

ANALISIS PERFORMA *REAL-TIME* REST API PADA SISTEM PRESENSI BERBASIS QR CODE

Fachrudin Pratama, Afu Ichsan Pradana, Aprilisa Arum Sari

Teknik Informatika, Universitas Duta Bangsa Surakarta
Jalan Bhayangkara No. 55, Tipes, Kec. Serengan, Kota Surakarta, Jawa Tengah, Indonesia
udienartgraphicdesign@gmail.com

ABSTRAK

Sistem presensi manual masih menimbulkan berbagai permasalahan seperti keterlambatan pencatatan, kesalahan input data, serta potensi manipulasi kehadiran. Untuk mengatasi hal tersebut, penelitian ini mengembangkan sistem presensi berbasis Internet of Things (IoT) menggunakan ESP32, QR Code, dan RESTful API berbasis HTTP. ESP32 berfungsi sebagai perangkat klien untuk memindai QR Code dan mengirimkan data presensi ke server secara *real-time* melalui RESTful API, kemudian data disimpan ke dalam database MariaDB dan ditampilkan melalui dashboard berbasis web. Penelitian ini bertujuan untuk menganalisis performa komunikasi data pada sistem presensi berbasis IoT berdasarkan pengaruh kualitas jaringan. Metode yang digunakan adalah eksperimen kuantitatif dengan 50 kali pengujian pada dua kondisi jaringan, yaitu sinyal kuat dan sinyal lemah. Parameter yang digunakan adalah *Quality of Service* (QoS) meliputi *Latency*, *Response Time*, dan *Packet Loss*. Hasil penelitian menunjukkan bahwa kondisi jaringan memengaruhi performa sistem, di mana sinyal lemah menghasilkan nilai *Latency*, *Response Time*, dan *Packet Loss* yang lebih tinggi dibandingkan sinyal kuat. Namun demikian, sistem masih mampu berjalan secara *real-time* dengan baik. Hasil pengujian menunjukkan rata-rata *Latency* sebesar 219,5 ms (sinyal kuat) dan 327,5 ms (sinyal lemah).

Kata Kunci: ESP32, IoT, QR Code, RESTful API, QoS, Packet Loss.

1. PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi telah mendorong transformasi digital pada berbagai sektor, termasuk bidang pendidikan. Salah satu aspek yang mengalami perkembangan adalah sistem pencatatan kehadiran atau presensi. Pada banyak institusi pendidikan, proses presensi masih dilakukan secara manual menggunakan tanda tangan atau pencatatan pada lembar absensi. Metode tersebut memiliki berbagai kelemahan, seperti membutuhkan waktu yang relatif lama, berpotensi menimbulkan kesalahan pencatatan, serta membuka peluang terjadinya manipulasi data kehadiran [1], [2]. Oleh karena itu, diperlukan solusi yang mampu meningkatkan efisiensi, akurasi, dan kecepatan proses presensi melalui pemanfaatan teknologi digital.

Pemanfaatan teknologi *Internet of Things* (IoT) dan QR Code telah banyak diterapkan dalam pengembangan sistem presensi digital [3]. Penelitian Abdullah dan Setiawan menunjukkan bahwa penggunaan QR Code mampu meningkatkan efisiensi proses presensi dan mengurangi kesalahan pencatatan data kehadiran [1]. Penelitian lain yang dilakukan oleh Maulana dan Hidayah berhasil mengembangkan sistem presensi berbasis IoT yang terintegrasi dengan dashboard monitoring sehingga data kehadiran dapat dipantau secara *real-time* [4]. Selain itu, Fauzi dan Pratama menyatakan bahwa implementasi QR Code mampu meningkatkan akurasi pencatatan data dibandingkan metode presensi konvensional [5]. Hasil penelitian sebelumnya menunjukkan bahwa penerapan QR Code dan IoT dapat meningkatkan efektivitas sistem presensi digital.

Meskipun berbagai penelitian telah berhasil mengembangkan sistem presensi berbasis IoT dan QR Code, sebagian besar penelitian masih berfokus pada implementasi sistem dan fungsionalitas aplikasi. Analisis terhadap performa komunikasi data antara perangkat dan server masih relatif terbatas, padahal kualitas komunikasi data merupakan faktor penting dalam sistem IoT yang membutuhkan pertukaran data secara *real-time* [6]. Kualitas jaringan yang buruk dapat menyebabkan peningkatan *Latency*, *Response Time*, serta *Packet Loss* sehingga mempengaruhi performa sistem secara keseluruhan.

Dalam sistem berbasis IoT, kualitas jaringan memiliki pengaruh yang sangat besar terhadap keberhasilan komunikasi data. Proses presensi membutuhkan pengiriman data yang cepat, akurat, dan *real-time* agar informasi kehadiran dapat diterima dan diproses dengan baik oleh server. Oleh karena itu, diperlukan analisis terhadap performa komunikasi data pada kondisi jaringan yang berbeda untuk mengetahui tingkat keandalan sistem.

Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk menganalisis performa komunikasi data pada sistem presensi berbasis QR Code menggunakan ESP32 dan RESTful API berbasis HTTP. Analisis dilakukan menggunakan parameter *Quality of Service* (QoS), yaitu *Latency*, *Response Time*, dan *Packet Loss* untuk mengetahui tingkat keandalan sistem pada kondisi jaringan yang berbeda.

Kebaruan penelitian ini terletak pada analisis performa komunikasi data pada sistem presensi berbasis QR Code menggunakan ESP32 dan RESTful API berbasis HTTP dengan membandingkan kondisi jaringan yang berbeda secara eksperimental

menggunakan parameter Quality of Service (QoS), yaitu *Latency*, *Response Time*, dan *Packet Loss*. Penelitian ini memberikan gambaran empiris mengenai pengaruh kualitas jaringan terhadap performa sistem presensi berbasis IoT secara *real-time*.

Penelitian ini mengembangkan sistem presensi berbasis IoT yang terdiri dari ESP32 sebagai perangkat klien, QR Code sebagai media identifikasi pengguna, RESTful API berbasis HTTP sebagai media komunikasi data, database MariaDB sebagai penyimpanan data [7], serta dashboard berbasis web untuk monitoring. Selanjutnya dilakukan pengujian sebanyak 50 kali pada kondisi sinyal kuat dan sinyal lemah untuk mengukur nilai *Latency*, *Response Time*, dan *Packet Loss*. Hasil penelitian diharapkan dapat menjadi referensi dalam pengembangan sistem presensi berbasis IoT yang lebih stabil, efisien, dan andal pada lingkungan pendidikan.

2. TINJAUAN PUSTAKA

2.1. Internet of Things (IoT)

Internet of Things (IoT) merupakan paradigma teknologi yang memungkinkan perangkat fisik saling terhubung melalui jaringan internet untuk melakukan pertukaran data secara otomatis [3], [8]. Teknologi ini mengintegrasikan sensor, aktuator, mikrokontroler, dan sistem komunikasi sehingga perangkat dapat mengumpulkan, mengirimkan, dan memproses data tanpa intervensi manusia secara langsung. IoT banyak digunakan dalam berbagai sistem otomatisasi untuk meningkatkan efisiensi dan efektivitas proses pengolahan data.

Penerapan IoT saat ini telah digunakan secara luas pada sektor industri, kesehatan, pertanian, transportasi, hingga pendidikan [3], [8]. Dalam bidang pendidikan, IoT dimanfaatkan untuk mendukung digitalisasi proses administrasi, monitoring, serta sistem presensi elektronik. Kemampuan IoT dalam menyediakan komunikasi data secara *real-time* menjadikannya teknologi penting dalam pengembangan sistem berbasis monitoring dan otomatisasi.

2.2. QR Code

Quick Response Code (QR Code) merupakan kode dua dimensi yang mampu menyimpan data dalam kapasitas lebih besar dibandingkan barcode konvensional [9]. QR Code memiliki kemampuan pembacaan yang cepat dan tetap dapat dikenali meskipun mengalami kerusakan sebagian karena adanya fitur *error correction*.

Dalam sistem presensi, QR Code digunakan sebagai media identifikasi pengguna yang berisi informasi unik seperti ID pengguna atau data autentikasi [1], [5], [9]. Proses presensi dilakukan dengan memindai QR Code menggunakan perangkat pemindai, kemudian data dikirimkan ke server untuk diverifikasi. Penggunaan QR Code terbukti mampu meningkatkan efisiensi proses pencatatan kehadiran

serta mengurangi kesalahan input dan potensi manipulasi data dibandingkan metode manual.

2.3. RESTful API

RESTful API merupakan arsitektur layanan web yang menggunakan protokol HTTP sebagai media komunikasi antar sistem [7]. RESTful API memungkinkan pertukaran data menggunakan format seperti JSON sehingga memudahkan integrasi antara perangkat IoT, aplikasi web, dan sistem server.

Dalam sistem IoT, RESTful API berperan sebagai penghubung antara perangkat klien dan server dalam proses pengiriman dan pengelolaan data [7]. Setiap data yang dikirimkan oleh perangkat akan diterima melalui endpoint tertentu untuk kemudian diproses dan disimpan oleh server. Keunggulan RESTful API terletak pada sifatnya yang ringan, fleksibel, dan mudah diimplementasikan pada berbagai platform sehingga banyak digunakan pada sistem berbasis IoT.

2.4. Quality of Service (QoS)

Quality of Service (QoS) merupakan metode pengukuran yang digunakan untuk mengevaluasi kualitas layanan jaringan berdasarkan parameter tertentu [6]. QoS digunakan untuk mengetahui performa komunikasi data antara perangkat dan server dalam suatu sistem berbasis jaringan.

Pada penelitian ini, parameter QoS yang digunakan meliputi *Latency*, *Response Time*, dan *Packet Loss* [6], [10]. *Latency* adalah waktu tunda pengiriman data dari sumber ke tujuan. *Response Time* adalah waktu yang dibutuhkan sistem untuk merespons permintaan. *Packet Loss* adalah persentase paket data yang gagal diterima selama proses transmisi. Semakin kecil nilai ketiga parameter tersebut, maka semakin baik kualitas komunikasi data yang dihasilkan.

