P-ISSN : 2580-4316
https://doi.org/10.37817/ikraith-informatika.v10i2 E-ISSN : 2654-8054

Sistem Pemantauan EKG 1-Lead Berbasis ESP32 dan Wireless Fidelity dengan Visualisasi Real Time Pada Smartphone

[1]Nugroho Budhisantosa, [2]Noval Rizky Ramadhan, [3]Habibullah Akbar, Gerry Firmansyah[4]
[1, 2, 3, 4] Teknik Informatika, Universitas Esa Unggul, Kota Jakarta

E-mail: [1]nugroho.budhisantosa@esaunggul.ac.id,
[2]novalrizky428@student.esaunggul.ac.id, [3]habibullah.akbar@esaunggul.ac.id,
[4]gerry@esaunggul.ac.id

ABSTRAK

Penelitian ini membahas pengembangan sistem pemantauan detak jantung berbasis Internet of Things (IoT) menggunakan mikrokontroler ESP32, sensor ECG AD8232, protokol MQTT, dan platform Node-RED sebagai antarmuka visual. Sistem ini dirancang untuk melakukan akuisisi sinyal jantung, mengolah data menjadi nilai Beats Per Minute (BPM), serta menampilkan informasi secara real-time melalui dashboard berbasis web yang dapat diakses menggunakan smartphone maupun perangkat lain. Metodologi yang digunakan adalah Waterfall, dengan tahapan analisis kebutuhan, perancangan, implementasi, dan pengujian sistem. Hasil penelitian menunjukkan bahwa sistem mampu menampilkan sinyal EKG dengan latensi rata-rata ±180 ms, sesuai dengan target real-time (<200 ms). Selain itu, sistem berhasil mengklasifikasikan kondisi jantung ke dalam kategori normal, tachycardia, bradycardia, atrial flutter, dan arrhythmia, dengan tingkat akurasi mencapai 92% berdasarkan pengujian menggunakan data simulasi (dummy) maupun data sensor aktual. Fitur tambahan seperti notifikasi visual dan audio, serta fungsi reset grafik (Clear Graph) berjalan sesuai rancangan dan meningkatkan pengalaman pengguna. Validasi dengan alat referensi (pulse oximeter) menunjukkan selisih nilai BPM rata-rata ±5 bpm, yang masih dapat diterima untuk aplikasi non-klinis. Dengan hasil tersebut, sistem ini dapat digunakan sebagai solusi monitoring mandiri untuk kesehatan jantung, serta memiliki potensi pengembangan lebih lanjut melalui integrasi penyimpanan data historis, enkripsi komunikasi, maupun implementasi aplikasi mobile.

Kata Kunci: Internet of Things, ESP32, Node-RED, MQTT, EKG, monitoring jantung

ABSTRACT

This study presents the development of an Internet of Things (IoT)-based heart monitoring system utilizing an ESP32 microcontroller, an ECG AD8232 sensor, the MQTT protocol, and Node-RED as a web-based interface. The system is designed to capture cardiac signals, process them into beats per minute (BPM), and display the results in real-time through a responsive dashboard accessible via smartphones or other devices. The research employs the Waterfall methodology, covering system requirements analysis, design, implementation, and evaluation. Experimental results demonstrate that the system can display ECG signals with an average latency of ±180 ms, which meets the target for real-time performance (<200 ms). Moreover, the system effectively classifies heart conditions into normal, tachycardia, bradycardia, atrial flutter, and arrhythmia, achieving an accuracy rate of 92% when tested with both simulated (dummy) and real sensor data. Additional features such as visual and audio alerts and the Clear Graph function performed reliably, enhancing overall usability. Validation against a reference device (pulse oximeter) indicated an average BPM deviation of only ±5 bpm, which remains acceptable for non-clinical monitoring purposes. The findings confirm that this system provides a reliable solution for self-monitoring of cardiac activity. Furthermore, it holds significant potential for future

P-ISSN : 2580-4316
E-ISSN : 2654-8054 https://doi.org/10.37817/ikraith-informatika.v10i2

enhancements, including the integration of historical data storage, stronger communication security, and the development of dedicated mobile applications.

Keywords: *Internet of Things, ESP32, Node-RED, MQTT, ECG, heart monitoring*

1. PENDAHULUAN

Penyakit kardiovaskular masih menjadi penyebab utama kematian di dunia, dengan angka mencapai sekitar 17,9 juta jiwa setiap tahun (WHO, 2021). Di Indonesia, prevalensinya juga cukup tinggi, yakni 1,5% dari populasi atau sekitar 4,8 juta jiwa berdasarkan data Riskesdas 2018 (Kemenkes RI, 2018). Kondisi ini menegaskan pentingnya deteksi dini dan pemantauan kesehatan jantung untuk mencegah komplikasi serius yang berpotensi fatal.

Elektrokardiogram (EKG) merupakan metode diagnostik non-invasif yang umum digunakan untuk memantau aktivitas listrik jantung serta mendeteksi berbagai kelainan seperti aritmia, iskemia, dan infark miokard (Goldberger, 2017). Salah satu konfigurasi yang banyak digunakan adalah EKG 1-lead karena praktis dan mampu memberikan informasi dasar mengenai aktivitas jantung. Namun, perangkat EKG konvensional masih menghadapi kendala, antara lain harga yang tinggi, keterbatasan portabilitas, serta ketergantungan pada fasilitas medis dan tenaga profesional (Fahim et al., 2019).

Perkembangan teknologi Internet of Things (IoT) dan mikrokontroler menghadirkan peluang baru untuk menciptakan perangkat EKG yang lebih murah, portabel, dan mudah diakses. Mikrokontroler ESP32 menjadi salah satu pilihan potensial karena memiliki kemampuan pemrosesan tinggi, konektivitas nirkabel Wi-Fi dan Bluetooth, serta harga yang relatif terjangkau (Espressif, 2023). Sejalan dengan itu, meningkatnya penetrasi smartphone di Indonesia lebih dari 197 juta pengguna pada 2023 (Statista, 2023) memungkinkan integrasi visualisasi sinyal EKG secara real-time melalui aplikasi berbasis website.

Sejumlah penelitian terdahulu telah mengonfirmasi potensi ini. Rani et al. (2022) mengembangkan sistem pemantauan EKG real-time berbasis ESP32 dengan visualisasi sinyal yang akurat melalui Wi-Fi. Nugroho et al. (2023) juga berhasil merancang alat pemantau detak jantung berbasis ESP32 yang terintegrasi dengan smartphone Android. Sementara itu, Karthik et al. (2021) menunjukkan stabilitas dan akurasi sistem EKG 1-lead nirkabel berbasis ESP32. Temuan-temuan tersebut menguatkan urgensi pengembangan sistem EKG yang lebih portabel, efisien, dan mendukung telemedisin.

Berdasarkan latar belakang tersebut, penelitian ini bertujuan untuk merancang, mengimplementasikan, dan menguji sistem pemantauan EKG 1-lead berbasis mikrokontroler ESP32 yang dilengkapi konektivitas Wi-Fi dan visualisasi real-time pada smartphone. Secara khusus, penelitian ini diarahkan untuk menghasilkan sistem akuisisi sinyal jantung yang dapat diproses dan ditransmisikan secara efisien, menampilkan visualisasi sinyal EKG melalui aplikasi berbasis website, serta mengevaluasi performa sistem dalam hal akurasi dan kestabilan dengan membandingkannya terhadap perangkat EKG konvensional.

2. LANDASAN TEORI

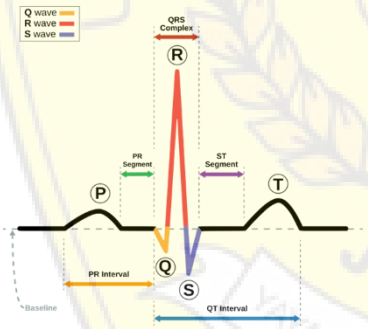
2.1 Elektrokardiogram (EKG)

Elektrokardiogram (EKG) merupakan salah satu metode pemeriksaan medis non-invasif yang digunakan untuk merekam aktivitas listrik jantung. Prinsip kerja EKG adalah mendeteksi perbedaan potensial listrik

https://doi.org/10.37817/ikraith-informatika.v10i2
P-ISSN : 2580-4316
E-ISSN : 2654-8054

yang timbul akibat kontraksi otot jantung melalui elektroda yang ditempelkan pada permukaan tubuh. Sinyal listrik tersebut kemudian ditampilkan dalam bentuk gelombang yang terdiri atas P wave, QRS complex, dan T wave, yang masing-masing mewakili fase aktivitas jantung tertentu (Goldberger, 2017).

EKG menjadi alat penting untuk mendiagnosis berbagai gangguan jantung, seperti aritmia, iskemia miokard, hingga infark miokard akut. Pemeriksaan standar biasanya menggunakan konfigurasi 12-lead untuk memberikan gambaran menyeluruh aktivitas listrik jantung dari berbagai sudut pandang. Namun, untuk pemantauan sederhana atau penggunaan portabel, konfigurasi EKG 1-lead sudah cukup memadai. Meskipun tidak sekomprehensif 12-lead, sistem 1-lead mampu memberikan informasi dasar terkait ritme jantung dan detak per menit, sehingga sering digunakan dalam perangkat pemantauan mandiri dan sistem telemedisin.



