

PENGEMBANGAN APLIKASI WEB UNTUK MONITORING REKAM MEDIS PADA SISTEM E-HEALTH TRACKER BERBASIS IOT

Martogi Jekson C. Siagian¹, Geraldo Alvin Sugiarto²,
I Gusti Agung Putu Raka Agung³, Gede Sukadarmika⁴

^{1,2}Mahasiswa Program Studi Teknik Elektro, Fakultas Teknik, Universitas Udayana

^{3,4}Dosen Program Studi Teknik Elektro, Fakultas Teknik, Universitas Udayana

Jalan Raya Kampus Unud Jimbaran, Kec. Kuta Selatan, Kabupaten Badung, Bali 80361

Email: martogisiagian17@gmail.com¹, geraldoalvin123@gmail.com²,

rakaagung@unud.ac.id³, sukadarmika@unud.ac.id⁴

ABSTRAK

Transformasi digital dalam sektor kesehatan terus mengalami perkembangan signifikan, terutama dalam pencatatan dan pengelolaan rekam medis. Penelitian ini bertujuan untuk pengembangan aplikasi web *monitoring* rekam medis pada sistem *E-Health Tracker* berbasis *Internet of Things* (IoT) dengan pendekatan *User-Centered Design* (UCD). Sistem ini dibangun untuk memberikan kemudahan akses informasi kesehatan bagi pengguna, melalui antarmuka yang intuitif dan kompatibilitas lintas perangkat. Integrasi perangkat IoT menyediakan transmisi data vital seperti detak jantung, suhu tubuh, dan kadar oksigen ke *server* pusat, yang kemudian ditampilkan dalam *platform* web yang mudah dipahami. Keunggulan aplikasi ini terletak pada fleksibilitas akses serta kemampuannya dalam menjembatani kesenjangan layanan kesehatan, terutama bagi masyarakat di wilayah dengan keterbatasan fasilitas medis. Penelitian ini diharapkan dapat memberikan kontribusi nyata dalam mendukung pemerataan akses layanan kesehatan dan meningkatkan literasi serta kesadaran kesehatan masyarakat melalui implementasi teknologi berbasis web dan IoT.

Kata kunci : *Internet of Things* (IoT), *Monitoring* Rekam Medis, Aplikasi Web.

ABSTRACT

The digital transformation in the healthcare sector continues to advance significantly, particularly in the recording and management of electronic medical records. This study proposes the development of a web-based application for monitoring medical records within an *E-Health Tracker* system that leverages the *Internet of Things* (IoT), designed using a *User-Centered Design* (UCD) approach. The system aims to provide seamless access to health information for users through an intuitive interface and cross-device compatibility. The integration of IoT devices enables the transmission of vital health indicators—such as heart rate, body temperature, and oxygen levels—to a central server, which are then visualized through a web-based platform. A key advantage of this application lies in its flexible access and its potential to bridge gaps in healthcare services, especially in regions with limited medical infrastructure. This research is expected to contribute meaningfully to the equitable distribution of healthcare access and to raise public awareness of health literacy through the implementation of IoT-integrated web technologies.

Key Words: *Internet of Things* (IoT), *Medical Record Monitoring*, *Web Application*.

1. PENDAHULUAN

Perkembangan teknologi informasi telah mendorong transformasi digital di sektor kesehatan, terutama dalam

pengelolaan data rekam medis. Sistem konvensional yang mengandalkan pencatatan manual terbukti kurang efisien dan rentan terhadap kesalahan serta

kehilangan data. Oleh karena itu, pengembangan aplikasi web untuk pencatatan dan *monitoring* rekam medis menjadi solusi strategis dalam meningkatkan kualitas layanan kesehatan secara menyeluruh [1].

Aplikasi rekam medis berbasis web memungkinkan akses data pasien secara terpusat, sehingga memudahkan tenaga medis dalam melakukan diagnosis, perawatan, dan tindak lanjut. Sistem ini juga mendukung interoperabilitas antar fasilitas kesehatan, mempercepat proses pencarian data, dan meningkatkan akurasi informasi medis [1],[2]. Dengan dukungan teknologi web, sistem dapat diakses dari berbagai perangkat dan lokasi, menjadikannya fleksibel dan adaptif terhadap kebutuhan pengguna.

Integrasi *Internet of Things* (IoT) dalam sistem *E-Health Tracker* memperluas cakupan *monitoring* kesehatan dengan menghubungkan perangkat IoT yang sudah ada ke *platform* web. Data vital pasien seperti detak jantung, suhu tubuh, dan kadar oksigen dapat dikirim secara otomatis ke *server* dan ditampilkan dalam antarmuka web yang mudah dipahami oleh tenaga medis maupun keluarga pasien [3].

Salah satu keunggulan utama dari sistem berbasis web adalah kemampuannya untuk diakses oleh siapa pun, kapan pun, dan di mana pun. Hal ini sangat penting dalam konteks layanan kesehatan modern yang menuntut mobilitas tinggi dan respons cepat terhadap kondisi pasien. Dengan sistem ini, dokter, perawat, dan bahkan anggota keluarga dapat memantau kondisi pasien secara simultan tanpa harus berada di lokasi yang sama [4].

Dengan adanya teknologi IoT yang terintegrasi dengan aplikasi berbasis web, sistem ini memastikan bahwa data rekam medis hasil pengukuran indikator vital ditransmisikan secara langsung ke *server* pusat dan dapat diakses melalui berbagai perangkat yang terhubung ke *internet*. Aplikasi web memungkinkan siapa saja yang memiliki akun, baik pasien, anggota keluarga, maupun tenaga medis, untuk memantau rekam medis pasien secara jarak jauh. Hal ini tidak hanya memberikan kemudahan akses, tetapi juga mendukung pengawasan kesehatan yang lebih kolaboratif antara pasien, keluarga, dan tenaga medis [5].

Penelitian ini bertujuan pada pengembangan aplikasi web untuk

monitoring rekam medis dalam sistem *E-Health Tracker* berbasis *Internet of Things* (IoT), dengan pendekatan *User-Centered Design* (UCD) guna memastikan tampilan yang informatif, kemudahan penggunaan, dan aksesibilitas lintas perangkat. Aplikasi ini dirancang agar masyarakat, termasuk mereka yang berada di daerah dengan keterbatasan fasilitas kesehatan, dapat melakukan pemeriksaan kesehatan secara rutin dan mandiri. Sementara itu, tenaga medis maupun keluarga pasien dapat melakukan pemantauan kondisi kesehatan secara langsung dari lokasi mana pun. Dengan memanfaatkan teknologi web dan IoT, sistem ini diharapkan dapat mendukung transformasi digital layanan kesehatan sekaligus memperkuat pemerataan akses dan meningkatkan kesadaran kesehatan masyarakat luas.