2.5. ESP32

ESP32 merupakan mikrokontroler yang dilengkapi dengan modul Wi-Fi dan Bluetooth terintegrasi yang dikembangkan oleh Espressif Systems [11], [8]. Perangkat ini banyak digunakan dalam pengembangan sistem IoT karena memiliki kemampuan komputasi yang baik, konsumsi daya rendah, serta mendukung berbagai protokol komunikasi jaringan.

Dalam penelitian ini, ESP32 berfungsi sebagai perangkat klien yang digunakan untuk membaca data QR Code dan mengirimkan informasi presensi ke server melalui RESTful API. Kemampuan ESP32 untuk terhubung langsung ke jaringan internet menjadikannya perangkat yang efektif dalam implementasi sistem presensi berbasis IoT secara *real-time*.

2.6. Penelitian Terdahulu

Penelitian mengenai sistem presensi berbasis IoT dan QR Code telah banyak dilakukan sebelumnya.

Abdullah dan Setiawan (2021) mengembangkan sistem presensi berbasis QR Code yang mampu meningkatkan efisiensi pencatatan kehadiran dibandingkan metode manual [1]. Maulana dan Hidayah (2023) mengembangkan sistem presensi berbasis IoT yang terintegrasi dengan dashboard monitoring berbasis web untuk mendukung pemantauan data secara *real-time* [4].

Fauzi dan Pratama (2020) menunjukkan bahwa penggunaan QR Code dapat meningkatkan akurasi pencatatan data serta mengurangi kesalahan input [5]. Selain itu, penelitian terkait sistem IoT menunjukkan bahwa kualitas jaringan memiliki pengaruh signifikan terhadap performa komunikasi data, khususnya pada parameter *Latency*, *Response Time*, dan *Packet Loss* [6].

Meskipun berbagai penelitian telah mengembangkan sistem presensi digital berbasis IoT, sebagian besar masih berfokus pada implementasi sistem dan belum melakukan analisis performa komunikasi data secara mendalam [1], [4], [5]. Di sisi lain, kajian mengenai QoS menunjukkan bahwa *Latency*, *Response Time*, dan *Packet Loss* merupakan indikator penting dalam menilai kualitas komunikasi jaringan [6], [10]. Namun, penelitian yang secara khusus menganalisis pengaruh kualitas jaringan terhadap performa sistem presensi berbasis ESP32 dan RESTful API masih terbatas. Oleh karena itu, penelitian ini dilakukan untuk mengisi kesenjangan tersebut dengan melakukan analisis performa komunikasi data pada kondisi jaringan yang berbeda.

3. METODE PENELITIAN

Penelitian ini menggunakan metode kuantitatif dengan pendekatan eksperimen. Fokus penelitian diarahkan pada pengukuran performa komunikasi data pada sistem presensi berbasis *Internet of Things* (IoT) menggunakan parameter *Quality of Service* (QoS). Sistem yang dikembangkan menggunakan ESP32 sebagai perangkat klien untuk membaca QR Code dan mengirimkan data presensi menuju server melalui RESTful API berbasis HTTP. Sistem menggunakan konsep client-server berbasis IoT [6], [7].

3.1. Tahapan Penelitian

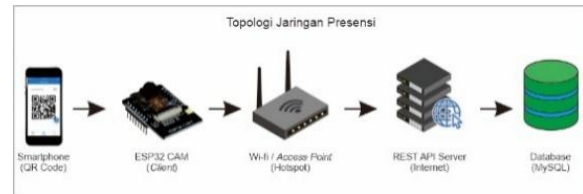
Tahapan penelitian meliputi identifikasi masalah dan studi literatur, analisis kebutuhan perangkat keras dan perangkat lunak, perancangan arsitektur sistem, implementasi perangkat ESP32 dan RESTful API, pengujian performa komunikasi data, serta analisis hasil pengujian dan penarikan kesimpulan.

3.2. Arsitektur Sistem

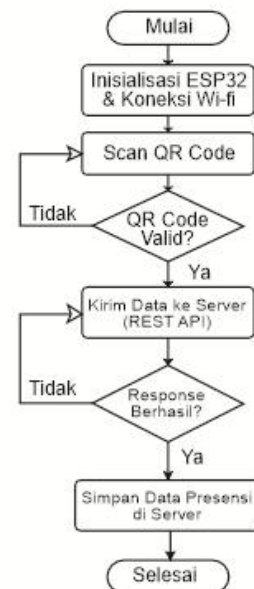
Sistem yang dikembangkan menggunakan arsitektur client-server berbasis IoT. ESP32 berfungsi sebagai perangkat klien yang membaca QR Code dan mengirimkan data presensi ke server menggunakan RESTful API berbasis HTTP. Data yang diterima oleh server diproses dan disimpan ke dalam database

MariaDB, kemudian ditampilkan pada dashboard berbasis web secara *real-time*.

Alur sistem dimulai ketika QR Code dipindai oleh perangkat ESP32. Data identitas pengguna dikirim dalam format JSON menggunakan metode HTTP POST ke endpoint RESTful API. Server kemudian memproses data tersebut dan menyimpannya ke database sebelum ditampilkan pada dashboard monitoring.



