lihat akses

Pengujian pada fungsi lihat akses dilaksanakan untuk memeriksa kinerja dari fungsi tersebut. Proses lihat akses dikatakan berhasil jika *client* mendapatkan respon *success* dari *server*. hasil dari pengujian fungsi lihat akses dapat dilihat pada Tabel 2.18.

Tabel 2. Hasil pengujian fungsi lihat akses

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama** | **Bentuk Pengujian** | **Respon** | **Hasil** |
| 1 | Lihat akses dengan token | Melakukan *request* lihat akses menggunakan token yang sesuai | *success* | Berhasil |
| 2 | Lihat akses tanpa token | Melakukan *request* lihat akses tanpa menggunakan token | *Unauthenticated* | Gagal |

Pada Tabel 2.18 terlihat bahwa proses lihat akses akan berhasil jika permaintaan dilakukan dengan menambahkan token, jika permintaan tidak menggunakan token maka permintaan akan ditolak.

1. Lihat pintu

Pengujian pada fungsi lihat pintu dilaksanakan untuk memeriksa kinerja dari fungsi tersebut. Proses lihat pintu dikatakan berhasil jika *client* mendapatkan respon *success* dari *server*. hasil dari pengujian fungsi lihat pintu dapat dilihat pada Tabel 2.19.

Tabel 2. Hasil pengujian fungsi lihat pintu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama** | **Bentuk Pengujian** | **Respon** | **Hasil** |
| 1 | Lihat pintu tanpa token | Melakukan *request* lihat akses tanpa menggunakan token yang sesuai | *Unauthenticated* | Gagal |
| 2 | Lihat pintu dengan token pengguna | Melakukan *request* lihat akses menggunakan token yang dimiliki pengguna | *Unauthenticated* | Gagal |
| 3 | Lihat Pintu dengan token operator | Melakukan *request* lihat akses menggunakan token yang dimiliki operator | *success* | Berhasil |

Pada Tabel 2.19 terlihat bahwa proses lihat pintu berhasil jika permintaan disertai dengan token operator, jika permintaan tidak menggunakan token atau menggunakan token yang dimiliki oleh pengguna biasa maka permintaan akan ditolak.

1. Riwayat akses

Pengujian pada fungsi lihat riwayat akses dilaksanakan untuk memeriksa kinerja dari fungsi tersebut. Proses lihat riwayat akses dikatakan berhasil jika *client* mendapatkan respon *success* dari *server*. hasil dari pengujian fungsi lihat riwayat akses dapat dilihat pada Tabel 2.20.

Tabel 2. Hasil pengujian riwayat akses

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama** | **Bentuk Pengujian** | **Respon** | **Hasil** |
| 1 | Lihat riwayat dengan token | Melakukan *request* lihat riwayat menggunakan token yang sesuai | *success* | Berhasil |
| 2 | Lihat riwayat tanpa token | Melakukan *request* lihat riwayat tanpa menggunakan token | *Unauthenticated* | Gagal |

Pada Tabel 2.20 terlihat bahwa proses lihat riwayat akses berhasil jika permintaan disertai dengan token yang sesuai, jika permintaan tidak menggunakan token maka permintaan akan ditolak.

1. Verifikasi akses

Pengujian pada fungsi verifikasi akses dilaksanakan untuk memeriksa kinerja dari fungsi tersebut. Proses verifikasi akses dikatakan berhasil jika *client* mendapatkan respon *success* dari *server*. hasil dari pengujian fungsi lihat riwayat akses dapat dilihat pada Tabel 2.21.

Tabel 2. Hasil pengujian fungsi verifikasi akses

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama** | **Bentuk Pengujian** | **Respon** | **Hasil** |
| 1 | Verifikasi akses tanpa token | Melakukan *request* lihat verifikasi akses tanpa menggunakan token | *Unauthenticated* | Gagal |
| 2 | Verifikasi akses dengan token | Melakukan *request* verifikasi akses menggunakan token yang sesuai | *success* | Berhasil |
| 3 | Verifikasi akses dengan id pintu salah | Melakukan *request* verifikasi akses menggunakan identitas pintu yang salah | *no\_data* | Gagal |

Pada Tabel 2.21 terlihat bahwa proses verifikasi akses berhasil jika permintaan disertai dengan token dan identitas pintu sesuai, jika identitas pintu tidak sesuai maka proses verifikasi akses akan gagal karena pintu tidak ditemukan. Jika permintaan tidak disertai dengan token maka permintaan tersebut akan ditolak.

1. *Signature*

Pengujian pada fungsi *signature* dilaksanakan untuk memeriksa kinerja dari fungsi tersebut. Proses pembuatan *signature* dikatakan berhasil jika *client* mendapatkan respon *success* dari *server*. hasil dari pengujian fungsi buat signature dapat dilihat pada Tabel 2.22.

Tabel 2. Hasil Pengujian Fungsi Signature

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama** | **Bentuk Pengujian** | **Respon** | **Hasil** |
| 1 | *Signature* tanpa token | Melakukan *request* *signature* tanpa menggunakan token | *Unauthenticated* | Gagal |
| 2 | *Signature* dengan token | Melakukan *request* *signature* menggunakan token dan data lengkap | *success* | Berhasil |
| 3 | *Signature* data kurang | Melakukan *request* *signature* menggunakan data yang kurang | *missing\_parameter* | Gagal |

Pada Tabel 2.22 terlihat bahwa proses fungsi *signature* berhasil jika permintaan yang dikirimkan disertai dengan token dan data yang dikirimkan lengkap, jika permintaan yang dikirimkan tanpa menggunakan token atau ada data yang kurang maka permintaan akan gagal.

1. *Websocket*

Pengujian komunikasi *websocket* dilakukan untuk mengetahui kinerja dari koneksi *subscribtion* pada *channel* *websocket*. Hasil dari pengujian *websocket* dapat dilihat pada Tabel 2.23.

Tabel 2. Hasil iengujian *websocket*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama** | **Bentuk Pengujian** | **Respon** | **Hasil** |
| 1 | *Subscription* dengan *signature* | Melakukan *subscription* ke *channel websocket* menggunakan *signature* yang sesuai | *Subscription Succeed* | Berhasil |
| 2 | *Subscription* tanpa *signature* | Melakukan *subscription* ke *channel websocket* tanpa menggunakan *signature* | *Invalid Signature* | Gagal |
| 3 | *Subscribtion* dengan *signature* salah | Melakukan *subscription* ke *channel websocket* menggunakan *signature* yang tidak sesuai | *Invalid* *Signature* | Gagal |

Pada Tabel 2.23 terlihat bahwa proses *subscription* pada *channel* *websocket* berhasil jika menggunakan kode *signature* yang sesuai, jika tidak menggunakan kode *signature* atau menggunakan kode *signature* yang salah maka proses *subscription* akan gagal.

1. *Update* status pintu

Pengujian pada fungsi *update* status pintu dilaksanakan untuk memeriksa kinerja dari fungsi tersebut. Proses *update* status pintu dikatakan berhasil jika *client* mendapatkan respon *success* dari *server*. hasil dari pengujian fungsi *update* status pintu dapat dilihat pada Tabel 2.24.

Tabel 2. Hasil pengujian fungsi *update* status pintu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama** | **Bentuk Pengujian** | **Respon** | **Hasil** |
| 1 | *Update* status tanpa token | Melakukan *update* status pintu tanpa menggunakan token | *Unauthenticated* | Gagal |
| 2 | *Update* status dengan token | Melakukan *update* status pintu dengan menggunaka token dan data lengkap | *success* | Berhasil |

**Tabel 2.24** (lanjutan)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama** | **Bentuk Pengujian** | **Respon** | **Hasil** |
| 3 | *Update* status data tidak lengkap | Melakukan *upate* status pintu dengan menggunakan data yang tidak lengkap | *missing\_parameter* | Gagal |
| 4 | *Update* status id pintu salah | Melakukan *update* status pintu menggunakan data identitas pintu yang salah | *failed* | Gagal |

Pada Tabel 2.24 terlihat bahwa proses *update* status pintu berhasil jika *request* dikirimkan dengan token dan data yang lengkap, jika ada data yang kurang lengkap atau tidak disertai dengan token akan *request* tersebut akan gagal.

1. Peringatan pintu

Pengujian pada fungsi peringatan pintu dilaksanakan untuk memeriksa kinerja dari fungsi tersebut. Proses peringatan pintu dikatakan berhasil jika *client* mendapatkan respon *success* dari *server*. hasil dari pengujian fungsi peringatan pintu dapat dilihat pada Tabel 2.25.

