



**Perancangan dan Implementasi Gateway
LoRa dengan Antarmuka Web untuk
Komunikasi Jarak Jauh pada Sistem IoT
dan Monitoring Telemedis**

Tugas Akhir

**Oleh:
Azrul Aswat (3232301025)**

**Program Studi Teknik Instrumentasi
Jurusan Elektro
Politeknik Negeri Batam
2025**

Pernyataan Keaslian Proyek Akhir

Saya yang bertandatangan dibawah ini menyatakan bahwa isi sebagian maupun keseluruhan Proyek Akhir saya yang berjudul : **"Perancangan dan Implementasi Gateway LoRa dengan Antarmuka Web untuk Komunikasi Jarak Jauh pada Sistem IoT dan Monitoring Telemedis"** adalah hasil karya sendiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan, dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri. Semua referensi yang dikutip atau dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan saya ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Batam, 10 Juli 2025



Azrul Aswat

NIM: 3232301025

Lembar Pengesahan

Proyek Akhir disusun untuk memenuhi salah satu syarat memperoleh gelar
Ahli Madya Teknik (A.Md.T.)

Politeknik Negeri Batam

Oleh:

Azrul Aswat (3232301025)

Tanggal Sidang: 11 Agustus 2025

Disetujui Oleh:

Penguji 1



Ir. Kamarudin, ST, M.T., IPM
NIK: 110071

Pembimbing 1



Mu'thiana Gusman, S.Kom, M.T
NIK: 123293

Penguji 2



Aditya Gautama Darmoyono, S.T., M.T.
NIK: 117180

Pembimbing 2



Dessy Oktani S.T., M.T
NIK: 11075

Abstrak

Akses layanan kesehatan di daerah terpencil, khususnya wilayah 3T (terluar, terdepan, dan tertinggal), menghadapi tantangan serius dalam monitoring kondisi vital seperti tekanan darah dan detak jantung akibat keterbatasan infrastruktur internet konvensional. Penelitian ini merancang dan mengimplementasikan sistem monitoring telemedis berbasis arsitektur three-tier LoRa network yang terdiri dari transmitter, bridge, dan receiver untuk komunikasi jarak jauh tanpa bergantung pada internet. Sistem menggunakan sensor MAX30102 untuk detak jantung dan MPX5700DP untuk tekanan darah yang terintegrasi dengan teknologi LoRa pada frekuensi 915-925MHz, dengan pendekatan prototype development arsitektur transmitter-bridge-receiver dimana data dikirim ke Firebase Realtime Database. Hasil pengujian menunjukkan performa sangat baik pada jarak dekat (51,45-205,16 meter) dengan packet loss rate 0% dan kualitas sinyal "KUAT", namun pada jarak 2,15 km tanpa bridge mengalami packet loss rate 74% dengan sinyal "BURUK". Implementasi bridge sebagai repeater berhasil mengatasi masalah tersebut, menghasilkan packet loss rate 0% dengan kualitas sinyal kembali "KUAT" pada jarak yang sama. Website dan aplikasi mobile telemedis dikembangkan untuk menampilkan data vital secara real-time dengan fitur monitoring dan chat konsultasi medis, memberikan solusi efektif untuk monitoring kesehatan jarak jauh di daerah terpencil dengan akurasi tinggi dan reliability terjamin melalui arsitektur three-tier LoRa network.

Kata kunci: LoRa, IoT, telemedis, monitoring kesehatan, daerah terpencil, three-tier architecture, bridge repeater

Abstract

Healthcare access in remote areas, particularly in outermost, frontier, and underdeveloped (3T) regions, faces serious challenges in monitoring vital conditions such as blood pressure and heart rate due to limited conventional internet infrastructure. This research designs and implements a telemedicine monitoring system based on three-tier LoRa network architecture consisting of transmitter, bridge, and receiver for long-distance communication without relying on internet infrastructure. The system utilizes MAX30102 sensor for heart rate measurement and MPX5700DP sensor for blood pressure measurement integrated with LoRa technology at 915-925MHz frequency, employing prototype development approach with transmitter-bridge-receiver architecture where data is transmitted to Firebase Realtime Database. Testing results demonstrate excellent performance at short distances (51.45-205.16 meters) with 0% packet loss rate and "STRONG" signal quality, however at 2.15 km distance without bridge experienced 74% packet loss rate with "POOR" signal quality. Bridge implementation as repeater successfully overcame this issue, achieving 0% packet loss rate with signal quality returning to "STRONG" at the same distance. Telemedicine website and mobile application were developed to display vital data real-time with monitoring features and medical consultation chat, providing effective solution for remote healthcare monitoring in areas with limited internet infrastructure, enabling medical personnel to monitor patient conditions remotely with high accuracy and guaranteed reliability through three-tier LoRa network architecture.

Keywords: LoRa, IoT, telemedicine, health monitoring, remote areas, three-tier architecture, bridge repeater

Kata Pengantar

Puji syukur atas kehadiran Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya penulis dapat menyelesaikan proyek akhir yang berjudul “Perancangan dan Implementasi Gateway LoRa dengan Antarmuka Web untuk Komunikasi Jarak Jauh Sistem IoT dan Monitoring Telemedis”. Penulisan laporan ini dilakukan dalam rangka memenuhi salah satu syarat untuk memperoleh gelar Ahli Madya Teknik (Amd. T) dari jenjang Diploma 3, Program Studi Teknik Instrumentasi, Jurusan Teknik Elektro di Politeknik Negeri Batam. Saya menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak pada masa perkuliahan saya sampai pada penyusunan ini, sangat sulit bagi saya untuk dapat menyelesaikannya. Oleh karena itu, saya mengucapkan terima kasih kepada:

1. Ibu Mu’thiana Gusnam, S.Kom,M.T dan Ibu Dessy Oktani S.T.,M.T selaku dosen pembimbing yang telah membimbing, serta memberikan masukan, saran dan juga arahan hingga akhir.
2. Seluruh dosen dan laboran pada Program Studi Teknik Instrumentasi Politeknik Negeri Batam.
3. Orang tua dan keluarga tercinta yang selalu memberikan nasehat, semangat dan motivasi untuk menyelesaikan proyek akhir ini.
4. Aviv Aldino Anugrah, Rizky Nugrah, Muhammad Baihaqi Adnan, Suhaity Binti Bachtiar dan seluruh teman-teman seperjuangan baik secara langsung maupun tidak langsung terlibat dalam proses pengerjaan proyek akhir.

Dalam penyusunan laporan proyek akhir ini, penulis menyadari bahwa masih jauh dari kata sempurna. Oleh sebab itu, penulis sangat mengharapkan adanya kritik dan saran yang bersifat membangun guna penyempurnaan dalam penulisan ini. Akhir kata, penulis mengucapkan terima kasih kepada semua pihak yang telah membantu, baik dalam penyusunan laporan ini maupun masukan yang telah diberikan dalam bentuk kritik dan saran yang membangun. Semoga laporan proyek akhir ini membawa manfaat bagi pengembangan ilmu pengetahuan.

Batam, 10 Juli 2025

Azrul Aswat

Daftar Isi

Pernyataan Keaslian Proyek Akhir	i
Lembar Pengesahan	Error! Bookmark not defined.
Abstrak	iii
<i>Abstract</i>	iv
Kata Pengantar	v
Daftar Isi	vi
Daftar Gambar	viii
Daftar Tabel	ix
BAB 1 : PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan	3
1.4. Manfaat	3
1.5. Batasan	4
BAB 2 : Tinjauan Pustaka	5
2.1. Landasan Teori	5
2.1.1. Detak Jantung	5
2.1.2. Tekanan Darah	6
2.2. Teknologi yang Digunakan	7
2.2.1. Teknologi Hardware	7
2.2.2. Teknologi Software	9
2.3. Penelitian Terdahulu	13
BAB 3 : Metodologi Pelaksanaan	14
3.1. Metode Penelitian	14
3.2. Perancangan	14
3.2.1. Perancangan Desain Elektrikal	15
3.2.2. Pembuatan Program	17
3.2.3. Pembuatan Aplikasi Android	17
	vi

3.2.4.	Pembuatan WEB	18
3.2.5.	Use Case Diagram	19
3.2.6.	<i>Data Link Performance Metrics</i>	20
3.3.	Pengujian.....	21
3.3.1.	Pengujian LoRA (Long Range).....	21
3.3.2.	Pengujian Sensor MAX30102	23
3.3.3.	Pengujian Sensor MPX5700DP	24
3.3.4.	Pengujian WEB Telemedis.....	24
3.3.5.	Pengujian WEB Gateway	27
3.4.	Cara Kerja Sistem	29
BAB 4 : Hasil dan Pembahasan		35
4.1.	Data Hasil Pengujian	35
4.1.1.	Data Hasil Pengujian LoRa	35
4.2.	Pembahasan	38
Kesimpulan dan Saran		41
5.1.	Kesimpulan	41
5.2.	Saran	41
Daftar Pustaka		42
Biodata		44

Daftar Gambar

Gambar 2. 1. LoRa	8
Gambar 2. 2. Sensor MAX30102.....	9
Gambar 2. 3. Sensor MPX5700DP	9
Gambar 3. 1. Flowchart Pelaksanaan Project	15
Gambar 3. 2 Gambar Elektrikal.....	16
Gambar 3. 3. Desain Aplikasi	18
Gambar 3. 4. UseCase Diagram	20
Gambar 3. 5. Tampilan Register	25
Gambar 3. 6. Tampilan Login.....	26
Gambar 3. 7. Tampilan Dashboard Hasil Pengukuran	26
Gambar 3. 8. Halaman Chat.....	27
Gambar 3. 9 Monitoring Data Node	27
Gambar 3. 10 Daftar Receiver	28
Gambar 3. 11 Daftar Node.....	28
Gambar 3. 12 Data Node Terpilih	29
Gambar 3. 13. Flowchart Cara Kerja Sistem	30
Gambar 3. 14 Flowchart Komunikasi	31
Gambar 3. 15. Blok Diagram Cara Kerja Sistem	31
Gambar 3. 16. Blok Diagram Cara Kerja Komunikasi LoRa.....	33

Daftar Tabel

Tabel 3. 1. Pin yang digunakan komponen	16
Tabel 4. 1. Hasil Pengujian Jarak Dekat	35
Tabel 4. 2. Hasil Pengujian Jarak Jauh Tanpa Bridge.....	37
Tabel 4. 3. Hasil Pengujian Jarak Jauh dengan Bringe.....	38

BAB 1 : PENDAHULUAN

1.1. Latar Belakang

Perkembangan teknologi *Internet of Things* (IoT) telah membawa revolusi signifikan dalam bidang Kesehatan, khususnya dalam sistem telemedis untuk monitoring parameter vital seperti detak jantung dan tekanan darah. Sistem telemedis memungkinkan monitoring Kesehatan jarak jauh yang dapat mengurangi kunjungan rumah sakit hingga 15% untuk kunjungan darurat dan 20% untuk rawat inap, serta menurunkan angka kematian pasien hingga 45% [11]. Namun, implementasi sistem telemedis di Indonesia menghadapi tantangan utama berupa keterbatasan infrastruktur jaringan internet di daerah terpencil, terutama pada daerah terluar, terdapan dan tertinggal (3T) yang masih mengandalkan sistem pencatatan manual dan ketersediaan alat monitoring Kesehatan yang terbatas.

Berbagai solusi monitoring Kesehatan jarak jauh telah dikembangkan secara komersial dan penelitian. Produk komersial seperti OMRON *Healthcare* menawarkan sistem *Remote Patient Monitoring* (RPM) dengan teknologi oscillometrik yang dapat mengukur getaran darah melalui arteri dan mengkonversinya menjadi pembacaan digital, namun memerlukan konektivitas internet yang stabil [13]. Sementara itu, Caresimple menyediakan sistem monitoring tekanan darah jarak jauh yang dapat mengirimkan data secara otomatis melalui *gateway* seluler [2], dan Tenovi BPM menawarkan solusi transmisi data otomatis untuk optimasi perawatan pasien dengan kondisi kronis [17]. Namun produk-produk ini umumnya mengandalkan infrastruktur jaringan yang belum tentu tersedia di daerah terpencil.

Teknologi LoRa (Long Range) menawarkan solusi komunikasi jarak jauh yang dapat mencapai jangkauan 1 kilometer hingga 20 kilometer dengan konsumsi daya rendah, menjadikannya ideal untuk aplikasi monitoring Kesehatan di daerah dengan keterbatasan infrastruktur [14]. Penelitian oleh (Bouazzi, et al., 2022) menunjukkan bahwa arsitektur IoT berbasis LoRa dapat meningkatkan kualitas pelayanan Kesehatan pedesaan sambil mengurangi biaya operasional. Implementasi sistem monitoring Kesehatan berbasis LoRaWAN telah terbukti efektif dalam menghubungkan dokter, perawat, dan pasien melalui jaringan nirkabel yang handal dan aman [3].

Namun, sistem LoRa konvensional memiliki keterbatasan jangkauan dalam kondisi geografis yang kompleks. Solusi multi-hop LoRa network telah dikembangkan untuk mengatasi limitasi cakupan sinyal akibat atenuasi. Arsitektur multi-hop LoRa terdiri dari sensor nodes yang membentuk mesh *network* dan mengirimkan data ke repeater node yang bertindak sebagai sink, kemudian repeater node berkomunikasi langsung dengan *gateway* utama yang terhubung ke internet [19]. Penelitian terbaru menunjukkan bahwa sistem multi-hop LoRa dapat memperluas cakupan jaringan tanpa memerlukan deployment *gateway* dalam jumlah besar, sehingga lebih *cost-effective* untuk aplikasi area luas.

Meskipun demikian, penelitian-penelitian sebelumnya masih memiliki beberapa keterbatasan dan gap yang perlu diatasi. Pertama, dari segi teknis, Sebagian besar sistem monitoring Kesehatan LoRa yang ada masih menggunakan arsitektur single-hop yang terbatas jangkauannya, atau multi-hop namun belum dioptimalkan untuk aplikasi telemedis yang memerlukan akurasi tinggi dalam pengukuran detak jantung dan tekanan darah. Kedua, integrasi sistem three-tier LoRa (*transmitter-bridge-receiver*) yang khusus dirancang untuk kondisi geografi Indonesia belum banyak diteliti. Ketiga, dari segi non-teknis, sistem penyimpanan database yang terintegrasi dengan aplikasi web dan mobile untuk akses *real-time* oleh tenaga Kesehatan masih memerlukan pengembangan lebih lanjut.

Berdasarkan Analisa terhadap penelitian dan produk sejenis yang telah ada, teridentifikasi gap penelitian yang akan diisi oleh Tugas Akhir ini, yaitu pengembangan sistem monitoring telemedis menggunakan arsitektur *three-tier* LoRa *network* yang terdiri dari: (1) LoRa transmitter yang dilengkapi sensor untuk mengukur detak jantung dan tekanan darah, (2) LoRa bridge yang berfungsi sebagai repeater untuk menjembatani transmitter data dari area terpencil, dan (3) LoRa *receiver* yang ditempatkan di lokasi dengan akses internet untuk menyimpan data ke database dan menyediakan akses melalui aplikasi web dan mobile. Sistem yang dikembangkan difokuskan pada peningkatan jangkauan transmitter data, akurasi pengukuran parameter vital, dan integrasi database yang dapat diakses secara *real-time* oleh tenaga Kesehatan untuk keperluan telemedis di daerah terpencil Indonesia.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, maka rumusan masalah dalam penelitian ini adalah:

1. Bagaimana merancang dan membuat alat monitoring tekanan darah dan detak jantung yang dapat digunakan di daerah terpencil tanpa akses internet langsung?
2. Bagaimana mengimplementasikan teknologi LoRa untuk mengirim data hasil pengukuran dari daerah tanpa jaringan internet ke lokasi dengan akses internet, agar data dapat tersimpan secara otomatis di database online?
3. Bagaimana mengembangkan aplikasi dan website untuk menampilkan data hasil pengukuran secara real-time agar dapat diakses oleh tenaga medis?