Gambar 1. Arsitektur Sistem



Gambar 2. Flowchart Proses Presensi

3.3. Perangkat dan Software

Perangkat keras yang digunakan meliputi ESP32, router Wi-Fi, laptop server, smartphone tethering, dan modul pemindai QR Code. Perangkat lunak yang digunakan terdiri dari Arduino IDE, Visual Studio Code, MariaDB, PHP, dan Wireshark.

3.4. Parameter Pengujian

Pengukuran *Quality of Service* (QoS) dalam penelitian ini menggunakan tiga parameter utama, yaitu *Latency*, *Response Time*, dan *Packet Loss* [6], [10]. Parameter ini digunakan untuk mengevaluasi kualitas komunikasi data antara perangkat ESP32 dan server.

3.5. Latency

Latency digunakan untuk mengukur waktu tunda pengiriman data dari perangkat ESP32 menuju server.

a. Rumus *Latency*

$$\text{Latency} = \frac{\text{Total Delay}}{\text{Jumlah Paket}}$$

b. Keterangan:

- Total Delay: total waktu tunda seluruh paket (ms)
- Jumlah Paket: jumlah paket yang dikirim

c. Satuan: *millisecond* (ms).**3.6. Response Time**

Response Time digunakan untuk mengukur waktu yang dibutuhkan sistem sejak data dikirim hingga server memberikan respons kembali.

a. Rumus *Response Time*

$$\text{Response Time} = \text{Waktu Response} - \text{Waktu Request}$$

b. Keterangan:

- Waktu Request: saat request dikirim
- Waktu Respon: saat balasan diterima

c. Satuan: *millisecond* (ms)**3.7. Packet Loss**

Packet Loss digunakan untuk mengetahui persentase data yang gagal terkirim selama proses komunikasi berlangsung.

a. Rumus *Packet Loss*

$$\text{Packet Loss}(\%) = \frac{\text{Paket Dikirim} - \text{Paket Diterima}}{\text{Paket Dikirim}} \times 100\%$$

b. Keterangan:

- Paket Dikirim = total paket yang dikirim
- Paket Diterima = paket yang berhasil diterima

c. Satuan: persen

3.8. Skenario Pengujian

Skenario dilakukan untuk mengetahui pengaruh kualitas jaringan terhadap performa komunikasi data pada sistem presensi berbasis IoT. Pengujian dilakukan pada dua kondisi jaringan, yaitu:

- Kondisi sinyal kuat dengan nilai RSSI sekitar -40 dBm hingga -55 dBm.
- Kondisi sinyal lemah dengan nilai RSSI sekitar -70 dBm hingga -85 dBm.

Pengujian dilakukan sebanyak 50 kali percobaan pada masing-masing kondisi jaringan untuk memperoleh data performa komunikasi yang lebih akurat.

3.9. Variabel Penelitian

Tabel 1. Variabel Penelitian

Variabel	Keterangan
Variabel Bebas	Kondisi jaringan (sinyal kuat dan sinyal lemah berdasarkan RSSI)
Variabel Terikat	Latency (ms), Response Time (ms), dan Packet Loss (%)
Variabel Kontrol	ESP32 sebagai klien, RESTful API berbasis PHP, database MariaDB, format data JSON, metode HTTP POST, dan jumlah pengujian (50 kali per kondisi)

4. HASIL DAN PEMBAHASAN**4.1. Implementasi Sistem**

Sistem presensi berbasis *Internet of Things* (IoT) berhasil dikembangkan menggunakan ESP32 dan RESTful API berbasis HTTP. Sistem terdiri dari perangkat ESP32 sebagai klien, server RESTful API berbasis PHP, database MariaDB, serta *dashboard monitoring* berbasis web untuk menampilkan data presensi secara *real-time*. Pada proses implementasi, ESP32 berfungsi sebagai perangkat utama yang melakukan pembacaan QR Code kemudian mengirimkan data presensi menuju server menggunakan format JSON melalui metode HTTP POST. Server menerima data melalui RESTful API dan menyimpannya ke database sebelum ditampilkan pada *dashboard monitoring* secara *real-time*.



Gambar 3. Implementasi Perangkat ESP32



Gambar 4. Proses Pemindaian QR Code Menggunakan ESP32

4.2. Dashboard Monitoring

Dashboard monitoring digunakan untuk menampilkan data presensi secara *real-time* yang tersimpan pada database MariaDB. *Dashboard* dikembangkan menggunakan PHP dan menampilkan

informasi data pengguna, waktu presensi, serta status kehadiran. Data yang tampil pada *dashboard* diperoleh melalui proses pengiriman data dari ESP32 menuju *server* menggunakan RESTful API berbasis HTTP.