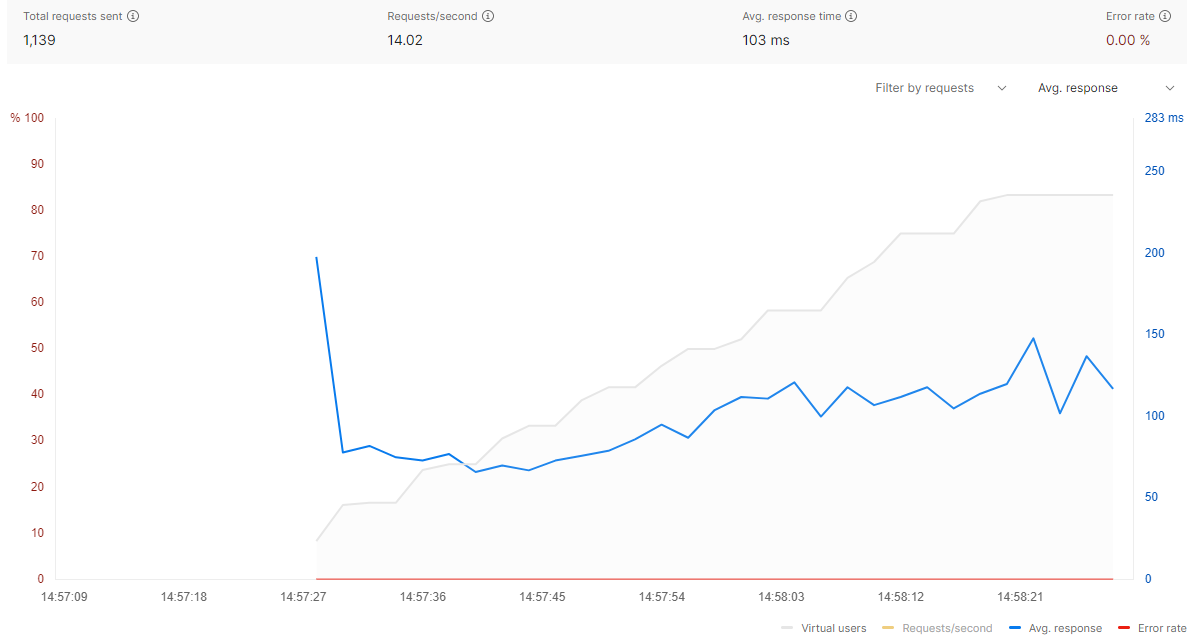
Tabel 2. Hasil pengujian fungsi peringatan pintu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama** | **Bentuk Pengujian** | **Respon** | **Hasil** |
| 1 | Peringatan pintu tanpa token | Mengirim peringatan pintu tanpa menggunakan token | *Unauthenticated* | Gagal |
| 2 | Peringatan pintu sesuai | Mengirim peringatan pintu dengan menggunaka token dan data lengkap | *success* | Berhasil |
| 3 | Peringatan pintu data tidak lengkap | Mengirim peringatan pintu dengan menggunakan data yang tidak lengkap | *missing\_parameter* | Gagal |
| 4 | Peringatan pintu id pintu salah | Mengirim peringatan pintu menggunakan data identitas pintu yang salah | *failed* | Gagal |

Pada Tabel 2.25 terlihat bahwa proses peringatan pintu berhasil jika *request* dikirimkan dengan token dan data yang lengkap, jika ada data yang kurang lengkap atau tidak disertai dengan token akan *request* tersebut akan gagal.

1. Pengujian Performa API

Pengujian pengujian performa API berfokus pada karakteristik dari sistem API yang telah dibuat seperti rasio *error*, waktu respon dan lain sebagainya yang menjadi indikator performa dari sistem tersebut. Pada pengujian performa ini dilakukan dengan menggunakan Postman dan Jmeter. Jemeter merupakan sebuah perangkat lunak yang digunakan untuk melakukan pengujian terutama pengujian beban dengan memberikan beberapa *request* secara bersamaan[1].



Gambar 2. Hasil Pengujian Performa API

Dapat dilihat pada Gambar 2.8, pada pengujian API secara keseluruhan dengan menggunakan 10 pengguna secara bersamaan menunjukkan waktu respon rata-rata 104 milidetik dengan waktu respon tertinggi 148 milidetik dan rasio error 0%. Pengujian performa juga dilakukan dengan melakukan simulasi beban pengguna pada API yang kemungkinan besar akan diakses secara bersamaan dan menentukan kinerja dari sistem keamanan kunci pintu tersebut, yaitu :

1. Performa verifikasi akses

Pada sistem penguncian pintu gedung ini kemungkinan pengguna melakukan permintaan akses secara bersamaan, secara teori semakin banyak pengguna yang mengirimkan *request* maka waktu respon akan semakin meningkat. Oleh karena itu dilakukan pengujian untuk melihat kemampuan sistem dalam menangani permintaan akses tersebut.

Gambar 2. Grafik hasil pengujian performa verifikasi akses

Gambar 2.9 merupakan hasil pengujian fungsi verifikasi akses menggunakan Jmeter. Pengujian dilakukan dengan melakukan simulasi beban permintaan verifikasi akses sebanyak 100 pengguna berbeda secara bersamaan. Dari hasil pengujian didapatkan bahwa pada puncak jumlah pengguna sistem membutuhkan waktu rata-rata 1.3 detik untuk memberikan respon ke pengguna dengan rasio *error* 0.0%.

1. Performa *update* status pintu

Setiap terjadi perubahan status pada pintu maka perangkat penguncian akan langsung mengirimkan status perubahan ke server. Dalam prosesnya dimungkinkan beberapa pintu mengirimkan status perubahan secara bersamaan sehingga dapat mempengaruhi kinerja dari sistem.

Gambar 2. Hasil Pengujian Performa Update Status Pintu

Gambar 2.10 merupakan hasil pengujian performa *update* status pintu menggunakan Jmeter. Proses pengujian mendapatkan hasil bahwa pada beban 100 pintu berbeda melakukan *update* status secara bersamaan maka sistem memerlukan waktu rata-rata 2.5 detik dengan rasio error 0.0%.

1. Performa penjadwalan

Pada proses penjadwalan maka sistem akan melakukan pemeriksaan data jadwal pada *database* setiap 1 menit, oleh karena itu metode pengecekan jadwal diharuskan selesai dilaksanakan sebelum proses pemeriksaan selanjutnya dijalankan. hasil dari proses pengujian performa penjadwalan dapat dilihat pada Gambar 2.11 dibawah.

Gambar 2. Hasil Pengujian Performa Penjadwalan

Pengujian dilakukan dengan mencatat waktu pemeriksaan untuk setiap jadwal, yaitu terdapat 11 titik pengujian dimulai dari 1 jadwal sampai 100 jadwal dengan masing-masing jadwal terdapat 20 pintu. Dapat dilihat pada Gambar 4.34 diatas, semakin besar jumlah jadwal yang ada maka waktu yang diperlukan pada proses penjadwalaa akan semakin lama dimana untuk 1 jadwal memerlukan waktu 96.00 milidetik dan 100 jadwal memerlukan waktu 3641.14 milidetik. Dari hasil pengujian menggunakan 100 jadwal waktu yang diperlukan yaitu 3.6 detik dimana nilai tersebut masih dibawah dari periode pengecekan jadwal (1 menit) sehingga sistem masih dapat menangani 100 proses penjadwalan dengan aman.

1. Pengujian Fungsionalitas *Website*

Proses pengujian fungsionalitas *website* berfokus pada fungsi dari setiap fitur yang telah diimplementasikan. Pengujian dilakukan dengan menggunakan metode *blackbox* dengan menggunakan *test*-*case* yang disesuaikan dengan berbagai kemungkinan interaksi pengguna pada fitur tersebut. Hasil dari proses pengujian fungsionalitas *website* dapat dilihat pada Tabel 2.26 di bawah.

Tabel 2. Hasil pengujian kebutuhan *website*

|  |  |  |
| --- | --- | --- |
| **No** | **Kriteria Kebutuhan** | **Hasil** |
| 1. | Tersedia Halaman *Login* dan *logout* *website* | Tersedia |
| 2. | Tersedia halaman *Reset Password* | Tersedia |
| 3. | Tersedia halaman untuk mengisi kode OTP | Tersedia |
| 4. | Tersedia menu dan halaman daftar pintu | Tersedia |
| 5. | Tersedia menu dan halaman daftar pengguna | Tersedia |
| 6. | Tersedia menu dan halaman penjadwalan | Tersedia |
| 7. | Tersedia menu dan halaman Riwayat akses | Tersedia |
| 8. | Tersedia menu dan halaman pengaturan akun | Tersedia |
| 9. | Tersedia menu dan halaman daftar Gedung | Tersedia |
| 10 | Tersedia menu dan halaman daftar operator | Tersedia |

Dapat dilihat pada Tabel 2.26 di atas, semua fitur atau menu yang dimiliki oleh *website* telah berhasil diimplementasikan menggunakan metode dan pemrograman yang sesuai. Proses pengujian lebih lanjut dilakukan pada semua fitur yang ada dengan hasil sebagai berikut :

1. Autentikasi

Pengujian pada halaman autentikasi dilaksanakan untuk menguji keberhasilan dari fungsi *login* dan *logout* yang bekerja pada halaman *website*. Hasil pengujian halaman autentikasi ditunjukkan pada tabel 2.27 dibawah.

Tabel 2. Hasil pengujian fungsi autentikasi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Nama Pengujian** | **Bentuk Pengujian** | | **Hasil Pengujian** | **Status** |
| 1. | *Login* dengan data benar | | Melakukan *login* menggunakan *email* dan *password* yang sesuai | masuk ke halaman *dashboard* | Berhasil |
| 2. | *Login* dengan data salah | | Melakukan *login* menggunakan *email* atau *password* yang salah | Pesan *login* gagal | Gagal |
| 3. | *Login* dengan data kurang | | Melakukan *login* menggunakan *email* saja atau *password* saja | Tampil peringatan untuk mengisi kolom yang masih kosong. | Gagal |
| 4. | *Login* dengan data tidak terdaftar | | Melakukan *login* menggunakan *email* yang belum terdaftar | Tampil peringatan “*login* gagal”. Data tidak ditemukan. | Gagal |

**Tabel 2.27** (lanjutan)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Nama Pengujian** | **Bentuk Pengujian** | | **Hasil Pengujian** | **Status** |
| 5. | *Login* dengan format tidak sesuai | | Melakukan *login* menggunakan *username* bukan *email* | Tampil peringatan untuk menyertakan “@” pada format penulisan *email*. | Gagal |
| 6. | *Login* dengan *email* belum terverifikasi | | Melakukan *login* menggunakan *email* yang berlum terverifikasi | Tampil peringatan *login* gagal dan data tidak ditemukan | Gagal |
| 7. | *Logout* dengan menekan tombol *logout* | | Melakukan Tindakan untuk *logout* dengan menekan tombol *logout* | Akan Kembali keluar ke halaman *login* | Berhasil |

Terlihat pada Tabel 2.27 bahwa proses pengujian yang dilakukan pada fungsi autentikasi *website*, fungsi autentikasi akan berhasil jika data yang dimasukkan benar sesuai dengan ketentuan dan proses autentikasi akan gagal jika data tidak sesuai.