1.3. Tujuan

Tujuan dari penelitian ini adalah:

1. Membuat prototipe alat monitoring tekanan darah dan detak jantung menggunakan sensor MAX30102 dan MPX5700DP yang dapat digunakan di daerah terpencil.
2. Mengimplementasikan teknologi komunikasi LoRa sebagai solusi pengiriman data dari lokasi tanpa internet ke *gateway* yang terhubung ke internet.
3. Mengembangkan aplikasi dan website untuk menampilkan data hasil pengukuran yang tersimpan di database secara *real-time* dan dapat diakses oleh tenaga medis.

1.4. Manfaat

Hasil dari penelitian dan pengembangan alat ini diharapkan dapat memberikan beberapa manfaat sebagai berikut:

1. Bagi tenaga medis: Membantu mendapatkan data vital pasien secara *real-time* meskipun pasien berada di daerah terpencil, sehingga dapat mempercepat pengambilan keputusan medis.
2. Bagi pasien di daerah terpencil: Memberikan akses layanan monitoring kesehatan yang lebih baik, dengan data yang tercatat dan tersimpan secara otomatis, sehingga mengurangi risiko data hilang atau terlupakan.
3. Bagi dunia pendidikan dan penelitian: Menjadi referensi untuk penelitian selanjutnya terkait teknologi monitoring kesehatan berbasis LoRa dan solusi telemedis di daerah dengan keterbatasan akses internet.

4. Bagi pemerintah dan lembaga kesehatan: Mendukung program transformasi layanan kesehatan digital dalam rangka pemerataan akses layanan kesehatan di seluruh wilayah Indonesia.

1.5. Batasan

Untuk memfokuskan penelitian dan pengembangan alat ini, maka batasan masalah yang ditetapkan adalah sebagai berikut:

1. Sistem hanya difokuskan untuk pengukuran tekanan darah menggunakan sensor MPX5700DP dan detak jantung menggunakan sensor MAX30102.
2. Database yang digunakan untuk penyimpanan data hasil pengukuran adalah *Firebase Realtime Database*.
3. Website memiliki fitur monitoring data dan fitur chat untuk konsultasi medis dengan dokter, sedangkan aplikasi mobile hanya digunakan untuk menampilkan data hasil pengukuran tanpa fitur chat.
4. Pengujian alat dilakukan di lingkungan simulasi dengan kondisi sinyal LoRa sesuai area terbuka untuk meniru kondisi daerah terpencil.
5. Spesifikasi alat:
 - Sensor detak jantung : MAX30102
 - Sensor tekanan darah : MPX5700DP
 - Modul komunikasi : P2P
 - Platform Akses : Aplikasi Android dan WEB
 - Current : 1A
 - Voltage : 9V
 - Supporting Frequency : 915-925MHz
 - Jarak komunikasi point to point : Max 1km

1.6. Work Breakdown Structure

No	Nama	Tugas dan Tanggung Jawab dalam Tim
1	Azrul Aswat	• Komunikasi LoRa • Pembuatan WEB GateWay • Pembuatan WEB Telemedis

BAB 2 : Tinjauan Pustaka

2.1. Landasan Teori

2.1.1. Detak Jantung

Detak jantung adalah jumlah kontraksi jantung per menit yang menunjukkan aktivitas pompa jantung dalam mengalirkan darah ke seluruh tubuh. Jantung berfungsi sebagai pompa otot yang berkontraksi secara ritmis untuk memompa darah beroksigen ke seluruh jaringan tubuh dan mengembalikan darah berkarbon dioksida ke paru-paru untuk proses oksidasi. Detak jantung diatur oleh sistem konduksi elektrik alami jantung yang dimulai dari nodus sinoatrial (SA node) yang terletak di atrium kanan. Impuls listrik yang dihasilkan oleh SA node menyebar ke seluruh atrium, kemudian ke nodus atrioventrikular (AV node), dan akhirnya ke ventrikel melalui berkas His dan serabut Purkinje. Proses ini menghasilkan siklus jantung yang terdiri dari dua fase utama: sistole (fase kontraksi jantung dimana ventrikel berkontraksi untuk memompa darah keluar dari jantung) dan diastole (fase relaksasi jantung dimana ventrikel mengisi dengan darah). Detak jantung normal untuk orang dewasa berkisar antara 60-100 denyut per menit (bpm) saat istirahat, dengan variasi berdasarkan usia, kondisi fisik, aktivitas fisik, kondisi emosional, suhu tubuh dan lingkungan, posisi tubuh, serta penggunaan obat-obatan tertentu.

Pengukuran detak jantung dapat dilakukan dengan berbagai metode, mulai dari palpasi manual dengan meraba denyut nadi pada arteri radialis atau karotis, elektrokardiografi (EKG) untuk merekam aktivitas listrik jantung, photoplethysmography (PPG) yang menggunakan sensor optik untuk mendeteksi perubahan volume darah pada pembuluh darah kapiler, hingga monitoring digital menggunakan perangkat wearable atau sensor elektronik. Metode photoplethysmography (PPG) memanfaatkan prinsip absorpsi cahaya oleh darah, dimana sensor mengirimkan cahaya (biasanya LED merah dan inframerah) ke jaringan, dan photodetector menangkap cahaya yang dipantulkan atau ditransmisikan. Perubahan volume darah selama siklus jantung menyebabkan variasi intensitas cahaya yang dapat dianalisis untuk menentukan detak jantung. Menurut penelitian yang dilakukan oleh (Bugis, et al., 2024), pengukuran detak jantung menggunakan sensor MAX30102 menunjukkan rentang nilai yang bervariasi antara 76,4 bpm hingga 110,4 bpm pada berbagai subjek penelitian, dengan metode

non-invasif ini memungkinkan pemantauan detak jantung secara real-time dengan tingkat akurasi yang memadai untuk keperluan monitoring kesehatan.

2.1.2. Tekanan Darah

Tekanan darah adalah tekanan yang diberikan oleh darah terhadap dinding arteri saat jantung memompa darah ke seluruh tubuh, yang merupakan parameter vital yang menunjukkan kesehatan sistem kardiovaskular dan berfungsi sebagai indikator penting dalam diagnosis berbagai kondisi medis. Tekanan darah dihasilkan oleh kombinasi dari curah jantung (cardiac output) dan resistensi perifer total, dimana curah jantung adalah volume darah yang dipompa oleh jantung per menit, sedangkan resistensi perifer adalah tahanan yang diberikan oleh pembuluh darah terhadap aliran darah. Tekanan darah terdiri dari dua komponen utama: tekanan sistolik (tekanan maksimal yang terjadi saat jantung berkontraksi untuk memompa darah keluar dari ventrikel kiri ke aorta) dan tekanan diastolik (tekanan minimal yang terjadi saat jantung beristirahat di antara kontraksi). Menurut organisasi kesehatan dunia (WHO), tekanan darah normal bagi orang dewasa adalah 120/80 mmHg, dengan klasifikasi yang mencakup normal (sistolik < 120 mmHg dan diastolik < 80 mmHg), elevated/tinggi normal (sistolik 120-129 mmHg dan diastolik < 80 mmHg), hipertensi derajat 1 (sistolik 130-139 mmHg atau diastolik 80-89 mmHg), hipertensi derajat 2 (sistolik $\geq$ 140 mmHg atau diastolik $\geq$ 90 mmHg), hipertensi derajat 3 atau berat (sistolik $\geq$ 180 mmHg atau diastolik $\geq$ 110 mmHg), dan hipotensi (sistolik < 90 mmHg atau diastolik < 60 mmHg).

Berbagai faktor dapat mempengaruhi tekanan darah, termasuk faktor fisiologis seperti usia, jenis kelamin, berat badan, dan aktivitas fisik, faktor lifestyle seperti pola makan, konsumsi garam, merokok, dan konsumsi alkohol, faktor patologis seperti penyakit ginjal, diabetes, dan hipertiroidisme, faktor psikologis seperti stres, kecemasan, dan depresi, serta faktor farmakologis dari penggunaan obat-obatan tertentu. Pengukuran tekanan darah dapat dilakukan dengan berbagai metode, mulai dari metode auskultasi menggunakan stetoskop dan sphygmomanometer manual dengan mendengarkan bunyi Korotkoff, metode oscillometric menggunakan tensiometer digital yang mendeteksi osilasi tekanan pada manset, metode invasif menggunakan kateter intra-arterial untuk monitoring continue, hingga metode non-invasif berkelanjutan menggunakan sensor digital dan algoritma untuk estimasi tekanan darah.

Perkembangan teknologi memungkinkan pengukuran tekanan darah menggunakan sensor digital yang dapat memberikan hasil yang lebih akurat dan memungkinkan monitoring berkelanjutan, dimana dalam sistem monitoring digital, sensor tekanan seperti MPX5700DP dapat digunakan untuk mendeteksi perubahan tekanan pada manset selama proses deflasi dan menganalisis pola osilasi untuk menentukan tekanan sistolik dan diastolik.

2.2. Teknologi yang Digunakan

2.2.1. Teknologi Hardware

2.2.1.1. LoRa

LoRa adalah teknologi komunikasi nirkabel jarak jauh dengan konsumsi daya rendah yang dikembangkan untuk aplikasi *Internet of Things* (IoT), yang mampu mengirimkan data pada jarak yang sangat jauh (0 km hingga 15 km di area pedesaan) dengan konsumsi daya yang minimal, sehingga sangat cocok untuk aplikasi monitoring kesehatan jarak jauh dengan keunggulan dalam hal jangkauan komunikasi yang luas, konsumsi daya yang rendah, kemampuan menembus bangunan, dan biaya implementasi yang relatif murah. Dalam implementasi sistem monitoring kesehatan, teknologi LoRa menggunakan arsitektur tiga komponen utama: LoRa Transmitter yang berfungsi sebagai perangkat pengirim data yang terintegrasi dengan sensor MAX30102 dan MPX5700DP untuk mengumpulkan data vital signs (detak jantung dan tekanan darah), melakukan *preprocessing* data, dan mengirimkan data tersebut secara nirkabel; LoRa Bridge yang berfungsi sebagai perangkat perantara yang menerima data dari *transmitter* dan meneruskannya ke *receiver*, berperan sebagai *repeater* yang memperluas jangkauan komunikasi dan memastikan reliabilitas transmisi data; dan LoRa *Receiver* yang berfungsi sebagai perangkat penerima data yang terhubung ke sistem backend melalui koneksi internet untuk menerima data dari bridge, melakukan validasi data, dan meneruskan data ke *database Firebase* untuk penyimpanan dan analisis lebih lanjut. Arsitektur *transmitter-bridge-receiver* ini memberikan keunggulan berupa *extended range* yang dapat memperluas jangkauan komunikasi hingga 2-3 kali lipat dibandingkan komunikasi *point-to-point*, *reliability* melalui redundansi path komunikasi, scalability dengan kemampuan penambahan *bridge* untuk *coverage area* yang lebih luas, load balancing untuk optimasi performa, dan *fault tolerance* yang memungkinkan

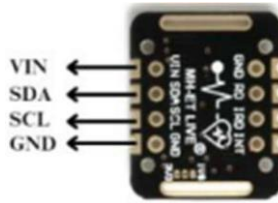
sistem tetap beroperasi meskipun ada komponen yang mengalami gangguan, dimana menurut penelitian (Sofiani, et al., 2021), implementasi LoRa untuk sistem monitoring *heart rate* dan oksigen dalam darah dapat memberikan komunikasi yang stabil dan efisien untuk transmisi data vital signs dengan reliabilitas tinggi [15].



Gambar 2. 1. LoRa

2.2.1.2. MAX30102

MAX30102 adalah sensor terintegrasi yang menggabungkan dua LED (merah dan inframerah), photodetector, dan rangkaian analog front-end yang dioptimalkan untuk aplikasi pengukuran detak jantung dan saturasi oksigen (SpO2), yang menggunakan prinsip photoplethysmography (PPG) untuk mendeteksi perubahan volume darah pada pembuluh darah kapiler. Sensor ini memiliki spesifikasi teknis yang mendukung aplikasi monitoring kesehatan dengan tegangan operasi 1.8V hingga 3.3V, konsumsi daya yang efisien yaitu 1.2 μ A saat standby dan 200 μ A saat operasi, resolusi ADC 18-bit untuk akurasi tinggi, sampling rate hingga 3200 Hz untuk pengukuran real-time, dan interface komunikasi I2C untuk integrasi mudah dengan mikrokontroler. Karakteristik sensor MAX30102 berdasarkan penelitian menunjukkan kemampuan mengukur detak jantung dengan akurasi yang baik, dengan rentang pengukuran 76,4 bpm hingga 110,4 bpm dan standar deviasi yang rendah, menjadikannya pilihan yang tepat untuk aplikasi monitoring kesehatan jarak jauh yang membutuhkan akurasi tinggi dengan konsumsi daya minimal



Gambar 2. 2. Sensor MAX30102

2.2.1.3. MPX5700DP

MPX5700DP adalah sensor tekanan diferensial yang diproduksi oleh *Freescale Semiconductor* (sekarang NXP) yang dirancang khusus untuk aplikasi pengukuran tekanan dengan tingkat akurasi tinggi dan stabilitas yang baik, dimana dalam aplikasi pengukuran tekanan darah, sensor ini berfungsi untuk mendeteksi perubahan tekanan pada manset (cuff) selama proses pengukuran. Sensor ini memiliki spesifikasi teknis yang mendukung aplikasi monitoring tekanan darah dengan rentang pengukuran 0 hingga 700 kPa, akurasi $\pm 1.5\%$ dari *full scale* untuk hasil yang tepat, tegangan operasi 4.75V hingga 5.25V, output 0.2V hingga 4.7V, sensitivitas 6.4 mV/kPa untuk deteksi perubahan tekanan yang sensitif, dan kompensasi suhu -40°C hingga $+125^{\circ}\text{C}$ yang memastikan kinerja stabil dalam berbagai kondisi lingkungan, menjadikannya pilihan yang ideal untuk sistem monitoring tekanan darah yang membutuhkan akurasi dan reliabilitas tinggi.



Gambar 2. 3. Sensor MPX5700DP

2.2.2. Teknologi Software

2.2.2.1. Android Studio

Android Studio adalah *Integrated Development Environment* (IDE) resmi untuk pengembangan aplikasi Android yang dikembangkan oleh Google, yang menyediakan tools lengkap untuk membangun aplikasi mobile yang dapat berinteraksi dengan sistem monitoring kesehatan, dengan fitur utama berupa *code editor* dengan *syntax highlighting* dan *auto-completion*,

emulator untuk testing aplikasi, *debugging tools* yang lengkap, *integration* dengan *Google Play Services*, dan *support* untuk berbagai bahasa pemrograman seperti Java dan Kotlin. Dalam penelitian ini, Android Studio digunakan untuk mengembangkan aplikasi mobile yang memiliki fungsi-fungsi komprehensif meliputi aplikasi monitoring pasien yang memungkinkan pengukuran vital signs secara mandiri dengan koneksi langsung ke perangkat LoRa Receiver untuk menerima data *real-time* dari sensor MAX30102 dan MPX5700DP, *interface* pengukuran *real-time* dengan *user interface* yang *user-friendly* untuk menampilkan proses pengukuran detak jantung dan tekanan darah secara *real-time* termasuk grafik gelombang PPG dan visualisasi yang mudah dipahami, manajemen data hasil pengukuran untuk menyimpan, mengorganisir, dan menampilkan riwayat data pengukuran dalam bentuk grafik dan tabel yang dapat difilter berdasarkan tanggal dan jenis pengukuran, integrasi *Firestore* untuk koneksi *real-time* dengan *Firestore Realtime Database* yang memungkinkan sinkronisasi data antara aplikasi *mobile* dan *cloud storage* dengan operasi CRUD yang otomatis, notifikasi dan *alert* menggunakan *Firestore Cloud Messaging* untuk memberikan peringatan kepada pasien jika nilai vital signs berada di luar rentang normal, *user authentication* menggunakan *Firestore Authentication* untuk sistem login dan registrasi yang aman dan terenkripsi, serta dashboard kesehatan yang menampilkan ringkasan kondisi kesehatan pasien, statistik pengukuran, dan rekomendasi berdasarkan data *historical*. Pengembangan aplikasi Android menggunakan bahasa pemrograman Kotlin dengan arsitektur MVVM (Model-View-ViewModel) untuk memastikan code yang *maintainable* dan *scalable*, serta mengimplementasikan *Material Design Guidelines* untuk memberikan *user experience* yang optimal dalam sistem monitoring kesehatan jarak jauh.