Gambar 1. Diagram Mesin EKG Konvensional

2.2 Mikrokontroler ESP32

ESP32 merupakan mikrokontroler modern yang banyak digunakan dalam berbagai aplikasi Internet of Things (IoT), termasuk sistem pemantauan kesehatan berbasis sensor. Dikembangkan oleh Espressif Systems, mikrokontroler ini dikenal karena integrasi fitur komunikasi nirkabel, efisiensi daya, serta kemampuan pemrosesan yang tinggi dalam satu chip (Espressif Systems, 2023).

Sebagai sebuah System-on-Chip (SoC), ESP32 menggabungkan CPU dual-core Tensilica Xtensa LX6, konektivitas Wi-Fi dan Bluetooth, serta berbagai periferal seperti ADC, DAC, SPI, dan I²C. Mikrokontroler ini mampu beroperasi hingga 240 MHz dan memiliki fitur *deep sleep* yang memungkinkan konsumsi daya sangat rendah, sehingga sangat cocok untuk perangkat portabel berbasis baterai (Espressif Systems, 2023).

Dalam sistem pemantauan EKG, ESP32 berfungsi sebagai unit pemrosesan utama. Mikrokontroler ini menerima sinyal analog dari sensor EKG, mengubahnya menjadi data digital melalui ADC internal, dan kemudian mentransmisikannya secara nirkabel ke antarmuka pengguna. Menurut Islam dan Khan (2021), "ESP32 memiliki kemampuan pemrosesan sinyal yang cukup untuk menangani data fisiologis seperti EKG, serta mendukung protokol komunikasi MQTT yang ringan dan efisien untuk transmisi data real-time."

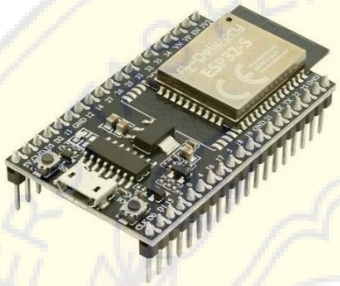
Keunggulan ESP32 dalam aplikasi kesehatan telah terbukti melalui berbagai penelitian. Rani et al. (2022) mengembangkan sistem pemantauan EKG real-time menggunakan ESP32 dengan antarmuka berbasis web, yang menunjukkan stabilitas dan akurasi tinggi dalam pengiriman data via Wi-Fi. Hal ini menunjukkan bahwa ESP32 dapat menjadi alternatif yang efisien dan portabel dibandingkan mikrokontroler konvensional dalam aplikasi medis.

Selain itu, ESP32 kompatibel dengan berbagai sensor biomedis seperti AD8232, MAX30100, dan ADS1292R, yang mendukung pengembangan sistem monitoring multi-parametrik. Styazhkina et al. (2020) menyatakan bahwa penggunaan mikrokontroler dengan kemampuan komunikasi nirkabel dan pemrosesan sinyal lokal sangat penting

P-ISSN : 2580-4316
E-ISSN : 2654-8054 https://doi.org/10.37817/ikraith-informatika.v10i2

untuk perangkat medis yang dapat digunakan di luar fasilitas klinis.

Islam dan Khan (2021) juga menekankan bahwa CPU dual-core pada ESP32 memungkinkan penanganan tugas kompleks seperti akuisisi dan pemrosesan sinyal EKG secara efisien. Fitur konsumsi daya rendah, terutama dalam mode sleep, membuat ESP32 ideal untuk perangkat portabel bertenaga baterai.



Gambar 2. Mikrokontroler ESP32

Pada proyek ini, ESP32 berperan sebagai "otak" perangkat EKG portabel, menerima sinyal listrik dari jantung melalui sensor, memprosesnya, dan mengirimkan data secara nirkabel ke smartphone. Mikrokontroler ini juga umum digunakan pada perangkat wearable, sistem smart home, dan drone.

Kelebihan ESP32:

1. **Harga Terjangkau**: Memungkinkan pengembangan perangkat medis yang ekonomis.
2. **Portabilitas**: Ukuran kecil dan konsumsi daya rendah cocok untuk perangkat yang mudah dibawa.
3. **Konektivitas Lengkap**: Integrasi Wi-Fi dan Bluetooth memudahkan komunikasi nirkabel.
4. **Pemrosesan Data**: CPU dual-core mampu memproses sinyal EKG secara efisien.

a) Spesifikasi Teknis

Tabel 1. Spesifikasi ESP32

Parameter	Nilai/Deskripsi
Prosesor	Dual-core Tensilica LX6
Frekuensi CPU	Hingga 240 MHz
Memori	520 KB RAM
ADC	12-bit, hingga 18 kanal
Komunikasi	Wi-Fi 802.11 b/g/n, Bluetooth v4.2
Tegangan Operasional	3.0V – 3.6V
Protokol IoT	Mendukung MQTT, HTTP, WebSocket

b) Peran dalam Sistem

ESP32 memiliki beberapa fungsi penting:

1) Akuisisi Data: Membaca sinyal analog dari sensor AD8232 melalui pin ADC.
2) Pemrosesan Awal: Melakukan filter digital, perhitungan BPM, dan deteksi lead-off.
3) Pengemasan Data: Mengubah data menjadi format JSON untuk komunikasi.
4) Pengiriman Data: Mengirim data ke broker MQTT untuk diteruskan ke Node-RED.

c) Alur Komunikasi

Gambar 3. *Payload JSON*

Sebagai *publisher* dalam arsitektur MQTT, ESP32 mengirim data yang telah diproses ke topik tertentu di broker MQTT. Subscriber

seperti Node-RED kemudian dapat mengakses data tersebut, memungkinkan komunikasi asinkron dan efisien.

d) Keunggulan dan Keterbatasan

Keunggulan:

1) Konektivitas nirkabel tanpa modul tambahan.
2) Kompatibel dengan berbagai protokol IoT.
3) Konsumsi daya rendah dan ukuran ringkas.
4) Komunitas pengembang luas dan dokumentasi lengkap.

Keterbatasan:

1) ADC internal lebih berisik dibanding ADC eksternal.
2) Penggunaan Wi-Fi dan Bluetooth bersamaan dapat mempengaruhi stabilitas.
3) Membutuhkan manajemen memori yang efisien untuk aplikasi real-time.

2.3 Komunikasi *Wireless Fidelity* (Wi-Fi)

Wi-Fi atau Wireless Fidelity merupakan teknologi komunikasi nirkabel yang memungkinkan pertukaran data antarperangkat melalui gelombang radio tanpa membutuhkan kabel fisik. Teknologi ini sangat penting dalam pengembangan perangkat Internet of Things (IoT), termasuk sistem pemantauan kesehatan, karena memudahkan pengiriman data secara real-time dari sensor ke antarmuka pengguna.

ESP32 dilengkapi modul Wi-Fi yang mendukung standar 802.11 b/g/n, memungkinkan konektivitas cepat dan stabil dalam jaringan lokal maupun internet (Espressif Systems, 2023). Dalam konteks sistem pemantauan EKG, Wi-Fi berperan sebagai jalur utama untuk mentransmisikan data yang telah diproses dari mikrokontroler ke perangkat lain, seperti smartphone atau server berbasis web. Menurut Rani et al. (2022), penggunaan Wi-Fi dalam sistem monitoring kesehatan berbasis ESP32 terbukti efektif dalam menjaga kestabilan pengiriman data secara real-time, sehingga informasi fisiologis dapat diakses secara akurat dan cepat oleh pengguna.

Keunggulan Wi-Fi pada ESP32 antara lain:

1. Transmisi Data Cepat: Memungkinkan pengiriman sinyal EKG dan data sensor lainnya secara real-time.
2. Kompatibilitas Luas: Bisa terhubung dengan berbagai perangkat, termasuk laptop, smartphone, dan server.
3. Mendukung Jaringan Lokal dan Internet: Data dapat dikirimkan dalam skala lokal maupun secara global melalui internet.

2.4 Protokol MQTT

MQTT (*Message Queuing Telemetry Transport*) adalah protokol komunikasi ringan yang dirancang untuk pertukaran data secara efisien di jaringan dengan bandwidth terbatas. Protokol ini banyak digunakan dalam aplikasi Internet of Things (IoT) karena kemampuannya mengirimkan data secara real-time dengan konsumsi daya dan sumber daya yang minimal (Islam & Khan, 2021).

Dalam sistem pemantauan EKG berbasis ESP32, MQTT berfungsi sebagai mekanisme untuk mentransfer data dari mikrokontroler ke broker yang kemudian dapat diakses oleh perangkat lain, seperti Node-RED atau antarmuka berbasis web. Menurut Rani et al. (2022), penggunaan MQTT memungkinkan pengiriman data fisiologis secara asinkron dan andal, sehingga pemantauan kondisi pasien

P-ISSN : 2580-4316
E-ISSN : 2654-8054 https://doi.org/10.37817/ikraith-informatika.v10i2

dapat dilakukan secara kontinu tanpa gangguan pada konektivitas.