2. TINJAUAN PUSTAKA

2.1 React

React merupakan pustaka *JavaScript* yang banyak digunakan untuk membangun antarmuka pengguna (*user interface*) yang interaktif dan responsif. *React* menggunakan pendekatan berbasis komponen yang memudahkan pengelolaan kode dan pemeliharaan sistem. Salah satu keunggulannya adalah penggunaan *Virtual DOM*, yang memungkinkan pembaruan tampilan dilakukan secara efisien sehingga meningkatkan performa aplikasi web [6], [7].

2.2 Progressive Web App (PWA)

Progressive Web App (PWA) adalah pendekatan pengembangan aplikasi web modern yang memadukan keunggulan aplikasi web tradisional dengan pengalaman seperti aplikasi *mobile*. PWA memungkinkan aplikasi berjalan lebih cepat, dapat diakses secara *offline*, dan dapat dipasang di layar utama perangkat pengguna layaknya aplikasi *native*. Hal ini dicapai melalui teknologi seperti *service worker* dan manifest aplikasi web, yang meningkatkan keandalan akses meskipun pada jaringan *internet* yang tidak stabil [8], [9].

2.3 FastAPI

FastAPI adalah kerangka kerja web berbasis *Python* yang dirancang untuk membangun *RESTful API* dengan kinerja tinggi. *FastAPI* mendukung pemrograman asinkron, validasi data otomatis, dan pembuatan dokumentasi API secara instan melalui *OpenAPI*. Kemudahan penggunaan dan efisiensinya membuat *FastAPI* ideal untuk menghubungkan data sensor IoT dengan antarmuka aplikasi web [10], [11].

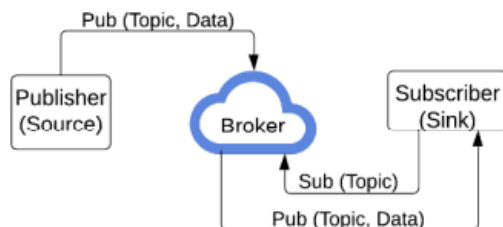
2.4 MongoDB

MongoDB merupakan basis data NoSQL berorientasi dokumen yang mendukung penyimpanan data yang fleksibel dan mudah diskalakan. Arsitektur MongoDB memungkinkan penanganan data tidak terstruktur dalam skala besar secara efisien, sehingga cocok untuk aplikasi modern berbasis *cloud* yang menangani data rekam medis pasien secara dinamis [12].

2.5 Mosquitto Broker

Mosquitto Broker adalah salah satu implementasi dari protokol *Message Queuing Telemetry Transport* (MQTT), yang merupakan protokol komunikasi ringan berbasis *publish-subscribe*. Mosquitto berperan sebagai perantara antara penerbit (*publisher*) dan pelanggan (*subscriber*), memastikan komunikasi antar perangkat IoT berjalan secara efisien. Mosquitto sering digunakan pada aplikasi IoT karena keandalannya dalam mengelola lalu lintas pesan antar perangkat dengan latensi rendah [13], [14].

Protokol MQTT berfungsi di atas infrastruktur TCP/IP, sehingga memerlukan lapisan transport untuk mengirimkan perintah dalam bentuk aliran *byte* antara *client* dan *server* secara dua arah. Pada Gambar 1 dapat dilihat proses komunikasi protokol MQTT [15].



Gambar 1. Protokol MQTT

Keterangan:

1. Topik (UTF-8) berperan sebagai saluran komunikasi yang digunakan oleh client untuk melakukan *subscribe*, serta berfungsi sebagai mekanisme penyaring bagi *broker* dalam mendistribusikan pesan ke *client* yang relevan.
2. *Broker (cloud)* bertugas mengelola proses *publish* dan *subscribe* data, sehingga memungkinkan pemisahan antara pengirim (*publisher*) dan penerima (*subscriber*) tanpa perlu saling mengetahui identitas satu sama lain (*space decoupling*).

Pada Gambar 1, diperlihatkan dua jenis *client*, yaitu *publisher* dan *subscriber*. Keduanya dapat saling terhubung melalui topik tertentu dengan perantara *broker* yang menjadi pusat distribusi pesan [15].

2.6 System Usability Scale (SUS)

System Usability Scale (SUS) adalah salah satu metode evaluasi *usability* yang diperkenalkan oleh John Brooke pada tahun 1986. SUS banyak digunakan untuk menilai tingkat kebergunaan suatu sistem, termasuk perangkat lunak, aplikasi web, dan sistem informasi lainnya. Metode ini dikenal valid dan reliabel meskipun digunakan pada jumlah responden yang relatif kecil, serta memiliki keunggulan berupa kemudahan pengukuran, perhitungan yang sederhana, dan bersifat bebas biaya [16].

Konsep *usability* atau kebergunaan sendiri merupakan atribut kualitas yang mengukur sejauh mana suatu antarmuka mudah digunakan oleh pengguna. Berdasarkan definisi dari *International Organization for Standardization* (ISO), *usability* mencakup tiga aspek utama: efektivitas (kemampuan pengguna mencapai tujuan tertentu), efisiensi (seberapa cepat dan hemat sumber daya untuk mencapai tujuan), dan kepuasan (kenyamanan serta pengalaman positif pengguna saat menggunakan sistem). Antarmuka yang dirancang dengan baik dapat meningkatkan interaksi antara pengguna dengan sistem, sehingga mendukung keberhasilan sistem secara keseluruhan [16].

SUS terdiri dari sepuluh item pernyataan yang dijawab menggunakan skala *Likert* 5 poin, mulai dari "Sangat Tidak Setuju" hingga "Sangat Setuju". Pernyataan dengan nomor ganjil bersifat positif, sedangkan nomor genap bersifat negatif. Skor dihitung dengan cara mengkonversi setiap respons menjadi skor kontribusi, menjumlahkannya kemudian dikalikan 2,5 untuk menghasilkan nilai akhir pada rentang 0 hingga 100. Skor kemudian diinterpretasikan menggunakan berbagai kategori, misalnya *acceptability range* (skor ≥ 70 dianggap dapat diterima), *grade scale* (skor ≥ 90 mendapat nilai A), dan *adjective rating* (skor $\geq 70,4$ dikategorikan sebagai "Good") [16].