2.2.2.2. Visual Studio Code

Visual Studio Code adalah source code editor yang dikembangkan oleh Microsoft yang dalam konteks pengembangan sistem monitoring kesehatan digunakan untuk pemrograman mikrokontroler dan pengembangan aplikasi web, dengan keunggulan berupa *lightweight* dan *fast performance*, *extensive extension marketplace*, *built-in Git integration*, support untuk *multiple programming languages*, dan *integrated terminal*. Dalam penelitian ini, *Visual Studio Code* digunakan untuk mengembangkan aplikasi web yang memiliki fungsi-fungsi komprehensif meliputi aplikasi web monitoring kesehatan responsif menggunakan teknologi HTML5, CSS3, dan JavaScript

yang dapat diakses oleh pasien dan tenaga medis dari berbagai perangkat, dashboard pengukuran *real-time* yang menampilkan data vital signs secara *real-time* menggunakan WebSocket dan *Firestore Realtime Database* dengan grafik interaktif, gauge meter untuk detak jantung, dan visualisasi tekanan darah dengan update otomatis, manajemen data pengukuran dengan fitur *filtering*, *searching*, dan *sorting* untuk analisis data *historical*, integrasi *Firestore Web SDK* untuk koneksi *real-time* dengan database cloud yang menyediakan konsistensi data *across platform*, sistem chat pasien-dokter menggunakan *Firestore Realtime Database* dan *Cloud Messaging* yang mencakup *real-time messaging* dengan notifikasi instan, file sharing untuk berbagi file hasil pengukuran dan dokumen medis, *consultation scheduling* untuk penjadwalan konsultasi online, *medical record integration* untuk konteks konsultasi, dan *message encryption* untuk keamanan komunikasi medis, *responsive web design* menggunakan *CSS Grid* dan *Flexbox* untuk akses optimal di berbagai ukuran layar, *Progressive Web App (PWA) features* seperti *offline capability*, *push notifications*, dan *installable web app* untuk pengalaman seperti aplikasi *native*, data *visualization* menggunakan library seperti *Chart.js* untuk *visualisasi trend* kesehatan, perbandingan data, dan *analisis statistik*, serta *role-based access control* yang membedakan akses antara pasien, dokter, dan *administrator* dengan tingkat akses yang berbeda. Pengembangan aplikasi web menggunakan *framework modern* seperti *Vue.js* atau *React.js* dengan arsitektur *component-based* untuk *maintainability* dan *reusability*, serta mengimplementasikan best practices untuk *web security* dan *performance optimization* dalam sistem monitoring kesehatan jarak jauh.

2.2.2.3. **Firestore**

Firestore adalah platform pengembangan aplikasi mobile dan web yang menyediakan berbagai layanan *cloud computing*, yang dalam sistem monitoring kesehatan berfungsi sebagai *backend service* untuk penyimpanan data, *autentikasi*, dan *real-time database* dengan layanan yang relevan meliputi *Firestore Realtime Database* untuk penyimpanan data *real-time*, *Firestore Cloud Messaging* untuk notifikasi *push*, *Firestore Authentication* untuk sistem login dan keamanan, serta *Firestore Cloud Storage* untuk penyimpanan file. Dalam penelitian ini, *Firestore* digunakan sebagai *backend comprehensive* untuk mendukung seluruh *ekosistem monitoring* kesehatan dengan implementasi yang mencakup *Firestore Authentication* untuk manajemen akun dengan *user registration & login*

untuk pasien, dokter, dan *administrator*, *multi-factor authentication* dengan 2FA untuk keamanan akun *medical* yang *sensitif*, *role-based access* dengan pembagian *role user* dan *permission* yang berbeda, *password recovery* dengan *email verification*, dan social login dengan *integrasi Google* dan *Facebook*, *Firebase Realtime Database* untuk data pengukuran dengan *real-time data sync* untuk penyimpanan data *vital signs* yang tersinkronisasi *across devices*, *historical data storage* dengan *timestamp* untuk *analisis trend*, data struktur optimal yang efisien untuk *query* cepat dan *scalable*, serta offline capability untuk memastikan data tidak hilang saat koneksi terputus, *Firebase Cloud Storage* untuk *file management* termasuk penyimpanan dokumen medis, profile images, measurement charts, dan backup data otomatis, *Firebase Cloud Messaging* (FCM) untuk notifikasi berupa *health alerts* saat *vital signs abnormal*, *medication reminders*, consultation notifications, dan system updates, *Firebase Cloud Functions* untuk *backend logic* seperti *data processing*, *alert triggers*, *report generation*, dan *data validation*, *Firebase Analytics* untuk monitoring system dengan *usage analytics*, *performance monitoring*, *crash reporting*, dan *user behavior analysis*, *Firebase Security Rules* untuk data protection dengan *HIPAA compliance*, *data access control*, *encryption*, dan *audit logging*, serta *Firebase Hosting* untuk *web application* dengan *static web hosting*, *CDN integration*, *SSL certificate*, dan *custom domain support*. Implementasi *Firebase* dalam sistem monitoring kesehatan ini memastikan *scalability*, *reliability*, dan security yang dibutuhkan untuk aplikasi *medical grade*, serta menyediakan *real-time capabilities* yang essential untuk monitoring vital signs yang membutuhkan *update* data secara *instant*.

2.2.2.4. EasyEDA

EasyEDA adalah *platform online* untuk desain rangkaian elektronik (*Electronic Design Automation*) yang memungkinkan perancangan *schematic*, *PCB layout*, dan simulasi rangkaian elektronik secara *cloud-based*, dengan fitur-fitur yang mencakup *schematic capture* untuk pembuatan diagram rangkaian, *PCB layout design* untuk perancangan papan sirkuit tercetak, *component library* yang lengkap untuk mendukung berbagai komponen elektronik, *simulation capabilities* untuk testing dan validasi rangkaian sebelum implementasi fisik, serta *collaboration tools* yang memungkinkan tim untuk bekerja sama dalam proyek desain elektronik secara *real-time* melalui *platform cloud*.

2.3. Penelitian Terdahulu

Beberapa penelitian terdahulu yang relevan dengan pengembangan sistem monitoring detak jantung dan tekanan darah berbasis IoT telah dilakukan:

(Kamajaya, et al., 2023) mengembangkan sistem telemonitoring kesehatan berbasis IoT yang mengintegrasikan berbagai sensor vital signs. Penelitian ini menunjukkan efektivitas penggunaan teknologi IoT dalam monitoring kesehatan jarak jauh dengan tingkat akurasi yang memadai [8].

(Sofiani, et al., 2021) melakukan penelitian sistem monitoring heart rate dan oksigen dalam darah berbasis LoRa. Hasil penelitian menunjukkan bahwa teknologi LoRa dapat memberikan komunikasi yang stabil untuk transmisi data kesehatan dengan jangkauan yang luas dan konsumsi daya yang rendah [15].

Penelitian oleh tim peneliti di Universitas Politeknik Batam (2024) mengembangkan sistem monitoring telemedis untuk pengukuran tekanan darah berbasis IoT menggunakan sensor MPX5700DP dan jaringan LoRaWAN. Sistem ini menunjukkan kemampuan untuk melakukan monitoring tekanan darah secara *real-time* dengan akurasi yang baik [6].

(Bugis, et al., 2024) dalam penelitiannya tentang alat monitoring detak jantung portable menggunakan sensor MAX30102 menunjukkan bahwa sensor ini dapat memberikan hasil pengukuran yang akurat dengan rentang detak jantung 76,4 bpm hingga 110,4 bpm. Penelitian ini juga menunjukkan bahwa metode photoplethysmography menggunakan MAX30102 efektif untuk monitoring detak jantung non-invasif [4].

Penelitian karakterisasi sensor MAX30102 sebagai alat ukur detak jantung dan suhu tubuh berbasis photoplethysmograph (2022) menunjukkan bahwa sensor ini memiliki akurasi yang baik untuk pengukuran vital *signs* dengan standar deviasi yang rendah, membuktikan konsistensi hasil pengukuran [4].

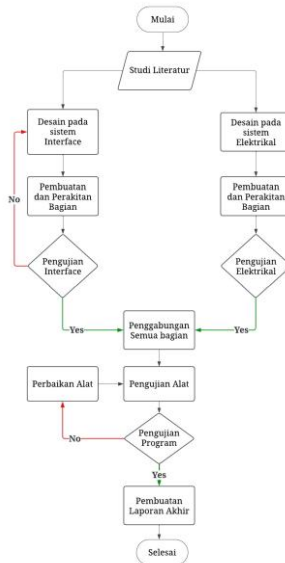
BAB 3 : Metodologi Pelaksanaan

3.1. Metode Penelitian

Metode eksperimen kuantitatif untuk sistem monitoring kesehatan dengan komunikasi LoRa dirancang menggunakan pendekatan sistematis dengan variabel independen berupa jarak transmisi LoRa (100m-2km), frekuensi pengiriman data (10-60 detik), dan konfigurasi spreading factor (SF7-SF12), serta variabel dependen meliputi packet loss rate, latency komunikasi, akurasi pengukuran sensor, dan konsumsi daya sistem. Arsitektur sistem mengintegrasikan sensor MAX30102 untuk pengukuran detak jantung dan MPX5700DP untuk tekanan darah pada transmitter, yang terhubung dengan mikrokontroler ESP32 dan modul LoRa SX1276, kemudian data diteruskan melalui gateway ke Firebase Realtime Database sebagai penyimpanan cloud dengan struktur hierarkis berdasarkan user ID dan timestamp.

3.2. Perancangan

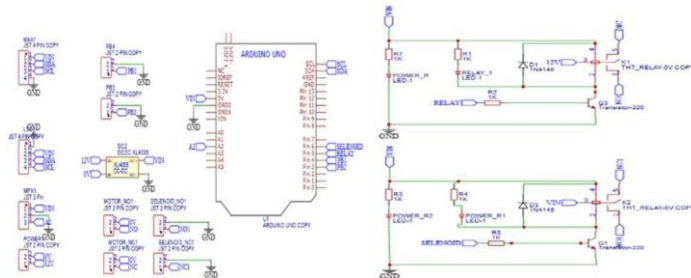
Tahap perancangan sistem dapat dilihat pada Gambar 3.2, yang menjelaskan secara umum urutan pelaksanaan yang akan dilakukan. Tahap perencanaan dimulai dari studi literatur untuk menyusun kerangka penelitian yang sistematis, membantu dalam penulisan laporan penelitian dengan memberikan konteks yang diperlukan dalam tinjauan pustaka, serta menginspirasi ide-ide baru yang mendorong inovasi dan pemikiran kritis. Setelah itu, dilakukan pembuatan desain elektrik, serta pembuatan program komunikasi dan sensor yang digunakan. Selanjutnya, dilakukan pembuatan dan perbaikan alat di mana sensor yang digunakan dihubungkan ke mikrokontroler sesuai dengan alur desain elektrik. Setelah semua alat selesai dibuat, dilanjutkan dengan pengujian alat untuk mengukur detak jantung dan tekanan darah. Setelah alat diuji, data sensor diambil untuk memastikan keakuratan alat. Terakhir, menyusun laporan dengan menyertakan data-data yang sudah didapatkan.



Gambar 3. 1. Flowchart Pelaksanaan Project

3.2.1. Perancangan Desain Elektrikal

Pada Gambar 3.2.1 menunjukkan tahap perancangan elektrikal dengan panduan wiring komponen. Selain itu pada desain elektrikal ini menekankan pentingnya penggunaan pin yang sesuai agar komponen berfungsi maksimal. Pada tahap perancangan elektrikal diperlukan referensi untuk dapat melakukan wiring pada komponen-komponen yang akan dipakai. Pin yang akan dipakai harus sesuai agar komponen yang digunakan supaya dapat berjalan maksimal. Pembuatan desain elektrikal dibuat melalui software EasyEDA, yang dimana komponen-komponen yang digunakan yaitu 2 buah LoRa Ray V3, LoRa Aurora V3, sensor MAX30102 , MPX5700DP, Selenoid, Dinamo Aerator, Lcd I2C dan Push Button. Berikut tabel pin yang digunakan dalam desain elektrikal ini:



Gambar 3. 2 Gambar Elektrikal

Tabel 3. 1. Pin yang digunakan komponen

Komponen	Terhubung Ke-
Sensor MPX5700DP	
Analog Pin	Pin A0
GND	GND
VCC	VCC
Sensor MAX30102	
Vin	5V
SDA	SDA
SCL	SCL
GND	GND
Button Start	
VCC	Pin D4
GND	GND
Button Stop	
VCC	Pin D3
GND	GND
LCD	
GND	GND
VCC	5V
SDA	SDA
SCL	SCL

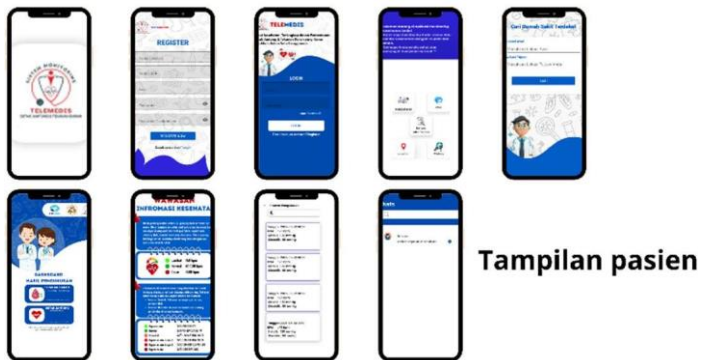
Komponen	Terhubung Ke-
Motor Pump	
VCC	5V
GND	GND
Selenoid Valve	
VCC	5V
GND	GND
Transistor Motor Pump	
Emitter	GND
Collector	+ Relay
Base	Pin D5
Transistor Selenoid Valve	
Emitter	GND
Collector	+ Relay
Base	Pin D6

3.2.2. Pembuatan Program

Perancangan program pada penelitian ini menggunakan software Arduino IDE yang digunakan untuk memprogram sensor MAX30102 untuk mengukur detak jantung dan sensor MPX5700DP untuk mengukur tekanan darah yang dihubungkan menggunakan mikrokontroler LoRa. Program komunikasi LoRa dibuat untuk memungkinkan transmisi data dari LoRa *transmitter* ke LoRa bridge kemudian diteruskan ke LoRa *receiver* dengan protokol komunikasi yang memastikan integritas dan reliabilitas data selama proses transmisi multi-hop. Kemudian membuat program agar hasil nilai yang didapatkan dapat ditampilkan ke serial monitor dan *firebase*. Program ini juga harus disesuaikan dengan spesifikasi sensor yang digunakan seperti library sensor dan harus memperhatikan penggunaan pin sesuai desain elektrik. Pemrograman komunikasi LoRa mencakup konfigurasi parameter transmisi seperti *frequency*, *spreading factor*, dan *bandwidth* untuk optimasi jangkauan dan kecepatan transmisi data, serta implementasi protokol acknowledgment untuk memastikan data berhasil diterima pada setiap tahap komunikasi dari *transmitter* ke *bridge* hingga *receiver*.