Protokol MQTT bekerja dengan konsep publish-subscribe, di mana ESP32 bertindak sebagai publisher yang mengirim data ke topic tertentu di broker, dan perangkat lain yang berperan sebagai subscriber akan menerima data tersebut. Metode ini membuat komunikasi menjadi lebih fleksibel dan hemat sumber daya dibandingkan komunikasi langsung satu-ke-satu (point-to-point) (Styazhkina et al., 2020). Keunggulan penggunaan MQTT dalam sistem monitoring kesehatan antara lain:

1) Ringan dan Efisien: Memerlukan bandwidth rendah dan cocok untuk perangkat dengan sumber daya terbatas.

2) Komunikasi Asinkron: Data dapat dikirim dan diterima secara fleksibel tanpa menunggu respon langsung.

3) Skalabilitas Tinggi: Memudahkan penambahan perangkat baru ke jaringan tanpa mengganggu sistem yang sudah berjalan.

2.5 Node Red

Node-RED adalah platform pengembangan berbasis alur kerja (*flow-based development*) yang memungkinkan integrasi dan pengolahan data secara visual menggunakan *drag-and-drop* tanpa memerlukan pemrograman yang kompleks (Rani et al., 2022). Platform ini banyak digunakan dalam aplikasi Internet of Things (IoT) karena kemampuannya untuk menghubungkan berbagai perangkat, protokol, dan layanan secara efisien.

Dalam sistem pemantauan EKG berbasis ESP32, Node-RED berperan sebagai *subscriber* yang menerima data dari *broker* MQTT. Data EKG yang telah dikirim oleh ESP32 kemudian dapat diproses lebih lanjut, divisualisasikan, dan ditampilkan pada antarmuka berbasis web atau perangkat lain. Menurut Styazhkina et al. (2020), penggunaan Node-RED mempermudah pengembangan sistem monitoring kesehatan karena menyediakan *interface* yang intuitif untuk memanipulasi dan menampilkan data sensor secara real-time.

Beberapa keunggulan Node-RED dalam sistem monitoring kesehatan meliputi:

1. **Visualisasi Data**: Data yang diterima dapat langsung ditampilkan dalam bentuk grafik atau indikator real-time.
2. **Integrasi Mudah**: Memungkinkan koneksi dengan berbagai protokol IoT seperti MQTT, HTTP, dan WebSocket.
3. **Pengembangan Cepat**: Alur kerja berbasis *drag-and-drop* mempercepat implementasi sistem tanpa pemrograman kompleks.
4. **Fleksibilitas**: Mempermudah penambahan fitur baru atau integrasi dengan layanan cloud dan database.

2.6 Pemrosesan Sinyal EKG

Elektrokardiogram (EKG) merupakan sinyal listrik yang berasal dari aktivitas jantung dan digunakan untuk menggambarkan kondisi fisiologis pasien. Agar dapat dianalisis dengan baik, sinyal EKG perlu melalui tahapan pemrosesan mulai dari akuisisi, konversi data, hingga visualisasi. Menurut Islam dan Khan (2021), pemrosesan sinyal ini penting untuk memastikan bahwa data yang diterima dari sensor memiliki kualitas yang cukup sehingga dapat ditampilkan dan dianalisis secara real-time.

Pada sistem berbasis ESP32, sinyal analog dari sensor EKG seperti AD8232 pertama-tama dibaca melalui kanal *Analog to Digital Converter* (ADC).

Proses ini mengubah sinyal listrik analog dari jantung menjadi data digital agar dapat diproses lebih lanjut oleh mikrokontroler. Setelah itu, dilakukan pemrosesan awal seperti filter digital untuk mengurangi *noise*, perhitungan BPM (*beats per minute*), serta deteksi kondisi *lead-off* ketika elektroda tidak terpasang dengan benar.

Rani et al. (2022) menunjukkan bahwa pemrosesan sinyal EKG berbasis ESP32 mampu memberikan hasil yang stabil dan akurat, bahkan ketika data dikirim secara nirkabel melalui Wi-Fi. Hal ini membuktikan bahwa mikrokontroler ini memiliki kapasitas pemrosesan cukup untuk menangani sinyal fisiologis secara efisien.

Selain itu, pemrosesan sinyal EKG yang dilakukan di sisi perangkat (lokal) sangat penting untuk mengurangi beban jaringan dan meningkatkan kecepatan respon. Styazhkina et al. (2020) menekankan bahwa pemrosesan lokal pada perangkat portabel memberikan keuntungan dalam pemantauan kesehatan di luar fasilitas medis, karena memungkinkan pasien tetap termonitor tanpa harus selalu berada di rumah sakit.
Dengan demikian, tahapan pemrosesan sinyal EKG dalam penelitian ini mencakup:

1. **Akuisisi Data** – Membaca sinyal listrik jantung melalui sensor EKG.
2. **Konversi Analog ke Digital** – Menggunakan ADC internal ESP32 untuk menghasilkan data digital.
3. **Pemrosesan Awal** – Meliputi filter digital, penghitungan BPM, dan deteksi *lead-off*.
4. **Pengemasan Data** – Mengubah data ke dalam format JSON agar sesuai dengan protokol komunikasi IoT.
5. **Transmisi Data** – Mengirim hasil pemrosesan ke broker MQTT dan divisualisasikan melalui Node-RED.

2.7 Pengembangan Antar Muka Berbasis Website

Antarmuka berbasis website menjadi komponen penting dalam sistem EKG berbasis IoT karena menjadi media interaksi utama antara pengguna dan sistem. Melalui dashboard ini, pengguna dapat memantau sinyal EKG secara *real-time*, melihat jumlah detak jantung per menit (BPM), hingga menerima notifikasi jika terjadi gangguan pada aktivitas jantung.

Pada penelitian ini, antarmuka dibangun menggunakan Node-RED, sebuah platform visual berbasis *flow* yang mendukung integrasi dengan berbagai protokol komunikasi, termasuk MQTT. Pemilihan Node-RED didasari beberapa keunggulan: mudah diintegrasikan dengan ESP32, mendukung visualisasi data real-time, pengembangannya cepat tanpa kode kompleks, serta dashboard yang dihasilkan dapat langsung diakses melalui browser.
Antarmuka yang dirancang terdiri atas beberapa komponen utama, yaitu grafik EKG untuk menampilkan sinyal secara real-time, indikator BPM, status lead-off untuk mengecek pemasangan elektroda, waktu dan tanggal data, serta notifikasi anomali jika BPM berada di luar batas normal. Semua komponen diatur dalam layout dashboard yang responsif sehingga nyaman diakses melalui desktop maupun perangkat mobile.

2.8 Metode *Waterfall*

Model *Waterfall* merupakan salah satu metode pengembangan perangkat lunak yang paling klasik dan terstruktur. Pendekatan ini menekankan alur kerja yang bersifat

P-ISSN : 2580-4316
E-ISSN : 2654-8054 https://doi.org/10.37817/ikraith-informatika.v10i2

linear, di mana setiap tahap harus diselesaikan terlebih dahulu sebelum melanjutkan ke tahap berikutnya (Pressman, 2015). Tahapannya biasanya meliputi analisis kebutuhan, desain sistem, implementasi, pengujian, hingga pemeliharaan.

Kelebihan utama metode ini adalah dokumentasi yang lengkap serta alur kerja yang jelas, sehingga memudahkan pengembang dalam memantau setiap proses. Hal ini sangat sesuai untuk proyek dengan kebutuhan yang sudah terdefinisi dengan baik sejak awal, termasuk dalam penelitian ini yang berfokus pada perancangan sistem EKG portabel. Dengan menggunakan pendekatan Waterfall, proses pengembangan dapat berjalan lebih sistematis dan mengurangi risiko kesalahan pada tahap implementasi (Sommerville, 2016).

3. METODOLOGI

Jenis Penelitian

Penelitian ini menggunakan metode Research and Development (R&D). Fokusnya adalah menghasilkan produk berupa sistem pemantauan EKG 1-lead yang portabel, terhubung dengan smartphone, dan menampilkan data secara real-time. Produk yang dikembangkan meliputi perangkat keras (sensor, mikrokontroler) dan perangkat lunak (aplikasi berbasis website).

Tahapan Perancangan Sistem

Pengembangan sistem menggunakan model Waterfall, karena sesuai untuk proyek dengan kebutuhan yang jelas. Tahapannya mencakup:

a) Analisis kebutuhan: Mengidentifikasi kebutuhan fungsional seperti akuisisi sinyal EKG, transmisi data melalui Wi-Fi, dan visualisasi real-time, serta kebutuhan non-fungsional seperti keamanan dan kestabilan sistem.
b) Desain sistem: Meliputi rancangan perangkat keras (sensor EKG, ESP32, komunikasi Wi-Fi) dan perangkat lunak (firmware ESP32, antarmuka berbasis website dengan Node-RED, dan format data).
c) Implementasi: Mewujudkan desain ke dalam perakitan perangkat keras dan penulisan kode perangkat lunak.
d) Pengujian: Meliputi uji unit, integrasi, dan sistem secara menyeluruh untuk memastikan akurasi sinyal, kestabilan koneksi, serta kesesuaian fitur dengan kebutuhan.

Metode Pengumpulan Data

Data penelitian diperoleh melalui dua cara:

Studi literatur, dengan menelaah buku, jurnal, artikel ilmiah, serta penelitian terdahulu yang relevan.

