

TUGAS AKHIR

**IMPLEMENTASI SISTEM *MULTIUSER* UNTUK
TENSIMETER OMRON DENGAN KONEKTIVITAS
BLUETOOTH DAN *DASHBOARD MOBILE***

Oleh:

GILANG RAMADHANI ALIFianto

1202220013



PROGRAM STUDI STRATA 1 SISTEM INFORMASI

FAKULTAS REKAYASA INDUSTRI

UNIVERSITAS TELKOM

2026

ABSTRAK

Hipertensi merupakan salah satu penyakit tidak menular yang menyebabkan sekitar 10,8 juta kematian per tahun secara global, sehingga diperlukan pendekatan pemantauan tekanan darah yang lebih modern melalui sistem *Remote Patient Monitoring* (RPM) berbasis *Internet of Things* (IoT). Namun, perangkat tensimeter berbasis *Bluetooth Low Energy* (BLE) seperti Omron HEM-7155T umumnya tidak menyediakan mekanisme autentikasi bawaan untuk memisahkan data antar pengguna, sehingga berpotensi menimbulkan pencampuran data pengukuran dalam skenario multiuser. Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem multiuser yang mampu mengelola koneksi BLE pada tensimeter Omron, mengisolasi data pengukuran tekanan darah antar pengguna melalui integrasi *API backend*, serta mengevaluasi efektivitas sistem dari segi fungsionalitas, penerimaan pengguna, dan performa di bawah beban konkuren. Metode pengembangan yang digunakan mengikuti pendekatan *Software Development Life Cycle* (SDLC) dengan arsitektur tiga lapis yang terdiri dari aplikasi mobile berbasis React Native (Expo), backend berbasis FastAPI dengan autentikasi JWT dan basis data SQLite, serta komunikasi BLE ke perangkat tensimeter Omron. Evaluasi dilakukan melalui tiga metode, yaitu *Blackbox Testing*, *User Acceptance Testing* (UAT), dan Load Testing menggunakan Grafana k6. Hasil pengujian menunjukkan bahwa seluruh 11 skenario *Blackbox Testing* dinyatakan lulus dengan tingkat kelulusan 100%, UAT memperoleh indeks kepuasan pengguna sebesar 93,2% yang termasuk kategori sangat baik dari 5 responden, dan Load Testing dengan 100 *Virtual Users* selama 5 menit mencatat 0% *error rate* dengan rata-rata *response time* 113,59 ms. Penelitian ini memberikan kontribusi berupa kerangka arsitektur multiuser untuk perangkat medis BLE tanpa modifikasi firmware, yang dapat menjadi referensi bagi pengembangan sistem RPM di lingkungan domestik maupun klinik kecil.

Kata kunci—*multiuser, tensimeter Omron, Bluetooth Low Energy, Remote Patient Monitoring, dashboard mobile*

ABSTRACT

Hypertension is one of the leading non-communicable diseases responsible for approximately 10.8 million deaths annually worldwide, necessitating a more modern approach to blood pressure monitoring through Internet of Things (IoT)-based Remote Patient Monitoring (RPM) systems. However, Bluetooth Low Energy (BLE)-based blood pressure monitors such as the Omron HEM-7155T generally do not provide built-in authentication mechanisms to separate data between users, potentially causing data mixing in multiuser scenarios. This research aims to design and implement a multiuser system capable of managing BLE connections to the Omron blood pressure monitor, isolating blood pressure measurement data between users through backend API integration, and evaluating the system's effectiveness in terms of functionality, user acceptance, and performance under concurrent load. The development method follows the Software Development Life Cycle (SDLC) approach with a three-layer architecture consisting of a React Native (Expo)-based mobile application, a FastAPI-based backend with JWT authentication and SQLite database, and BLE communication to the Omron blood pressure monitor device. Evaluation was conducted through three methods: Blackbox Testing, User Acceptance Testing (UAT), and Load Testing using Grafana k6. The test results demonstrate that all 11 Blackbox Testing scenarios passed with a 100% pass rate, UAT achieved a user satisfaction index of 93.2% categorized as very good from 5 respondents, and Load Testing with 100 Virtual Users over 5 minutes recorded a 0% error rate with an average response time of 113.59 ms. This research contributes a multiuser architectural framework for BLE medical devices without firmware modification, serving as a reference for developing RPM systems in domestic environments and small clinics.