Gambar 2. Interpretasi skor SUS

3. METODE PENELITIAN

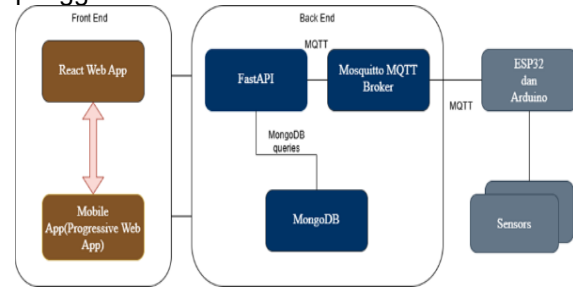
Penelitian ini menggunakan pendekatan *research and development* dengan tujuan mengembangkan aplikasi web untuk *monitoring* rekam medis berbasis *Internet of Things* (IoT). Aplikasi ini dirancang untuk menampilkan hasil pengukuran empat indikator vital kesehatan yang dikirim dari Perangkat IoT ke *server*, kemudian ditampilkan melalui antarmuka berbasis web. Pengembangan aplikasi web untuk *monitoring* rekam medis pada sistem *E-Health Tracker* berbasis IoT dilakukan melalui beberapa tahapan terstruktur untuk memastikan sistem dapat menampilkan data kesehatan dengan baik.

3.1 Gambaran Umum Sistem

Arsitektur sistem *monitoring* rekam medis berbasis IoT yang dikembangkan dapat dilihat pada Gambar 3, terdiri dari tiga bagian utama: *front end*, *back end*, dan perangkat IoT. Bagian *front end* mencakup aplikasi berbasis web yang dikembangkan dengan *React* dan dioptimalkan untuk seluler melalui *Progressive Web App* (PWA). Antarmuka ini memungkinkan pengguna mengakses data rekam medis melalui *browser* atau perangkat *mobile* dengan tampilan yang interaktif dan mudah digunakan. Bagian *back end* menggunakan *FastAPI* sebagai *server* untuk menangani komunikasi data antara antarmuka pengguna, basis data, dan broker MQTT. Data yang diterima disimpan dalam basis data *MongoDB*, yang memungkinkan penyimpanan dan pengambilan data secara efisien. *Mosquitto MQTT Broker* berperan sebagai penghubung antara perangkat IoT dan *server*, mengelola pengiriman data sensor menggunakan protokol MQTT.

Bagian perangkat IoT terdiri dari *ESP32 dan Arduino* yang dilengkapi dengan sensor untuk mengukur indikator vital kesehatan. Data hasil pengukuran dikirim ke *server* melalui broker MQTT dan ditampilkan melalui aplikasi web. Desain sistem ini memastikan komunikasi yang efisien dari perangkat IoT ke *server*, penyimpanan data

yang andal, serta penyajian informasi kesehatan yang mudah diakses oleh pengguna.



Gambar 3. Gambaran Umum Sistem

3.2 Pendekatan *User-Centered Design* (UCD)

Pengembangan aplikasi web *monitoring* rekam medis berbasis IoT pada penelitian ini menerapkan pendekatan *User-Centered Design* (UCD) untuk memastikan solusi yang dihasilkan sesuai kebutuhan pengguna. UCD dilakukan secara iteratif melalui empat tahap utama, mulai dari pemahaman konteks hingga evaluasi desain.

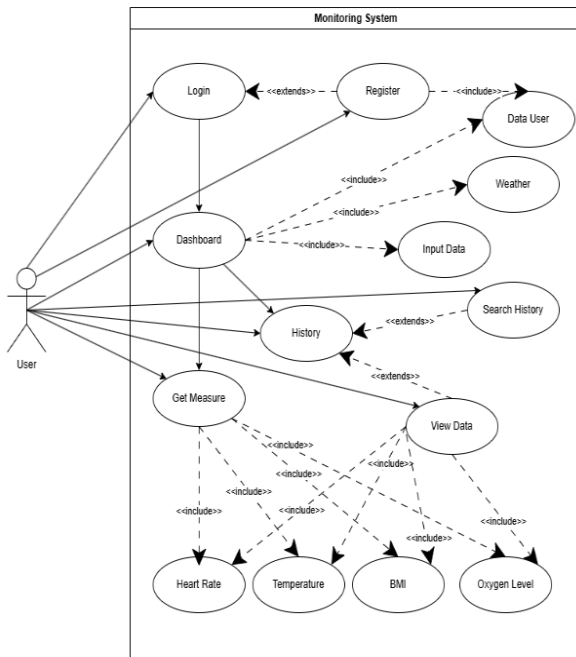
1. Pengguna utama sistem adalah masyarakat desa dengan tingkat literasi digital beragam. Permasalahan yang teridentifikasi meliputi ketiadaan rekam medis digital, pencatatan manual yang rentan kesalahan, keterbatasan akses informasi riwayat kesehatan, dan ketergantungan pada kunjungan fisik ke fasilitas kesehatan. Sistem dikembangkan untuk diakses secara mandiri maupun oleh anggota keluarga melalui perangkat *mobile* atau computer.
2. Ditetapkan tujuan sistem untuk menyediakan akses data kesehatan secara langsung, riwayat pengukuran yang terdigitalisasi, transparansi bagi keluarga, serta eliminasi pencatatan manual. Fitur utama yang dikembangkan meliputi antarmuka *login*, registrasi, *dashboard* dengan data indikator kesehatan (*heart rate*, suhu, tekanan darah, saturasi oksigen), informasi pasien, data cuaca, serta halaman riwayat pengukuran yang dapat diakses dan dicari dengan mudah.
3. Antarmuka aplikasi dirancang berbasis *React* dengan prinsip PWA untuk mendukung akses lintas perangkat. *Dashboard* menampilkan metrik kesehatan dan informasi pasien dalam tata letak berbasis kartu. Fitur navigasi seperti tombol *Get Measure* untuk mengambil data terbaru dan *History*

untuk melihat riwayat disediakan untuk kemudahan pengguna.

4. Pengujian *usability* dilakukan dengan metode *System Usability Scale* (SUS). Responden mencoba aplikasi secara langsung atau melalui video panduan sebelum mengisi kuesioner.