Pengujian sistem, dengan menguji perangkat yang sudah dikembangkan menggunakan sinyal simulasi dan hasil pembacaan sensor.

Pendekatan Pengujian Performa

a) Pengujian performa dilakukan dengan menilai:
b) Akurasi sinyal: membandingkan hasil sistem dengan sinyal dari simulator atau alat EKG standar.
c) Stabilitas sinyal dan koneksi: mengamati noise, gangguan koneksi, serta keterlambatan data.
d) Metrik performa: kualitas gelombang PQRST, keakuratan BPM, kestabilan garis dasar, dan kejelasan sinyal.

https://doi.org/10.37817/ikraith-informatika.v10i2
P-ISSN : 2580-4316
E-ISSN : 2654-8054

Alat dan Bahan Penelitian

Alat yang digunakan meliputi: mikrokontroler ESP32, modul sensor EKG (AD8232), elektroda, breadboard, kabel jumper, baterai, smartphone Android, komputer/laptop, serta alat EKG standar sebagai pembanding. Perangkat lunak yang digunakan antara lain Arduino IDE/ESP-IDF untuk ESP32, Node-RED untuk dashboard, dan bahasa pemrograman C++ atau Java/Kotlin untuk aplikasi. Data yang digunakan berupa literatur relevan serta hasil pengujian sinyal simulasi dan aktual.

4. HASIL DAN PEMBAHASAN

1. Tahap Analisis Sistem

Tahap analisis merupakan langkah awal dalam metode Waterfall dan berfungsi sebagai pijakan penting sebelum masuk ke tahap perancangan. Pada tahap ini, peneliti berusaha merumuskan secara jelas apa saja yang dibutuhkan oleh sistem, baik dari sisi fungsi utama yang harus dijalankan maupun kualitas kinerja yang diharapkan.

Sistem yang dikembangkan dalam penelitian ini adalah aplikasi pemantauan detak jantung yang memanfaatkan mikrokontroler ESP32 dan platform Node-RED. Sistem ini dirancang agar mampu membaca sinyal jantung dari sensor, mengolahnya menjadi nilai BPM, lalu menampilkannya secara real-time melalui antarmuka web yang bisa diakses lewat smartphone.

1.1.1 Kebutuhan Fungsional

Kebutuhan fungsional berkaitan langsung dengan apa saja yang harus bisa dilakukan sistem agar berjalan sesuai tujuan. Beberapa fungsi utama antara lain: membaca sinyal dari sensor AD8232 dan mengubahnya menjadi BPM, mengirimkan data tersebut secara rutin ke broker MQTT, kemudian memprosesnya di Node-RED agar bisa divisualisasikan dalam bentuk grafik. Selain itu, sistem juga harus mampu mengklasifikasikan kondisi jantung berdasarkan nilai BPM, misalnya normal, terlalu cepat (tachycardia), terlalu lambat (bradycardia), atau aritmia. Jika kondisi tidak normal terdeteksi, sistem wajib memberikan peringatan berupa notifikasi visual maupun suara, serta menyediakan tombol untuk menghapus grafik agar pengguna bisa memulai sesi pemantauan baru.

1.1.2 Kebutuhan Non-Fungsional

Di luar fungsi utama, sistem juga dituntut memenuhi aspek kualitas tertentu agar benar-benar layak digunakan. Misalnya, data harus ditampilkan secara real-time dengan keterlambatan maksimal 200 milidetik, dashboard harus bisa diakses dengan mudah melalui browser smartphone tanpa instalasi tambahan, dan koneksi data harus stabil menggunakan protokol MQTT. Selain itu, sistem harus mampu berjalan dalam waktu lama tanpa gangguan, memiliki tampilan antarmuka yang sederhana dan mudah dipahami, serta menjaga keamanan data. Karena bersifat prototipe, sistem juga dirancang modular agar di masa depan bisa dikembangkan lebih lanjut, misalnya menambahkan fitur penyimpanan data historis atau analisis lanjutan.

1.1.3 Asumsi dan Kendala

Dalam tahap analisis ini, ada beberapa asumsi dan keterbatasan yang perlu dicatat. Pertama, sistem hanya berfokus pada pengukuran BPM tanpa menganalisis bentuk gelombang ECG secara detail. Kedua, pengujian dilakukan di lingkungan umum, bukan

P-ISSN : 2580-4316
E-ISSN : 2654-8054 https://doi.org/10.37817/ikraith-informatika.v10i2

dalam setting klinis, sehingga hasil yang diperoleh hanya bersifat indikatif. Ketiga, diasumsikan koneksi internet tersedia dengan baik saat sistem digunakan. Terakhir, sensor AD8232 digunakan tanpa proses kalibrasi medis, sehingga data yang dihasilkan tidak bisa dijadikan dasar diagnosis medis resmi.

1.2 Tahap Desain Sistem

dilakukan setelah analisis kebutuhan selesai, dengan tujuan untuk menggambarkan rancangan sistem secara menyeluruh sebelum masuk ke tahap implementasi. Pada bagian ini, dirancang arsitektur, alur data, komponen perangkat keras, perangkat lunak, hingga tampilan antarmuka pengguna. Dengan adanya desain yang matang, proses implementasi menjadi lebih terarah dan sistem dapat dipastikan sesuai dengan kebutuhan fungsional maupun non-fungsional yang telah ditentukan.

Sistem yang dirancang dalam penelitian ini merupakan pemantauan detak jantung berbasis ESP32 dengan dukungan Node-RED sebagai pengolah dan penyaji data. Hasil pemantauan ditampilkan melalui dashboard berbasis web yang bisa diakses secara real-time menggunakan smartphone.

1.2.1 Arsitektur Umum

Arsitektur sistem menggambarkan hubungan antar komponen utama. Sensor ECG (AD8232) berfungsi menangkap sinyal listrik jantung dari tubuh, kemudian mikrokontroler ESP32 mengolah sinyal tersebut menjadi data digital dan menghitung nilai BPM. Data ini dikirimkan melalui broker MQTT yang bertindak sebagai perantara komunikasi dengan Node-RED. Node-RED kemudian menerima, memproses, dan mengklasifikasikan data, sebelum menampilkannya dalam bentuk grafik dan notifikasi pada dashboard web. Dengan alur ini, pengguna dapat memantau kondisi jantung secara langsung melalui perangkat smartphone.

1.2.2 Desain Perangkat Keras

Perangkat keras utama yang digunakan dalam sistem ini antara lain:

a) ESP32 DevKit V1 sebagai pengendali utama dengan kemampuan Wi-Fi.
b) Sensor AD8232 untuk menangkap sinyal jantung.
c) Elektroda dan kabel jumper untuk menghubungkan sensor ke tubuh dan mikrokontroler.
d) Breadboard atau PCB sebagai media perakitan rangkaian.

1.2.4 Desain Perangkat Lunak

Perangkat lunak dalam sistem ini terdiri dari dua bagian, yaitu firmware pada ESP32 dan alur (flow) pada Node-RED.

a) Firmware ESP32: dikembangkan menggunakan Arduino IDE. Fungsinya meliputi inisialisasi koneksi Wi-Fi dan MQTT, membaca sinyal analog dari sensor, menghitung BPM, serta mengirimkan data ke broker MQTT dalam format tertentu.

b) Flow Node-RED: digunakan untuk memproses data yang masuk, baik berupa data dummy (untuk simulasi) maupun data asli dari sensor. Komponen Node-RED akan menerima data MQTT, menyimpannya sementara, menghitung rata-rata, menentukan kondisi jantung, lalu menampilkannya dalam grafik. Selain itu, Node-RED juga dilengkapi dengan notifikasi visual dan suara, serta tombol untuk menghapus grafik dan memulai sesi baru. Komponen utama dalam flow:

c) mqtt in: menerima data dari topik tertentu melalui jaringan (Wi-Fi).

P-ISSN : 2580-4316
https://doi.org/10.37817/ikraith-informatika.v10i2 E-ISSN : 2654-8054

d) function: menyimpan buffer data, menghitung rata-rata, dan menentukan kondisi jantung.

e) ui_chart: menampilkan grafik BPM secara real-time.

f) ui_toast dan ui_audio: memberikan notifikasi visual dan suara.

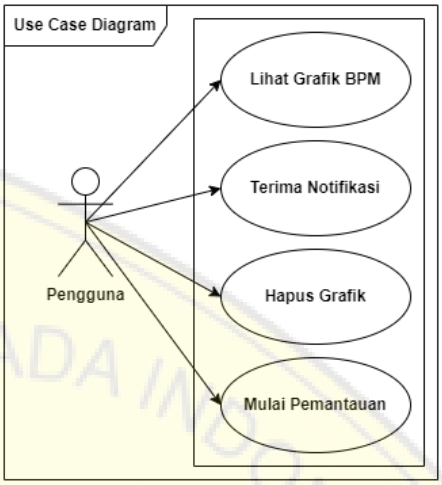
g) ui_button: digunakan untuk menghapus grafik dan memulai sesi baru.