1. *Reset* *password*

Pengujian pada halaman *reset password* dilaksanakan untuk menguji keberhasilan dari fungsi tampilan dan akses yang bekerja pada halaman *reset password* pada *website*. Hasil pengujian halaman *reset password* ditunjukkan pada Tabel 2.28 di bawah.

Tabel 2. Hasil pengujian *reset* *password*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Pengujian** | **Bentuk Pengujian** | **Hasil Pengujian** | **Status** |
| 1. | Menampilkan halaman r*eset password* | Melakukan akses dengan menekan tombol lupa *password* | Halaman *reset password b*erhasil ditampilkan | Berhasil |
| 2. | *reset password* sesuai | Mengisi formulir *reset password* dengan *email* yang terdaftar | Mendapat *email* balasan berupa *link* untuk mengakses *reset password* | Berhasil |
| 3. | Mengganti  *password* | Mengisi *password* baru dan konfirmasi *password* baru | Terkonfirmasi dan masuk ke halaman *login*. | Berhasil |
| 4. | *email* tidak terdaftar | Memasukkan *email* yang tidak terdaftar atau salah memasukkan alamat *email* | Tampil peringatan bahwa alamat *email* tidak ditemukan | Gagal |

Dapat dilihat pada Tabel 2.28 bahwa proses *reset* *password* akan berhasil jika prosesnya menggunakan alamat *email* yang terdaftar serta *password* baru yang dimasukkan sesuai dengan format yang telah ditentukan.

1. Verifikasi *email*

Pengujian pada halaman verifikasi email dilaksanakan untuk menguji keberhasilan dari fungsi tampilan dan akses untuk melakukan verifikasi *email* dengan memasukkan kode OTP yang bekerja pada halaman verifikasi *email* pada *website*. Hasil pengujian halaman verifikasi email ditunjukkan pada tabel 2.29 di bawah.

Tabel 2. Hasil pengujian verifikasi email

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No**. | **Nama Pengujian** | **Bentuk Pengujian** | **Hasil Pengujian** | **Status** |
| 1. | Verifikasi *email* tidak sesuai | Melakukan verifikasi *email* menggunakan kode OTP yang salah | Tampil peringatan kode OTP tidak sesuai | Gagal |
| 2. | Verifikasi *email* kadaluwarsa. | Melakukan verifikasi *email* menggunakan kode OTP yang sudah kadaluwarsa | Tampil peringatan OTP sudah kedaluwarsa | Gagal |
| 3. | Verifikasi *email* sesuai | Melakukan verifikasi *email* menggunakan kode OTP yang sesuai | Verifikasi *email* berhasil | Berhasil |
| 4. | Verifikasi *email* *token* salah | Melakukan verifikasi *email* menggunakan *token* yang sudah terhapus | Kode OTP tidak diketahui | Gagal |
| 5. | Verifikasi *email* *token* lebih dari 6 digit | Melakukan verifikasi *email* menggunakan *token* yang lebih dari 6 digit | Tampil peringatan bahwa *token* lebih dari 6 digit | Gagal |

Dapat dilihat pada Tabel 2.29 bahwa proses verifikasi *email* akan berhasil jika menggunakan kode OTP yang sesuai dan masih aktif yang telah diterima pada alamat *email* yang terdaftar.

1. Daftar pintu

Pengujian pada halaman daftar pintu dilaksanakan untuk menguji keberhasilan dari fungsi tampilan dan akses pada halaman daftar pintu pada *website*. Hasil pengujian halaman daftar pintu ditunjukkan pada Tabel 2.30 di bawah.

Tabel 2. Hasil pengujian daftar pintu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Pengujian** | **Bentuk Pengujian** | **Hasil Pengujian** | **Status** |
| 1. | Membuka menu daftar pintu pada halaman *dashboard* | Melakukan akses dengan menekan menu daftar pintu | Halaman daftar pintu berhasil ditampilkan | Berhasil |
| 2. | Menampilkan daftar pintu yang terdaftar dalam aplikasi | Melakukan akses untuk melihat daftar pintu | Daftar pintu berhasil ditampilkan | Berhasil |
| 3. | Menambahkan daftar pintu | Melakukan akses untuk menambah daftar pintu dengan menekan tombol tambah pada halaman daftar pintu | Tampil formulir untuk mengisi nama pintu yang akan ditambahkan | Berhasil |
| 4. | Menambahkan daftar pintu dengan nama yang sama | Melakukan akses untuk menambah daftar pintu dengan menggunakan nama yang sudah ada dalam daftar pintu sebelumnya | Daftar pintu baru berhasil ditambahkan | Gagal |
| 5. | Mencari daftar pintu yang tersedia | Mencari daftar pintu yang sudah terdaftar dengan cepat menggunakan menu cari pintu | Daftar pintu berhasil ditemukan setelah dicari dengan kata kunci yang dituliskan | Berhasil |
| 6. | Membuka dan menutup akses penguncian pintu | Melakukan akses untuk membuka dan menutup daftar pintu dengan menekan tombol buka atau tutup | Muncul formulir untuk konfirmasi buka pintu atau batal buka pintu | Berhasil |

**Tabel 2.30** (lanjutan)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Pengujian** | **Bentuk Pengujian** | **Hasil Pengujian** | **Status** |
| 5. | Mencari daftar pintu yang tersedia | Mencari daftar pintu yang sudah terdaftar dengan cepat menggunakan menu cari pintu | Daftar pintu berhasil ditemukan setelah dicari dengan kata kunci yang dituliskan | Berhasil |
| 6. | Membuka dan menutup akses penguncian pintu | Melakukan akses untuk membuka dan menutup daftar pintu dengan menekan tombol buka atau tutup | Muncul formulir untuk konfirmasi buka pintu atau batal buka pintu | Berhasil |

1. Daftar pengguna

Pengujian pada halaman daftar pengguna dilaksanakan untuk menguji keberhasilan dari fungsi tampilan dan akses pada halaman daftar pengguna pada *website*. Hasil pengujian halaman daftar pengguna ditunjukkan pada Tabel 2.31 di bawah.

Tabel 2. Hasil pengujian daftar pengguna

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Pengujian** | **Bentuk Pengujian** | **Hasil Pengujian** | **Status** |
| 1. | Membuka menu daftar pengguna pada halaman *dashboard* | Melakukan akses dengan menekan menu daftar pengguna | Halaman daftar pengguna berhasil ditampilkan | Berhasil |
| 2. | Menampilkan daftar pengguna yang terdaftar dalam aplikasi | Melakukan akses untuk melihat daftar pengguna | Daftar pengguna berhasil ditampilkan | Berhasil |
| 3. | Menampilkan daftar pengguna ketika belum ada pintu yang terdaftar | Melakukan akses untuk melihat daftar pengguna | Daftar pengguna tidak ditemukan | Gagal |
| 4. | Menambahkan daftar pengguna | Menambah daftar pengguna dengan menekan tombol tambah pada halaman daftar pintu | Tampil formulir untuk mengisi nama pengguna yang akan ditambahkan | Berhasil |
| 5. | Menambahkan daftar pengguna dengan nama yang berbeda | Melakukan akses untuk menambah daftar pengguna dengan menggunakan nama yang belum ada dalam daftar pengguna sebelumnya | Daftar pengguna baru berhasil ditambahkan | Berhasil |
| 6. | Menambahkan daftar pengguna dengan nama yang sama | Melakukan akses untuk menambah daftar pengguna dengan menggunakan nama yang sudah ada dalam daftar pengguna sebelumnya | Daftar pengguna baru berhasil ditambahkan | Gagal |
| 7. | Mencari daftar pengguna yang tersedia | Mencari daftar pengguna yang sudah terdaftar dengan cepat menggunakan menu cari pintu | Daftar pengguna berhasil ditemukan setelah dicari dengan kata kunci yang dituliskan | Berhasil |
| 8. | Mencari daftar pengguna yang tidak tersedia | Mencari daftar pengguna yang belom terdaftar dengan cepat menggunakan menu cari pintu | Daftar pengguna tidak ditemukan setelah dicari dengan kata kunci yang dituliskan | Gagal |
| 9. | Melihat daftar pengguna | Melakukan akses untuk melihat daftar pengguna dengan menekan tombol “lihat” | Dapat melihat *profile* pengguna dan melihat akses pintu pengguna | Berhasil |
| 10. | Menghapus daftar pengguna | Melakukan akses untuk menghapus daftar pengguna dengan menekan tombol “hapus” | Muncul formulir untuk mengkonfirmasi penghapusan | Berhasil |

**Tabel 2.31** (lanjutan)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Pengujian** | **Bentuk Pengujian** | **Hasil Pengujian** | **Status** |
| 6. | Menambahkan daftar pengguna dengan nama yang sama | Menambah daftar pengguna dengan menggunakan nama yang sudah ada dalam daftar pengguna sebelumnya | Daftar pengguna baru berhasil ditambahkan | Gagal |
| 7. | Mencari daftar pengguna yang tersedia | Mencari daftar pengguna yang sudah terdaftar dengan cepat menggunakan menu cari pintu | Daftar pengguna berhasil ditemukan setelah dicari dengan kata kunci yang dituliskan | Berhasil |
| 8. | Mencari daftar pengguna yang tidak tersedia | Mencari daftar pengguna yang belom terdaftar dengan cepat menggunakan menu cari pintu | Daftar pengguna tidak ditemukan setelah dicari dengan kata kunci yang dituliskan | Gagal |
| 9. | Melihat daftar pengguna | Melakukan akses untuk melihat daftar pengguna dengan menekan tombol “lihat” | Dapat melihat *profile* pengguna dan melihat akses pintu pengguna | Berhasil |
| 10. | Menghapus daftar pengguna | Melakukan akses untuk menghapus daftar pengguna dengan menekan tombol “hapus” | Muncul formulir untuk mengkonfirmasi penghapusan | Berhasil |

1. Penjadwalan

Pengujian pada halaman penjadwalan dilaksanakan untuk menguji keberhasilan dari fungsi tampilan dan akses pada halaman penjadwalan pada *website*. Hasil pengujian halaman penjadwalan ditunjukkan pada Tabel 2.32 di bawah.