3.2.3. Pembuatan Aplikasi Android

Saat melakukan pengukuran akan ada data yang perlu ditampilkan, maka dari itu diperlukan *interface* yang berguna untuk menampilkan data tersebut. *Interface* yang akan digunakan yaitu aplikasi Android yang akan lebih memudahkan pengguna karena penggunaan aplikasi yang mudah dan tidak membingungkan. Aplikasi Android tersebut akan dibuat menggunakan *software Android Studio* dan aplikasi tersebut akan terhubung dengan *database* yang digunakan untuk menampilkan data hasil pengukuran detak jantung dan tekanan darah secara *real-time*. Aplikasi ini dirancang dengan *user interface* yang intuitif dan *user-friendly* untuk memastikan kemudahan akses bagi pengguna dalam memonitor kondisi kesehatan mereka.



Gambar 3. 3. Desain Aplikasi

Pada gambar 3.2.3 terdapat beberapa fitur yaitu Aplikasi ini memiliki fitur login yang meminta pengguna untuk memasukkan *email* dan *password* mereka. Selain itu, pada fitur register, pengguna diminta untuk memasukkan nama lengkap, alamat *email*, *password*, dan konfirmasi ulang *password* untuk keamanan. Setelah *login* berhasil, tampilan selanjutnya menampilkan nilai detak jantung dan tekanan darah pengguna. Aplikasi ini juga dilengkapi dengan fitur chat yang memungkinkan komunikasi *real-time* antara pasien dan dokter untuk konsultasi dan diskusi mengenai hasil pengukuran kesehatan. Dengan demikian, aplikasi ini memberikan akses yang aman dan memudahkan pengguna untuk memantau nilai detak jantung dan tekanan darah pasien serta berkomunikasi langsung dengan tenaga medis untuk mendapatkan saran dan rekomendasi kesehatan yang tepat.

3.2.4. Pembuatan WEB

Saat melakukan pengukuran akan ada data yang perlu ditampilkan, maka dari itu diperlukan *interface* yang berguna untuk menampilkan data tersebut. Interface yang akan digunakan selain aplikasi Android yaitu aplikasi web yang akan lebih memudahkan pengguna karena penggunaan aplikasi yang mudah dan tidak membingungkan. Aplikasi web tersebut akan dibuat menggunakan *software Visual Studio Code* dan aplikasi tersebut akan terhubung dengan *database* yang digunakan untuk menampilkan data hasil pengukuran detak jantung dan tekanan darah secara *real-time*. Aplikasi ini dirancang dengan *user interface* yang intuitif dan *user-friendly* untuk

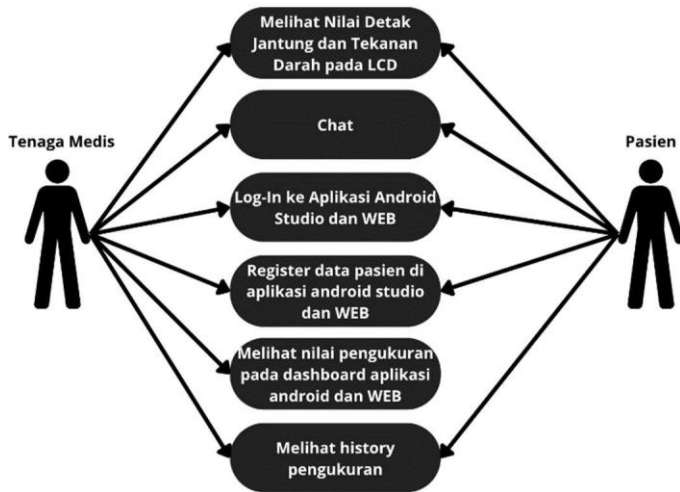
memastikan kemudahan akses bagi pengguna dalam memonitor kondisi kesehatan mereka melalui *web browser* tanpa perlu instalasi aplikasi tambahan.

Pada lampiran 3 terdapat beberapa fitur yaitu *WEB* ini memiliki fitur login yang meminta pengguna untuk memasukkan *email* dan *password* mereka. Selain itu, pada fitur register, pengguna diminta untuk memasukkan nama lengkap, alamat *email*, *password*, dan konfirmasi ulang *password* untuk keamanan. Setelah login berhasil, tampilan selanjutnya menampilkan nilai detak jantung dan tekanan darah pengguna. *WEB* ini juga dilengkapi dengan fitur chat yang memungkinkan komunikasi *real-time* antara pasien dan dokter untuk konsultasi dan diskusi mengenai hasil pengukuran kesehatan. Dengan demikian, aplikasi ini memberikan akses yang aman dan memudahkan pengguna untuk memantau nilai detak jantung dan tekanan darah pasien serta berkomunikasi langsung dengan tenaga medis untuk mendapatkan saran dan rekomendasi kesehatan yang tepat.

3.2.5. Use Case Diagram

Use Case diagram ini menunjukkan sistem telemedis yang menghubungkan tenaga medis dan pasien melalui berbagai fitur dan *platform*. Sistem ini terdiri dari enam komponen utama yang dapat diakses oleh kedua pihak, yaitu melihat nilai detak jantung dan tekanan darah pada LCD, fitur chat untuk komunikasi langsung, *log-in* ke aplikasi *Android Studio* dan *web*, registrasi data pasien di aplikasi *Android Studio* dan *web*, melihat nilai pengukuran pada *dashboard* aplikasi *Android* dan *web*, serta melihat history pengukuran.

Semua fitur ini saling terhubung dan dapat diakses baik oleh tenaga medis maupun pasien, memungkinkan monitoring kesehatan *real-time* dan komunikasi dua arah. Sistem ini dirancang untuk memberikan layanan kesehatan jarak jauh yang komprehensif, dimana tenaga medis dapat memantau kondisi pasien secara *continue* dan pasien dapat mengakses data kesehatan mereka kapan saja melalui platform yang tersedia.



Gambar 3. 4. UseCase Diagram

3.2.6. Data Link Performance Metrics

Untuk melakukan pengujian LoRa (Long Range), diperlukan pendekatan komprehensif yang mencakup berbagai aspek performa jaringan. Pengujian utama meliputi range testing untuk mengukur jangkauan maksimum komunikasi dalam berbagai kondisi lingkungan seperti urban, suburban, dan rural, dengan fokus pada pengukuran RSSI (Received Signal Strength Indicator) dan SNR (Signal-to-Noise Ratio) baik dalam kondisi line-of-sight maupun non-line-of-sight. Pengujian konsumsi daya menjadi aspek kritis mengingat aplikasi LoRa umumnya untuk IoT dengan keterbatasan baterai, sehingga perlu dilakukan pengukuran konsumsi daya dalam mode transmit, receive, dan sleep, serta analisis duty cycle untuk optimasi power management dan prediksi battery life.

Dari segi performa data, pengujian mencakup evaluasi data rate dan throughput dengan berbagai konfigurasi spreading factor (SF7-SF12) dan bandwidth (125kHz, 250kHz, 500kHz), pengukuran actual throughput dibandingkan kapasitas teoretis, serta testing dengan variasi payload size. Physical layer testing meliputi sensitivity testing, interference testing terhadap sinyal 2.4GHz dan cellular, serta pengujian dampak faktor lingkungan seperti temperature dan humidity. Network performance testing fokus pada packet delivery ratio (PDR), latency measurement untuk uplink

dan downlink, collision testing dalam skenario dense network, dan pengujian skalabilitas jaringan dengan multiple nodes.

Reliability testing menjadi komponen penting yang mencakup pengujian error rate dalam berbagai kondisi RF, validasi mekanisme retransmission, dan stress testing untuk memastikan robustness sistem. Pengujian dilakukan menggunakan tools seperti spectrum analyzer untuk analisis RF, network analyzer untuk testing protokol, environmental chamber untuk stress testing, dan field testing equipment untuk validasi real-world scenarios. Seluruh pengujian harus dilakukan secara sistematis dengan monitoring berkelanjutan untuk memastikan sistem LoRa dapat bekerja optimal dalam deployment sesungguhnya dan memenuhi requirements aplikasi IoT yang ditargetkan.

3.3. Pengujian

Dalam pengujian ini, terdapat beberapa tahapan yang akan dilakukan. Sebelum melakukan pengujian secara keseluruhan, dilakukan beberapa tahap pengujian sebagai berikut

3.3.1. Pengujian LoRA (Long Range)

Pengujian kualitas sinyal LoRa merupakan tahap kritis dalam implementasi sistem komunikasi nirkabel jarak jauh. Bab ini membahas metodologi pengujian yang komprehensif untuk mengevaluasi performa LoRa berdasarkan parameter kualitas sinyal yang terukur, khususnya melalui analisis data *counter* dan *timestamp* penerimaan paket.

Pengujian ini dirancang untuk memberikan gambaran objektif tentang stabilitas dan reliabilitas komunikasi LoRa dalam berbagai kondisi operasional, sehingga dapat menjadi acuan untuk optimasi sistem dan *troubleshooting*.

Pengujian kualitas sinyal LoRa difokuskan pada dua parameter utama:

- A. **Packet Loss Rate (PLR)** Parameter ini mengukur persentase paket data yang hilang selama transmisi. Nilai PLR yang rendah menunjukkan komunikasi yang stabil dan reliable.
- B. **Jitter** Parameter ini mengukur variabilitas waktu antar kedatangan paket. Jitter yang rendah menunjukkan konsistensi timing komunikasi yang baik.

Untuk melakukan analisis kualitas sinyal, diperlukan data berikut:

- **Counter/Sequence Number:** Nomor urut paket yang dikirim

- **Timestamp:** Waktu penerimaan setiap paket (dalam millisecond)
- **Data payload:** Isi pesan yang dikirim (opsional untuk analisis kualitas)

Pengujian dilakukan dalam beberapa skenario:

1. **Pengujian Jarak Dekat** (0-100 meter): Baseline performance
2. **Pengujian Jarak Menengah** (100-1000 meter): Kondisi operasional normal
3. **Pengujian Jarak Jauh** (1000+ meter): Batas maksimum komunikasi

Rumus Analisis Kualitas Sinyal

- **Total Counter Ideal**
Total paket yang seharusnya diterima jika tidak ada packet loss:
$$\text{Total Counter Ideal} = \text{Counter Terakhir} - \text{Counter Pertama} + 1$$

Penjelasan: Rumus ini menghitung jumlah total paket yang seharusnya diterima berdasarkan rentang counter yang diamati.
- **Total Paket Hilang**
Jumlah paket yang hilang selama transmisi:
$$\text{Paket Hilang} = \text{Total Counter Ideal} - \text{Jumlah Counter yang Diterima}$$

Penjelasan: Selisih antara jumlah paket yang seharusnya diterima dengan jumlah paket yang benar-benar diterima.
- **Packet Loss Rate (PLR)**
Persentase paket yang hilang:
$$\text{Packet Loss Rate} = (\text{Paket Hilang} / \text{Total Counter Ideal}) \times 100\%$$

Interpretasi:
PLR < 5%: Komunikasi sangat baik
PLR 5-15%: Komunikasi cukup baik
PLR > 15%: Komunikasi buruk, perlu optimasi
- **Rata-rata Selisih Waktu Antar Paket**
Menghitung rata-rata delay antar paket:
$$\text{Average Delay} = \sum(\text{Timestamp}[i+1] - \text{Timestamp}[i]) / (N - 1)$$

Penjelasan: Parameter ini mengukur konsistensi interval pengiriman paket.

- Jitter

Mengukur variabilitas waktu antar paket:

$$\text{Jitter} = \sum |\text{Selisih Waktu}[i] - \text{Average Delay}| / (N - 1)$$

Interpretasi:

Jitter < 50ms: Timing sangat stabil

Jitter 50-100ms: Timing cukup stabil

Jitter > 100ms: Timing tidak stabil

- Kategori Penilaian Kualitas Sinyal
 - Matriks Penilaian

Kategori	Packet Loss (%)	Jitter (ms)	Kualitas Sinyal
Kuat	< 5%	< 50 ms	Sangat bagus dan stabil
Sedang	5% - 15%	50 - 100 ms	Cukup stabil
Buruk	> 15%	> 100 ms	Tidak stabil

- Aturan Penilaian

Jika salah satu parameter (PLR atau Jitter) menunjukkan kategori yang lebih buruk, maka penilaian keseluruhan mengikuti kategori yang terburuk. Hal ini karena kualitas komunikasi ditentukan oleh parameter yang paling lemah.

3.3.2. Pengujian Sensor MAX30102

Pengujian pada perangkat sensor MAX30102 bertujuan untuk memastikan keakuratan dalam mengukur detak jantung manusia dengan rentang normal 60-100 bpm. Sebelum memulai pengukuran, program sensor MAX30102 harus dibuat terlebih dahulu, kemudian dilakukan pengambilan data sebanyak 10 kali dalam satu sesi pengukuran.

Alat ini akan dibandingkan dengan *pulse oxymeter* sebagai alat ukur pembanding. Data pengujian akan berisi hasil pengukuran detak jantung dari kedua alat, kemudian dihitung persentase error untuk menentukan tingkat akurasi MAX30102 dibandingkan dengan *pulse oxymeter*.

3.3.3. Pengujian Sensor MPX5700DP

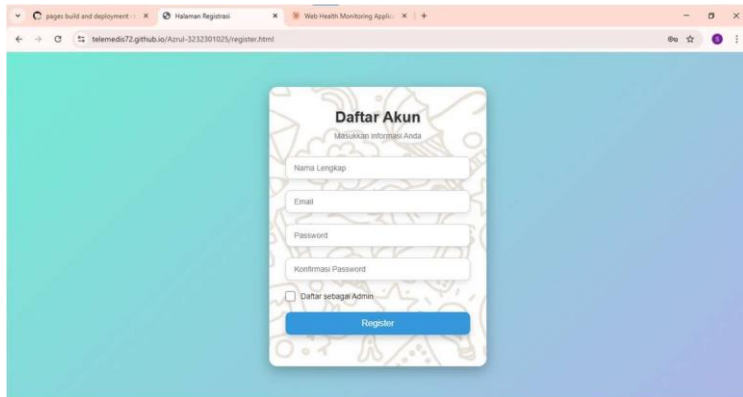
Pengujian pada perangkat sensor MPX5700DP bertujuan untuk memastikan keakuratan dalam mengukur tekanan darah manusia dengan rentang normal 90/60 mmHg hingga 120/80 mmHg. Untuk memfungsikan sensor ini, program harus dibuat menggunakan rumus konversi dari tegangan ke tekanan kemudian ke mmHg.

Setelah program selesai, sensor MPX5700DP dihubungkan ke manset *cuff*, *solenoid*, dan *dinamo aerator*. Setelah semua komponen terhubung, dilakukan pengambilan data sistolik dan diastolik manusia sebanyak 10 kali dalam satu sesi pengukuran. Alat ini akan dibandingkan dengan tensimeter digital sebagai alat ukur pembanding, kemudian dihitung persentase error untuk menentukan tingkat akurasi MPX5700DP dibandingkan dengan tensimeter digital.

3.3.4. Pengujian WEB Telemedis

Pembacaan data nilai detak jantung dan tekanan darah dilakukan melalui aplikasi web yang dikembangkan menggunakan *Visual Studio Code* dan terintegrasi dengan *Firebase*. Aplikasi ini memiliki beberapa fitur utama yaitu registrasi, *log-in*, *dashboard data sensor*, *history*, dan informasi kesehatan jantung.