1.2.5 Desain Antarmuka Pengguna

Antarmuka pengguna dirancang agar sederhana, informatif, dan mudah diakses melalui browser smartphone. Beberapa elemen utama pada dashboard meliputi: grafik BPM real-time dengan rentang 10–200 bpm, notifikasi pop-up dan suara jika kondisi jantung tidak normal, tombol "Clear Graph" untuk memulai sesi baru, serta label dan warna yang jelas agar mudah dipahami oleh pengguna.

1.2.5 Pemodelan Sistem (UML)

Untuk memperjelas rancangan, sistem juga digambarkan menggunakan diagram UML. Diagram Use Case digunakan untuk menunjukkan interaksi antara pengguna dengan sistem. Dalam hal ini, aktor utamanya adalah pengguna yang dapat melakukan beberapa fungsi, seperti melihat grafik BPM, menerima notifikasi, menghapus grafik, dan memulai pemantauan baru. Diagram ini membantu memperjelas hubungan antara kebutuhan pengguna dengan fungsi yang ada pada sistem.



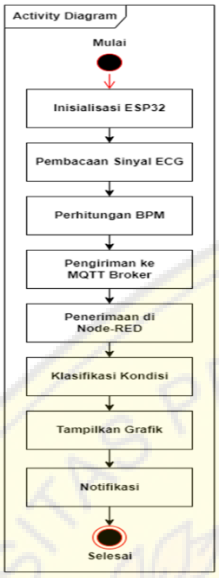
Gambar 4. UML-Use Case Diagram

1.2.6 Activity Diagram

Activity Diagram digunakan untuk menggambarkan alur aktivitas dalam sistem yang sedang dirancang. Diagram ini membantu memetakan urutan proses dari awal hingga akhir, sehingga memudahkan pemahaman bagaimana sistem bekerja. Setiap aktivitas direpresentasikan dengan simbol tertentu yang dihubungkan oleh panah sebagai penunjuk arah alur kerja.

Pada sistem ini, alur aktivitas dimulai dari proses inisialisasi ESP32, kemudian berlanjut ke pembacaan sinyal ECG, dilanjutkan dengan perhitungan BPM. Set elah itu, data yang diperoleh dikirimkan ke broker MQTT, kemudian diteruskan ke Node-RED untuk diproses. Node-RED melakukan klasifikasi kondisi jantung, lalu hasilnya divisualisasikan dalam bentuk grafik real-time, serta memberikan notifikasi apabila terdeteksi kondisi tidak normal. Dengan demikian, diagram ini memperlihatkan gambaran umum bagaimana data diproses mulai dari sensor hingga ditampilkan ke pengguna.

P-ISSN : 2580-4316
E-ISSN : 2654-8054 https://doi.org/10.37817/ikraith-informatika.v10i2

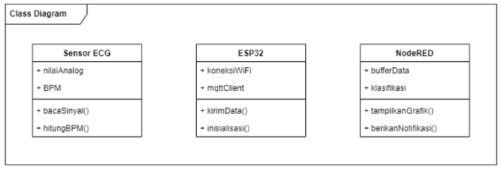


Gambar 5. UML-Activity Diagram

1.2.7 Class Diagram

Class Diagram adalah salah satu jenis diagram UML yang digunakan untuk menggambarkan struktur objek dalam sistem. Diagram ini menjelaskan kelas-kelas utama, atribut, metode, serta hubungan antar kelas yang bersifat statis. Artinya, diagram ini tidak memperlihatkan alur proses secara dinamis, melainkan menunjukkan bagaimana komponen dalam sistem saling terhubung secara konseptual. Diagram ini memodelkan struktur kelas dalam sistem untuk membantu merancang struktur kode dan relasi antar komponen, yaitu:

a) SensorECG: memiliki atribut nilaiAnalog dan bpm, serta metode bacaSinyal() dan hitungBPM().
b) ESP32: memiliki koneksiWiFi dan mqttClient, serta metode kirimData() dan inisialisasi().
c) NodeRED: menyimpan bufferData dan klasifikasi, serta metode tampilkanGrafik() dan beriNotifikasi().



Gambar 5. UML-Class Diagram

1.3 Tahap Implementasi Sistem

Tahap implementasi merupakan proses menerjemahkan rancangan sistem ke bentuk nyata, baik pada sisi perangkat keras maupun perangkat lunak. Pada tahap ini seluruh komponen digabungkan sehingga sistem pemantauan detak jantung dapat berfungsi secara utuh. Proses implementasi dilakukan secara bertahap, dimulai dari perakitan hardware, penulisan firmware pada ESP32, konfigurasi komunikasi MQTT, hingga pembuatan antarmuka pengguna berbasis Node-RED.

1.3.1 Implementasi Perangkat Keras

Bagian perangkat keras diawali dengan merakit sensor ECG **AD8232** yang dilengkapi elektroda untuk menangkap sinyal listrik jantung. Sensor ini dihubungkan ke mikrokontroler **ESP32** melalui pin ADC untuk mengubah sinyal analog menjadi data digital. Proses perakitan menggunakan breadboard dan kabel jumper agar koneksi antar komponen mudah diatur. Setelah elektroda dipasang pada tubuh (RA, LA, RL), kestabilan sinyal diuji menggunakan *serial plotter*. Dengan cara ini, diperoleh data detak jantung dalam bentuk sinyal analog yang siap diolah lebih lanjut.

Implementasi Firmware ESP32

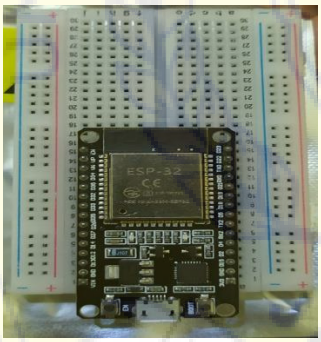
Pada sisi firmware, ESP32 diprogram menggunakan **Arduino IDE** dengan bahasa pemrograman C++. Beberapa fungsi utama firmware meliputi:

1. Inisialisasi koneksi Wi-Fi menggunakan SSID dan password lokal.

https://doi.org/10.37817/ikraith-informatika.v10i2
P-ISSN : 2580-4316
E-ISSN : 2654-8054

2. Menyambungkan modul ke broker MQTT publik (*mqtt-dashboard.com*).
3. Membaca sinyal analog dari sensor melalui pin ADC.
4. Menghitung nilai BPM berdasarkan deteksi puncak gelombang (R-peak).
5. Mengirim data BPM ke topik **esp32-ecg/bpm** dalam format JSON lengkap dengan *timestamp* dan *device ID*.

Gambar 6. True Heart Rate Board



Gambar 7. Dummy Heart Rate Board

1.3.3 Implementasi Firmware ESP32

Firmware dikembangkan menggunakan Arduino IDE dengan bahasa pemrograman C++. Fungsi utama firmware meliputi:

a) Inisialisasi koneksi Wi-Fi menggunakan SSID dan password lokal
b) Inisialisasi koneksi MQTT ke broker publik (mqtt-dashboard.com)
c) Pembacaan sinyal analog dari pin ADC
d) Perhitungan BPM berdasarkan interval detak (peak detection)
e) Pengiriman data BPM ke topik MQTT esp32-ecg/bpm

Berikut ini adalah potongan kode untuk Arduino atau PlatformIO sketch: inisialisasi Wi-Fi, MQTT, dan timer sampling, yaitu:

```
// 1_init_esp32.ino (snippet)
#include <WiFi.h>
#include <PubSubClient.h>

const char* SSID = "YourSSID";
const char* PASS = "YourPassword";

const char* MQTT_SERVER = "mqtt.example.com";
const uint16_t MQTT_PORT = 1883;
const char* MQTT_TOPIC = "esp32-ecg/bpm";

WiFiClient espClient;
PubSubClient mqtt(espClient);

void setupWifi() {
  Serial.print("Connecting to Wi-Fi ");
  Serial.print(SSID);
  WiFi.begin(SSID, PASS);
  unsigned long start = millis();
  while (WiFi.status() != WL_CONNECTED && millis() - start < 10000) {
    delay(200);
    Serial.print(".");
  }
  if (WiFi.status() == WL_CONNECTED) {
    Serial.println("\nWi-Fi connected. IP: " +
WiFi.localIP().toString());
    Serial.println("\nWi-Fi connect failed. Proceeding offline.");
  }
}

void connectMQTT() {
  mqtt.setServer(MQTT_SERVER, MQTT_PORT);
  while (!mqtt.connected()) {
    Serial.print("Connecting MQTT... ");
    if (mqtt.connect("esp32-ecg-01")) {
      Serial.println("connected");
    } else {
      Serial.print("failed, rc=");
      Serial.print(mqtt.state());
      delay(1000);
    }
  }
}

void setup() {
  Serial.begin(115200);
  setupWifi();
  connectMQTT()
}

void loop() {
  if (!mqtt.connected()) connectMQTT();
  mqtt.loop();
}
```

Gambar 8. Inisialisasi ESP32

```
// 2_read_ecg.ino (snippet)
const int ECG_PIN = 34;          // ADC input
const int SAMPLING_HZ = 250;     // target sampling frequency
const unsigned long SAMPLING_US = 1000000UL / SAMPLING_HZ;

#define BUFFER_SIZE 1024
volatile int ecgBuffer[BUFFER_SIZE];
volatile int bufIndex = 0;

void sampleECGonce() {
  static unsigned long nextMicros = 0;
  unsigned long now = micros();
  if (now >= nextMicros) {
    int raw = analogRead(ECG_PIN); // 12-bit ADC typically
    ecgBuffer[bufIndex++] = raw;
    if (bufIndex >= BUFFER_SIZE) bufIndex = 0;
    nextMicros = now + SAMPLING_US;
  }
}
```