Tabel 2. Hasil pengujian penjadwalan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Pengujian** | **Bentuk Pengujian** | **Hasil Pengujian** | **Status** |
| 1. | Membuka menu daftar pengguna pada halaman *dashboard* | Melakukan akses dengan menekan menu daftar pengguna | Halaman daftar pengguna berhasil ditampilkan | Berhasil |
| 2. | Menambahkan penjadwalan pada kunci pintu | Melakukan akses untuk menambah penjadwalan menekan tombol tambah | Tampipl formulir untuk mengatur jadwal penguncian | Berhasil |
| 3. | Mencari daftar pintu yang sudah terjadwal | Mencari daftar pintu yang sudah terjadwal pada sistem | Daftar pintu yang terjadwal berhasil ditemukan menggunakan kata kunci yang sesuai | Berhasil |
| 4. | Pengaturan jadwal kunci pintu | Mengatur penjadwalan pada formulir dengan memasukkan data nama pintu, tanggal, durasi harian, dan perulangan hari | Penjadwalan penguncian terkonfirmasi dan dapat terjadwal sesuai pengaturan | Berhasil |

**Tabel 2.32** (lanjutan)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Pengujian** | **Bentuk Pengujian** | **Hasil Pengujian** | **Status** |
| 5. | Memilih tanggal dan durasi pengaturan jadwal pintu | Menentukan tanggal dan durasi penguncian pintu sesuai permintaan | Pintu hanya dapat dibuka sesuai jadwal yang telah ditetapkan | Berhasil |
| 6. | Konfirmasi Penjadwalan | Melakukan akses dengan menekan tombol “Tambahkan” | Penjadwalan yang telah diatur berhasil terkonfirmasi | Berhasil |
| 7. | Pembatarlan penjadwalan akses kunci pintu | Melakukan akses dengan menekan tombol “Batal” | Penjadwalan yang telah diatur batal terkonfirmasi | Berhasil |

1. Riwayat akses

Pengujian pada halaman riwayat akses dilaksanakan untuk menguji keberhasilan dari fungsi tampilan dan akses pada halaman riwayat akses pada *website*. Hasil pengujian halaman riwayat akses ditunjukkan pada Tabel 2.33 di bawah.

Tabel 2. Hasil pengujian riwayat akses

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Pengujian** | **Bentuk Pengujian** | **Hasil Pengujian** | **Status** |
| 1. | Membuka menu riwayat akses | Melakukan akses dengan menekan menu riwayat akses | Halaman riwayat akses berhasil ditampilkan | Berhasil |
| 2. | Menampilkan riwayat akses yang terdaftar dalam sistem | Melakukan akses untuk melihat halaman riwayat akses | Halaman riwayat akses berhasil ditampilkan | Berhasil |

**Tabel 2.33** (lanjutan)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Pengujian** | **Bentuk Pengujian** | **Hasil Pengujian** | **Status** |
| 4. | Mencari riwayat pintu yang telah diakses oleh pengguna | Melakukan akses dengan mengetikkan kata kunci pada kolom untuk mencari riwayat pintu yang telah diakses oleh pengguna | Riwayat pintu yang telah diakses berhasil ditemukan | Berhasil |

1. Pengaturan akun

Pengujian pada halaman riwayat akses dilaksanakan untuk menguji keberhasilan dari fungsi tampilan dan akses pada halaman riwayat akses pada website. Hasil pengujian halaman riwayat akses ditunjukkan pada Tabel 2.34 di bawah.

Tabel 2. Hasil pengujian pengaturan akun

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Pengujian** | **Bentuk Pengujian** | **Hasil Pengujian** | **Status** |
| 1. | Membuka halaman *profile* | Melakukan akses dengan menekan menu *profile* | Halaman *profile* berhasil ditampilkan | Berhasil |
| 2. | Melakukan *upload* foto | Melakukan akses dengan menekan tombol *upload* foto | Tersedia formulir untuk menambahkan foto profil | Berhasil |
| 3. | Memilih file foto | Memilih file foto dengan menekan tombol pilih file dan memilih file yang sesuai dengan format dan ukuran | Tampil ke halaman penyimpanan perangkat untuk memilih foto profil | Berhasil |

**Tabel 2.34** (lanjutan)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Pengujian** | **Bentuk Pengujian** | **Hasil Pengujian** | **Status** |
| 4. | Memilih foto dengan format dan ukuran yang tidak sesuai | Memilih file foto dengan format dan ukuran yang tidak sesuai dengan ketentuan | Foto profil tidak berhasil di *upload* | Gagal |
| 5. | Mengedit profil | Menekan tombol “edit profil” | Berhasil muncul formulir untuk melakukan update profil | Berhasil |
| 6. | Konfirmasi edit profil | Melakukan akses dengan menekan tombol “simpan” | Berhasil menyimpan pembaruan profil | Berhasil |
| 7. | Mambatalkan edit profil | Melakukan akses dengan menekan tombol “batal” | Pembaruan profil dibatalkan | Berhasil |
| 8. | Mengganti *password* | Melakukan akses dengan menekan tombol ganti *password* | Berhasil muncul formulir untuk melakukan ganti *password* | Berhasil |
| 9. | Memasukkan *password* saat ini dengan benar | Mengetikkan *password* saat ini dengan benar | Berhasil tampil formulir untuk mengisi *password* baru | Berhasil |
| 10. | Salah memasukkan *password* saat ini | Mengetikkan *password* saat ini dengan salah | Tidak dapat untuk mengganti *password* baru | Gagal |
| 11. | Konfirmasi *password* baru | Memasukkan *password* baru dan konfirmasi *password* baru pada form | *Password* baru berhasil terkonfirmasi | Berhasil |
| 12. | Membuka tombol menu *logout* | Melakukan akses dengan menekan tombol *logout* | Berhasil untuk keluar dari aplikasi | Berhasil |

1. Daftar gedung

Pengujian pada halaman daftar gedung dilaksanakan untuk menguji keberhasilan dari fungsi tampilan dan akses pada halaman daftar gedung pada website. Hasil pengujian halaman daftar gedung ditunjukkan pada Tabel 2.35 di bawah.

Tabel 2. Hasil pengujian daftar gedung

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Pengujian** | **Bentuk Pengujian** | **Hasil Pengujian** | **Status** |
| 1. | Membuka menu daftar gedung pada halaman *dashboard* | Melakukan akses dengan menekan menu daftar gedung | Halaman daftar gedung berhasil ditampilkan | Berhasil |
| 2. | Menambahkan daftar gedung dengan nama yang berbeda | Melakukan akses untuk menambah daftar gedung dengan menggunakan nama yang belom ada dalam daftar gedung sebelumnya | Daftar gedung baru berhasil ditambahkan | Berhasil |
| 3. | Menambahkan daftar gedung dengan nama yang sama | Melakukan akses untuk menambah daftar gedung dengan menggunakan nama yang sudah ada dalam daftar pintu sebelumnya | Daftar gedung baru berhasil ditambahkan | Gagal |
| 4. | Mencari daftar gedung yang tersedia | Mencari daftar gedung yang sudah terdaftar dengan cepat menggunakan menu cari gedung | Daftar gedung berhasil ditemukan setelah dicari dengan kata kunci yang dituliskan | Berhasil |
| 5. | Mencari daftar gedung yang tidak tersedia | Mencari daftar gedung yang belom terdaftar dengan cepat menggunakan menu cari gedung | Daftar gedung tidak ditemukan setelah dicari dengan kata kunci yang dituliskan | Gagal |
| 6. | Membuka dan menutup akses penguncian pintu | membuka dan menutup daftar pintu dengan menekan tombol buka atau tutup | Muncul formulir untuk konfirmasi buka pintu atau batal buka pintu | Berhasil |
| 7. | Mengedit daftar gedung | Melakukan akses untuk mengedit daftar gedung dengan menekan tombol “edit” | Muncul formulir untuk mengedit nama dan operator gedung | Berhasil |
| 8. | Menghapus daftar gedung | Melakukan akses untuk menghapus daftar gedung dengan menekan tombol “hapus” | Muncul formulir untuk mengkonfirmasi penghapusan daftar gedung | Berhasil |

1. Daftar operator

Pengujian pada halaman daftar operator dilaksanakan untuk menguji keberhasilan dari fungsi tampilan dan akses pada halaman daftar operator pada website. Hasil pengujian halaman daftar operator ditunjukkan pada Tabel 2.36 dibawah.