Keywords — multiuser, Omron blood pressure monitor, Bluetooth Low Energy, Remote Patient Monitoring, mobile dashboard

LEMBAR PENGESAHAN

Tugas Akhir dengan judul :

**IMPLEMENTASI SISTEM *MULTIUSER* UNTUK
TENSIMETER OMRON DENGAN KONEKTIVITAS
BLUETOOTH DAN *DASHBOARD MOBILE***

Telah disetujui dan disahkan pada Sidang Tugas Akhir

Program Studi Strata 1 Sistem Informasi

Fakultas Rekayasa Industri Universitas Telkom

Oleh :

Gilang Ramadhani Alifianto

1202220013

Bandung, 10 Agustus 2026

Disetujui oleh,

Pembimbing 1,



Dr. Tien Fabrianti Kusumasari, S.T., M.T.

14790008

Pembimbing 2,



Dr. Sinung Suakanto, S.T., M.T

22820012

LEMBAR PERNYATAAN ORISINALITAS



Nama : Gilang Ramadhani Alifianto

NIM : 1202220013

Alamat : Jl. Raya Doro - Pekalongan, Kasotengah,
Doro, Kec. Doro, Kabupaten Pekalongan,
Jawa Tengah 51191

No. Tlp : 083844457333

Email : langilang@student.telkomuniversity.ac.id

Menyatakan bahwa Tugas Akhir ini merupakan karya orisinal saya sendiri. Atas pernyataan ini, saya siap menanggung risiko atau sanksi yang dijatuhkan kepada saya apabila kemudian ditemukan adanya pelanggaran terhadap kejujuran akademik atau etika keilmuan dalam karya ini, atau ditemukan bukti yang menunjukkan ketidakaslian karya ini.

Bandung, 27 Juli 2026

Gilang Ramadhani Alifianto

KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadiran Allah SWT atas rahmat dan karunia-Nya sehingga Tugas Akhir berjudul "Implementasi Sistem *Multiuser* untuk Tensimeter Omron dengan Konektivitas *Bluetooth* dan *Dashboard Mobile*" ini dapat diselesaikan sebagai salah satu syarat kelulusan Program Studi S1 Sistem Informasi, Fakultas Rekayasa Industri, Universitas Telkom.

Dalam proses penyusunannya, penulis mendapatkan banyak bimbingan, dorongan, dan bantuan dari berbagai pihak. Oleh karena itu, penulis ingin mengucapkan terima kasih kepada:

1. Kedua Orang Tua (Ayah dan Ibu) atas doa, kasih sayang, dan dukungan luar biasa yang tak pernah putus.
2. Ibu Dr. Tien Fabrianti Kusumasari, S.T., M.T. dan Bapak Dr. Sinung Suakanto, S.T., M.T. atas arahan, waktu, dan bimbingan yang sangat berharga.
3. Bapak/Ibu Dosen S1 Sistem Informasi Universitas Telkom atas ilmu dan wawasan yang telah diberikan.
4. Seluruh pihak dan rekan-rekan seperjuangan yang telah memberikan dukungan secara langsung maupun tidak langsung.

Penulis menyadari Tugas Akhir ini masih jauh dari sempurna, namun diharapkan dapat memberikan manfaat bagi pembacanya. Semoga segala kebaikan yang telah diberikan dibalas oleh Tuhan Yang Maha Esa.