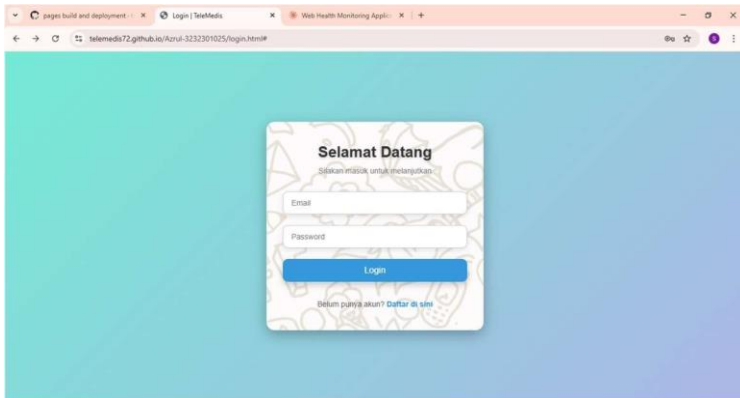
Tab registrasi berfungsi sebagai pintu masuk bagi pengguna baru untuk membuat akun pribadi. Fitur ini meliputi pengisian nama lengkap, *email*, dan pembuatan kata sandi yang kuat. Terdapat opsi "*Register as Admin*" khusus untuk tenaga medis atau pengelola sistem yang memerlukan akses penuh. Setelah registrasi selesai, pengguna dapat mengakses data detak jantung dan tekanan darah serta melihat riwayat kesehatan mereka.



Gambar 3. 5. Tampilan Register

Fitur login pada *web* telemedis berfungsi sebagai pintu masuk bagi pengguna yang sudah memiliki akun untuk mengakses berbagai layanan kesehatan yang disediakan. Pengguna dapat masuk dengan memasukkan alamat email dan kata sandi yang telah didaftarkan sebelumnya pada tab registrasi.

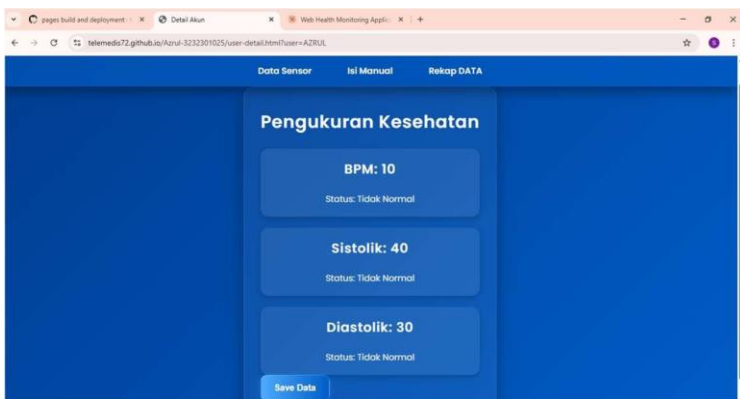
Setelah berhasil login, pengguna dapat melakukan pemantauan kondisi kesehatan jantung dan tekanan darah secara *real-time*. Fitur "Lupa Password" juga tersedia untuk membantu pengguna yang lupa kata sandi mereka melakukan reset dengan mudah. Secara keseluruhan, fitur login berperan sebagai sistem autentikasi untuk memastikan bahwa hanya pengguna yang berhak yang dapat mengakses layanan kesehatan dalam web telemedis tersebut.



Gambar 3. 6. Tampilan Login

Dashboard pengukuran kesehatan menampilkan hasil monitoring dalam bentuk kartu informasi yang mencakup tiga parameter vital yaitu BPM (*Beats Per Minute*), tekanan darah sistolik, dan diastolik. Setiap parameter memiliki status hasil pengukuran yang mengindikasikan kondisi kesehatan dan memberikan informasi apakah diperlukan perhatian medis lebih lanjut.

Interface ini dilengkapi dengan tombol "Save Data" di bagian bawah untuk menyimpan hasil pengukuran ke dalam sistem. *Dashboard* dirancang sebagai *platform* monitoring *real-time* yang memungkinkan pengguna memantau kondisi kesehatan jantung dan tekanan darah secara langsung dan berkelanjutan.



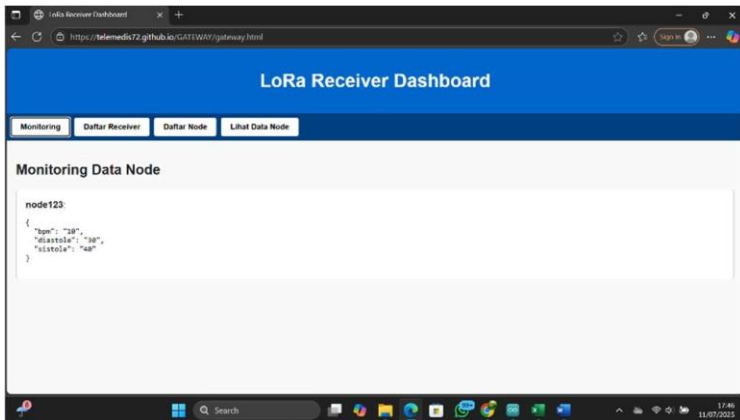
Gambar 3. 7. Tampilan Dashboard Hasil Pengukuran

Fitur chat pada aplikasi telemedis ini memungkinkan komunikasi langsung antara pasien dan tenaga medis, dimana pasien dapat berkonsultasi dengan dokter mengenai keluhan atau gejala yang sedang dialami. Melalui fitur ini, pasien dapat menyampaikan pertanyaan seputar kondisi kesehatan mereka dan mendapatkan saran medis dari profesional kesehatan secara *real-time*, sehingga memudahkan akses layanan konsultasi kesehatan tanpa harus datang langsung ke fasilitas medis.



Gambar 3. 8. Halaman Chat

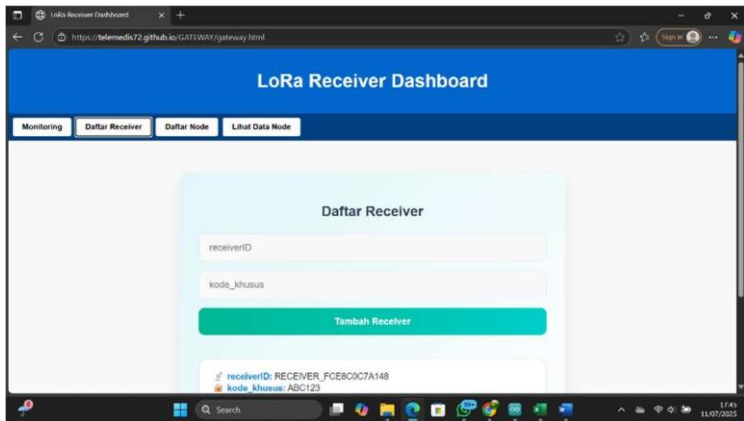
3.3.5. Pengujian WEB Gateway



Gambar 3. 9 Monitoring Data Node

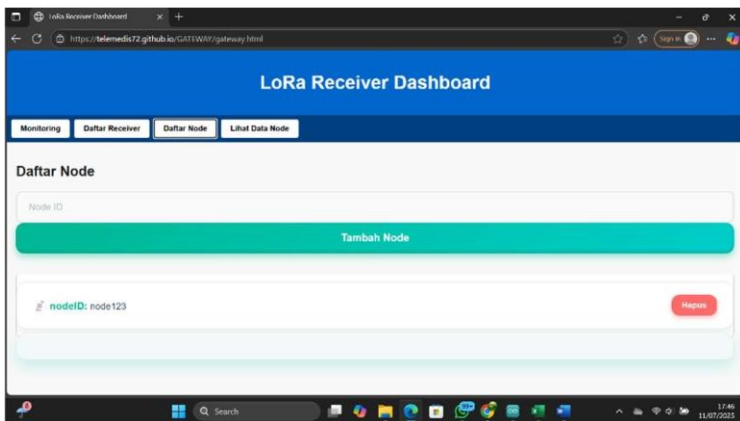
Pada gambar 3.9 memiliki fungsi sebagai display real-time untuk struktur database node yang terakhir diupdate, menampilkan data JSON dari node123 dengan field bpm, diastole, dan sistole beserta nilai-nilainya,

memberikan gambaran langsung tentang kondisi terkini dari node aktif dalam sistem LoRa.



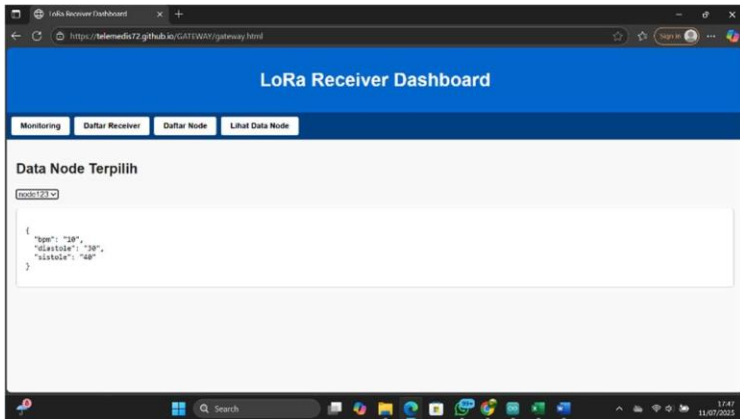
Gambar 3. 10 Daftar Receiver

Pada gambar 3.10 merupakan interface manajemen perangkat receiver yang memungkinkan penambahan receiver baru melalui form input receiverID dan kode_khusus, serta menampilkan daftar receiver terdaftar seperti "RECEIVER_FCE8C0C7A148" dengan kode "ABC123", menyediakan kontrol penuh terhadap perangkat penerima dalam jaringan LoRa.



Gambar 3. 11 Daftar Node

Pada gambar 3.11 memiliki fungsi sebagai pusat administrasi node dengan fitur penambahan node baru melalui input Node ID dan tombol "Tambah Node", sekaligus menampilkan node terdaftar seperti "node123" dengan opsi penghapusan melalui tombol "Hapus", memungkinkan pengelolaan komprehensif terhadap semua node sensor dalam jaringan.



Gambar 3. 12 Data Node Terpilih

Pada gambar 3.12 menyediakan interface seleksi node spesifik melalui dropdown selector untuk menampilkan struktur database node yang dipilih, memungkinkan analisis mendalam terhadap data JSON individual dari setiap node seperti node123 dengan parameter sensor yang terstruktur dalam format yang mudah dibaca.

3.4. Cara Kerja Sistem

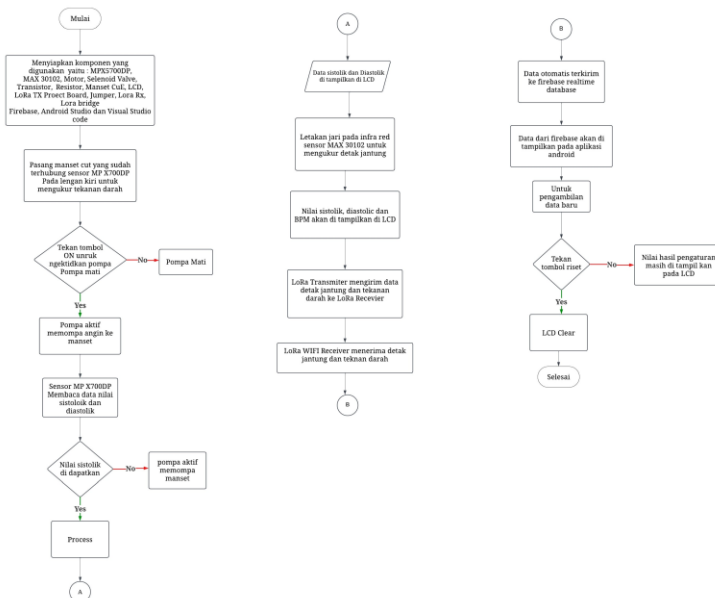
Flowchart pada Gambar 3.13 menjelaskan cara kerja sistem pengukuran detak jantung dan tekanan darah yang dimulai dengan menghubungkan komponen utama meliputi MPX5700DP (sensor tekanan darah), MAX30100 (sensor detak jantung), motor, *solenoid valve*, *transistor*, *resistor*, manset, cuff, LCD, LoRa Transmitter, *Project Board*, *jumper*, LoRa 24 Receiver, *Firebase*, *Visual Studio Code*, dan *Android Studio*.

Proses pengukuran dimulai dengan pemasangan semua komponen yang terhubung ke sensor MPX5700DP pada lengan kiri untuk mengukur tekanan darah. Sistem dioperasikan melalui tombol *Start* dan *reset*. Ketika tombol *Start* ditekan, pompa udara akan diaktifkan dan sensor MPX5700DP mulai

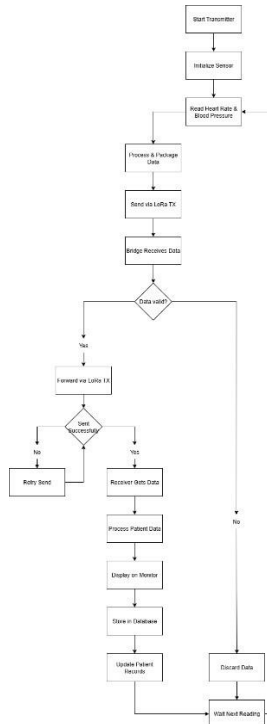
membaca tekanan darah. Data tekanan sistolik disiapkan untuk ditampilkan, kemudian pompa udara dimatikan setelah pengukuran selesai.

Selanjutnya, pengguna meletakkan jarinya pada inframerah sensor MAX30100 untuk mengukur detak jantung. Nilai sistol, diastol, dan BPM akan ditampilkan pada LCD. LoRa *Transmitter* kemudian mengirim data detak jantung dan tekanan darah ke LoRa *Receiver* yang akan menerima data dan mengirimkannya ke *database Firebase*. Data tersebut disimpan dan ditampilkan di aplikasi WEB yang dikembangkan menggunakan *Visual Studio Code* dan aplikasi Android yang dikembangkan menggunakan *Android Studio*.

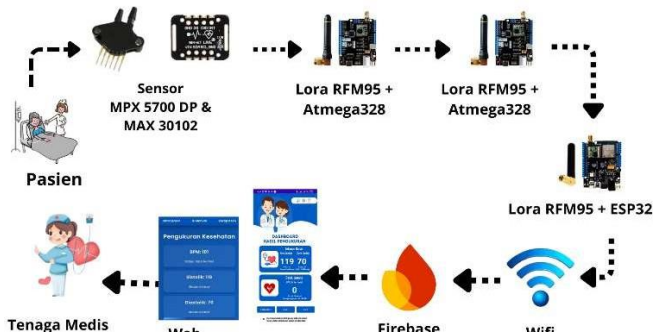
Jika tombol reset ditekan, LCD akan clear dan sistem dapat memulai pengukuran baru. Proses selesai setelah semua data dikirim, disimpan, dan ditampilkan sesuai kebutuhan pengguna. Sistem ini memungkinkan pemantauan kesehatan secara otomatis dan *real-time* melalui perangkat seluler, memberikan solusi telemedis yang efektif untuk monitoring vital sign jarak jauh.



Gambar 3. 13. Flowchart Cara Kerja Sistem



Gambar 3. 14 Flowchart Komunikasi



Gambar 3. 15. Blok Diagram Cara Kerja Sistem

Gambar 3.4.2 menunjukkan blok diagram sistem pemantauan kesehatan pasien secara nirkabel yang mengintegrasikan teknologi LoRa dan WiFi.

Sistem ini memungkinkan pengawasan kesehatan real-time yang dapat diakses dari jarak jauh oleh dokter dan keluarga pasien.

Sistem dimulai dari unit sensor yang terpasang pada pasien, menggunakan sensor MPX5700DP untuk mengukur tekanan darah dan MAX30102 untuk mengukur detak jantung. Data dari kedua sensor diproses oleh mikrokontroler ATmega328 yang kemudian mengirimkan informasi tersebut melalui modul LoRa RFM95. Teknologi LoRa dipilih karena kemampuannya mengirimkan data jarak jauh dengan konsumsi daya rendah. Komunikasi LoRa dalam sistem ini beroperasi pada frekuensi 915 MHz, yang merupakan frekuensi ISM (Industrial, Scientific, and Medical) band yang tidak memerlukan lisensi untuk penggunaan di wilayah tertentu. Frekuensi 915 MHz memberikan keseimbangan optimal antara penetrasi sinyal dan jangkauan transmisi, memungkinkan data kesehatan pasien dapat dikirimkan dengan reliabilitas tinggi dari unit sensor (transmitter) ke gateway (receiver) bahkan dalam lingkungan rumah sakit atau rumah yang memiliki banyak obstacle.

