|  |  |
| --- | --- |
|  | 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 | **6 Februari 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,  7 April 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 Thongs*” proses pengujian yang dilakukan meliputi pengujian secara fungsional dan pengujian untuk mengukur performa sistem. Dokumen ini menjelaskan metode derta alat yang digunakan dalam proses pengujian. Proses pengujian dilakukan untuk melakukan verifikasi dari hasil proses implementasi dengan spesifikasi dan standar 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 dilakuakan.
2. Sebagai penjelasan mengenai metode dan alat yang digunakan dalam proses pengujian.
3. Sebagai acuan keberhasilan sistem sesuai dengan spesifikasi dan standar 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.

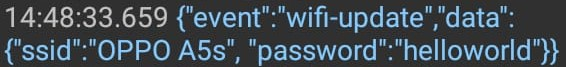


Gambar 2. Status Smartphone Terhubung ke HC-05

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 kunic pintu. Hasil dari proses pengujian pengiriman data dapat dilihat pada Gambar 2.2 dibawah.

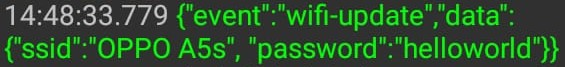


Gambar 2. Pengujian Pengiriman Data dari Smartphone ke Modul Bluetooth

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 |

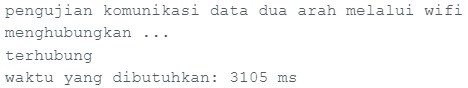
Gambar 2. Grafik Hasil Perhitungan Nilai QoS Komunikasi Bluetooth

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 :

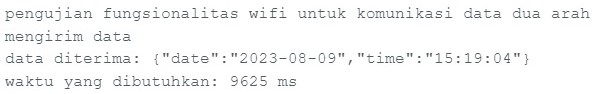
1. Pengujian koneksi ke *router*



Gambar 2. Status Perangkat WiFi (ESP32) 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** |
| 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% |
| 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 |
| 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 |
| 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* | Melakukan aksi membuka pintu dalam waktu lama. | **Tanda Bunyi Pengingat**: *Buzzer* dapat digunakan sebagai pengingat. | Berhasil |
| 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 |
| 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 |
| 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* dengan 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 |
| 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 |
| 3 | Update avatar bukan gambar | Melakukan update avatar menggunakan file salin 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 update avatar dilaksanakan untuk memeriksa kinerja dari fungsi tersebut. Proses update avatar 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 Lihat 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 gaga 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 Pengujian Websocket

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

Pada Tabel 2.23 terlihat bahwa proses *subscribtion* 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 |
| 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.

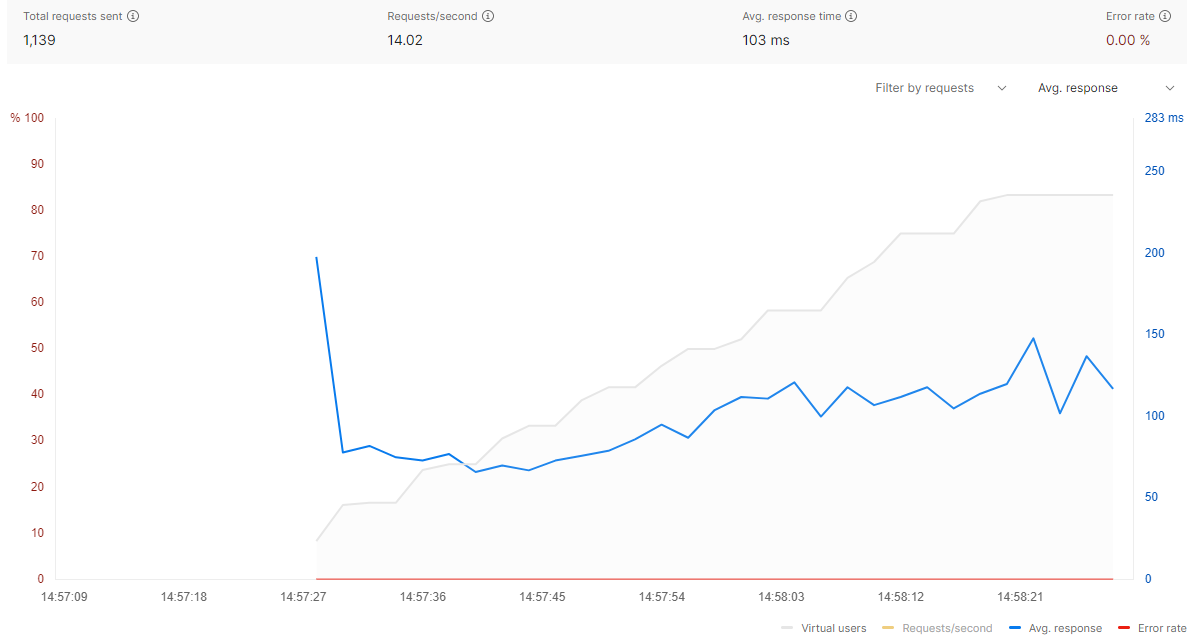
Tabel 2. Hasil Pengujian Fungsi Peringatan Pintu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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 mensimulasikan 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 mensimulasikan beban permintaan verifikasi akses sebanyak 100 pengguna berbeda secara bersamaan. Dari hasil pengujian didapatkan bahwa pada puncak jumlah pengguna sistem membutuhkan waktu rata-rata 1.5 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
2. Pengujian Performa Website
3. Pengujian Fungsionalitas Aplikasi Mobile
4. Pengujian Performa Aplikasi Mobile
5. 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 telash dicapai serta kemampuan dari sistem yang dihasilkan.

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