Penulis,

Gilang Ramadhani Alifianto

DAFTAR ISI

ABSTRAK	ii
LEMBAR PERNYATAAN ORISINALITAS	v
KATA PENGANTAR	vi
DAFTAR ISI	vii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xii
DAFTAR ISTILAH	xiii
DAFTAR SINGKATAN	xvi
DAFTAR SIMBOL	xvii
DAFTAR LAMPIRAN	xviii
Bab I PENDAHULUAN	1
I.1 Latar Belakang	1
I.2 Rumusan Masalah	3
I.3 Tujuan Penelitian	4
I.4 Batasan Penelitian	4
I.5 Manfaat Penelitian	5
I.6 Sistematika Penulisan	6
Bab II TINJAUAN PUSTAKA	8
II.1 <i>Remote Patient Monitoring</i> (RPM)	8
II.2 Bluetooth Low Energy (BLE)	8
II.3 Pemanfaatan Internet of Things (IoT) untuk Kesehatan	10
II.4 Pemantauan Tanda Vital	10
II.5 Tensimeter Omron	12
II.6 Manajemen Multiuser pada Bluetooth	12

II.7	Pengembangan Aplikasi Mobile dengan Framework Expo	14
II.8	Pengembangan Application Programming Interface (API) Berbasis Framework FastAPI	15
II.9	Pengembangan Sistem RPM Berbasis Mobile	16
II.9.1	Lightweight Methodologies	17
II.9.2	Heavyweight Methodologies	18
II.10	Pengujian Perangkat Lunak	23
II.10.1	<i>User Acceptance Test (UAT)</i>	23
II.10.2	<i>Blackbox Testing</i>	24
II.10.3	<i>Load Testing</i>	25
II.11	<i>State of the art</i>	26
Bab III	METODE PENYELESAIAN MASALAH	36
III.1	Metode Konseptual	36
III.2	Sistematika Penyelesaian Masalah	37
III.2.1	Identifikasi	38
III.2.2	Tahap Pengembangan	38
III.2.3	Tahap Kesimpulan dan Saran	39
III.3	Alasan Pemilihan Metode	39
III.4	Metode Pengumpulan data	40
III.5	Metode Evaluasi	41
Bab IV	Hasil PENELITIAN	42
IV.1	<i>Planning and Requirements</i>	42
IV.1.1	Analisis Sistem Eksisting	42
IV.1.2	Gap Analisis	43
IV.1.3	Pengumpulan Data dan Wawancara	45
IV.1.4	Analisis Kebutuhan Sistem	47

IV.1.5	Arsitektur Model Konseptual	48
IV.2	<i>Analysis and Design</i>	52
IV.2.1	<i>Entity Relationship Diagram (ERD)</i>	54
IV.2.2	<i>Use Case Scenario</i>	55
IV.2.3	<i>Use Case Diagram</i>	60
IV.2.4	<i>Sequence Diagram</i>	60
IV.2.5	<i>Activity Diagram</i>	67
IV.2.6	<i>Wireframe</i>	74
IV.2.7	<i>Mockup High-fidelity</i>	75
IV.3	Tahap Implementasi	82
IV.3.1	Login	82
IV.3.2	Register	82
IV.3.3	Lupa Password	83
IV.3.4	Manajemen Profile	83
IV.3.5	Measurement Flow	84
IV.4	Pengujian	85
IV.4.1	<i>Blackbox Testing</i>	86
IV.4.2	<i>User Acceptance Test (UAT)</i>	90
IV.4.3	<i>Load Testing</i>	94
Bab V	VALIDASI HASIL DAN DISKUSI	98
V.1	Evaluasi Penelitian	98
V.2	Diskusi	102
V.3	Dampak Hasil Penelitian	103
Bab VI	KESIMPULAN DAN SARAN	106
VI.1	Kesimpulan	106
VI.2	Saran	107