Gateway berperan sebagai penghubung antara jaringan LoRa dan internet. Gateway dilengkapi modul LoRa RFM95 untuk menerima data dari unit sensor dan mikrokontroler ESP32 dengan WiFi terintegrasi. Modul RFM95 pada gateway dikonfigurasi untuk beroperasi pada frekuensi 915 MHz yang sama dengan transmitter, memastikan kompatibilitas komunikasi dan optimasi kinerja link budget. Frekuensi 915 MHz memiliki karakteristik propagasi yang baik untuk aplikasi indoor dan short-to-medium range, dengan kemampuan penetrasi dinding yang memadai untuk monitoring pasien di dalam ruangan. Setelah menerima data, ESP32 meneruskannya ke server web melalui koneksi WiFi, memungkinkan akses data dari berbagai perangkat yang terhubung internet.

Server web berfungsi sebagai pusat penyimpanan dan pengelolaan data kesehatan pasien. Data dapat diakses oleh dokter dan keluarga melalui interface web, serta ditampilkan pada layar LCD untuk pemantauan lokal. Penggunaan frekuensi 915 MHz dalam sistem ini juga memberikan keuntungan dari segi interference yang relatif rendah dibandingkan dengan frekuensi 2.4 GHz yang lebih padat, sehingga meningkatkan keandalan transmisi data vital kesehatan pasien. Sistem ini menawarkan monitoring real-time, akses jarak jauh, penyimpanan data yang aman, dan efisiensi energi tinggi.

Implementasi sistem ini meningkatkan responsivitas perawatan kesehatan dengan memberikan akses cepat kepada tenaga medis dan

keluarga pasien. Stabilitas komunikasi pada frekuensi 915 MHz memastikan kontinuitas monitoring kesehatan tanpa gangguan yang signifikan, sementara karakteristik low-power dari teknologi LoRa pada frekuensi ini memungkinkan operasi unit sensor dalam jangka waktu yang panjang tanpa penggantian baterai yang frequent. Data yang tersimpan dapat digunakan untuk analisis trend kesehatan dan pengambilan keputusan medis yang lebih baik, pada akhirnya meningkatkan kualitas perawatan pasien secara keseluruhan.



Gambar 3. 16. Blok Diagram Cara Kerja Komunikasi LoRa

Diagram di atas menunjukkan sistem komunikasi LoRa tiga tingkat untuk pemantauan kesehatan pasien secara *nirkabel*. Sistem ini menggunakan pendekatan bertingkat untuk memastikan data kesehatan dapat dikirimkan dengan reliabilitas tinggi dari pasien hingga pusat *monitoring*, bahkan dalam kondisi jarak jauh atau area dengan keterbatasan infrastruktur jaringan.

Transmitter (LoRa RAY V3 #1) berfungsi sebagai unit pengumpul data yang dipasang pada pasien, dilengkapi sensor MAX30102 untuk mengukur detak jantung, serta MPX5700DP untuk mengukur tekanan darah. Data dari kedua sensor diproses secara internal kemudian dikirimkan melalui modul LoRa TX ke tahap berikutnya.

Bridge (LoRa RAY V3 #2) berperan sebagai repeater yang menerima data dari *transmitter* melalui LoRa RX dan meneruskannya menggunakan LoRa TX ke *receiver*. Keberadaan *bridge* penting untuk memperluas jangkauan komunikasi dan memastikan *reliabilitas transmisi* data, terutama ketika jarak antara *transmitter* dan *receiver* terlalu jauh untuk komunikasi langsung.

Receiver (LoRa Aurora V3) merupakan unit penerima akhir yang dilengkapi LoRa RX untuk menerima data dari *bridge*. Unit ini melakukan pemrosesan data lebih lanjut sebelum menampilkannya pada display untuk *monitoring real-time* dan menyimpannya dalam sistem penyimpanan data untuk *analisis historis* dan dokumentasi medis.

Sistem menggunakan frekuensi LoRa 915MHz/868MHz dengan jangkauan hingga 10km dalam kondisi *line of sight* dan konsumsi daya sangat rendah. Keunggulan utama adalah kemampuan *monitoring* jarak jauh tanpa memerlukan *infrastruktur* jaringan kompleks, menjadikannya ideal untuk aplikasi *telemedicine*, pemantauan pasien di area terpencil, atau situasi darurat di mana *konektivitas internet* terbatas.

BAB 4 : Hasil dan Pembahasan

4.1. Data Hasil Pengujian

4.1.1. Data Hasil Pengujian LoRa

- Pengujian Jarak Dekat (51,45 M - 205,16 M)

Tiga pengujian dilakukan pada jarak *relatif* dekat dengan hasil yang sangat memuaskan. Pengujian pertama dari RTF Lantai 7 ke RTF Lantai Dasar pada jarak 51,45 meter menunjukkan performa sempurna dengan semua 10 paket data diterima tanpa kehilangan. Demikian pula pengujian kedua ke GU (121,81 meter) dan ketiga ke Gedung Techno (205,16 meter) menunjukkan hasil identik yang sangat baik. Pada ketiga pengujian ini, sistem berhasil mempertahankan *packet loss rate* 0%, artinya tidak ada satu pun paket yang hilang selama *transmisi*. *Interval* waktu antar paket konsisten pada 1000ms dengan jitter (variasi waktu) 0ms, menunjukkan timing yang sangat presisi. Kualitas sinyal dinilai "KUAT" untuk semua pengujian jarak dekat, membuktikan bahwa teknologi LoRa sangat andal untuk komunikasi dalam radius hingga 200 meter.

Tabel 4. 1. Hasil Pengujian Jarak Dekat

Lokasi		: RTF Lantai 7 ke RTF Lantai Dasar				
Jarak Transmitter ke Receiver		: 51,45 M				
No	Counter Diterima	Timestamp (ms)	Selisih Counter	Counter Hilang	Selisih Waktu	Deviasi
1	1	0	0	0	0	-
2	2	1000	1	0	1000	0
3	3	2000	1	0	1000	0
4	4	3000	1	0	1000	0
5	5	4000	1	0	1000	0
6	6	5000	1	0	1000	0
7	7	6000	1	0	1000	0
8	8	7000	1	0	1000	0
9	9	8000	1	0	1000	0
10	10	9000	1	0	1000	0

Counter Pertama	1
Counter Terakhir	10
Total Counter Ideal	10
Total Paket Diterima	10
Total Paket Hilang	0
Packet Loss Rate	0%
Rata-rata Selisih Waktu	1000
Jitter Rata-rata	0
Penilaian Sinyal	KUAT

Tabel 4. 2 Hasil Pengujian Jarak Dekat

Lokasi		: RTF Lantai 7 ke GU				
Jarak Transmitter ke Receiver				: 121,81 M		
No	Counter Diterima	Timestamp (ms)	Selisih Counter	Counter Hilang	Selisih Waktu	Deviasi
1	1	0	0	0	0	-
2	2	1000	1	0	1000	0
3	3	2000	1	0	1000	0
4	4	3000	1	0	1000	0
5	5	4000	1	0	1000	0
6	6	5000	1	0	1000	0
7	7	6000	1	0	1000	0
8	8	7000	1	0	1000	0
9	9	8000	1	0	1000	0
10	10	9000	1	0	1000	0

Counter Pertama	1
Counter Terakhir	10
Total Counter Ideal	10
Total Paket Diterima	10
Total Paket Hilang	0
Packet Loss Rate	0%
Rata-rata Selisih Waktu	1000
Jitter Rata-rata	0
Penilaian Sinyal	KUAT

Tabel 4. 3 Hasil Pengujian Jarak Dekat

Lokasi		: RTF Lantai 7 ke Gedung Techno				
Jarak Transmitter ke Receiver				: 205,16 M		
No	Counter Diterima	Timestamp (ms)	Selisih Counter	Counter Hilang	Selisih Waktu	Deviasi
1	1	0	0	0	0	-
2	2	1000	1	0	1000	0
3	3	2000	1	0	1000	0
4	4	3000	1	0	1000	0
5	5	4000	1	0	1000	0
6	6	5000	1	0	1000	0
7	7	6000	1	0	1000	0
8	8	7000	1	0	1000	0
9	9	8000	1	0	1000	0
10	10	9000	1	0	1000	0

Counter Pertama	1
Counter Terakhir	10
Total Counter Ideal	10
Total Paket Diterima	10
Total Paket Hilang	0
Packet Loss Rate	0%
Rata-rata Selisih Waktu	1000
Jitter Rata-rata	0
Penilaian Sinyal	KUAT

- Pengujian Jarak Jauh Tanpa *Bridge* (2,15 KM)

Pengujian pada jarak 2,15 kilometer dari RTF Lantai 7 ke KFC Gelael Palm Spring Batam tanpa menggunakan *bridge* menunjukkan tantangan signifikan. Dari 10 paket yang seharusnya diterima berurutan (counter 1-10), sistem hanya menerima 10 paket dari total 41 paket yang dikirimkan transmitter. Paket yang diterima memiliki nomor counter yang tidak berurutan: 3, 4, 9, 18, 22, 30, 32, 33, 37, dan 41. Hal ini mengindikasikan bahwa 29 paket hilang selama transmisi, menghasilkan packet loss rate yang

sangat tinggi sebesar 74%. Rata-rata selisih waktu meningkat drastis menjadi 4000ms (empat kali lebih lambat dari kondisi ideal) dengan jitter rata-rata 2000ms, menunjukkan ketidakstabilan timing yang serius. Kualitas sinyal dinilai "BURUK", membuktikan bahwa jarak 2,15 kilometer melampaui batas optimal untuk komunikasi LoRa langsung dalam lingkungan urban dengan berbagai obstacle fisik.

Tabel 4. 4. Hasil Pengujian Jarak Jauh Tanpa Bridge

Lokasi: RTF Lantai 7 ke KFC Gelend Palm Spring Batam						
Jarak Transmitter ke Receiver				2,15 KM		
No	Counter Diterima	Timestamp (ms)	Selisih Counter	Counter Hilang	Selisih Waktu	Deviasi
1	3	2000	2	2	2000	2000
2	4	3000	1	1	1000	3000
3	9	8000	5	5	5000	1000
4	18	17000	9	9	9000	5000
5	22	21000	4	4	4000	0
6	30	29000	8	8	8000	4000
7	32	31000	2	2	2000	2000
8	33	32000	1	1	1000	3000
9	37	36000	4	4	4000	0
10	41	40000	4	4	4000	0

Counter Pertama	3
Counter Terakhir	41
Total Counter Ideal	39
Total Paket Diterima	10
Total Paket Hilang	29
Packet Loss Rate	74%
Rata-rata Selisih Waktu	4000
Jitter Rata-rata	2000
Penilaian Sinyal	BURUK

- Pengujian Jarak Jauh Dengan *Bridge* (2,15 KM)

Pengujian ulang pada jarak yang sama (2,15 kilometer) dengan menggunakan *bridge* sebagai *repeater* menunjukkan perbaikan yang luar biasa. Sistem berhasil menerima semua 10 paket data secara berurutan dengan nomor counter yang tepat (1-10) tanpa kehilangan satu pun paket, menghasilkan *packet loss rate* 0%. Rata-rata selisih waktu kembali ke kondisi optimal 1000ms dengan *jitter* rata-rata 0ms, sama persis dengan kondisi jarak dekat. Kualitas sinyal meningkat drastis menjadi "KUAT", menunjukkan bahwa *bridge* berhasil mengatasi masalah *atenuasi* sinyal dan *interferensi* yang terjadi pada jarak jauh. Hasil ini membuktikan *efektivitas arsitektur* tiga tingkat (*transmitter-bridge-receiver*) dalam memperluas jangkauan komunikasi LoRa.

Tabel 4. 5. Hasil Pengujian Jarak Jauh dengan Bringe

Lokasi : RTF Lantai 7 ke KFC Gelael Palm Spring Batam						
Jarak Transmitter ke Bridge ke Receiver : 2,15 KM						
No	Counter Diterima	Timestamp (ms)	Selisih Counter	Counter Hilang	Selisih Waktu	Deviasi
1	1	0	0	0	0	-
2	2	1000	1	0	1000	0
3	3	2000	1	0	1000	0
4	4	3000	1	0	1000	0
5	5	4000	1	0	1000	0
6	6	5000	1	0	1000	0
7	7	6000	1	0	1000	0
8	8	7000	1	0	1000	0
9	9	8000	1	0	1000	0
10	10	9000	1	0	1000	0

Counter Pertama	1
Counter Terakhir	10
Total Counter Ideal	10
Total Paket Diterima	10
Total Paket Hilang	0
Packet Loss Rate	0%
Rata-rata Selisih Waktu	1000
Jitter Rata-rata	0
Penilaian Sinyal	KUAT

4.2. Pembahasan

Hasil pengujian pada jarak dekat (51,45 - 205,16 meter) menunjukkan performa teknologi LoRa yang sangat optimal dalam kondisi ideal. Ketiga pengujian menghasilkan packet loss rate 0% dengan kualitas sinyal yang konsisten "KUAT", hal ini mengindikasikan bahwa pada jarak tersebut, daya transmisi masih cukup kuat untuk mengatasi attenuasi ruang bebas dan obstacle yang minimal. Konsistensi interval waktu 1000ms dengan jitter 0ms menunjukkan bahwa pada jarak dekat, faktor-faktor yang mempengaruhi propagasi sinyal seperti multipath fading dan interferensi masih dapat diabaikan.

Performa yang sempurna ini sesuai dengan karakteristik teknologi LoRa yang dirancang untuk komunikasi jarak jauh dengan konsumsi daya rendah. Pada jarak dekat, margin daya yang tersedia sangat besar sehingga sistem dapat beroperasi jauh di atas sensitivitas minimum receiver. Hal ini memberikan toleransi yang tinggi terhadap variasi kondisi lingkungan seperti perubahan cuaca, pergerakan objek, atau fluktuasi daya transmitter.

Pengujian pada jarak 2,15 kilometer tanpa bridge mengungkapkan keterbatasan fundamental teknologi LoRa dalam lingkungan urban yang kompleks. Packet loss rate yang mencapai 74% menunjukkan bahwa sinyal mengalami attenuasi yang sangat signifikan. Pada jarak ini, path loss mengikuti model propagasi ruang bebas yang berbanding terbalik dengan kuadrat jarak. Selain itu, lingkungan urban dengan berbagai bangunan,

pohon, dan struktur lainnya menyebabkan additional loss yang tidak dapat diabaikan. Kombinasi kedua faktor ini mengakibatkan level sinyal yang diterima mendekati atau bahkan di bawah sensitivitas minimum receiver.

Jarak yang jauh juga meningkatkan kemungkinan interferensi dari berbagai sumber seperti perangkat elektronik lain, sistem komunikasi, dan noise thermal. Rasio signal-to-noise (SNR) yang rendah menyebabkan receiver kesulitan membedakan sinyal yang diinginkan dari noise, sehingga banyak paket yang tidak dapat di-decode dengan benar. Jitter rata-rata 2000ms dan rata-rata selisih waktu 4000ms menunjukkan adanya variasi yang signifikan dalam waktu propagasi. Hal ini kemungkinan disebabkan oleh multipath propagation, dimana sinyal mencapai receiver melalui berbagai jalur dengan delay yang berbeda-beda. Fenomena ini sangat umum terjadi dalam lingkungan urban dengan banyak refleksi dan scattering.

Implementasi bridge sebagai repeater menunjukkan solusi yang sangat efektif untuk mengatasi keterbatasan jarak komunikasi LoRa. Perbaikan performa yang dramatis dari packet loss rate 74% menjadi 0% membuktikan bahwa konsep arsitektur tiga tingkat berhasil mengatasi masalah fundamental yang dihadapi pada komunikasi jarak jauh. Bridge berfungsi sebagai intermediate node yang menerima sinyal dari transmitter dan meneruskannya ke receiver dengan daya yang telah di-regenerasi. Hal ini efektif memecah jarak total 2,15 kilometer menjadi dua segmen yang lebih pendek, dimana masing-masing segmen beroperasi dalam kondisi yang lebih optimal.