3.3 Use Case Diagram

Pada sistem *monitoring* rekam medis berbasis IoT yang dikembangkan, hanya terdapat satu aktor utama yaitu *user* yang berinteraksi langsung dengan aplikasi web, sebagaimana ditunjukkan pada Gambar 4.



Gambar 4. Use Case Diagram Sistem

Fitur-fitur utama yang dapat diakses pengguna antara lain autentikasi melalui *login* dan registrasi, kemudian akses ke halaman *dashboard* yang menjadi pusat tampilan data kesehatan. Dari *dashboard*, pengguna dapat menggunakan fitur *Get Measure* untuk mengambil data hasil pengukuran indikator kesehatan seperti detak jantung, suhu tubuh, tekanan darah, dan saturasi oksigen. Fitur-fitur ini saling terhubung melalui relasi ketergantungan untuk memastikan data yang ditampilkan akurat. Selain itu, pengguna juga dapat mengakses fitur *History* untuk melihat riwayat pengukuran sebelumnya, termasuk melakukan pencarian cepat melalui *Search History*. Sistem juga menyajikan informasi cuaca lokal sebagai pelengkap melalui fitur *Weather*. Terdapat pula fitur *Input Data*, yang memungkinkan setiap orang yang

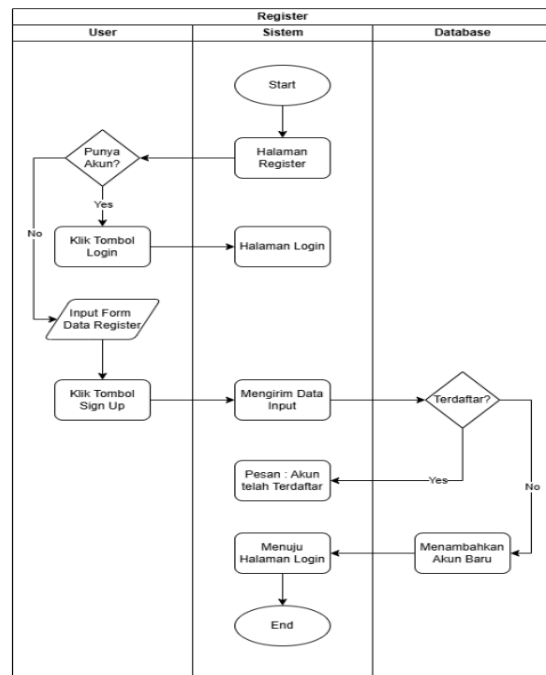
melakukan pengecekan kesehatan untuk menginput data mereka ke sistem, meskipun menggunakan akun yang berbeda. Seluruh fitur ini dirancang untuk mendukung kebutuhan pengguna dalam memantau rekam medis secara jarak jauh dengan akses yang mudah dan informatif.

3.4. Activity Diagram

Activity Diagram ini menyajikan alur aktivitas pada setiap halaman dan fitur utama dalam aplikasi web monitoring rekam medis berbasis IoT yang dikembangkan. Diagram ini bertujuan untuk memperjelas proses bisnis dan interaksi pengguna dalam sistem secara terstruktur.

1. Activity Diagram Halaman Register

Proses dimulai dari halaman registrasi, di mana pengguna memilih untuk *login* jika sudah memiliki akun atau melanjutkan registrasi dengan mengisi form data, sebagaimana ditunjukkan pada Gambar 5.



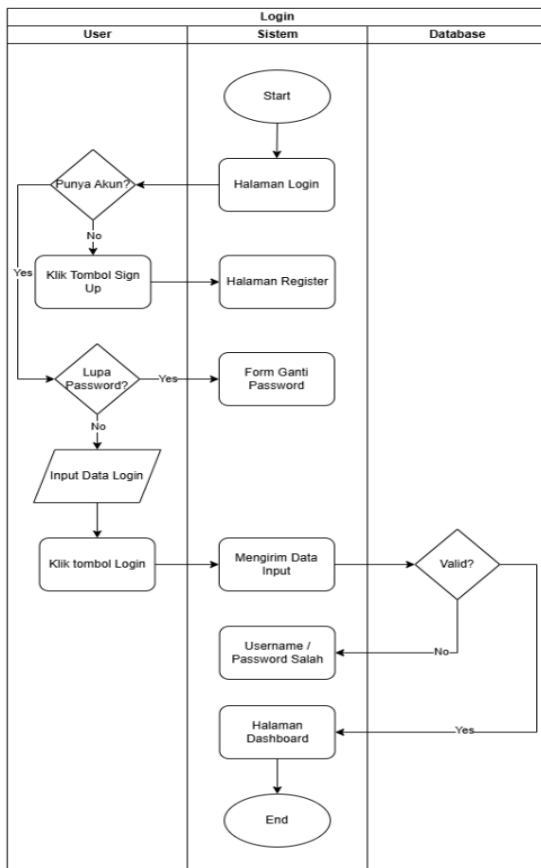
Gambar 5. Activity Diagram Halaman Register

Setelah data diisi dan tombol *Sign Up* ditekan, sistem memproses data dan memeriksa apakah sudah terdaftar. Jika sudah ada, pengguna diarahkan ke halaman *login* dengan pesan bahwa akun telah terdaftar. Jika belum, sistem menambahkan akun baru ke *database*, memberikan konfirmasi, dan mengarahkan pengguna ke halaman *login*. Proses ini melibatkan

pengguna, sistem, dan *database* secara terintegrasi.

2. Activity Diagram Halaman Login

Proses dimulai dengan pengguna mengakses halaman *login*. Jika pengguna belum memiliki akun, mereka diarahkan ke halaman registrasi. Jika lupa password, pengguna dapat mengakses form ganti password, sebagaimana ditunjukkan pada Gambar 6.



Gambar 6. Activity Diagram Halaman Login

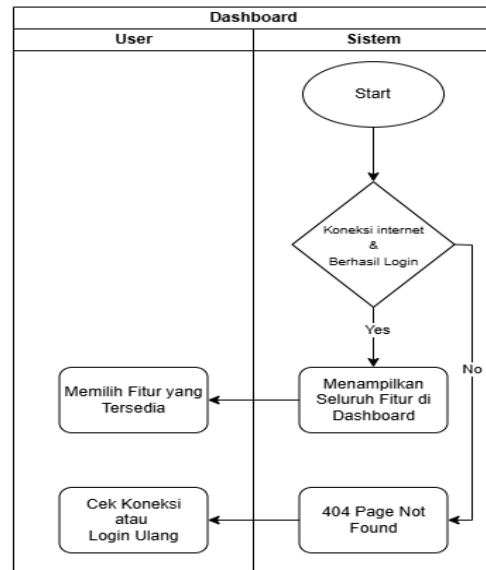
4. Activity Diagram Get Measure

Proses dimulai ketika pengguna melakukan pengecekan menggunakan alat ukur. Setelah itu, pengguna menginputkan data diri ke dalam sistem

Selanjutnya, pengguna memasukkan data *login* dan menekan tombol *Login*. Sistem memproses data dan memvalidasi dengan *database*. Jika valid, pengguna diarahkan ke halaman *dashboard*. Jika tidak valid, sistem memberikan notifikasi bahwa username atau password salah, dan pengguna diminta untuk mencoba Kembali.