DAFTAR PUSTAKA	108
LAMPIRAN.....	117

DAFTAR GAMBAR

Gambar I-1 <i>Deaths attributable to high systolic blood pressure (1990 and 2019)</i>	1
Gambar II-1 <i>Software Development Life Cycle (SDLC)</i>	17
Gambar II-2 <i>Iterative Incremental</i>	20
Gambar III-1 <i>Model Konseptual</i>	36
Gambar III-2 <i>Sistematika Penyelesaian Masalah</i>	38
Gambar IV-1 <i>Component Diagram</i>	48
Gambar IV-2 <i>Entity Relationship Diagram</i>	54
Gambar IV-3 <i>Use Case Diagram</i>	60
Gambar IV-4 <i>Sequence Diagram Register</i>	61
Gambar IV-5 <i>Sequence Diagram Login</i>	62
Gambar IV-6 <i>Sequence Diagram Manajemen Profile</i>	63
Gambar IV-7 <i>Sequence Diagram Measurement Flow</i>	64
Gambar IV-8 <i>Sequence Diagram Alur Komunikasi BLE</i>	66
Gambar IV-9 <i>Activity Diagram Register</i>	68
Gambar IV-10 <i>Activity Diagram Login</i>	69
Gambar IV-11 <i>Activity Manajemen Profile</i>	70
Gambar IV-12 <i>Activity Measurement Flow</i>	72
Gambar IV-13 <i>Wireframe</i>	75
Gambar IV-14 <i>High-Fidelity Login</i>	76
Gambar IV-15 <i>High-Fidelity Register</i>	77
Gambar IV-16 <i>High-Fidelity Lupa Password</i>	78
Gambar IV-17 <i>High-Fidelity Manajemen Profile</i>	79
Gambar IV-18 <i>High-Fidelity Measurement Flow</i>	80
Gambar IV-19 <i>Implementasi Login</i>	82
Gambar IV-20 <i>Implementasi Register</i>	83
Gambar IV-21 <i>Implementasi Lupa Password</i>	83
Gambar IV-22 <i>Implementasi Manajemen Profile</i>	84
Gambar IV-23 <i>Implementasi Prediksi Gula Darah</i>	84
Gambar IV-24 <i>Load Testing K6</i>	95
Gambar IV-25 <i>Treshold Load Testing</i>	96
Gambar IV-26 <i>Result Load Testing</i>	97

DAFTAR TABEL

Tabel II-1 Tahap Pengembangan <i>Iterative Incremental</i>	21
Tabel II-2 <i>State of The Art</i>	26
Tabel III-1 <i>Perbandingan Pendekatan Metodologi Penelitian dan Pengembangan Sistem RPM</i>	39
Tabel IV-1 Gap Analisis	44
Tabel IV-2 Kebutuhan Fungsional	47
Tabel IV-3 Endpoint dan Fungsi	53
Tabel IV-4 <i>Use Case Scenario Login</i>	55
Tabel IV-5 <i>Use Case Scenario Register</i>	56
Tabel IV-6 Use Case Scenario Manajemen Profile	57
Tabel IV-7 <i>Use Case Scenario Measurement Flow</i>	58
Tabel IV-8 Temuan Pengujian	85
Tabel IV-9 <i>Blackbox Testing Fase 1</i>	86
Tabel IV-10 <i>Blackbox Testing Fase 2</i>	88
Tabel IV-11 <i>User Acceptance Test Scenario Fase 1</i>	90
Tabel IV-12 <i>User Acceptance Test Scenario Fase 2</i>	92
Tabel IV-13 Hasil Penilaian User Acceptance Test (UAT)	93
Tabel V-1 Hasil Evaluasi	101

DAFTAR ISTILAH

Istilah	Deskripsi	Halaman Pertama Kali Digunakan
Tensimeter	Alat medis (<i>sphygmomanometer</i>) yang digunakan untuk mengukur tekanan darah dalam arteri (pembuluh darah) manusia	1
	Tekanan darah saat jantung berkontraksi memompa darah ke seluruh tubuh, dinyatakan dalam satuan mmHg	11
Sistolik (SYS)	Tekanan darah saat jantung berelaksasi di antara detak, dinyatakan dalam satuan mmHg	11
Diastolik (DIA)	Jumlah detak jantung per menit (<i>Beats Per Minute</i>)	12
Denyut Nadi (BPM)	Persentase hemoglobin dalam darah yang terikat dengan oksigen, normalnya 95–100%	10
Saturasi Oksigen (SpO ₂)	Perangkat perantara yang menjembatani komunikasi antara perangkat sensor/IoT dengan <i>server</i> atau <i>cloud</i>	10
Gateway	Perangkat lunak perantara yang menghubungkan antarmuka pengguna (<i>frontend</i>) dengan basis data dan logika bisnis (<i>backend</i>)	15
Middleware	Titik akses URL pada API yang menerima permintaan (<i>request</i>) dan mengembalikan respons (<i>response</i>)	22
Endpoint		