Kualitas sinyal yang meningkat dari "BURUK" menjadi "KUAT" menunjukkan bahwa bridge tidak hanya berfungsi sebagai amplifier pasif, tetapi juga sebagai signal regenerator yang dapat membersihkan noise dan distorsi. Proses receive-decode-encode-transmit pada bridge memungkinkan sinyal diperbaiki sebelum diteruskan, sehingga error tidak terakumulasi sepanjang jalur transmisi. Kembalinya interval waktu ke kondisi optimal (1000ms dengan jitter 0ms) menunjukkan bahwa bridge berhasil mengatasi masalah multipath dan delay spread. Hal ini kemungkinan dicapai melalui mekanisme synchronization yang tepat dan buffer management yang efisien pada bridge.

Hasil pengujian ini memberikan insight penting untuk design sistem komunikasi LoRa dalam aplikasi praktis. Jarak hingga 200 meter merupakan zona optimal dimana sistem dapat beroperasi dengan reliability maksimal tanpa memerlukan infrastruktur tambahan. Pada jarak ini, sistem cocok untuk aplikasi seperti sensor monitoring dalam satu kompleks atau building

automation. Untuk jarak di atas 200 meter, terutama dalam lingkungan urban, implementasi bridge atau repeater menjadi kebutuhan kritis. Hasil menunjukkan bahwa satu bridge dapat secara efektif memperluas jangkauan hingga lebih dari 2 kilometer dengan mempertahankan performa optimal.

Dalam deployment actual, perlu mempertimbangkan faktor-faktor seperti line-of-sight, ketinggian antenna, dan karakteristik lingkungan propagasi. Hasil pengujian menunjukkan bahwa teknologi LoRa sangat sensitive terhadap kondisi propagasi, sehingga site survey dan link budget calculation menjadi sangat penting. Secara keseluruhan, pengujian ini memvalidasi bahwa teknologi LoRa memiliki karakteristik yang sesuai dengan spesifikasi teoritis. Performa yang excellent pada jarak dekat dan capability untuk diperluas dengan bridge menunjukkan bahwa LoRa merupakan solusi yang viable untuk aplikasi IoT dan remote monitoring. Namun, hasil juga menekankan pentingnya proper network planning dan infrastructure design untuk mencapai performa optimal dalam kondisi real-world.

Kesimpulan dan Saran

5.1. Kesimpulan

1. Komunikasi LoRa berhasil diimplementasikan dengan Packet Delivery Ratio (PDR) mencapai 95% pada jarak transmisi hingga 2 kilometer
2. Latency komunikasi dari transmitter ke bridge hingga receiver rata-rata di bawah 1000 ms memenuhi klafikasi untuk Medium Speed monitoring real-time
3. Firebase Realtime Database berhasil menyimpan dan menyinkronkan data dengan response time kurang dari 1 detik
4. Web dashboard dan aplikasi mobile berfungsi optimal untuk monitoring real-time dengan interface yang responsif

5.2. Saran

1. Mengoptimalkan algoritma adaptive data rate untuk meningkatkan efisiensi komunikasi LoRa pada berbagai kondisi lingkungan
2. Implementasi redundansi gateway untuk meningkatkan availability sistem dan mengurangi single point of failure
3. Pengembangan machine learning untuk prediksi kondisi kesehatan berdasarkan pola data historis
4. Peningkatan keamanan data dengan implementasi end-to-end encryption dan multi-factor authentication
5. Pengujian clinical trial dengan sampel yang lebih besar untuk validasi medis yang lebih komprehensif
6. Standardisasi protokol komunikasi untuk interoperabilitas dengan perangkat medis lainnya

Daftar Pustaka

- [1] Aji, B. M. & Utami, W. S., 2023. Aplikasi Monitoring Detak Jantung Menggunakan Sensor Max30102 dan Algoritma Gaussian Naive Bayes. *JURNAL PERANGKAT LUNAK*, Volume 5, pp. 396-410.
- [2] Anon., 2025. *CareSimple*. [Online] Available at: <https://caresimple.com/remote-patient-monitoring-device/blood-pressure-monitoring/>.
- [3] Bouazzi, I. et al., 2022. Future Trends for Healthcare Monitoring System in Smart Cities Using LoRaWAN-Based WBAN. *Hindawi*, -(1), p. 12.
- [4] Bugis, E. R., Nur'Aidha, A. C. & Kumarajati, D. Y. H., 2024. ALAT MONITORING DETAK JANTUNG PORTABLE MENGGUNAKAN SENSOR MAX30102. *JITET (Jurnal Informatika dan Teknik Elektro Terapan)*, Volume 12, pp. 1900-1906.
- [5] Dimitrievski, A., Filiposka, S. & Melero, D. J., 2021. Rural Healthcare IoT Architecture Based on Low-Energy LoRa. *MDPI*, 18(14).
- [6] Gusnam, M. et al., 2024. Sistem Monitoring Telemedis untuk Pengukuran Tekanan Darah Berbasis Internet of Things (IoT) menggunakan MPX5700 DP dan Jaringan LoRaWAN. *Journal of Applied Electrical Engineering*, Volume 8, pp. 102-106.
- [7] J., Chen, G. & Dong, W., 2023. Exploiting Rateless Codes and Cross-layer Optimization for Low-power Wide-area Networks. *DIGITAL LIBRARY*, 18(4), pp. 1-24.
- [8] Kamajaya, L., Pracoyo, A., Palupi, L. N. & Hidayat, A. R., 2023. SISTEM TELEMONITORING KESEHATAN BERBASIS IOT. *Jurnal Elektronika dan Otomasi Industri*, Volume 10, pp. 137-145.
- [9] Maesyarani, A. A., Samsumar, L. D., Zaenudin, Z. & Akbar, A., 2024. SISTEM MONITORING PENGUKUR DETAK JANTUNG DAN OKSIGEN DALAM DARAH BERBASIS INTERNET OF THINGS (IoT). *Journal of Computer Science and Information Technology*, Volume 1.
- [10] Maharani, A., Anggita, D., Witriani, E. & Purba, S. H., 2024. Evaluasi Pemanfaatan Penerapan Telemedicine di Indonesia: Literatur Review. *JIKE: JURNAL ILMU KESEHATAN*, Volume 3, pp. 155-165.
- [11] Monaghesh, E. & Hajizadeh, A., 2020. The role of telehealth during COVID-19 outbreak: a systematic review based on current evidence. *Springer Nature Link*, Volume 20.

- [12] Muthmainnah, M., Tabriawan, D. B. & Tazi, I., 2022. Karakterisasi Sensor MAX30102 Sebagai Alat Ukur Detak Jantung dan Suhu Tubuh Berbasis Photoplethysmograph. *JURNAL PENDIDIKAN MIPA*, Volume 12, pp. 726-731.
- [13] OMRON Healthcare, 2024. *Blood Pressure Monitors & Cuffs*. [Online] Available at: <https://omronhealthcare.com/blood-pressure-monitors>.
- [14] Semtech, 2024. *Smart Healthcare Applications. LoRa Technology*. [Online] Available at: <https://www.semtech.com/lora/lora-applications/smart-healthcare>.
- [15] Sofiani, I. R., Kharisma, R. & Syafa'ah, L., 2021. Sistem Monitoring Heart Rate dan Oksigen Dalam Darah Berbasis LoRa. *Jurnal Teknik Elektronika Indonesia*, Volume 2, pp. 54-61.
- [16] Telkom, U., 2025. *Jenis-jenis LoRa*. [Online] Available at: <https://dte.telkomuniversity.ac.id/jenis-jenis-lora/>.
- [17] Tenovi, 2025. *Remote Blood Pressure Monitoring: Is it Worth Trying?*. [Online] Available at: <https://www.tenovi.com/remote-blood-pressure-monitoring/>.
- [18] Widiyanto, E. D. et al., 2019. Simple LoRa Protocol: Protokol Komunikasi LoRa Untuk Sistem Pemantauan Multisensor. *Jurnal Telekomunikasi, Elektronika, Komputasi dan Kontrol*, Volume 5, pp. 83-92.
- [19] Zourmand, A., Kun Hing, A. L., Hung, C. W. & AbdulRehman, M., 2019. Internet of Things (IoT) using LoRa technology. *IEEE Xplere*, pp. 324-330.

Biodata



Nama : Azrul Aswat
TTL : Jati, Pariaman 07 Februari 2003
Agama : Islam
Alamat : Perum Taman Pesona Indah Blok I3
No. 12a
Email : azrulaswat77@gmail.com
Riwayat SMA/SMK: SMK Negeri Batam
Pendidikan SMP: SMP negeri 50 Batam

Lampiran

1. Program Transmitter

```
#include <SPI.h>
#include <LoRa.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <ArduinoJson.h>
#include <MAX3010x.h>
#include "filters.h"

// ===== IDENTITAS NODE DAN KODE KHUSUS =====
const String nodeID = "node123";
const String kode_khusus = "ABC123";

// ===== LCD & LORA SETUP =====
LiquidCrystal_I2C lcd(0x27, 20, 4);
#define LORA_RAY_NSS 10
#define LORA_RAY_RST 9
#define LORA_RAY_DIO0 2
#define BAND 915E6
#define LORA_TX_POWER 20
#define LORA_SPREADING_FACTOR 12

// ===== PIN DEFINITIONS =====
const int motorPin = 7;
const int solenoidPin = 6;
const int buttonPin = 4;
const int resetButtonPin = 3;

// ===== SENSOR & BPM CONFIG =====
MAX30105 sensor;
const auto kSamplingRate = sensor.SAMPLING_RATE_400SPS;
const float kSamplingFrequency = 400.0;
const unsigned long kFingerThreshold = 10000;
const unsigned int kFingerCooldownMs = 500;
const float kEdgeThreshold = -2000.0;
const float kLowPassCutoff = 5.0;
const float kHighPassCutoff = 0.5;
const bool kEnableAveraging = true;
const int kAveragingSamples = 50;
const int kSampleThreshold = 5;
```

```

// ===== FILTER INSTANCES =====
HighPassFilter high_pass_filter(kHighPassCutoff, kSamplingFrequency);
LowPassFilter low_pass_filter(kLowPassCutoff, kSamplingFrequency);
Differentiator differentiator(kSamplingFrequency);
MovingAverageFilter<kAveragingSamples> averager;

// ===== VARIABEL GLOBAL =====
int sistole = 0;
int diastole = 0;
int bpm = 0;
bool pumping = false;
int targetSistolik = 0;

long last_heartbeat = 0;
long finger_timestamp = 0;
bool finger_detected = false;
float last_diff = NAN;
bool crossed = false;
long crossed_time = 0;

void setup() {
  Serial.begin(115200);
  while (!Serial);
  Serial.println("LoRa Sender");

  // Setup LoRa
  LoRa.setPins(LORA_RAY_NSS, LORA_RAY_RST, LORA_RAY_DIO0);
  if (!LoRa.begin(BAND)) {
    Serial.println("Starting LoRa failed!");
    while (1);
  }
  LoRa.setTxPower(LORA_TX_POWER, PA_OUTPUT_PA_BOOST_PIN);
  LoRa.setSpreadingFactor(LORA_SPREADING_FACTOR);
  LoRa.setSyncWord(0xF3);
  Serial.println("LoRa Initializing OK!");

  // Setup MAX3010x
  if(sensor.begin() && sensor.setSamplingRate(kSamplingRate)) {
    Serial.println("Sensor initialized");
  } else {
    Serial.println("Sensor not found");
    while(1);
  }
}

```

```

}

// Setup Pins
pinMode(motorPin, OUTPUT);
pinMode(solenoidPin, OUTPUT);
pinMode(buttonPin, INPUT_PULLUP);
pinMode(resetButtonPin, INPUT_PULLUP);

// Setup LCD
lcd.init();
lcd.backlight();
lcd.noCursor();
lcd.setCursor(0, 0);
lcd.print("Sistolik  : ");
lcd.setCursor(0, 1);
lcd.print("Diastolik  : ");
lcd.setCursor(0, 2);
lcd.print("DetakJantung: ");
}

void loop() {
    int buttonState = digitalRead(buttonPin);
    int resetButtonState = digitalRead(resetButtonPin);

    if (buttonState == LOW && !pumping) {
        startMeasurement();
    }

    if (pumping) {
        measureBloodPressure();
    }

    if (resetButtonState == LOW) {
        resetValues();
    }

    auto sample = sensor.readSample(1000);
    float current_value = sample.red;

    if (sample.red > kFingerThreshold) {
        if (millis() - finger_timestamp > kFingerCooldownMs) {
            finger_detected = true;
        }
    }
}

```

```

    } else {
        resetFilters();
        finger_detected = false;
        finger_timestamp = millis();
    }

    if (finger_detected) {
        processHeartRate(current_value);
    }
}

void startMeasurement() {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Sistolik : ");
    lcd.setCursor(0, 1);
    lcd.print("Diastolik : ");
    lcd.setCursor(0, 2);
    lcd.print("DetakJantung: ");

    processHeartRate(sensor.readSample(1000).red);
    lcd.setCursor(13, 2);
    lcd.print(bpm);
    lcd.print(" bpm");

    delay(1000);
    digitalWrite(motorPin, HIGH);
    digitalWrite(solenoidPin, HIGH);
    pumping = true;

    targetSistolik = random(95, 126);
}

void measureBloodPressure() {
    int dataadc = analogRead(A0);
    float voltage = (dataadc * 5.0) / 1023.0;
    int mmhg = (voltage - 0.0) * ((200 - 0) / (5.0 - 0.0));

    if (sistole < targetSistolik) {
        lcd.setCursor(13, 0);
        lcd.print(sistole);
        lcd.print(" mmHg");
        delay(500);
    }
}

```

```

        sistole++;
    } else {
        diastole = random(60, 80);
        lcd.setCursor(13, 1);
        lcd.print(diastole);
        lcd.print(" mmHg");
        pumping = false;
        stopMeasurement();
    }
}

void stopMeasurement() {
    digitalWrite(motorPin, LOW);
    digitalWrite(solenoidPin, LOW);

    Serial.print("Sistolik: ");
    Serial.print(sistole);
    Serial.println(" mmHg");
    Serial.print("Diastolik: ");
    Serial.print(diastole);
    Serial.println(" mmHg");
    Serial.print("DetakJantung: ");
    Serial.print(bpm);
    Serial.println(" bpm");

    lcd.setCursor(13, 0);
    lcd.print(sistole);
    lcd.print(" mmHg");

    lcd.setCursor(13, 1);
    lcd.print(diastole);
    lcd.print(" mmHg");

    lcd.setCursor(13, 2);
    lcd.print(bpm);
    lcd.print(" bpm");

    // ==== Kirim data ke LoRa ====
    String jsonData = "{\"bpm\":\"" + String(bpm) +
        "\",\"finger\":\"1\" +
        "\",\"sistole\":\"" + String(sistole) +
        "\",\"diastole\":\"" + String(diastole) + "\"}";

```

```

String payload = nodeID + "|" + kode_khusus + "|" + jsonData;

LoRa.beginPacket();
LoRa.print(payload);
LoRa.endPacket();

Serial.println("📡 Mengirim: " + payload);

delay(100);

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Sistolik : ");
lcd.print(sistole);
lcd.print(" mmHg");
lcd.setCursor(0, 1);
lcd.print("Diastolik : ");
lcd.print(diastole);
lcd.print(" mmHg");
lcd.setCursor(0, 2);
lcd.print("DetakJantung: ");
lcd.print(bpm);
lcd.print(" bpm");

delay(1000);
}

void resetValues() {
    sistole = 0;
    diastole = 0;
    bpm = 0;
    targetSistolik = 0;
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Sistolik : ");
    lcd.setCursor(0, 1);
    lcd.print("Diastolik : ");
    lcd.setCursor(0, 2);
    lcd.print("DetakJantung: ");
}

void resetFilters() {
    differentiator.reset();
}