3. Activity Diagram Dashboard

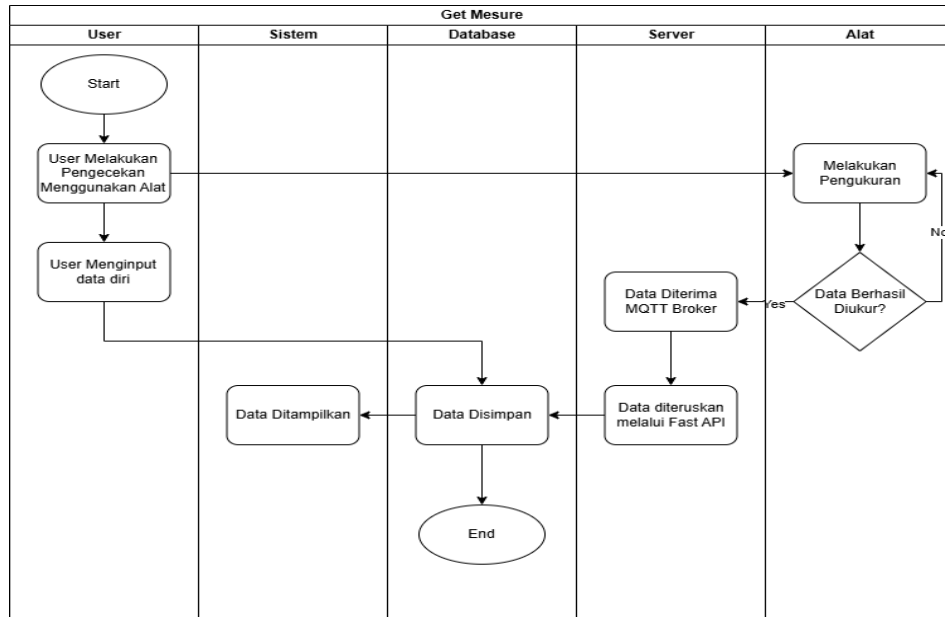
Proses dimulai dengan pengecekan koneksi *internet* dan status *login* pengguna, dapat dilihat pada gambar 7.



Gambar 7. Activity Diagram Dashboard

Jika koneksi stabil dan *login* berhasil, sistem menampilkan seluruh fitur di *dashboard* dan pengguna dapat memilih fitur yang tersedia. Namun, jika koneksi bermasalah atau *login* gagal, sistem menampilkan halaman *404 Page Not Found*, dan pengguna diminta untuk memeriksa koneksi atau *login* ulang.

sebelum proses pengukuran dilakukan oleh alat. proses dapat dilihat pada gambar 8.

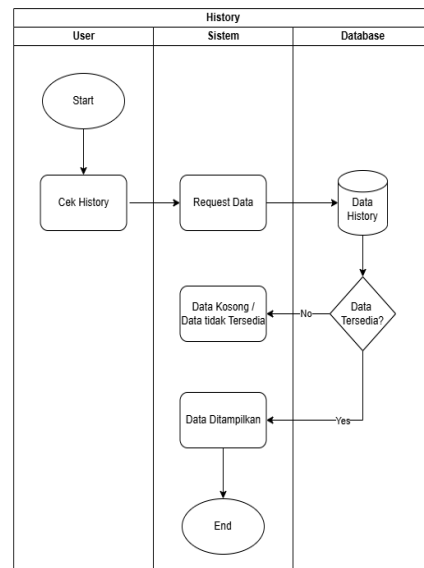


Gambar 8. Activity Diagram Get Measure

Alat kemudian melakukan pengukuran. Jika gagal, proses kembali ke langkah sebelumnya hingga berhasil. Jika berhasil, data dikirim ke *server* melalui MQTT Broker, diteruskan ke *FastAPI* untuk disimpan di *database*, lalu hasil pengukuran ditampilkan ke pengguna.

5. Activity Diagram History

Proses dimulai ketika pengguna memilih untuk mengecek Riwayat, dapat dilihat pada Gambar 9. Sistem mengirim permintaan data ke *database*. Jika data tidak tersedia, sistem memberikan notifikasi bahwa data kosong. Jika tersedia, sistem menampilkan data riwayat kepada pengguna. Diagram ini menggunakan tiga swimlane untuk memisahkan peran pengguna, sistem, dan *database*.



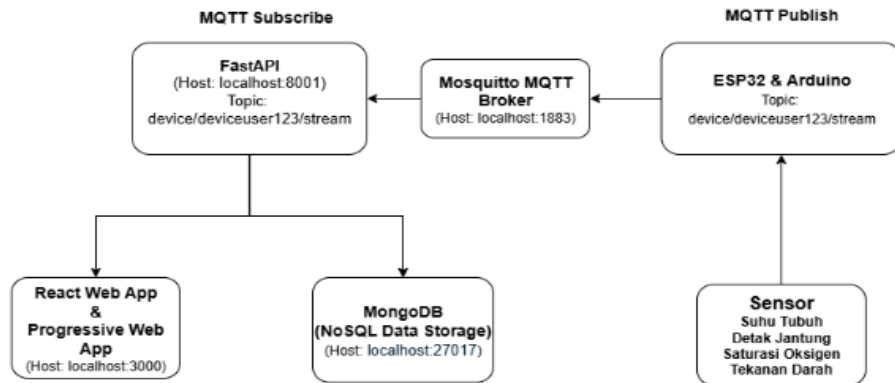
Gambar 9. Activity Diagram History

4. HASIL DAN PEMBAHASAN

4.1 Implementasi Komunikasi Aplikasi Web dengan Perangkat IoT

Arsitektur komunikasi pada sistem ini menggambarkan alur pertukaran data

antara perangkat IoT, *server*, dan aplikasi web, sebagaimana ditunjukkan pada Gambar 10.