	Proses mengubah data (seperti <i>password</i>) menjadi string acak dengan panjang tetap menggunakan algoritma kriptografi satu arah	45
<i>Hashing</i>		
	Titik kemacetan atau hambatan dalam sistem yang menyebabkan penurunan performa secara keseluruhan	15
<i>Bottleneck</i>		
	Jumlah permintaan (<i>request</i>) yang dapat diproses oleh server dalam satuan waktu tertentu (biasanya per detik)	12
<i>Throughput</i>		
	Waktu yang dibutuhkan server untuk memproses permintaan dan mengembalikan respons kepada klien	22
<i>Response Time</i>		
	Tahap peningkatan beban secara bertahap dalam pengujian performa, dari jumlah pengguna rendah ke tinggi	22
<i>Ramp-up</i>		
	Tahap penurunan beban secara bertahap dalam pengujian performa, dari jumlah pengguna tinggi ke rendah	22
<i>Ramp-down</i>		
	Perangkat elektronik yang dapat dikenakan di tubuh, seperti <i>smartwatch</i> atau <i>fitness tracker</i> , untuk memantau data kesehatan	9
<i>Wearable</i>		
	Model pemrosesan di mana operasi dapat berjalan secara bersamaan tanpa harus menunggu	14
<i>Asinkron (Async)</i>		

	operasi sebelumnya selesai (<i>non-blocking</i>)	
<i>Scope Creep</i>	Perluasan ruang lingkup proyek yang tidak terkendali di luar rencana awal	18
<i>Firmware</i>	Perangkat lunak tetap yang tertanam di dalam perangkat keras untuk mengendalikan fungsi dasarnya	96
<i>Indexing</i>	Teknik pengoptimalan basis data dengan membuat indeks pada kolom tertentu untuk mempercepat pencarian data	80
<i>Caching</i>	Teknik penyimpanan data sementara di memori (RAM) agar dapat diakses lebih cepat tanpa perlu mengambil ulang dari basis data	80

DAFTAR SINGKATAN

Singkatan	Deskripsi	Halaman Pertama Kali Digunakan
API	<i>Application Programming Interface</i>	15
UUID	<i>Universally Unique Identifier</i>	45
ECG/EKG	<i>Electrocardiogram</i> / Elektrokardiogram	9
SDLC	<i>Software Development Life Cycle</i>	16
WSGI	<i>Web Server Gateway Interface</i>	15
ASGI	<i>Asynchronous Server Gateway Interface</i>	15
SYS	<i>Systolic</i> (Tekanan Darah Sistolik)	11
DIA	<i>Diastolic</i> (Tekanan Darah Diastolik)	11
BPM	<i>Beats Per Minute</i> (Denyut Nadi per Menit)	12
IHB	<i>Irregular Heartbeat</i> (Detak Jantung Tidak Teratur)	44
MOV	<i>Movement</i> (Indikator Pergerakan Tubuh)	44
SpO ₂	Saturasi Oksigen Perifer	10
VU	<i>Virtual User</i> (Pengguna Virtual)	22

DAFTAR SIMBOL

Simbol	Deskripsi	Halaman Pertama Kali Digunakan
✓	Tanda centang menandakan threshold terpenuhi	37
$\geq$	Lebih besar atau sama dengan	11
%	Persentase digunakan untuk SpO ₂ , <i>error rate</i> , indeks UAT	11
°C	Derajat <i>Celsius</i> satuan suhu tubuh	11