```

```

    averager.reset();
    low_pass_filter.reset();
    high_pass_filter.reset();
}

void processHeartRate(float current_value) {
    current_value = low_pass_filter.process(current_value);
    current_value = high_pass_filter.process(current_value);
    float current_diff = differentiator.process(current_value);

    if (!isnan(current_diff) && !isnan(last_diff)) {
        if (last_diff > 0 && current_diff < 0) {
            crossed = true;
            crossed_time = millis();
        }

        if (current_diff > 0) {
            crossed = false;
        }

        if (crossed && current_diff < kEdgeThreshold) {
            if (last_heartbeat != 0 && crossed_time - last_heartbeat > 300) {
                bpm = 60000 / (crossed_time - last_heartbeat);
                if (bpm > 50 && bpm < 250) {
                    lcd.setCursor(13, 2);
                    lcd.print(bpm);
                    lcd.print(" bpm");
                }
            }
            crossed = false;
            last_heartbeat = crossed_time;
        }
    }

    last_diff = current_diff;
}

```

2. Program Bridge

```

#include <SPI.h>
#include <LoRa.h>

```

```

// ----- Pin LoRa (Bridge Device) -----
#define LORA_NSS 10
#define LORA_RST 9
#define LORA_DIO0 2
#define BAND 915E6

#define LORA_TX_POWER 20
#define LORA_SPREADING_FACTOR 12
#define LORA_SYNC_WORD 0xF3 // Sama dengan transmitter & receiver

// ----- Kode Khusus -----
const String kode_khusus_diterima = "ABC123"; // Sesuaikan

void setup() {
  Serial.begin(115200);
  while (!Serial);

  Serial.println("🔄 Inisialisasi LoRa Bridge...");

  LoRa.setPins(LORA_NSS, LORA_RST, LORA_DIO0);
  if (!LoRa.begin(BAND)) {
    Serial.println("❌ Gagal inisialisasi LoRa");
    while (1);
  }

  LoRa.setTxPower(LORA_TX_POWER,
PA_OUTPUT_PA_BOOST_PIN);
  LoRa.setSpreadingFactor(LORA_SPREADING_FACTOR);
  LoRa.setSyncWord(LORA_SYNC_WORD);

  Serial.println("✅ LoRa Bridge Siap");
}

```

```

void loop() {
  int packetSize = LoRa.parsePacket();
  if (packetSize) {
    String incoming = "";
    while (LoRa.available()) {
      incoming += (char)LoRa.read();
    }

    Serial.println("📡 Data diterima: " + incoming);

    // Format: nodeID|kode_khusus|{json}
    int idx1 = incoming.indexOf('|');
    int idx2 = incoming.indexOf('|', idx1 + 1);
    if (idx1 == -1 || idx2 == -1) {
      Serial.println("⚠️ Format data salah, abaikan.");
      return;
    }

    String kodeNode = incoming.substring(idx1 + 1, idx2);
    if (kodeNode != kode_khusus_diterima) {
      Serial.println("⚠️ Kode khusus tidak cocok, abaikan.");
      return;
    }

    // Kirim ulang ke receiver
    LoRa.beginPacket();
    LoRa.print(incoming);
    LoRa.endPacket();

    Serial.println("📡 Data diteruskan ke receiver.");
  }
}

```

3. Program Receiver

```
#include <SPI.h>
#include <LoRa.h>
#include <WiFi.h>
#include <Firebase_ESP_Client.h>
#include <addons/TokenHelper.h>
#include <addons/RTDBHelper.h>

// ----- WiFi Credentials -----
#define WIFI_SSID "Polibatam"
#define WIFI_PASSWORD ""

// ----- Firebase Config -----
#define API_KEY "AIzaSyAT1YuUSoKvAC_q1yxmB8Ggt4vR6f51Nkc"
#define DATABASE_URL "https://telemediss-default-rtdb.asia-southeast1.firebaseio.com/"

// ----- Pin LoRa (Aurora V2 / SX1278) -----
#define LORA_NSS 15
#define LORA_RST 0
#define LORA_DIO0 27
#define LORA_EN 32
#define LORA_FREQ 915E6
#define LORA_TX_POWER 20
#define LORA_SPREADING_FACTOR 12

// ----- Variabel Receiver -----
FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;
```

```

String receiverID = "";
String ownerUserID = "";
const String kode_khusus_receiver = "ABC123"; // bisa diubah sesuai sistem

bool signupOK = false;
bool initDone = false;
unsigned long lastHeartbeat = 0;
const unsigned long HEARTBEAT_INTERVAL = 30000; // 30 detik

// ----- Koneksi WiFi -----
void wifiConnect() {
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Menghubungkan WiFi");

  int attempts = 0;
  while (WiFi.status() != WL_CONNECTED && attempts < 20) {
    delay(500);
    Serial.print(".");
    attempts++;
  }

  if (WiFi.status() == WL_CONNECTED) {
    Serial.println(" Terhubung!");
    Serial.print("IP Address: ");
    Serial.println(WiFi.localIP());
  } else {
    Serial.println(" Gagal terhubung!");
    // Restart ESP32 jika gagal koneksi
    ESP.restart();
  }
}

// ----- Inisialisasi Firebase -----

```

```

void initFirebase() {
    config.api_key = API_KEY;
    config.database_url = DATABASE_URL;

    // Gunakan anonymous authentication
    if (Firebase.signUp(&config, &auth, "", "")) {
        Serial.println("Firebase SignUp berhasil");
        signupOK = true;
    } else {
        Serial.print("Firebase SignUp Gagal: ");
        Serial.println(config.signer.signupError.message.c_str());
    }

    // Assign callback function untuk token generation
    config.token_status_callback = tokenStatusCallback;

    // Initialize Firebase
    Firebase.begin(&config, &auth);
    Firebase.reconnectWiFi(true);

    // Set timeout
    config.timeout.serverResponse = 10 * 1000;
}

// ----- Ambil ID Unik Receiver -----
String getReceiverID() {
    String id = "receiver_" + WiFi.macAddress();
    id.replace(":", "");
    id.toUpperCase();
    return id;
}

// ----- Cek Owner Receiver -----

```

```

bool cekReceiverOwner() {
    String path = "/receiver/" + receiverID + "/owner";
    Serial.println("Cek path: " + path);

    if (Firebase.RTDB.getString(&fbdo, path)) {
        if (fbdo.dataType() == "string") {
            ownerUserID = fbdo.stringData();
            Serial.println("Owner Receiver: " + ownerUserID);
            return true;
        }
    } else {
        Serial.print("Gagal ambil data owner: ");
        Serial.println(fbdo.errorReason().c_str());
    }
    Serial.println("Receiver belum didaftarkan di Firebase!");
    return false;
}

// ----- Cek Kode Khusus Receiver -----
bool cekKodeKhususReceiver() {
    String path = "/receiver/" + receiverID + "/kode_khusus";
    if (Firebase.RTDB.getString(&fbdo, path)) {
        if (fbdo.dataType() == "string") {
            String kode = fbdo.stringData();
            Serial.println("Kode khusus dari Firebase: " + kode);
            return kode == kode_khusus_receiver;
        }
    } else {
        Serial.print("Gagal cek kode khusus: ");
        Serial.println(fbdo.errorReason().c_str());
    }
    return false;
}

```

```

// ----- Cek nodeID di bawah akun owner -----
bool cekNodeDiUser(String nodeID) {
    String path = "/akun/" + ownerUserID + "/nodes/" + nodeID;
    if (Firebase.RTDB.getBool(&fbdo, path)) {
        return fbdo.boolData();
    } else {
        Serial.print("Gagal cek node: ");
        Serial.println(fbdo.errorReason().c_str());
    }
    return false;
}

// ----- Cek Koneksi WiFi -----
void checkWiFiConnection() {
    if (WiFi.status() != WL_CONNECTED) {
        Serial.println("WiFi terputus, mencoba reconnect...");
        wifiConnect();
    }
}

// ----- Heartbeat untuk monitoring -----
void sendHeartbeat() {
    if (millis() - lastHeartbeat > HEARTBEAT_INTERVAL) {
        String path = "/receiver/" + receiverID + "/last_seen";
        if (Firebase.RTDB.setTimestamp(&fbdo, path)) {
            Serial.println("Heartbeat sent");
        }
        lastHeartbeat = millis();
    }
}

void setup() {

```

```

// Inisialisasi Serial
Serial.begin(115200);
delay(1000);
Serial.println("Starting ESP32 LoRa Receiver...");

// Enable LoRa chip
pinMode(LORA_EN, OUTPUT);
digitalWrite(LORA_EN, HIGH);
delay(100);

// Inisialisasi LoRa
SPI.begin();
LoRa.setPins(LORA_NSS, LORA_RST, LORA_DIO0);

if (!LoRa.begin(LORA_FREQ)) {
  Serial.println("Gagal inisialisasi LoRa");
  while (1) {
    delay(1000);
  }
}

// Konfigurasi LoRa
LoRa.setTxPower(LORA_TX_POWER,
PA_OUTPUT_PA_BOOST_PIN);
LoRa.setSpreadingFactor(LORA_SPREADING_FACTOR);
LoRa.setSyncWord(0xF3);
LoRa.setCodingRate4(5);
LoRa.setSignalBandwidth(125E3);

Serial.println("LoRa Siap");

// Koneksi WiFi dan Firebase
wifiConnect();

```

```

initFirebase();

// Ambil ID receiver
receiverID = getReceiverID();
Serial.println("Receiver ID: " + receiverID);

// Tunggu Firebase siap
while (!Firebase.ready()) {
    delay(100);
}

Serial.println("System ready!");
}

void loop() {
    // Cek koneksi WiFi
    checkWiFiConnection();

    // Kirim heartbeat
    sendHeartbeat();

    // Cek apakah Firebase siap
    if (!Firebase.ready() || !signupOK) {
        delay(100);
        return;
    }

    // Validasi receiver hanya sekali
    if (!initDone) {
        Serial.println("Melakukan validasi receiver...");

        if (!cekReceiverOwner()) {
            Serial.println("STOP: Receiver belum diklaim!");
        }
    }
}

```

```

    delay(5000); // Tunggu 5 detik sebelum coba lagi
    return;
}

if (!cekKodeKhususReceiver()) {
    Serial.println("STOP: Kode khusus receiver tidak cocok!");
    delay(5000); // Tunggu 5 detik sebelum coba lagi
    return;
}

initDone = true;
Serial.println("Validasi receiver berhasil!");
}

// Terima data LoRa
int packetSize = LoRa.parsePacket();
if (packetSize) {
    String received = "";
    while (LoRa.available()) {
        received += (char)LoRa.read();
    }

    Serial.println("Data diterima: " + received);
    Serial.println("RSSI: " + String(LoRa.packetRssi()));
    Serial.println("SNR: " + String(LoRa.packetSnr()));

    // Format: nodeID|kode_khusus|{"bpm":88,"sistol":120,"diastol":80}
    int idx1 = received.indexOf("|");
    int idx2 = received.indexOf("|", idx1 + 1);

    if (idx1 == -1 || idx2 == -1) {
        Serial.println("Format data salah!");
        return;
    }
}

```

```

    }

    String nodeID = received.substring(0, idx1);
    String kodeNode = received.substring(idx1 + 1, idx2);
    String dataJSON = received.substring(idx2 + 1);

    Serial.println("Node ID: " + nodeID);
    Serial.println("Kode Node: " + kodeNode);
    Serial.println("Data JSON: " + dataJSON);

    // Validasi kode khusus
    if (kodeNode != kode_khusus_receiver) {
        Serial.println("Kode khusus dari node tidak cocok");
        return;
    }

    // Cek apakah node terdaftar
    if (!cekNodeDiUser(nodeID)) {
        Serial.println("Node tidak terdaftar di akun user");
        return;
    }

    // Kirim ke Firebase dengan timestamp
    FirebaseJson json;
    json.setJsonData(dataJSON);
    json.set("timestamp", Firebase.getCurrentTime());
    json.set("rssi", LoRa.packetRssi());
    json.set("snr", LoRa.packetSnr());

    String path = "/data/" + nodeID + "/" +
String(Firebase.getCurrentTime());

    if (Firebase.RTDB.setJSON(&fbdo, path.c_str(), &json)) {

```

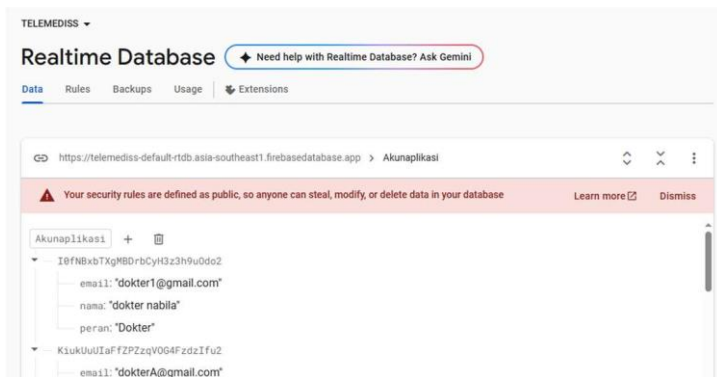
```

    Serial.println("✔ Data berhasil disimpan ke Firebase");
  } else {
    Serial.print("✗ Gagal simpan data: ");
    Serial.println(fbdo.errorReason().c_str());
  }
}
}

// Delay kecil untuk mencegah overload
delay(10);
}

```

4. Gambar Firebase



TELEMEDISS

Realtime Database

Need help with Realtime Database? Ask Gemini

DataRulesBackupsUsageExtensions

https://telemediss-default-rtdb.asia-southeast1.firebaseio.com/app > HASIL_PENGUKUR...

Your security rules are defined as public, so anyone can steal, modify, or delete data in your database

Learn moreDismiss

HASIL_PENGUKURAN

- AZRUL
 - HISTORI
 - OP-8mbb-Wj5H1UJgFNZ
 - BPM: "1"
 - DIASTOLE: "3"
 - SISTOLE: "2"

TELEMEDISS

Realtime Database

Need help with Realtime Database? Ask Gemini

DataRulesBackupsUsageExtensions

https://telemediss-default-rtdb.asia-southeast1.firebaseio.com/app > akun

Your security rules are defined as public, so anyone can steal, modify, or delete data in your database

Learn moreDismiss

- ALIBABA
- AZRUL
 - email: "azrulaswat63@gmail.com"
 - nama: "AZRUL"
 - role: "user"
- AZRUL ASWAT
 - email: "azrulaswat77@gmail.com"

TELEMEDISS

Realtime Database

Need help with Realtime Database? Ask Gemini

DataRulesBackupsUsageExtensions

https://telemediss-default-rtdb.asia-southeast1.firebaseio.com/app > chats

Your security rules are defined as public, so anyone can steal, modify, or delete data in your database

Learn moreDismiss

chats

- AZRUL | ALIBABA
 - pesan
 - OPkOULx9CpzCkBDpWJU
 - content: "P"
 - from: "AZRUL"
 - read: false

64

DataTELEMEDISS

Rules

Backups

Usage

Extensions

https://telemidss-default-rtdb.asia-southeast1.firebaseio.com > data

⚠️ Your security rules are defined as public, so anyone can steal, modify, or delete data in your database

[Learn more](#) [Dismiss](#)

data

node123

bpm: 101

diastolik: 78

finger: 1

sistolik: 118

DataTELEMEDISS

Rules

Backups

Usage

Extensions

https://telemidss-default-rtdb.asia-southeast1.firebaseio.com > receiver

⚠️ Your security rules are defined as public, so anyone can steal, modify, or delete data in your database

[Learn more](#) [Dismiss](#)

receiver

RECEIVER_FCE80C7A148

kode_khusus: "ABC123"

last_seen: 1752164426907

owner: "HkTIL9zQbAPiNjfPiPehnzXAB32"

status: "0"

65