Tabel 2. Hasil pengujian daftar operator

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Pengujian** | **Bentuk Pengujian** | **Hasil Pengujian** | **Status** |
| 1. | Membuka menu daftar operator pada halaman *Dashboard* khusus moderator | Melakukan akses dengan menekan menu daftar operator | Halaman daftar operator berhasil ditampilkan | Berhasil |
| 2. | Menampilkan daftar operator yang terdaftar dalam aplikasi | Melakukan akses untuk melihat daftar operator | Daftar operator berhasil ditampilkan | Berhasil |
| 3. | Menampilkan daftar operator ketika belum ada pintu yang terdaftar | Melakukan akses untuk melihat daftar operator | Daftar operator tidak ditemukan | Gagal |
| 4. | Menambahkan daftar operator | Melakukan akses untuk menambah daftar operator dengan menekan tombol tambah pada halaman daftar operator | Tampil formulir untuk mengisi nama operator yang akan ditambahkan | Berhasil |
| 5. | Menambahkan daftar operator dengan nama yang berbeda | Melakukan akses untuk menambah daftar operator dengan menggunakan nama yang belom ada dalam daftar operator sebelumnya | Daftar operator baru berhasil ditambahkan | Berhasil |
| 6. | Menambahkan daftar operator dengan nama yang sama | Menambah daftar operator dengan menggunakan nama yang sudah ada dalam daftar operator sebelumnya | Daftar operator baru berhasil ditambahkan | Gagal |
| 7. | Mencari daftar operator yang tersedia | Mencari daftar operator yang sudah terdaftar dengan cepat menggunakan menu cari pintu | Daftar operator berhasil ditemukan setelah dicari dengan kata kunci yang dituliskan | Berhasil |
| 8. | Mencari daftar operator yang tidak tersedia | Mencari daftar operator yang belom terdaftar dengan cepat menggunakan menu cari operator | Daftar operator tidak ditemukan setelah dicari dengan kata kunci yang dituliskan | Gagal |
| 9. | Menghapus daftar operator | Melakukan akses untuk menghapus daftar operator dengan menekan tombol “hapus” | Muncul formulir untuk mengkonfirmasi penghapusan | Berhasil |

1. Pengujian Performa Website

Pengujian performa website bertujuan untuk mengetahui kerakeristik dari website yang telah dikembangkan seperti rasio error, waktu respon dan lain sebagainya. Pengujian dilakukan dengan menggunakan metode load testing dengan menggunakan Jmeter. Hasil dari pengujian performa website adalah sebagai berikut :

1. Performa *landing* *page*

Halaman *landing page* diuji untuk memastikan bahwa tujuan utamanya tercapai dengan efektif dan memberikan pengalaman yang positif kepada pengunjung.Pengujian halaman website melibatkan analisis interaksi antara variabel-variabel penting seperti latensi (waktu respons *server*), pengguna (jumlah pengunjung), sampel (data yang dimuat), dan waktu (kecepatan pemuatan). Pengujian ini bertujuan untuk memahami bagaimana kinerja halaman berubah saat beban pengguna meningkat, yang dapat mempengaruhi latensi dan waktu pemuatan halaman. Hasil dari pengujian pada halaman *landing page* aplikasi website dapat dilihat pada Gambar 2.12 di bawah.

Gambar 2. Performa landing page

Gambar 2.12 diatas menampilkan grafik yang menggambarkan interaksi antara latensi (waktu respons *server*), pengguna (jumlah pengunjung), sampel (data yang dimuat), dan waktu (kecepatan pemuatan) pada *landing page*. Pada pengujian ini menggunakan 200 sampel pengujian. Ketika jumlah pengunjung naik maka waktu respons *server* juga akan naik, waktu yang dibutuhkan untuk mengakses data juga naik. Pada Gambar 4.46 diatas puncak dari latensi ditunjukkan ketika sample di angka 90 *user* dan waktu yang dibutuhkan sekitar 1200 milidetik. Hal itu menunjukkan bahwa website dapat diakses oleh banyak pengguna secara bersamaan dengan waktu respon yang masih wajar.

1. Performa *login*

Halaman *login* diuji untuk memahami bagaimana interaksi antara latensi, pengguna, sampel, dan waktu mempengaruhi pengalaman pengguna saat mencoba masuk ke dalam sistem. Pengujian ini penting karena halaman *login* adalah titik masuk utama bagi pengguna untuk mengakses aplikasi atau *platform*, dan kinerjanya dapat sangat mempengaruhi kesan awal pengguna terhadap sistem. Hasil dari pengujian pada halaman *login* aplikasi website dapat dilihat pada Gambar 2.13 dibawah.

Gambar 2. Performa halaman login

Gambar 2.13 diatas menampilkan grafik yang menggambarkan interaksi antara latensi (waktu respons *server*), pengguna (jumlah pengunjung), sampel (data yang dimuat), dan waktu (kecepatan pemuatan) pada halaman *login*. Dalam uji coba ini, dilibatkan sebanyak 200 sampel pengujian. Saat jumlah pengunjung meningkat, terlihat peningkatan waktu respons dari *server*, serta durasi untuk mengakses data. Dalam Diagram 4.47 yang tercantum di atas, titik puncak latensi mencapai nilai tertinggi ketika sampel mencapai 60 pengguna, dengan durasi respons *server* hanya sekitar 1000 milidetik. Situasi ini mengindikasikan kemampuan halaman landing page dalam mengelola proses dengan performa pemuatan dan respon yang optimal. Hal itu menunjukkan bahwa website dapat diakses oleh banyak pengguna secara bersamaan dengan waktu respon yang masih wajar.

1. Performa *reset password*

Halaman lupa *password* diuji untuk memahami bagaimana interaksi antara latensi (waktu respons *server*), pengguna (jumlah pengguna yang mengakses halaman tersebut), sampel (data yang diakses untuk mengelola permintaan lupa *password*), dan waktu mempengaruhi pengalaman pengguna saat menggunakan fitur ini. Pengujian ini penting karena halaman lupa *password* adalah bagian penting dari pengalaman pengguna dalam mengatasi masalah akses ke akun mereka. Hasil dari pengujian pada halaman *login* aplikasi website dapat dilihat pada Gambar 2.14 dibawah.

Gambar 2. Performa halaman reset password

Gambar 2.14 diatas menampilkan grafik yang menggambarkan interaksi antara latensi (waktu respons *server*), pengguna (jumlah pengunjung), sampel (data yang dimuat), dan waktu (kecepatan pemuatan) pada halaman lupa *password*. Dalam uji coba ini, dilibatkan sebanyak 200 sampel pengujian. Dalam skenario lonjakan pengunjung, terlihat lonjakan waktu respons *server* dan waktu akses data. Pada Diagram 4.48, puncak latensi terjadi saat sampel mencapai 20 pengguna, dengan respons *server* sekitar 1000 milidetik. Situasi ini mencerminkan performa baik halaman landing page dalam mengelola pemuatan dan respon secara optimal dan menunjukkan bahwa website dapat diakses oleh banyak pengguna secara bersamaan dengan waktu respon yang masih wajar.

1. Pengujian Fungsionalitas Aplikasi Mobile

Pengujian fungsionalitas aplikasi mobile bertujuan untuk melihat kinerja dari fitur yang disediakan oleh aplikasi apakah sudah sesuai atau belum. Proses pengujian menggunakan metode blackbox dengan menggunakan berbagai kemungkinan interaksi pengguna terhadap aplikasi mobile. Hasil dari proses pengujian adalah sebagai berikut :

1. Autentikasi

Pengujian pada fungsi *authentication* bertujuan untuk mengidentifikasi potensi kerentanan, kesalahan, atau masalah keamanan yang dapat dieksploitasi oleh pihak yang tidak berwenang. Pengujian pada halaman *authentication* dilaksanakan untuk menguji keberhasilan dari fungsi *login* dan *logout* yang bekerja pada halaman tersebut. Hasil pengujian halaman *authentication* ditunjukkan pada Tabel 4.3 dibawah.

Tabel 2. Hasil pengujian autentikasi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Nama** | | **Bentuk Pengujian** | **Hasil Pengujian** | **Status** |
| 1. | *Login* dengan data benar | Melakukan *Login* menggunakan *email* dan *password* yang sesuai | | Berhasil melakukan *Login* dan mendapatkan akses untuk masuk ke halaman *Dashboard* | Berhasil |
| 2. | *Login* dengan data salah | Melakukan *Login* dengan menggunakan *email* atau *password* yang salah | | Tidak mendapatkan akses masuk ke halaman *Dashboard* | Gagal |
| 3. | *Login* dengan data kurang | Melakukan *Login* dengan menggunakan *email* saja atau *password* saja | | Tampil peringatan untuk mengisi kolom yang masih kosong. | Gagal |
| 4. | *Login* dengan data tidak terdaftar | Melakukan *Login* dengan menggunakan *email* yang belum terdaftar | | Tampil peringatan “*Login* Gagal”. Data tidak ditemukan. | Gagal |
| 5. | *Login* dengan format tidak sesuai | Melakukan *Login* dengan menggunakan *User*name bukan *email* | | Tampil peringatan untuk menyertakan “@” pada format penulisan *email*. | Gagal |
| 6. | *Login* dengan *email* belum terverifikasi | Melakukan *Login* dengan menggunakan *email* yang berlum terverifikasi | | Tampil peringatan *Login* Gagal dan data tidak ditemukan | Gagal |
| 7. | *Logout* dengan menekan tombol *logout* | Melakukan Tindakan untuk *logout* dengan menekan tombol *logout* | | Akan Kembali keluar ke halaman *Login* | Berhasil |

Pada Tabel 2.37 terlihat bahwa proses authentication login berhasil apabila email dan password sesuai dengan format yang ditentukan. Proses login dilakukan dengan menekan tombol login pada formulir dan untuk proses logout dilakukan dengan menekan tombol logout.