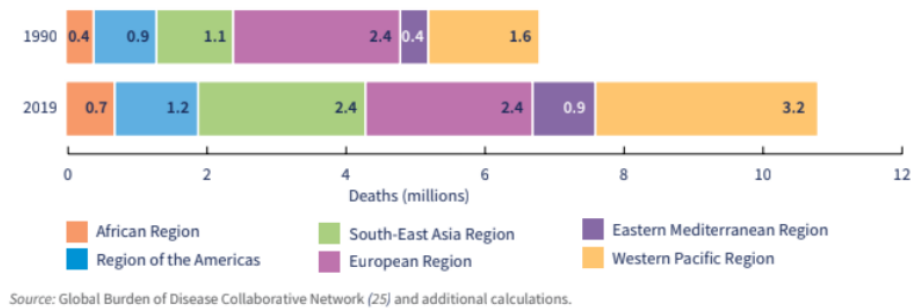
DAFTAR LAMPIRAN

Lampiran A-1 Dokter Spesialis Jantung	117
Lampiran A-2 Kepala IGD.....	117
Lampiran B-1 UAT User 1	118
Lampiran B-2 UAT User 2	118
Lampiran B-3 UAT User 3	119
Lampiran B-4 UAT User 4	119

BAB I PENDAHULUAN

I.1 Latar Belakang

Penyakit tidak menular (PTM) seperti hipertensi dan gagal jantung telah memengaruhi 1,4 miliar orang dewasa berusia 30–79 tahun, dengan hanya satu dari lima orang yang berhasil mengendalikan kondisinya (Ho dkk., 2024). Menurut laporan terbaru dari *World Health Organization* hipertensi menyebabkan sekitar 10,8 juta kematian yang dapat dicegah setiap tahunnya, berkontribusi pada 235 juta tahun kehidupan yang hilang atau hidup dengan kecacatan.



Gambar I-1 *Deaths attributable to high systolic blood pressure (1990 and 2019)*

Sumber: *Global report on hypertension* WHO

Pada Gambar I.1 menunjukkan bahwa terdapat peningkatan signifikan jumlah kematian akibat tekanan darah tinggi secara global pada tahun 1990 hingga 2019. Pada tahun 1990, total kematian tercatat sekitar 8,9 juta jiwa, sementara pada tahun 2019 meningkat menjadi 10,8 juta jiwa. Hipertensi terus menjadi penyebab utama kematian yang dapat dicegah secara global, dengan lonjakan signifikan terutama di Asia Tenggara, Pasifik Barat, dan Afrika. Peningkatan ini mencerminkan dampak perubahan gaya hidup dan modernisasi, serta menunjukkan bahwa penyakit tidak menular bukan lagi masalah negara maju saja. Meskipun beberapa wilayah seperti Eropa dan Mediterania Timur menunjukkan tren stabil atau menurun, data global menegaskan bahwa hipertensi tetap menjadi ancaman serius bagi kesehatan masyarakat dan membutuhkan penanganan lintas wilayah yang lebih sistematis.

Melihat peningkatan kematian akibat hipertensi secara global, pendekatan yang lebih modern dalam pengelolaan tekanan darah perlu dilengkapi dengan

inovasi digital. Salah satu solusi yang semakin relevan adalah *telemedicine*, yaitu sistem layanan kesehatan jarak jauh yang memanfaatkan teknologi digital untuk memantau dan mengelola kondisi pasien. Khususnya, bentuk *telemedicine* seperti medical *telemonitoring* memungkinkan pemantauan tekanan darah secara *real-time* dari jarak jauh. Fujiwara dkk. (2023) menegaskan bahwa tekanan darah merupakan parameter yang sangat cocok untuk dimonitor melalui sistem ini, karena memungkinkan pertukaran data yang akurat dan otomatis antara pasien dan tenaga kesehatan bahkan bagi lansia yang rentan atau tinggal di daerah terpencil. Selain itu, studi oleh Mackwood dkk. (2024) menunjukkan bahwa perluasan layanan *telehealth* sebagai bagian dari *telemedicine* pasca-pandemi berhasil meningkatkan akses bagi pasien di wilayah rural maupun urban dengan pendapatan rendah, tanpa mengganggu efektivitas pengelolaan hipertensi.