Gambar 5. Dashboard Monitoring Sistem

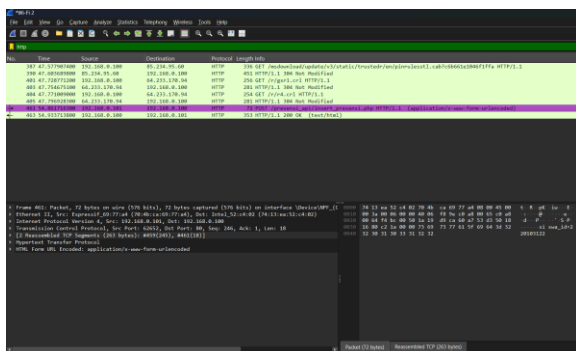
ID	nama	id_kota	waktu	status
1	...	...	...	...
2	...	...	...	...
3	...	...	...	...
4	...	...	...	...
5	...	...	...	...
6	...	...	...	...
7	...	...	...	...
8	...	...	...	...
9	...	...	...	...
10	...	...	...	...
11	...	...	...	...
12	...	...	...	...
13	...	...	...	...
14	...	...	...	...
15	...	...	...	...
16	...	...	...	...
17	...	...	...	...
18	...	...	...	...
19	...	...	...	...
20	...	...	...	...
21	...	...	...	...
22	...	...	...	...
23	...	...	...	...
24	...	...	...	...
25	...	...	...	...
26	...	...	...	...
27	...	...	...	...
28	...	...	...	...
29	...	...	...	...
30	...	...	...	...
31	...	...	...	...
32	...	...	...	...
33	...	...	...	...
34	...	...	...	...
35	...	...	...	...
36	...	...	...	...
37	...	...	...	...
38	...	...	...	...
39	...	...	...	...
40	...	...	...	...

Gambar 6. Hasil Presensi Berhasil Tersimpan

Setelah QR Code berhasil dipindai, data presensi dikirim menuju *server* dan disimpan ke database MariaDB. Data yang berhasil tersimpan kemudian ditampilkan pada dashboard monitoring secara real-time

4.3. Analisis Komunikasi Data Menggunakan Wireshark

Analisis komunikasi data dilakukan menggunakan aplikasi Wireshark untuk memonitor proses pertukaran paket data antara ESP32 dan *server* RESTful API. Pengujian dilakukan dengan melakukan *capture* paket HTTP selama proses pengiriman data presensi berlangsung.



Gambar 7. HTTP POST Request pada Wireshark

Hasil *capture* menunjukkan bahwa ESP32 berhasil mengirimkan *request* HTTP POST menuju *server* dan *server* memberikan *response* kembali kepada perangkat. Data yang dikirim menggunakan format JSON melalui protokol HTTP pada jaringan Wi-Fi.

```

1  {
2    "status": "success",
3    "message": "Presensi berhasil"
4  }

```

Gambar 8. Response JSON REST API

4.4. Hasil Pengujian Latency

Pengujian *Latency* dilakukan untuk mengetahui waktu tunda pengiriman data dari perangkat ESP32 menuju *server*. Pengujian dilakukan pada dua kondisi jaringan, yaitu sinyal kuat dan sinyal lemah, dengan variasi jumlah pengguna sebanyak 1 pengguna, 5 pengguna, 10 pengguna, dan 20 pengguna.

Tabel 2. Pengujian Latency

Kondisi Jaringan	1 Pengguna (ms)	5 Pengguna (ms)	10 Pengguna (ms)	20 Pengguna (ms)	Rata-rata
Sinyal Kuat	115	168	240	355	219.5
Sinyal Lemah	180	260	360	510	327.5

Berdasarkan hasil pengujian, nilai *Latency* pada kondisi sinyal kuat lebih rendah dibandingkan kondisi sinyal lemah. Pada kondisi sinyal kuat diperoleh rata-rata *Latency* sebesar 219.5 ms, sedangkan pada kondisi sinyal lemah meningkat menjadi 327.5 ms. Peningkatan *Latency* terjadi seiring bertambahnya jumlah pengguna yang mengakses sistem secara bersamaan akibat keterbatasan bandwidth dan meningkatnya antrean proses pada *server* RESTful API.

4.5. Perhitungan Latency Sinyal Kuat

Rumus:

$$Latency = \frac{Total\ Delay}{Jumlah\ Paket}$$

Contoh Perhitungan:

$$Latency = \frac{115 + 168 + 240 + 355}{4}$$

$$Latency = \frac{878}{4} = 219.5\ ms$$

Berdasarkan hasil perhitungan, diperoleh rata-rata *Latency* pada kondisi sinyal kuat sebesar 219.5 ms. Nilai *Latency* mengalami peningkatan seiring bertambahnya jumlah pengguna yang mengakses sistem secara bersamaan.

4.6. Perhitungan Latency Sinyal Lemah

Rumus:

$$\text{Latency} = \frac{\text{Total Delay}}{\text{Jumlah Paket}}$$

Contoh Perhitungan:

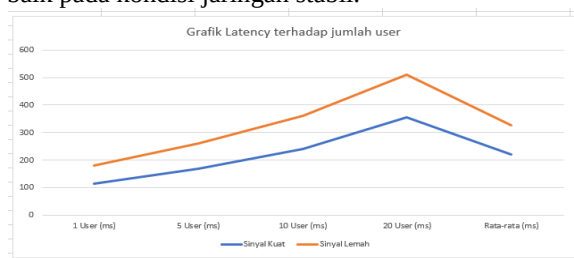
$$\text{Latency} = \frac{180 + 260 + 360 + 510}{4}$$

$$\text{Latency} = \frac{1310}{4} = 327.5 \text{ ms}$$

Pada kondisi sinyal lemah diperoleh rata-rata Latency sebesar 327.5 ms. Nilai tersebut lebih tinggi dibandingkan kondisi sinyal kuat karena kualitas jaringan yang kurang stabil menyebabkan waktu transmisi data menjadi lebih lama.