Gambar 9. Pembacaan sinyal ECG

P-ISSN : 2580-4316
E-ISSN : 2654-8054 https://doi.org/10.37817/ikraith-informatika.v10i2

SAMPLING_HZ = 250 adalah nilai umum untuk EKG 1-lead prototipe (bisa naik ke 500 Hz).

ecgBuffer menyimpan sample sementara; Anda bisa mengirim potongan (window) ke algoritma deteksi puncak.

```
// 3_calc_bpm.ino (snippet)
#include <vector>

const int MIN_PEAK_DISTANCE_MS = 250;  // batas minimal antar puncak ~ 240-300 ms

unsigned long lastPeakTime = 0;
std::vector<unsigned long> ibiList;    // simpan beberapa IBI untuk smoothing

int detectPeakInWindow(int* samples, int len, int threshold) {
  // sederhana: cari sample yang > threshold dan lebih besar dari tetangganya
  for (int i = 1; i < len-1; i++) {
    if (samples[i] > threshold && samples[i] > samples[i-1] && samples[i] >= samples[i+1]) {
      return i;
    }
  }
  return -1;
}

float computeBPMfromIBIList() {
  if (ibiList.size() == 0) return 0.0;
  // gunakan rata-rata IBI (ms)
  unsigned long sum = 0;
  for (auto &v : ibiList) sum += v;
  float meanIBI = (float)sum / (float)ibiList.size();
  if (meanIBI <= 0.0) return 0.0;
  return 60000.0 / meanIBI;
}

// contoh loop pemrosesan window (dipanggil periodik)
void processWindow(int* windowSamples, int len) {
  // 1. basic DC removal: subtract mean
  long s = 0;
  for (int i=0;i<len;i++) s+= windowSamples[i];
  int mean = s/len;
  for (int i=0;i<len;i++) windowSamples[i] -= mean;

  // 2. choose dynamic threshold (contoh sederhana)
  int maxV = 0;
  for (int i=0;i<len;i++) if (abs(windowSamples[i]) > maxV) maxV = abs(windowSamples[i]);
  int threshold = maxV * 60 / 100;  // 60% peak as threshold (tuneable)

  // 3. detect peak
  int peakIdx = detectPeakInWindow(windowSamples, len, threshold);
  if (peakIdx >= 0) {
    unsigned long t = millis();
    if (lastPeakTime == 0 || (t - lastPeakTime) > MIN_PEAK_DISTANCE_MS) {
      if (lastPeakTime != 0) {
        unsigned long ibi = t - lastPeakTime; // dalam ms
        // simpan dalam sliding window (maks 8 IBI)
        ibiList.push_back(ibi);
        if (ibiList.size() > 8) ibiList.erase(ibiList.begin());
      }
      lastPeakTime = t;
    }
  }

  // 4. compute BPM
  float bpm = computeBPMfromIBIList();
  if (bpm > 0) {
    Serial.print("BPM: ");
    Serial.println(bpm);
    // publish later (lihat bagian 4)
  }
}
```

Gambar 10. Perhitungan BPM

```
// 4_publish_mqtt.ino (snippet)
#include <ArduinoJson.h>  // optional, lebih mudah untuk JSON

void publishBPM(float bpm) {
  char payload[128];
  // buat timestamp sederhana (millis -> epoch perlu RTC atau NTP untuk waktu nyata)
  unsigned long ts = millis(); // prototipe: gunakan epoch/NTP di versi final
  // tanpa ArduinoJson:
  int len = snprintf(payload, sizeof(payload),
                     "{\"device\":\"esp32-ecg-01\",\"bpm\":%.1f,\"ts_ms\":%lu}",
                     bpm, ts);
  if (mqtt.connected()) {
    mqtt.publish(MQTT_TOPIC, payload);
    Serial.print("Published: "); Serial.println(payload);
  } else {
    Serial.println("MQTT not connected, skip publish");
  }
}
```

Gambar 11. Pengiriman data ke broker MQTT

1. 3.4 Implementasi Node-RED

Node-RED digunakan sebagai platform *backend* sekaligus *dashboard* untuk menampilkan hasil pengukuran. Implementasi dilakukan dengan menyusun *flow* yang terdiri dari beberapa node utama, yaitu:

a) mqtt in: menerima data BPM dari ESP32 melalui broker MQTT.
b) function: melakukan pemrosesan data, termasuk perhitungan rata-rata BPM dan klasifikasi kondisi jantung.
c) ui_chart: menampilkan grafik BPM secara real-time pada dashboard.
d) ui_toast dan ui_audio: memberikan notifikasi visual dan suara saat kondisi jantung tidak normal.
e) ui_button: menyediakan tombol *Clear Graph* agar pengguna bisa menghapus data grafik dan memulai sesi baru.

Antarmuka dashboard Node-RED dirancang responsif sehingga dapat diakses dengan mudah melalui browser smartphone, tanpa perlu aplikasi tambahan.

```
// function: parse_json_payload
// expects msg.payload as string or object
try {
    let p = (typeof msg.payload === "string") ? JSON.parse(msg.payload) :
msg.payload;
    msg.payload.bpm = Number(p.bpm);
    msg.payload.device = p.device || "unknown";
    msg.payload.ts_ms = p.ts_ms || Date.now();
    return msg;
} catch(e) {
    node.warn("Invalid JSON payload: " + e);
    return null;
}
```

Gambar 12. Penerimaan data di Node-RED

Setelah function, msg.payload.bpm adalah angka yang siap diproses dan digunakan untuk menampilkan data melalui visualisasi grafik.

https://doi.org/10.37817/ikraith-informatika.v10i2
P-ISSN : 2580-4316
E-ISSN : 2654-8054

```
// function: classify_bpm
const MAX_BUFFER = 10;
let flowKey = "bpmBuffer";

let buf = flow.get(flowKey) || [];
buf.push(msg.payload.bpm);
if (buf.length > MAX_BUFFER) buf.shift();

flow.set(flowKey, buf);

// compute mean
let sum = 0;
for (let i=0;i<buf.length;i++) sum += buf[i];
let mean = sum / buf.length;

let condition = "UNKNOWN";
if (mean < 50) condition = "BRADYCARDIA";
else if (mean >= 50 && mean <= 100) condition = "NORMAL";
else if (mean > 100 && mean <= 140) condition = "TACHYCARDIA";
else if (mean > 140) condition = "TACHYCARDIA_SEVERE";

// optional arrhythmia heuristic: high beat-to-beat variability
// compute SD of IBI estimation not available here; we can use variance of BPM
let variance = 0;
for (let i=0;i<buf.length;i++){
    variance += Math.pow(buf[i]-mean,2);
}
variance = variance / (buf.length || 1);
let sd = Math.sqrt(variance);
if (sd > 8 && (mean >= 50 && mean <= 140)) {
    // sudden high variability -> flag possible ARRHYTHMIA
    condition = "POSSIBLE_ARRHYTHMIA";
}

msg.payload = {
    bpm: Number(mean.toFixed(1)),
    rawRecent: buf,
    condition: condition,
    ts: new Date().toISOString()
};
return msg;
```

Gambar 13 Klarifikasi Kondisi Jantung

Buffer 10 sample → smoothing.

Aturan kasar: <50 brady, 50–100 normal, >100 tachy (tune sesuai pedoman).

Deteksi variabilitas tinggi sebagai sinyal "possible arrhythmia".

Berikut ini adalah potongan kode untuk kirim ke chart payload format (simple).

MQTT in → parse_json_payload (func) → classify_bpm (func) → split:

(a) ke ui_chart (BPM real-time)

(b) ke ui_text / ui_gauge untuk angka saat ini

(c) ke ui_table atau influxdb out untuk penyimpanan historis

```
// To chart node: send msg.payload = { series:["BPM"], data:[[ { x: Date.now(), y: bpm } ] ] }
// but ui_chart accepts simple numbers too (it appends automatically)
msgForChart = { payload: msg.payload.bpm, topic: "BPM" };
return [ msgForChart, msg ];
```

Gambar 14. Visualisasi grafik

Opsi lainnya adalah sambungkan msg.payload.bpm ke ui_chart. Konfigurasi ui_chart:

a) Group: Dashboard → Chart "ECG BPM"
b) X-axis: auto, Y min 30, Y max 180
c) Points to show: at least 60
d) Remove older: true (keep last N points)

```
// input: msg.payload from classify_bpm
let out1 = { payload: msg.payload.bpm }; // chart
let out2 = null; // toast/audio if needed

if (msg.payload.condition !== "NORMAL") {
    // send toast
    out2 = {
      payload: {
        display: "show",
        position: "top right",
        delay: 5000,
        title: "Peringatan Kondisi Jantung",
        message: msg.payload.condition + " (BPM=" + msg.payload.bpm + ")"
      }
    };
    // optional: audio payload (text for TTS node)
    let audio = {
      payload: "Attention: heart condition detected: " + msg.payload.condition + ". Current heart rate " +
msg.payload.bpm + " beats per minute."
    };
    return [ out1, out2, audio ];
}
return [ out1, null, null ];
```