1. Verifikasi email

Pengujian pada halaman verifikasi email dilaksanakan untuk menguji keberhasilan dari fungsi verifikasi email dengan memasukkan token yang sesuai. Selain itu juga untuk menguji keberhasilan fungsi yang bekerja pada halaman tersebut. Hasil pengujian halaman verifikasi *email* ditunjukkan oleh Tabel 2.38 dibawah.

Tabel 2. Hasil Pengujian verifikasi email

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No**. | **Nama Pengujian** | **Bentuk Pengujian** | **Hasil Pengujian** | **Status** |
| 1. | Verifikasi *email* tanpa *token* | Melakukan verifikasi *email* tanpa menggunakan *token* | Tampil Peringatan untuk mengisi *token* | Gagal |
| 2. | Verifikasi *email* tidak sesuai | Melakukan verifikasi *email* menggunakan kode OTP yang salah | Tampil peringatan kode OTP tidak sesuai | Gagal |
| 3. | Verifikasi *email* kadaluwarsa. | Melakukan verifikasi *email* menggunakan kode OTP yang sudah kadaluwarsa | Tampil peringatan OTP sudah kedaluwarsa | Gagal |
| 4. | Verifikasi *email* sesuai | Melakukan verifikasi *email* menggunakan kode OTP yang sesuai | Verifikasi *email* berhasil | Berhasil |
| 5. | Verifikasi *email* *token* salah | Melakukan verifikasi *email* menggunakan *token* yang sudah terhapus | Kode OTP tidak diketahui | Gagal |
| 6. | Verifikasi *email* *token* lebih dari 6 digit | Melakukan verifikasi *email* menggunakan *token* yang lebih dari 6 digit | Tampil peringatan bahwa *token* lebih dari 6 digit | Gagal |

Dapat dilihat pada Tabel 2.38 diatas bahwa untuk melakukan verifikasi email pengguna diwajibkan mengisi email yang sesuai pada halaman kirim email. Kemudian email akan mengirimkan token yang akan diisi ke halaman isi token. Verifikasi akan berhasil jika token yang dimasukkan benar dan juga pengisian token tidak melewati batas waktu.

1. Reset password

Pengujian fungsi reset password pada aplikasi Android bertujuan memverifikasi kemampuan pengguna untuk dengan berhasil mengubah kata sandi mereka dengan aman dan mudah, khususnya dalam situasi di mana mereka lupa atau ingin mengganti kata sandi. Pengujian ini memiliki signifikansi penting dalam menjaga keamanan dan ketersediaan aplikasi atau situs web tersebut. Hasil pengujian halaman *reset password* ditunjukkan oleh tabel 2.39 dibawah.

**Tabel 2.39**Pengujian Fungsi *Reset Password*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Pengujian** | **Bentuk Pengujian** | **Hasil Pengujian** | **Status** |
| 1. | Menampilkan halaman *Reset Password* | Melakukan akses dengan menekan tombol lupa *password* | Halaman *Reset Password b*erhasil ditampilkan | Berhasil |
| 2. | Pengujian *Reset Password* | Mengisi form *Reset Password* dengan *email* yang terdaftar | Mendapat *email* balasan berupa link untuk mengakses *Reset Password* | Berhasil |
| 3. | Pengujian melakukan *Reset Password* | Mengisi *password* baru dan konfirmasi *password* baru | Terkonfirmasi dan masuk ke halaman *Login* untuk melakukan akses *Login*. | Berhasil |
| 4. | Menggunakan *email* yang tidak terdaftar | Memasukkan *email* yang tidak terdaftar atau salah memasukkan Alamat *email* | Tampil peringatan bahwa alamat *email* tidak ditemukan | Gagal |

Dapat dilihat pada Tabel 2.39 diatas bahwa pengguna berhasil melakukan akses untuk *reset password* dengan menekan tombol lupa *password* kemudian mengisi formulir dengan email yang sesuai. Apabila format email tidak sesuai maka *reset password* gagal.

1. Update password

Pengujian fungsi *update password* pada pengujian aplikasi Android memiliki tujuan untuk memastikan bahwa pengguna dapat mengganti kata sandi yang ada dengan sukses, dan bahwa perubahan tersebut terjadi dengan aman dan sesuai dengan harapan. Hasil pengujian halaman *update password* ditunjukkan oleh Tabel 2.40 dibawah.

**Tabel 2.40** Hasil pengujian update password

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Pengujian** | **Bentuk Pengujian** | **Hasil Pengujian** | **Status** |
| 1. | Menampilkan formulir *Update Password* | Melakukan akses dengan menekan tombol ganti *password* | Halaman *Update Password b*erhasil ditampilkan | Berhasil |
| 2. | Pengujian *Update Password* | Mengisi form konfirmasi *password* dengan *password* saat ini, *password* baru, dan konfirmasi *password* dengan benar. | *Update Password* berhasil | Berhasil |
| 3. | Konfirmasi *Update Password* | Mengisi form konfirmasi *password* dengan *password* yang tidak sesuai | Tampil peringatan bahwa ada yang salah dengan itu *password* tidak dapat diperbarui dan konfirmasi *password* tidak sesuai | Gagal |

Dapat dilihat pada tabel 4.6 diatas bahwa pengguna berhasil melakukan akses untuk *update password* dengan menekan tombol ganti *password* kemudian mengisi formulir dengan email yang sesuai. Apabila format email tidak sesuai maka *update password* gagal.

1. Halaman dashboard

Pengujian fungsi halaman *dashboard* dalam pengujian aplikasi Android memiliki tujuan untuk memastikan bahwa halaman ini berfungsi dengan baik, memberikan informasi yang relevan, dan menawarkan pengalaman pengguna yang baik. Hasil pengujian halaman *dashboard* ditunjukkan oleh Tabel 2.41 dibawah.

**Tabel 2.41** Hasil pengujian halaman dashboard

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Pengujian** | **Bentuk Pengujian** | **Hasil Pengujian** | **Status** |
| 1. | Menampilkan halaman *Dashboard* | Melakukan akses dengan menyelesaikan tahapan *authentication* dengan benar | Halaman *Dashboard* berhasil ditampilkan | Berhasil |
| 2. | Menampilkan gambar informasi tentang aplikasi dan sistem penguncian | Menggeser gambar informasi dengan melakukan scroll horizontal | Halaman gambar informasi dapat di scrool secara horizontal saat halaman dibuka | Berhasil |

Dapat dilihat pada Tabel 2.41 diatas bahwa pengguna berhasil menampilkan halaman *dashdoard* dengan menyelesaikan tahapan *authentication* dengan benar. Setelah menyelesaikan tahap *authentication* maka akan masuk ke halaman *dashboard*  dengan beberapa menu di dalamnya dan juga papan informasi yang berada di bagian dashboard.

1. Daftar pintu

Pengujian fungsi halaman daftar pintu pada pengujian aplikasi Android memiliki tujuan untuk memastikan bahwa pengguna dapat dengan efektif melihat dan mengelola daftar pintu yang ada dalam aplikasi tersebut. Hasil pengujian halaman daftar pintu ditunjukkan oleh Tabel 2.42 dibawah.

**Tabel 2.42** Hasil pengujian daftar pintu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Pengujian** | **Bentuk Pengujian** | **Hasil Pengujian** | **Status** |
| 1. | Membuka menu daftar pintu pada halaman *Dashboard* | Melakukan akses dengan menekan menu daftar pintu | Halaman daftar pintu berhasil ditampilkan | Berhasil |
| 2. | Menampilkan daftar pintu yang terdaftar dalam aplikasi | Melakukan akses untuk melihat daftar pintu | Daftar pintu berhasil ditampilkan | Berhasil |
| 3. | Menampilkan daftar pintu Ketika belum ada pintu yang terdaftar | Melakukan akses untuk melihat daftar pintu | Daftar pintu tidak ditemukan | Gagal |

Dapat dilihat pada Tabel 2.42 diatas bahwa pengguna berhasil menampilkan halaman daftar pintu dengan menekan menu daftar pintu. Daftar pintu akan ditampilkan apabila daftar pintu sudah terdaftar dalam aplikasi. Apabila belum terdaftar maka daftar pintu tidak dapat ditampilkan.

1. Daftar kunci

Pengujian fungsi halaman daftar kunci pada pengujian aplikasi Android bertujuan untuk memverifikasi bahwa pengguna mampu dengan sukses mengakses, mengamati, dan mengelola daftar kunci secara efisien dalam aplikasi tersebut. Hasil pengujian halaman daftar pintu ditunjukkan oleh Tabel 2.43 dibawah.