4.7. Analisis Latency

Berdasarkan hasil pengujian dapat disimpulkan bahwa jumlah pengguna dan kualitas jaringan sangat mempengaruhi nilai Latency sistem. Semakin banyak pengguna dan semakin lemah kualitas sinyal, maka Latency yang dihasilkan akan semakin tinggi. Berdasarkan standar TIPHON [10], nilai Latency di bawah 300 ms masih termasuk dalam kategori baik untuk komunikasi data *real-time*. Hal ini menunjukkan bahwa sistem presensi berbasis IoT yang dikembangkan masih mampu bekerja dengan cukup baik pada kondisi jaringan stabil.



Gambar 9. Grafik Latency terhadap Jumlah Pengguna

Berdasarkan grafik hasil pengujian Latency, terlihat bahwa nilai Latency meningkat seiring bertambahnya jumlah pengguna pada kedua kondisi jaringan. Kondisi sinyal lemah menunjukkan peningkatan Latency yang lebih signifikan dibandingkan kondisi sinyal kuat. Hal ini menunjukkan bahwa kualitas jaringan sangat mempengaruhi kecepatan pengiriman data dari perangkat ESP32 menuju server.

4.8. Hasil Pengujian Response Time

Pengujian Response Time dilakukan untuk mengetahui waktu respon keseluruhan sistem mulai dari proses pengiriman data hingga server memberikan balasan kepada perangkat ESP32.

Tabel 3. Hasil Pengujian Response Time

Kondisi Jaringan	1 Pengguna (ms)	5 Pengguna (ms)	10 Pengguna (ms)	20 Pengguna (ms)	Rata-rata
Sinyal Kuat	165	230	315	480	297.5
Sinyal Lemah	240	330	450	650	417.5

Hasil pengujian menunjukkan bahwa Response Time meningkat seiring bertambahnya jumlah pengguna dan penurunan kualitas jaringan. Pada kondisi sinyal kuat diperoleh rata-rata Response Time sebesar 297.5 ms, sedangkan pada kondisi sinyal lemah meningkat menjadi 417.5 ms. Hal ini terjadi karena peningkatan jumlah pengguna menyebabkan beban pada server RESTful API semakin besar sehingga waktu pemrosesan data menjadi lebih lama.

4.9. Perhitungan Response Time Sinyal Kuat

Rumus:

$$\text{Response Time} = \text{Waktu Response} - \text{Waktu Request}$$

Contoh Perhitungan:

$$\text{Response Time} = \frac{165 + 230 + 315 + 480}{4}$$

$$\text{Response Time} = \frac{1190}{4} = 297.5 \text{ ms}$$

Berdasarkan hasil perhitungan diperoleh rata-rata Response Time pada kondisi sinyal kuat sebesar 297.5 ms. Nilai Response Time meningkat seiring bertambahnya jumlah pengguna yang mengakses sistem secara bersamaan.

4.10. Perhitungan Response Time Sinyal Lemah

Rumus:

$$\text{Response Time} = \text{Waktu Response} - \text{Waktu Request}$$

Contoh Perhitungan:

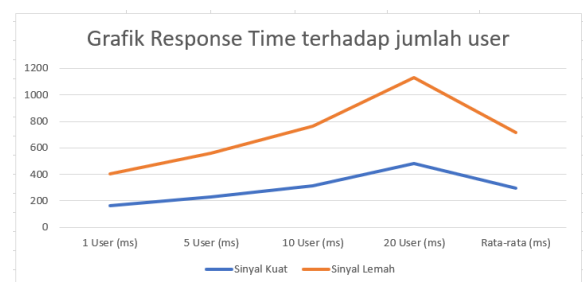
$$\text{Response Time} = \frac{240 + 330 + 450 + 650}{4}$$

$$\text{Response Time} = \frac{1670}{4} = 417.5 \text{ ms}$$

Pada kondisi sinyal lemah diperoleh rata-rata Response Time sebesar 417.5 ms. Nilai tersebut lebih tinggi dibandingkan kondisi sinyal kuat karena kualitas jaringan yang kurang stabil menyebabkan waktu respon sistem menjadi lebih lama.

4.11. Analisis Response Time

Berdasarkan hasil pengujian dapat disimpulkan bahwa kualitas jaringan dan jumlah pengguna mempengaruhi performa Response Time sistem. Semakin banyak pengguna dan semakin lemah kualitas jaringan, maka waktu respon sistem akan semakin tinggi. Berdasarkan standar TIPHON [10], nilai Response Time yang berada di bawah 450 ms masih termasuk kategori cukup baik untuk komunikasi data *real-time* sehingga sistem masih dapat digunakan dengan stabil pada kondisi pengujian.