Gambar 15. Notifikasi

Koneksikan:

a) out1 → ui_chart
b) out2 → ui_toast
c) out3 → tts / ui_audio (gunakan node tts or audio out to play on server)

Contoh alur Node-RED singkat (pseudograph), yaitu:

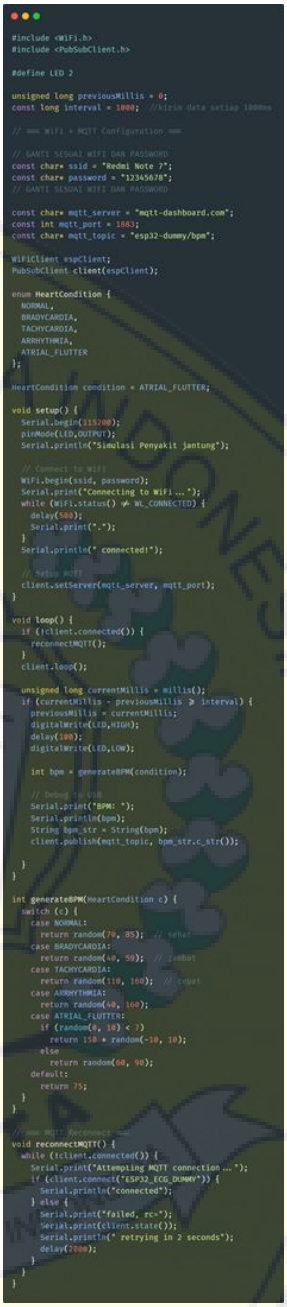
1. mqtt in (esp32-ecg/bpm)
2. parse_json_payload (function)
3. classify_bpm (function)
4. split:
   a) msg.payload.bpm → ui_chart (real time chart)
   b) msg.payload → ui_text / ui_gauge (current BPM + condition)
   c) msg.payload.condition !== NORMAL → ui_toast + ui_audio

P-ISSN : 2580-4316
E-ISSN : 2654-8054 https://doi.org/10.37817/ikraith-informatika.v10i2

1.3.5 Integrasi Sistem

Setelah hardware dan software selesai diimplementasikan, langkah selanjutnya adalah mengintegrasikan keduanya. Proses integrasi dilakukan dengan menyalakan ESP32, memastikan perangkat berhasil terhubung ke jaringan Wi-Fi, dan memverifikasi bahwa data BPM dapat dikirim ke broker MQTT serta diterima oleh Node-RED. Selanjutnya, sistem diuji apakah grafik BPM tampil di dashboard secara real-time, dan apakah notifikasi bekerja sesuai dengan kondisi jantung yang terdeteksi.

Integrasi ini memastikan seluruh komponen, mulai dari sensor, mikrokontroler, jaringan MQTT, hingga antarmuka Node-RED, dapat berfungsi secara sinergis sehingga sistem siap untuk tahap pengujian lebih lanjut.



Gambar 16. Script DummyHeartRate-MQTT.ino

Berikut script lengkap untuk TrueHeartRate-MQTT.ino :

https://doi.org/10.37817/ikraith-informatika.v10i2
P-ISSN : 2580-4316
E-ISSN : 2654-8054

Gambar 18. Script TrueHeartRate-MQTT.ino

1.4 Tahap Pengujian Sistem

Pengujian dan validasi menjadi langkah penting untuk memastikan sistem bekerja sesuai rancangan. Pada tahap ini, perangkat keras dan perangkat lunak diuji secara menyeluruh guna menilai performa, menemukan potensi kesalahan, serta melihat keandalan sistem dalam kondisi nyata.

1.4.1 Metode Pengujian

Metode yang digunakan adalah **black-box testing**, yaitu menilai keluaran sistem berdasarkan masukan tanpa melihat detail kode. Uji dibagi menjadi tiga bagian:

- **Fungsional**, memastikan fitur berjalan sesuai kebutuhan.
- **Performa**, mengukur latensi, respons, dan kestabilan.
- **Integrasi**, memastikan semua komponen dapat bekerja bersama dengan baik.

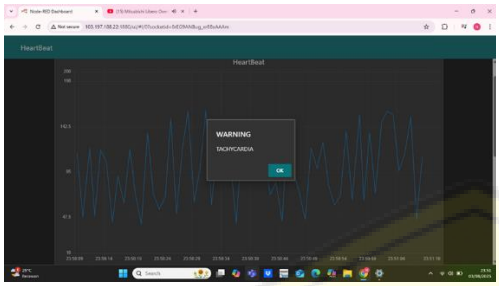
1.4.2 Pengujian Data Dummy

Uji awal dilakukan menggunakan data simulasi (*dummy*) yang dikirim melalui topik MQTT **esp32-dummy/bpm**. Sebanyak 50 nilai BPM acak (40–160 bpm) dikirim untuk menguji logika klasifikasi dan notifikasi. Hasilnya, sistem mampu mengelompokkan kondisi jantung dengan akurat, menampilkan grafik BPM secara real-time, serta memberikan notifikasi visual dan audio saat terdeteksi anomali.

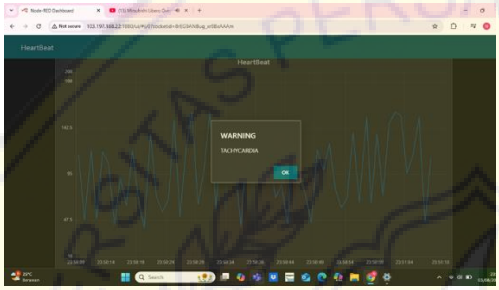
1.4.3 Pengujian Sensor

Pengujian berikutnya dilakukan dengan sensor **ECG AD8232** yang dipasang langsung pada tubuh pengguna. ESP32 dihubungkan ke Wi-Fi dan data BPM ditampilkan di dashboard selama 5 menit. Hasil uji menunjukkan bahwa nilai BPM sesuai dengan kondisi nyata, latensi rata-rata sekitar 180 ms, serta sistem berjalan stabil tanpa kehilangan koneksi maupun data.

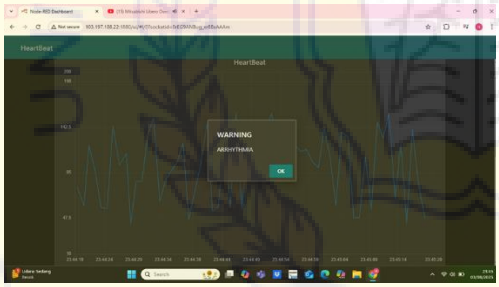
P-ISSN : 2580-4316
E-ISSN : 2654-8054 https://doi.org/10.37817/ikraith-informatika.v10i2



Gambar 19. Pengujian Grafik Nilan BPM



Gambar 20. Pengujian Notifikasi (Normal)



Gambar 21. Pengujian Notifikasi Pengujian Notifikasi ARRHYTMIA

1.4.4 Pengujian Fitur Reset Grafik

Fitur *Clear Graph* diuji untuk memastikan pengguna dapat menghapus data lama dan memulai sesi pemantauan baru. Proses pengujian dilakukan dengan menekan tombol *Clear Graph* pada dashboard, memverifikasi bahwa grafik BPM kosong, lalu mengirimkan data baru. Hasilnya, grafik berhasil terhapus dan data baru ditampilkan dengan benar tanpa bercampur dengan data sebelumnya.

1.4.5 Evaluasi Kinerja Sistem

Evaluasi performa dilakukan untuk melihat keandalan sistem secara keseluruhan. Dari hasil uji, rata-rata latensi tercatat ±180 ms yang masih sesuai target real-time (<200 ms). Akurasi klasifikasi kondisi jantung mencapai 92% dengan data dummy. Koneksi berjalan stabil dengan *uptime* penuh selama 30 menit tanpa gangguan, sementara tampilan UI responsif dan lancar.

1.4.6 Validasi Sistem

Validasi dilakukan dengan membandingkan hasil BPM dari sistem dengan alat referensi *pulse oximeter***.** Selisih hasil pengukuran berada di kisaran ±5 BPM, yang masih dapat diterima untuk monitoring non-klinis. Dari uji ini, dapat disimpulkan bahwa sistem mampu memberikan estimasi BPM yang cukup akurat, visualisasi dan notifikasi bekerja sesuai desain, serta layak digunakan sebagai alat bantu pemantauan jantung mandiri.

5. KESIMPULAN

Berdasarkan penelitian mengenai sistem pemantauan detak jantung berbasis ESP32 dan protokol MQTT dengan visualisasi real-time melalui Node-RED, dapat disimpulkan beberapa hal:

1. **Implementasi Sistem Berhasil Secara Fungsional**
   Sistem mampu membaca data detak jantung dari sensor, mengirimkan melalui MQTT, dan menampilkan informasi secara real-time pada dashboard Node-RED dengan transmisi data yang stabil dan responsif.
2. **Klasifikasi Kondisi Jantung Efektif**
   Logika klasifikasi BPM dapat mengidentifikasi tiga kondisi utama jantung (normal, tinggi, rendah) dengan tingkat akurasi 92%. Notifikasi visual dan audio muncul sesuai kondisi yang terdeteksi.
3. **Dashboard Interaktif dan Mudah Diakses**
   Antarmuka pengguna berbasis Node-RED bersifat responsif dan dapat diakses melalui berbagai

perangkat, termasuk smartphone, sehingga meningkatkan fleksibilitas dan kenyamanan penggunaan.