**Tabel 2.43** Hasil pengujian daftar kunci

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Pengujian** | **Bentuk Pengujian** | **Hasil Pengujian** | **Status** |
| 1. | Membuka menu daftar kunci pada halaman *Dashboard* | Melakukan akses dengan menekan menu daftar kunci | Halaman daftar kunci berhasil ditampilkan | Berhasil |
| 2. | Menampilkan daftar kunci yang terdaftar dalam aplikasi | Melakukan akses untuk melihat daftar kunci | Daftar kunci berhasil ditampilkan | Berhasil |
| 3. | Menampilkan daftar kunci ketika belum ada daftar kunci yang terdaftar | Melakukan akses untuk melihat daftar kunci | Daftar kunci tidak ditemukan | Gagal |

Dapat dilihat pada tabel 4.8 diatas bahwa pengguna berhasil menampilkan halaman daftar kunci dengan menekan menu daftar kunci. Daftar kunci akan ditampilkan apabila sudah terdaftar dalam aplikasi. Apabila belum terdaftar maka daftar kunci tidak dapat ditampilkan.

1. Riwayat akses

Pengujian fungsi halaman riwayat akses pada pengujian aplikasi Android memiliki tujuan untuk memastikan bahwa pengguna dapat mengakses dan melihat riwayat akses yang telah terjadi dengan akurat dan mudah di dalam aplikasi. Hasil pengujian halaman daftar pintu ditunjukkan oleh Tabel 2.44 dibawah.

**Tabel 2.44** Hasil pengujian riwayat akses

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Pengujian** | **Bentuk Pengujian** | **Hasil Pengujian** | **Status** |
| 1. | Membuka menu riwayat akses | Melakukan akses dengan menekan menu riwayat akses | Halaman riwayat akses berhasil ditampilkan | Berhasil |
| 2. | Menampilkan Riwayat akses yang terdaftar dalam aplikasi | Melakukan akses untuk melihat halaman riwayat akses | Halaman riwayat akses berhasil ditampilkan | Berhasil |
| 3. | Menampilkan halaman riwayat akses ketika belum adilakukan akses ke pintu. | Melakukan akses untuk melihat halaman riwayat akses | Halaman riwayat akses tidak ditemukan | Gagal |

Dapat dilihat pada Tabel 2.44 diatas bahwa pengguna berhasil menampilkan halaman riwayat akses dengan menekan menu riwayat akses. Riwayat akses akan ditampilkan apabila sudah terdaftar dalam aplikasi. Apabila belum terdaftar maka riwayat akses tidak dapat ditampilkan.

1. Utilitas

Pengujian fungsi halaman utilitas pada pengujian aplikasi Android memiliki tujuan untuk memastikan bahwa pengguna dapat menggunakan berbagai utilitas atau alat bantu untuk mengatur kunci Iot, pengaturan wifi dan melakukan register ke server. Hasil pengujian halaman daftar pintu ditunjukkan oleh Tabel 2.45 dibawah.

**Tabel 2.45** Hasil pengujian utilitas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Pengujian** | **Bentuk Pengujian** | **Hasil Pengujian** | **Status** |
| 1. | Membuka menu utilitas | Melakukan akses dengan menekan menu utilitas | Halaman utilitas berhasil ditampilkan | Berhasil |
| 2. | Melakukan pengaturan wifi | Mengisi SSID dan *password* dengan benar | Pengaturan wifi terkonfirmasi dan terkirim | Berhasil |
| 3. | Melakukan pengaturan wifi dengan SSID dan *password* salah | Melakukan akses pengaturan dengan memasukkan SSID dan *Password* yang salah | Tampil peringatan bahwa perangkat kunci kosong |  |
| 4. | Melakukan register ke *server* | Menekan tombol Pindai QR Pendaftaran | Pindai QR pendafataran terkonfirmasi | Berhasil |

Dapat dilihat pada Tabel 2.45 diatas bahwa pengguna berhasil menampilkan halaman utilitas akun dengan menekan menu utilitas. Halaman utilitas akun berfungsi untuk melakukan akses pengaturan dengan memasukkan SSID dan *password.* Pengaturan wifi berhasil apabila SSID dan *password* yang dimasukkan sesuai.

1. Update profile

Pengujian fungsi halaman *update profil*e pada pengujian aplikasi Android adalah untuk memverifikasi bahwa pengguna dapat mengubah dan memperbarui informasi dalam profil mereka dengan akurat dan mudah. Hasil pengujian halaman daftar pintu ditunjukkan oleh Tabel 2.46 dibawah.

**Tabel 2.46** Pengujian Fungsi Update Profile

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Pengujian** | **Bentuk Pengujian** | **Hasil Pengujian** | **Status** |
| 1. | Membuka halaman update profil | Melakukan akses dengan menekan tombol *update profile* | Berhasil masuk ke halaman *update profile* | Berhasil |
| 2. | Memasukkan *email* dengan benar | Memasukkan alamat *email* ke formulir dengan format yang benar | Berhasil memperbarui *email* | Berhasil |
| 3. | Memasukkan *email* dengan salah | Memasukkan alamat *email* ke formulir dengan format yan gsalah | *Email* gagal diperbarui dan terdapat peringatan ada yang salah dari sistem | Gagal |
| 4. | Mengganti nama pengguna menjadi nama yang baru | Memasukkan nama pengguna dengan nama baru yang berbeda | Nama berhasil diperbarui menjadi nama baru | berhasil |
| 5. | Mengganti jenis kelamin pengguna | Mengganti jenis kelamin pengguna dengan memilih jenis kelaamin yang berbeda | Jeneis kelamin berhasi terganti dengan. | Berhasil |
| 6. | Mengganti nomor HP pengguna dengan nomor yang salah | Mengganti nomor HP pengguna dengan nomor yang salah dan belom terdaftar | Nomor HP pengguna tidak dapat diperbarui dan terdapat peringatan ada yang salah dari sistem | Gagal |
| 7. | Mengganti nomor HP pengguna dengan nomor yang benar | Mengganti nomor HP pengguna dengan nomor benar dan sudah terdaftar | Nomor HP pengguna berhasil diperbarui | Berhasil |

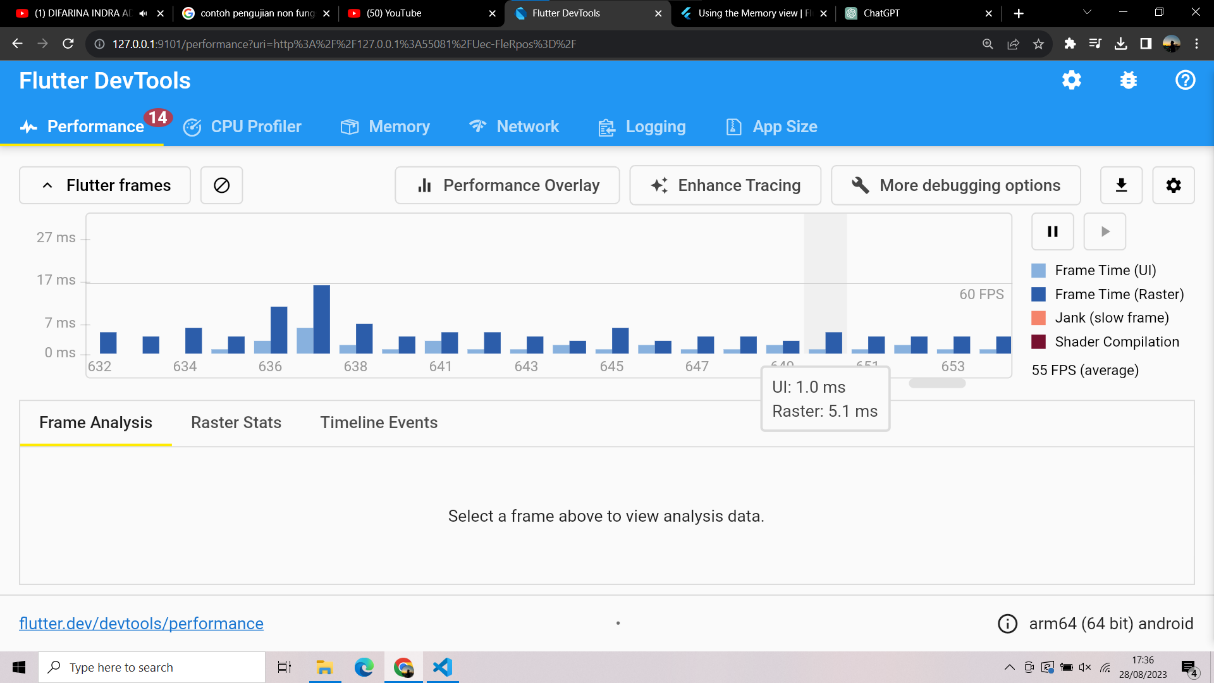
Dapat dilihat pada Tabel 2.46 diatas bahwa pengguna berhasil menampilkan halaman *update profil*e dengan menekan menu *update profil*e. Halaman profil berisi menu untuk mengganti *email*, nama pengguna*,* jenis kelamin*,* mengganti nomor HP*.* Beberapa menu tersebut berhasil diganti apabila menggunakan format dan data yang sesuai.

1. Pengujian Performa Aplikasi Mobile

Pengujian ini berfokus pada performa tampilan, alokasi memori, dan konektivitas internet dalam aplikasi mobile yang memiliki peran sentral dalam memastikan responsivitas antarmuka, manajemen memori yang efisien, serta performa yang konsisten dalam beragam kondisi jaringan. Hasil dari proses pengujian performa aplikasi mobile adalah sebagai berikut:

1. Performa tampilan

Pengujian performa tampilan pada aplikasi Android adalah proses untuk memastikan bahwa aplikasi Android dapat berjalan dengan lancar, responsif, dan memberikan pengalaman pengguna yang baik. Hasil dari pengujian performa tampilan aplikasi android dapat dilihat pada Gambar 2.15 dibawah.