Gambar 10. Arsitektur Komunikasi Sistem

Data kesehatan yang diukur oleh sensor meliputi suhu tubuh, detak jantung, saturasi oksigen, dan tekanan darah dikirimkan oleh perangkat *ESP32 & Arduino* ke topik *MQTT* melalui protokol *publish*. Pesan tersebut diterima oleh *Mosquitto MQTT Broker*, yang berfungsi sebagai perantara komunikasi berbasis *publish-subscribe*. *Backend* sistem menggunakan *FastAPI* untuk melakukan *subscribe* pada topik yang sama, kemudian menerima data yang diteruskan oleh broker. Selanjutnya, data disimpan ke basis data *MongoDB* untuk pencatatan dan keperluan riwayat rekam medis. Data yang sudah tersimpan juga disajikan kepada pengguna melalui antarmuka berbasis *React Web App* dan *Progressive Web App* (PWA), yang dapat diakses pada berbagai perangkat dengan koneksi *internet*. Arsitektur ini memastikan aliran data yang efisien, mulai dari pengukuran di perangkat IoT, penyimpanan terpusat di *server*, hingga penyajian data kepada pengguna melalui aplikasi web.

Tabel 1 merupakan kode program yang digunakan pada perangkat IoT untuk mengirimkan data Ke *MQTT Broker*

Tabel 1. Kode Program Pengiriman Data Perangkat IoT

```

void publishData() {
    StaticJsonDocument<256> jsonDoc;
    jsonDoc["ap_hi"] = sistolex;
    jsonDoc["ap_lo"] = diastolex;
    jsonDoc["saturation_data"] = so;
    jsonDoc["heart_rate_data"] =
    lockedBPM;
    jsonDoc["temp"] = suhu;

    char buffer[256];
    size_t n = serializeJson(jsonDoc,
    buffer);
  
```

```

client.publish("device/deviceuser123
/stream", buffer, n);
Serial.println("Data sent to
MQTT:");
Serial.println(buffer);
  
```

Tabel 2 merupakan kode program yang digunakan pada aplikasi Web untuk menerima data dari *MQTT Broker* yang telah dikirim oleh perangkat IoT.

Tabel 2. Kode Program Penerimaan data

```

try{
const client =
mqtt.connect("ws://localhost:9001", {
  username: "backenduser",
  password: "backenduser",
  rejectUnauthorized: false,
});

client.on("connect", () => {
  console.log("Connected to MQTT
broker");
  const topic =
"device+/stream";
  client.subscribe(topic, (err)
=> {
    if (err) {

console.error("Subscription error:",
err);

      setIsLoading(false);
      client.end();
    }
  });
});
  
```

Pada sistem ini, komunikasi data antara perangkat IoT dan aplikasi web dilakukan melalui *protokol MQTT* dengan perantara broker *Mosquitto*. Kode komunikasi penerimaan data bertugas untuk melakukan koneksi ke *broker MQTT* menggunakan protokol *WebSocket* pada alamat `ws://localhost:9001`, dengan *autentikasi* username dan password. Setelah koneksi berhasil, sistem melakukan

subscribe pada topik *device/+/stream*, yang memanfaatkan *wildcard* untuk menerima pesan dari semua perangkat IoT yang terhubung pada jalur tersebut. Apabila terjadi kegagalan saat proses *subscribe*, sistem akan menghentikan proses pemuatan data, mencetak pesan kesalahan pada *konsol*, dan menutup koneksi. Mekanisme ini memastikan data hasil pengukuran dari perangkat IoT dapat diterima oleh aplikasi web untuk ditampilkan kepada pengguna.

4.2 Implementasi Antarmuka Website

Hasil implementasi antarmuka website *monitoring* rekam medis berbasis IoT dapat dilihat pada Gambar 11 sampai dengan Gambar 14 yang terdiri dari empat halaman utama, yaitu *Register*, *Login*, *Dashboard*, dan *History*. Pada halaman *Register* yang dapat dilihat pada Gambar 10, pengguna dapat membuat akun baru dengan mengisi email, username, password, konfirmasi password, dan gender. Desain halaman konsisten dengan ilustrasi bertema kesehatan, serta dilengkapi validasi *input* untuk memastikan data yang dimasukkan benar. Setelah berhasil mendaftar, pengguna diarahkan ke halaman *Dashboard*.



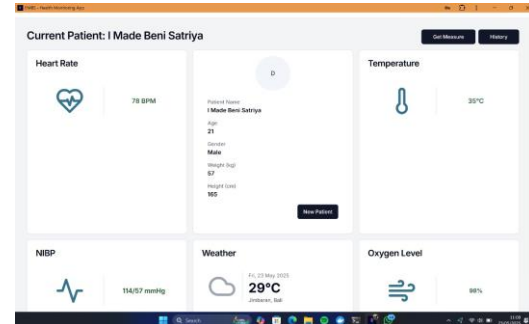
Gambar 11. Halaman *Register*

Pada halaman *Login* yang dapat dilihat pada Gambar 12, pengguna melakukan autentikasi dengan mengisi username dan password. Jika kredensial valid, *server* mengembalikan token autentikasi untuk sesi pengguna dan mengarahkan ke *dashboard*. Tautan ke halaman *register* juga disediakan bagi pengguna yang belum memiliki akun.



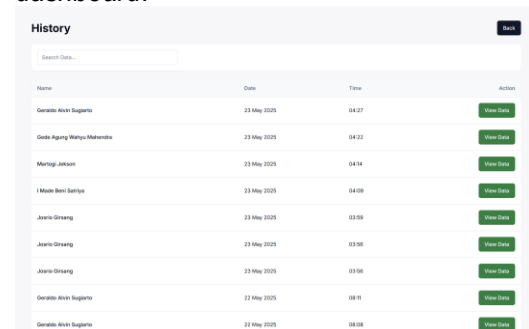
Gambar 12. Halaman *Login*

Halaman *Dashboard* menampilkan data kesehatan pasien dalam bentuk kartu yang dapat dilihat pada Gambar 13, mencakup detak jantung, suhu tubuh, tekanan darah, saturasi oksigen, cuaca lokal, dan informasi pasien. Tersedia tombol *Get Measure* untuk mengambil data terbaru dari perangkat IoT, serta tombol *History* untuk melihat riwayat pengukuran. Informasi cuaca diperoleh secara otomatis dari API eksternal.