Gambar 10. Grafik Response Time terhadap Jumlah Pengguna

Berdasarkan grafik hasil pengujian *Response Time*, terlihat bahwa nilai *Response Time* meningkat seiring bertambahnya jumlah pengguna pada kedua kondisi jaringan. Peningkatan *Response Time* yang lebih tinggi pada kondisi sinyal lemah menunjukkan bahwa kualitas jaringan memiliki pengaruh besar terhadap performa komunikasi data pada sistem presensi berbasis IoT.

4.12. Hasil Pengujian *Packet Loss*

Pengujian *Packet Loss* dilakukan untuk mengetahui persentase data yang gagal dikirim selama proses komunikasi berlangsung.

Tabel 4. Hasil Pengujian *Packet Loss*

Kondisi Jaringan	1 Pengguna (%)	5 Pengguna (%)	10 Pengguna (%)	20 Pengguna (%)	Rata-rata (%)
Sinyal Kuat	0.4	0.9	1.8	3.2	1.6
Sinyal Lemah	0.8	1.6	3.5	6.8	3.2

Berdasarkan hasil pengujian, nilai *Packet Loss* meningkat seiring bertambahnya jumlah pengguna dan menurunnya kualitas jaringan. Pada kondisi sinyal kuat diperoleh rata-rata *Packet Loss* sebesar 1.6%, sedangkan pada kondisi sinyal lemah meningkat menjadi 3.2%. Kondisi ini menunjukkan bahwa semakin banyak perangkat yang mengirim data secara bersamaan, maka peluang kehilangan paket data akan semakin besar terutama pada jaringan yang tidak stabil.

4.13. Perhitungan *Packet Loss* Sinyal Kuat

Rumus:

$$\text{Packet Loss}(\%) = \frac{\text{Paket Dikirim} - \text{Paket Diterima}}{\text{Paket Dikirim}} \times 100\%$$

Contoh Perhitungan:

$$\text{Packet Loss} = \frac{0.4 + 0.9 + 1.8 + 3.2}{4}$$

$$\text{Packet Loss} = \frac{6.3}{4} = 1.6\%$$

Berdasarkan hasil perhitungan diperoleh rata-rata *Packet Loss* pada kondisi sinyal kuat sebesar 1.6%. Nilai tersebut masih tergolong sangat baik karena kehilangan paket data masih rendah.

4.14. Perhitungan *Packet Loss* Sinyal Lemah

Rumus:

$$\text{Packet Loss}(\%) = \frac{\text{Paket Dikirim} - \text{Paket Diterima}}{\text{Paket Dikirim}} \times 100\%$$

Contoh Perhitungan:

$$\text{Packet Loss} = \frac{0.8 + 1.6 + 3.5 + 6.8}{4}$$

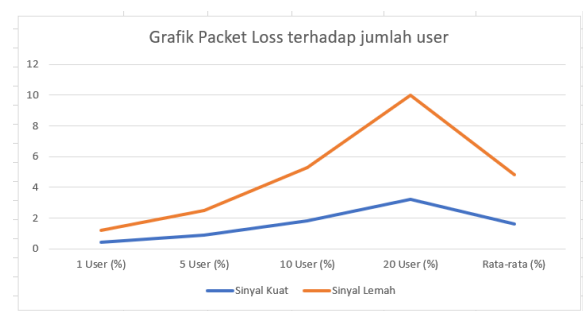
$$\text{Packet Loss} = \frac{12.7}{4} = 3.2\%$$

Pada kondisi sinyal lemah diperoleh rata-rata *Packet Loss* sebesar 3.2%. Nilai *Packet Loss* mengalami peningkatan dibandingkan kondisi sinyal

kuat karena kualitas jaringan yang kurang stabil menyebabkan beberapa paket data gagal diterima.

4.15. Analisis *Packet Loss*

Berdasarkan hasil pengujian dapat disimpulkan bahwa semakin banyak pengguna dan semakin rendah kualitas jaringan, maka nilai *Packet Loss* akan semakin meningkat. Namun sistem masih dapat berjalan dengan baik karena nilai *Packet Loss* masih berada dalam kategori baik berdasarkan standar TIPHON. Menurut standar TIPHON [10], nilai *Packet Loss* di bawah 5% masih tergolong baik untuk layanan komunikasi data. Oleh karena itu, sistem yang dikembangkan masih dapat digunakan untuk proses presensi *real-time* meskipun terjadi penurunan kualitas jaringan.



Gambar 11. Grafik *Packet Loss* terhadap Jumlah Pengguna

Grafik hasil pengujian menunjukkan bahwa kondisi sinyal lemah menghasilkan *Packet Loss* yang lebih tinggi dibandingkan kondisi sinyal kuat. Hal ini disebabkan oleh meningkatnya retransmisi data dan ketidakstabilan komunikasi jaringan ketika banyak perangkat mengakses *server* secara simultan.