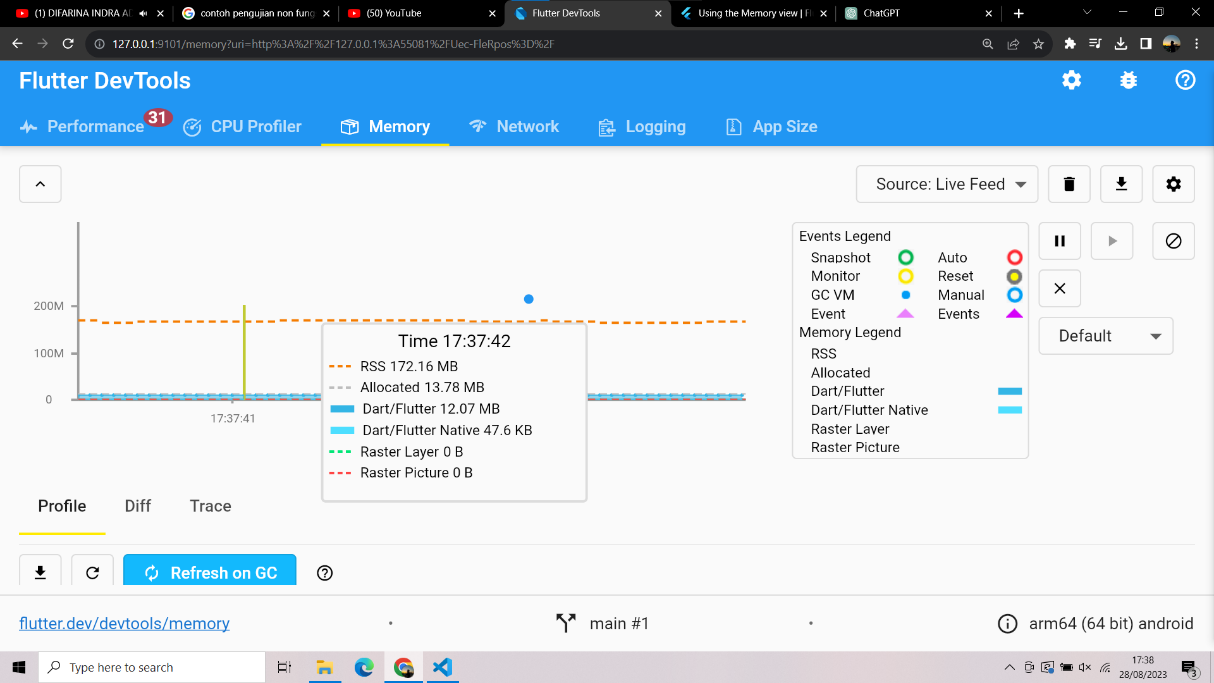


Gambar 2. Performa tampilan aplikasi

Gambar 2.15 diatas menampilkan hasil pengujian performa tampilan untuk mengukur kinerja aplikasi yang mengacu pada waktu yang dibutuhkan untuk memproses dan merender elemen-elemen antarmuka pengguna (UI) yang berupa *frame time* (UI) dan juga *frame time* (*raster*) untuk mengukur waktu yang diperlukan untuk merender elemen grafis menjadi gambar *raster* yang siap ditampilkan di *layer*. Selain itu juga menampilkan nilai rata-rata FPS (*Frame Per Second*) yang merupakan jumlah rata-rata *frame* yang ditampilkan per detik dalam suatu apllikasi. Pada gambar pengukuran tersebut ditampilkan *frame* *time* (UI) dengan rata-rata waktu 3 milisecond, *frame* *time* (*raster*) dengan rata-rata waktu 5 milisecond dan untuk nilai rata-rata FPS berkisar pada 55 FPS.

1. Penggunaan memori

Pengujian performa memori pada aplikasi Android merupakan langkah untuk mengamati dan mengukur bagaimana aplikasi menggunakan memori. Masalah dalam performa memori bisa menghasilkan konsekuensi seperti respons yang lambat, kegagalan (*crash*), atau penutupan aplikasi yang tiba-tiba. Tujuan dari pengujian ini adalah untuk memastikan bahwa aplikasi menggunakan sumber daya memori dengan efisien dan mencegah potensi permasalahan yang mungkin muncul karena penggunaan memori yang berlebihan. Hasil dari pengujian performa memori aplikasi Android dapat dilihat pada Gambar 2.16 dibawah.

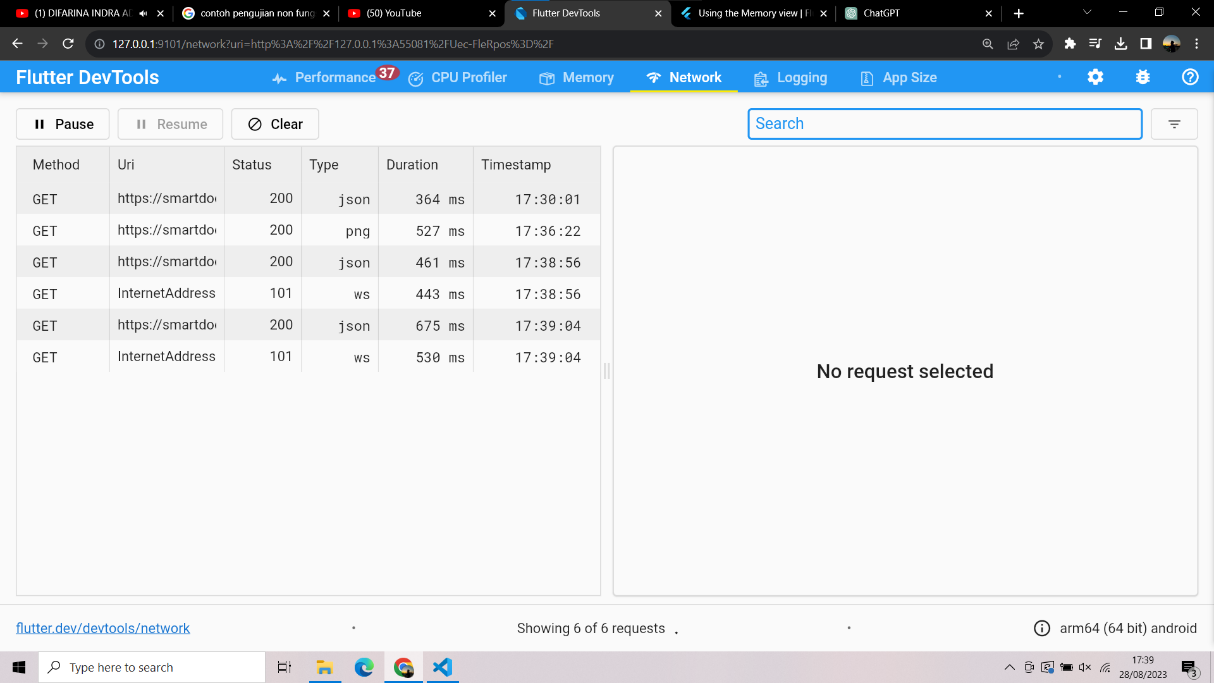


Gambar 2. Penggunaan memori

Gambar 2.16 diatas menampilkan hasil pengujian performa memory aplikasi. Pada pengujian performa tersebut, data yang ditampilkan seperti RSS (*Resident Set Size*), *allocated* (alokasi memori), *dart/flutter*, *dart/flutter native*, *raster layer*, dan *raster picture* menggambarkan berbagai aspek penggunaan memori dalam aplikasi. RSS menunjukkan seberapa banyak memori fisik yang aktif digunakan, sedangkan *allocated* mengukur total alokasi memori termasuk yang aktif dan tidak aktif. *dart/flutter* dan *dart/flutter native* mengacu pada alokasi memori yang terkait dengan bahasa pemrograman dan kerangka kerja. Penggunaan memori dalam *raster layer* dan *raster picture* mengindikasikan penggunaan memori oleh elemen grafis yang ditampilkan. Pada Gambar 4.44 diatas menampilkan nilai dari RSS (*Resident Set Size*) sebesar 172.16 MB, *allocated* (alokasi memori) sebesar 13.78 MB, penggunaan memori yang terkait dengan dart dan *flutter* sebesar 12.7 MB, alokasi memori untuk kode *Dart* yang dijalankan dalam *mode native* sebesar 47.6 KB dan untuk *raster layer* dan *raster picture* sebesar 0 B.

1. Performa internet

Hasil pengujian performa aplikasi Android terkait dengan kinerja internet mencakup sejumlah informasi penting yang menggambarkan bagaimana aplikasi berperilaku saat berinteraksi dengan jaringan. Hasil dari pengujian performa internet aplikasi Android dapat dilihat pada Gambar 2.17 dibawah.



Gambar 2. Performa koneksi internet

Gambar 2.17 diatas menampilkan hasil performa internet saat aplikasi digunakan. Pada gambar 4.45 diatas menampilkan tipe data yang diakses, durasi yang dibutuhkan dan juga waktu saat melakukan akses. Dapat disimpulkan untuk rata-rata durasi waktu yang dibutuhkan untuk mengakses data adalah 500 ms.

1. PENUTUP

Proses pengujian pada produk tugas akhir “Sistem Keamanan Kunci Pintu Gedung Berbasi *Internet of Things*” berhasil dijalankan dengan hasil yang sudah dijelaskan. Proses pengujian memberikan informasi terkait presentase keberhasil yang telah dicapai serta kemampuan dari sistem yang dihasilkan.

Berdasarkan pengujian yang telah dilakukan .... (tolong tambahin lagi)