Gambar 13. Halaman *Dashboard*

Terakhir, halaman *History* menyajikan daftar riwayat pengukuran kesehatan lengkap dengan nama pasien, tanggal, waktu, dan detail data yang dapat dilihat pada Gambar 14. Fitur pencarian memudahkan pengguna menemukan data tertentu, serta tombol *Back* untuk kembali ke *dashboard*.



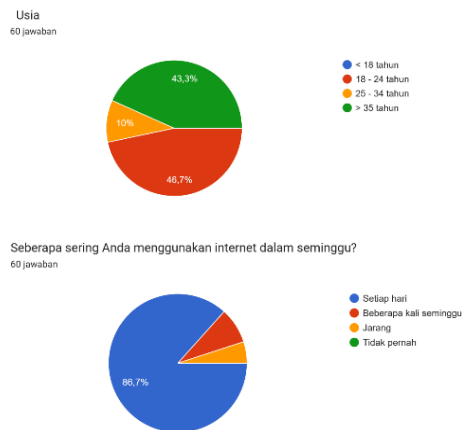
Gambar 14. Halaman *History*

4.3 Hasil dan Analisis Pengujian System Usability Scale (SUS)

Pengujian *usability* dilakukan untuk mengevaluasi tingkat kemudahan penggunaan dan kenyamanan aplikasi *E-Health Tracker* dari sudut pandang pengguna akhir. Metode yang digunakan adalah *System Usability Scale* (SUS), yaitu instrumen evaluasi berbasis kuesioner yang dikembangkan oleh Brooke (1986) dan terdiri dari 10 pernyataan dengan skala *Likert* 1–5. Tujuan pengujian ini adalah mengukur persepsi pengguna terhadap kemudahan penggunaan, tingkat

kompleksitas, dan kepercayaan diri saat berinteraksi dengan antarmuka aplikasi.

Pengujian melibatkan 60 responden dengan latar belakang beragam yang dapat dilihat pada Gambar 14, yaitu mahasiswa, tenaga medis, dan masyarakat umum dengan rentang usia: kurang dari 18 tahun (0%), 18–24 tahun (46,7%), 25–34 tahun (10%), dan lebih dari 35 tahun (43,3%). Sebanyak 14 responden berasal dari Desa Manduang, Klungkung sebagai lokasi uji coba awal, untuk memperoleh gambaran umum sebelum implementasi di Desa Tanglad, lokasi utama penelitian. Mayoritas responden (86,7%) juga tercatat sudah terbiasa menggunakan *internet* setiap hari, mendukung tingkat adaptasi terhadap aplikasi.



Gambar 14. Rentang usia dan penggunaan *internet*

Tabel 3. Hasil pertanyaan Kuisioner

| No | Pertanyaan |
|-----|--|
| P1 | Saya merasa akan sering ingin menggunakan aplikasi ini |
| P2 | Aplikasi ini terasa rumit |
| P3 | Saya merasa aplikasi ini mudah digunakan |
| P4 | Saya rasa saya akan butuh bantuan teknis untuk bisa menggunakan aplikasi ini |
| P5 | Fitur-fitur dalam aplikasi ini terintegrasi dengan baik |
| P6 | Saya merasa ada terlalu banyak inkonsistensi dalam aplikasi ini |
| P7 | Sebagian besar orang mungkin bisa menggunakan aplikasi ini dengan cepat |
| P8 | Aplikasi ini terasa membingungkan |
| P9 | Saya merasa percaya diri saat menggunakan aplikasi ini |
| P10 | Saya perlu belajar banyak sebelum bisa menggunakan aplikasi ini |

Kuesioner SUS berisi 10 pernyataan yang dapat dilihat pada Tabel 3, dengan komposisi 5 pernyataan positif (item ganjil) dan 5 pernyataan negatif (item genap). Nilai untuk pernyataan positif dihitung dengan mengurangi skor jawaban dengan 1,

sedangkan untuk pernyataan negatif dengan mengurangi jawaban dari 5. Semua skor dijumlahkan lalu dikalikan 2,5 untuk menghasilkan skor akhir per individu pada skala 0–100.

Dari hasil pengumpulan data kuesioner SUS terhadap 60 responden, diperoleh skor individual yang bervariasi antara 60 hingga 100, dengan skor mentah rata-rata sebesar 34,98. Setelah dikonversi ke dalam skala 0–100 menggunakan formula standar SUS, diperoleh rata-rata skor akhir sebesar 87,42. Nilai ini menunjukkan bahwa sistem memiliki tingkat *usability* yang sangat baik menurut persepsi pengguna. Lebih lanjut, sebanyak 48 dari 60 responden (80%) memberikan skor di atas 85, sedangkan tidak ada responden yang memberikan skor di bawah 60, menandakan bahwa sistem berhasil memenuhi ekspektasi pengguna dalam hal kemudahan penggunaan, kejelasan antarmuka, dan pengalaman interaksi secara keseluruhan.

Berdasarkan hasil pengumpulan data kuesioner, digunakan *System Usability Scale* (SUS) untuk melakukan perhitungan tingkat skala *Usability*. *System Usability Scale* (SUS) merupakan metode kuantitatif untuk mengukur tingkat *usability* suatu sistem melalui 10 pernyataan yang menggunakan skala *Likert* 1 sampai 5 [16]. Untuk mendapatkan skor SUS dari data kuesioner, digunakan rumus perhitungan sebagai berikut:

$$\text{Skor SUS} = \left(\sum_{i=1,3,5,7,9} (S_i - 1) + \sum_{j=2,4,6,8,10} (5 - S_j) \right) \times 2,5 \quad [16]$$

Dengan:

S_i adalah skor pada item ganjil (pernyataan positif)

S_j adalah skor pada item genap (pernyataan negatif)

Berikut merupakan perhitungan pada responden 1

Nilai responden: P1=4, P2=2, P3=5, P4=1, P5=5, P6=2, P7=5, P8=1, P9=5, P10=1

Item ganjil: $(4-1) + (5-1) + (5-1) + (5-1) + (5-1) = 3 + 4 + 4 + 4 + 4 = 19$

Item genap: $(5-2) + (5-1) + (5-2) + (5-1) + (5-1) = 3 + 4 + 3 + 4 + 4 = 18$

Skor mentah = $19 + 18 = 37$

Skor akhir SUS = $37 \times 2,5 = 92,5$

Setelah menghitung skor akhir SUS untuk 60 responden, nilai rata-rata dihitung dengan rumus:

$$\bar{SUS} = \frac{\sum_{k=1}^N SUS_k}{N} \quad [16]$$

Dengan:

SUSk adalah skor SUS individu responden ke- *k*

N = adalah jumlah total responden

Berdasarkan data, diperoleh:

$$\overline{SUS} = \frac{5245}{60} \approx 87,42$$

Berdasarkan hasil pengukuran *usability* menggunakan metode *System Usability Scale* (SUS), sistem memperoleh skor sebesar 87,42, yang menunjukkan tingkat kebergunaan berada dalam kategori Excellent menurut *adjective rating* dan termasuk dalam *Grade Scale A*, yaitu kategori nilai tertinggi dalam interpretasi SUS. Skor ini juga berada dalam kategori acceptable, karena telah melampaui ambang batas minimal 70 yang dianggap layak oleh pengguna. Secara rinci, hasil kuesioner menunjukkan bahwa skor pada pertanyaan-pertanyaan positif (nomor 1, 3, 5, 7, 9) cenderung tinggi, sedangkan skor pada pertanyaan negatif (nomor 2, 4, 6, 8, 10) cenderung rendah. Pola ini wajar dan sesuai dengan karakteristik SUS, di mana responden umumnya lebih menyetujui pernyataan positif yang mencerminkan kemudahan dan kepuasan penggunaan, serta tidak menyetujui pernyataan negatif yang merepresentasikan kerumitan atau kesulitan. Perlu dicatat bahwa perhitungan skor akhir SUS telah memperhitungkan pembalikan nilai pada item pertanyaan negatif sebagaimana dijelaskan oleh Brooke (1996), sehingga nilai akhir yang diperoleh telah mencerminkan tingkat *usability* sistem secara valid sesuai standar.

5. KESIMPULAN

Berdasarkan hasil pengembangan aplikasi web untuk *monitoring* rekam medis pada sistem *E-Health Tracker* berbasis *Internet of Things* (IoT), yang telah melalui tahapan perancangan, pengujian, dan analisis, maka diperoleh beberapa kesimpulan sebagai berikut:

1. Aplikasi web *monitoring* berhasil dikembangkan menggunakan *framework* React dengan pendekatan *User-Centered Design* (UCD), sehingga mampu menampilkan hasil pengukuran empat indikator vital tubuh yaitu detak jantung, saturasi oksigen, suhu tubuh, dan tekanan darah.
2. Evaluasi *usability* aplikasi web menggunakan metode *System Usability Scale* (SUS) menunjukkan skor sebesar 87,42, yang termasuk dalam kategori *Excellent* (*Grade Scale A*) dan

acceptable, menandakan tingkat kemudahan penggunaan dan kepuasan pengguna yang sangat baik terhadap antarmuka dan fungsionalitas sistem.

6. DAFTAR PUSTAKA

- [1] Biahdilah, A., & Septiana, Y. 2020. Rancang Bangun Aplikasi Rekam Medis Berbasis Web. *Jurnal Algoritma*, 17(2), 361–367.
- [2] Purnama, B. E., Rochmah, I. N., & Nurfauzi, M. 2024. Pengembangan Aplikasi Rekam Medis untuk Fasilitas Pelayanan Kesehatan. *Indonesian Journal on Networking and Security*, 13(1), 14–17.
- [3] Hasibuan, A. H. 2021. Pemantauan Kesehatan Pasien Berbasis IoT pada ESP32 WEB Server. *Skripsi*. Universitas Pembangunan Panca Budi.
- [4] Ramadhan, I. W., Firdaus, & Adinandra, S. 2024. Penerapan IoT dalam Sistem Monitoring Kesehatan: Inovasi dan Implementasi. *Techno.COM*, 23(4), 763–772.
- [5] Mohammed, B. G., & Hasan, D. S. 2023. Smart Healthcare Monitoring System Using IoT. *International Journal of Interactive Mobile Technologies*, 17(1), 141–152.
- [6] Sanjaya, N., & Sardjono, W. 2023. Aplikasi Konsultasi Dokter-Pasien Secara Online Berbasis Android Menggunakan React Native. *Jurnal Sistem Informasi Dan Informatika (Simika)*, 6(2), 153–164.
- [7] Muldoon, C., Görgü, L., O'Sullivan, J. J., Meijer, W. G., Masterson, B., & O'Hare, G. M. P. 2021. *Engineering testable and maintainable software with Spring Boot and React*.
- [8] Herman, & Frederick. 2022. Progressive Web Apps: Pengembangan dan Studi Penerimaan Pada Mahasiswa Indonesia menggunakan Scrum dan Utaut. *Jurnal Teknologi Terpadu*, 9(1), 22–28.
- [9] Hudianti, E., Maulana, D., & Nugroho, M. A. 2023. Implementasi Progressive Web Apps untuk Sistem Pengelolaan Potenso Desa Wisata Kali Opak Tujuh Bulan. *Journal Of Information System Management (JOISM)*, 4(2), 86–90.

- [10] Azhari, R. Y. 2022. Web Service Framework: flask dan fastAPI. *Technology and Informatics Insight Journal*, 1(1).
- [11] Chen, J. 2023. Model Algorithm Research based on Python Fast API. *Frontiers in Science and Engineering*, 3(9), 7-10.
- [12] Erraji, A., Maizate, A., & Ouzzif, M. 2023. An Integral Approach for Complete Migration from a Relational Database to MongoDB. *Journal of the Nigerian Society of Physical Sciences*, 5(2).
- [13] Rizkina Utamy, A. 2023. Prototype Wireless Sensor Network Sistem Pengukuran Debu dan Suhu Udara Berbasis MQTT Server. *Jurnal PROSISKO*, 10(2), 152–164.
- [14] Kumar, N. N., & Panchal, M. P. 2024. Building a Secure IoT Platform for Smart Home Automation: A Comprehensive Integration. *International Journal for Research in Applied Science and Engineering Technology*, 12(1), 302–309.
- [15] Revikasari, Y., Pramaita, N., & Nugraha, I. P. E. D. (2024). Evaluasi Website Monitoring Skuter Listrik Sewa Berbasis IoT dengan Protokol MQTT. *Jurnal SPEKTRUM*, 11(3), 59-68.
- [16] Kurniawan, E., Nofriadi, & Nata, A. 2022. Penerapan System Usability Scale (SUS) dalam Pengukuran Kebergunaan Website Program Studi di STMIK Royal. *Journal of Science and Social Research*, 5(1), 43–49.