4.16. Pembahasan

Berdasarkan hasil pengujian yang telah dilakukan, kualitas jaringan terbukti memiliki pengaruh signifikan terhadap performa komunikasi data pada sistem presensi berbasis IoT. Penurunan kualitas sinyal menyebabkan peningkatan *Latency*, *Response Time*, dan *Packet Loss*. Kondisi tersebut terjadi akibat keterbatasan bandwidth, meningkatnya retransmisi data, serta antrean proses pada *server* RESTful API. Implementasi RESTful API berbasis HTTP pada penelitian ini masih mampu mendukung komunikasi data secara *real-time* dengan baik. Hal ini ditunjukkan oleh keberhasilan sistem dalam melakukan pengiriman dan penyimpanan data ke *server* pada berbagai kondisi jaringan. Namun demikian, performa sistem sangat dipengaruhi oleh stabilitas jaringan yang digunakan selama proses komunikasi data berlangsung.

Secara keseluruhan, sistem presensi berbasis IoT menggunakan ESP32 dan RESTful API dapat diimplementasikan dengan baik pada lingkungan pendidikan. Sistem mampu melakukan pencatatan kehadiran secara otomatis serta menyediakan

monitoring data presensi secara *real-time* melalui dashboard berbasis web sehingga dapat meningkatkan efisiensi proses presensi dibandingkan metode manual.

5. KESIMPULAN DAN SARAN

Berdasarkan hasil implementasi dan pengujian yang telah dilakukan, sistem presensi berbasis QR Code menggunakan ESP32 dan RESTful API mampu berjalan dengan baik dalam proses pengiriman data secara *real-time*. Hasil pengujian menunjukkan bahwa kualitas jaringan memiliki pengaruh terhadap performa komunikasi data, terutama pada parameter *Latency*, *Response Time*, dan *Packet Loss*. Pada kondisi sinyal lemah, nilai ketiga parameter tersebut mengalami peningkatan dibandingkan kondisi sinyal kuat, yang menunjukkan bahwa kestabilan jaringan menjadi faktor penting dalam menjaga performa komunikasi data pada sistem berbasis IoT. Hasil pengujian menunjukkan rata-rata *Latency* sebesar 219,5 ms pada kondisi sinyal kuat dan 327,5 ms pada kondisi sinyal lemah, sedangkan *Response Time* rata-rata sebesar 297,5 ms dan 417,5 ms serta *Packet Loss* rata-rata sebesar 1,6% dan 3,2%. Secara keseluruhan, sistem yang dikembangkan masih dapat digunakan dengan baik sebagai alternatif sistem presensi digital pada lingkungan pendidikan dan mampu mendukung proses presensi secara *real-time* meskipun terjadi penurunan kualitas jaringan. Hasil penelitian menunjukkan bahwa parameter QoS dapat digunakan sebagai indikator untuk mengevaluasi keandalan komunikasi data pada sistem presensi berbasis IoT menggunakan ESP32 dan RESTful API.

DAFTAR PUSTAKA

- [1] R. Abdullah and D. Setiawan, "Implementasi sistem presensi mahasiswa berbasis QR code menggunakan teknologi web," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 8, no. 2, pp. 235–242, 2021.
- [2] T. Hidayat and B. Kurniawan, "Perancangan sistem presensi mahasiswa berbasis web menggunakan QR code," *Jurnal Teknologi dan Sistem Informasi*, vol. 6, no. 1, pp. 15–23, 2021.
- [3] V. D. Nguyen, H. Van Khoa, T. N. Kieu, and E. N. Huh, "Internet of Things-Based Intelligent Attendance System: Framework, Practice Implementation, and Application," *Electronics (Switzerland)*, vol. 11, no. 19, Oct. 2022, doi: 10.3390/electronics11193151.
- [4] R. Maulana and N. Hidayah, "Pengembangan sistem presensi digital berbasis QR code dan dashboard monitoring berbasis web," *Jurnal Teknologi Informasi dan Komunikasi*, vol. 14, no. 1, pp. 33–41, 2023.
- [5] M. Fauzi and R. Pratama, "Sistem presensi berbasis QR code untuk meningkatkan efisiensi pencatatan kehadiran," *Jurnal Informatika*, vol. 7, no. 3, pp. 120–128, 2020.
- [6] S. I. K. G. M. O. S. Author : Dilek, "QoS-Aware IoT Networks and Protocols: A Comprehensive Survey," *International Journal of Communication Systems*, vol. 35, no. 16, pp. 1–31, 2022.
- [7] Ramelan, A. Rahutomo, and F. Kiswanto, "Design and Implementation RESTful API for IoT Based Smart Home Systems," 2023, pp. 1–8.
- [8] D. Hercog, T. Lerher, M. Truntič, and O. Težak, "Design and Implementation of ESP32-Based IoT Devices," *Sensors*, vol. 23, no. 15, pp. 1–22, 2023.
- [9] T. N. , S. R. Wenny, "Development of Lecture Attendance System Using QR Code in Information and Computer Engineering Education Study Program of Universitas Negeri Makassar," *ELINVO (Electronics, Informatics, and Vocational Education)*, vol. 7, no. 2, pp. 101–109, 2022.
- [10] ETSI TIPHON, "Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON); General Aspects of Quality of Service (QoS)," Sophia Antipolis, 2019.
- [11] Espressif Systems, "ESP32 Series Datasheet v4.1," 2023.