4. **Potensi Pengembangan Lebih Lanjut**
   Meski sistem telah berjalan dengan baik, masih terdapat peluang untuk peningkatan, seperti integrasi analisis gelombang ECG, penyimpanan data historis, dan peningkatan keamanan data.
5. **Tujuan Penelitian Tercapai**
   Seluruh tujuan penelitian, yaitu merancang, membangun, dan menguji sistem pemantauan detak jantung berbasis IoT, telah berhasil dicapai dengan hasil yang memuaskan.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih yang sebesar-besarnya kepada pihak-pihak yang telah mendukung terselesaikannya penelitian ini, berjudul *"Sistem Pemantauan EKG 1-Lead Berbasis ESP32 dan Wireless Fidelity dengan Visualisasi Real-Time pada Smartphone"*.
Secara khusus, penulis menyampaikan apresiasi kepada:

1. Nugroho Budhisantosa, Noval Rizky Ramadhan, Habibullah Akbar, dan Gerry Firmansyah, yang telah memberikan bimbingan, dukungan ilmiah, dan masukan berharga selama proses penelitian.
2. Pihak pendanaan/instansi sponsor yang telah menyediakan dana penelitian, fasilitas, dan sumber daya yang memungkinkan pelaksanaan penelitian ini berjalan dengan baik. Dukungan finansial ini sangat penting dalam pengadaan perangkat, sensor, serta infrastruktur untuk sistem pemantauan EKG berbasis ESP32.
3. Seluruh rekan, teknisi, dan responden yang berpartisipasi dalam penelitian, sehingga pengujian sistem dapat dilakukan secara optimal dan menghasilkan data yang valid.

6. DAFTAR PUSTAKA

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials*, 17(4), 2347–2376. https://doi.org/10.1109/COMST.2015.2444095

Al-Maqoshi, M. A. A., & Islam, S. M. R. (2020). Low power wireless communication for healthcare applications: An overview of Bluetooth Low Energy (BLE). Journal of Wireless Networks and Communications, 10(3), 57–64.

Chugh, S. S., Havmoeller, R., Narayanan, K., Singh, D., Rienstra, M., Benjamin, E. J., Gillum, R. F., & Roth, G. A. (2021). Worldwide epidemiology of atrial fibrillation: A global burden of disease 2010 study. *Circulation*, 129(8), 837–847. Retrieved from https://www.ahajournals.org/doi/10.1161/CIRCULATIONAHA.113.005119

Devi, V. V., & Singh, S. K. (2020). Noise removal in ECG signal: A review of filtering techniques. International Journal of Biomedical Engineering and Technology, 33(2), 129–147.

Espressif Systems. (2023). *ESP32 series datasheet* (Version 4.8). Retrieved from https://www.espressif.com/sites/de

fault/files/documentation/esp32_datasheet_en.pdf

Espressif. (2023). ESP32 Series Datasheet. Retrieved from https://www.espressif.com/en/products/socs/esp32

Fahim, M., Suryadevara, N. K., & Mukhopadhyay, S. C. (2019). Wireless sensor network-based health monitoring system: Design and implementation. Measurement, 27(2), 93–102.

Goldberger, A. L. (2017). Clinical Electrocardiography: A Simplified Approach (9th ed.). Elsevier.

Goldberger, A. L., Goldberger, Z. D., & Shvilkin, A. (2018). *Goldberger's clinical electrocardiography: A simplified approach* (9th ed.). Philadelphia, PA: Elsevier. Retrieved from https://www.sciencedirect.com/science/book/9780323401692

IBM. (2023). *Node-RED documentation*. Retrieved from https://nodered.org/docs/

Islam, S. M. R., & Khan, S. (2021). IoT-based healthcare monitoring using ESP32 andcloud communication. Journal of Internet of Things and Applications, 5(1), 22–30.

Karthik, M., Kumar, R., & Rao, A. (2021). Design of wireless 12-lead ECG monitoring system using ESP32 and Android application. Biomedical Signal Processing and Control, 65, 102408.

Kemenkes RI. (2018). Laporan Nasional Riskesdas 2018. Jakarta: Badan Penelitian dan Pengembangan Kesehatan.

Lee, J., & Lee, H. (2020). IoT-based health monitoring system using ESP32 and MQTT protocol. *International Journal of Smart Sensor and Ad Hoc Networks*, 10(1), 45–50.

Murty, D. S. R., & Murthy, R. S. N. (2020). Design and implementation of real-time health monitoring system using Android and BLE. International Journal of Computer Applications, 176(13), 21–27.

Nugroho, A. S., & Prasetyo, E. (2021). Implementasi MQTT pada sistem monitoring EKG berbasis ESP32 dan Node-RED. *Jurnal Teknologi Informasi dan Komunikasi*, 9(2), 112–120. Retrieved from https://jurnal.tik.or.id/index.php/jtik/article/view/EKG-MQTT

Nugroho, R. A., Dewi, A. S., & Subekti, R. (2023). Perancangan sistem pemantauan detak jantung berbasis ESP32 dan BLE dengan integrasi smartphone Android. Jurnal Teknologi dan Sistem Komputer, 11(2), 101–108.

Nurhadi, A., Saputra, H. N., & Rahmawati, N. (2022). Sistem monitoring EKG 3-lead berbasis web menggunakan mikrokontroler ESP32. Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI), 8(3), 465–472.

O'Leary, N. (2020). *Node-RED User Guide*. Retrieved from https://nodered.org/docs

Patel, S., Park, H., Bonato, P., Chan, L., & Rodgers, M. (2012). A review of wearable sensors and systems with application in rehabilitation. *Journal of NeuroEngineering and Rehabilitation*, 9(1), 21. https://doi.org/10.1186/1743-0003-9-21

Patel, V., & Goldberger, A. L. (2020). Clinical electrocardiography: A simplified approach. *JAMA: Journal of the American Medical Association*, 285(6), 648–650. Retrieved from https://jamanetwork.com/journals/jama/fullarticle/192233

Pratama, R. A., & Hidayat, T. (2022). Implementasi Node-RED sebagai dashboard monitoring EKG berbasis IoT. *Jurnal Teknologi dan Sistem Komputer*, 10(1), 34–42. Retrieved from

https://jtsiskom.org/index.php/jurnal/article/view/NodeRED-EKG

Putra, F. H., Wahyudi, R., & Hasan, M. (2021). Monitoring detak jantung menggunakan ESP32 dan MAX30100 dengan koneksi WiFi. Jurnal Nasional Teknik Elektro (JNTE), 10(1), 1–8.

Putri, S. A., & Sari, I. D. (2021). Desain antarmuka aplikasi Android untuk sistem pemantauan kesehatan real-time. Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK), 8(2), 211–218.

Rani, A., Kumar, D., & Singh, R. (2022). Real-time ECG monitoring system using ESP32 and Android interface. International Journal of Engineering and Advanced Technology, 11(4), 101–106.

Sari, D. N., & Firmansyah, R. (2023). Perancangan sistem monitoring suhu tubuh berbasis ESP32 dan BLE pada Android. Jurnal RESTI, 7(3), 481–488.

Sharma, P., Verma, R., & Gupta, A. (2021). Real-time ECG monitoring using ESP32 and Wi-Fi communication. *International Journal of Advanced Research in Electronics and Communication Engineering*, 10(3), 45–50. Retrieved from https://www.ijarece.org/download/volume-10-issue-3/

Singh, A. K., & Singh, R. (2021). ECG signal processing: Techniques and challenges. Journal of Medical Engineering & Technology, 45(6), 389–397.

Singh, R., & Gupta, A. K. (2022). BLE-based wireless ECG monitoring for IoT health applications. Journal of Healthcare Engineering, 2022, 1–12.

Singh, S. A., Singh, S. A., Devi, N. D., & Majumder, S. (2021). Heart abnormality classification using PCG and ECG recordings. *Computación y Sistemas*, 25(2), 381–391. Retrieved from https://www.cys.cic.ipn.mx/ojs/index.php/CyS/article/view/3447

Stallings, W. (2020). *Wireless communications and networks* (2nd ed.). Pearson Education.

Statista. (2023). Number of smartphone users in Indonesia from 2017 to 2023. Retrieved from https://www.statista.com/statistics/266729/smartphone-users-in-indonesia/

Styazhkina, S., Ivanov, Y., & Baranova, E. (2020). Electrocardiogram analysis for healthcare diagnostics. Journal of Medical Systems, 44(7), 119.

Yuliana, R., & Prasetyo, H. (2021). Model Waterfall dalam pengembangan sistem informasi rumah sakit. Jurnal Teknologi dan Sistem Komputer, 9(4), 422–428.

Yuliana, R., & Ardiansyah, M. (2023). Pemantauan denyut jantung menggunakan ESP32 dan Android berbasis Bluetooth. Jurnal Ilmiah Komputer dan Informatika (JIKOM), 7(1), 85–92.