Efektivitas pendekatan ini semakin diperkuat oleh temuan Smith dkk. (2024), yang menunjukkan bahwa *Remote Patient Monitoring* (RPM), sebagai salah satu aplikasi *telemedicine* untuk pemantauan pasien jarak jauh, berkontribusi nyata terhadap peningkatan kontrol tekanan darah dan menjadi bukti kuat bahwa integrasi teknologi digital dapat memperbaiki manajemen penyakit kronis seperti hipertensi. Aniko dkk. (2024) *Remote Patient Monitoring* (RPM) *Systems* adalah pendekatan yang semakin populer dalam industri perawatan kesehatan yang memungkinkan pemantauan dan pengelolaan kondisi kesehatan pasien dari jarak jauh. RPM menyediakan pengamatan pasien yang lebih kuat dan fleksibel melalui sensor yang dapat dikenakan kapan saja dan di mana saja. Sistem pemantauan ini bertujuan untuk melacak individu secara tepat dan menghubungkan berbagai layanan serta objek di seluruh dunia melalui Internet untuk mengumpulkan, berbagi, mengintegrasikan, menyimpan, dan menganalisis data yang dihasilkan. RPM memanfaatkan berbagai teknologi termasuk *Internet of Things* (IoT), yang terdiri dari beberapa sensor, dan perangkat lunak lainnya. Komponen-komponen ini saling terhubung melalui internet untuk mengumpulkan dan berbagi data di antara mereka.

Meskipun Remote Patient Monitoring (RPM) berbasis IoT, seperti yang dijelaskan oleh Aniko dkk. (2024). *Remote Patient Monitoring* (RPM) telah menunjukkan potensi besar dalam meningkatkan pengelolaan hipertensi melalui

integrasi sensor dan aplikasi *mobile*, sistem ini umumnya dirancang untuk penggunaan individu atau lingkungan klinis dengan pengawasan terpusat. Dalam konteks penggunaan domestik, seperti di rumah tangga dengan beberapa anggota keluarga atau fasilitas kesehatan kecil, dukungan untuk *multiuser* sering kali terbatas. Tensimeter Omron berbasis *Bluetooth Low Energy* (BLE), misalnya, memungkinkan *multiple connections*, tetapi tidak menyediakan mekanisme autentikasi bawaan untuk memisahkan data antar pengguna. Hal ini menyebabkan potensi campur aduk data pengukuran tekanan darah antar pengguna, yang dapat mengurangi akurasi monitoring dan melanggar privasi data kesehatan. Penelitian oleh Smith dkk. (2024) menegaskan bahwa privasi dan isolasi data merupakan tantangan kritis dalam sistem IoT kesehatan, terutama untuk perangkat yang digunakan bersama oleh beberapa pengguna. Oleh karena itu, pengembangan sistem *multiuser* dengan autentikasi berbasis *backend* dan visualisasi data melalui *dashboard mobile* menjadi solusi yang relevan untuk meningkatkan skalabilitas dan keamanan pemantauan tekanan darah, sekaligus mendukung aksesibilitas di lingkungan dengan banyak pengguna, seperti keluarga atau komunitas kecil di Indonesia.

I.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, berikut adalah rumusan masalah yang dapat dirumuskan:

1. Bagaimana merancang dan mengimplementasikan sistem autentikasi *multiuser* berbasis *backend* untuk mengelola koneksi *Bluetooth Low Energy* (BLE) pada tensimeter Omron agar dapat digunakan oleh beberapa pengguna secara aman dan efisien?
2. Bagaimana cara mengisolasi data dari tensimeter Omron antar pengguna melalui integrasi API *backend* untuk mencegah pencampuran data dalam skenario *multiuser*?
3. Bagaimana mengevaluasi efektivitas tampilan sistem *multiuser* yang dikembangkan dalam hal kemudahan penggunaan untuk mendukung pemantauan kesehatan pasien?
4. Bagaimana merancang arsitektur terpisah yang mampu mengolah dan mendistribusikan data secara real-time?

I.3 Tujuan Penelitian

Berdasarkan rumusan masalah tersebut, maka tujuan dari penelitian ini adalah:

1. Merancang dan mengimplementasikan sistem autentikasi *multiuser* berbasis *backend* yang mampu mengelola koneksi *Bluetooth Low Energy* (BLE) pada tensimeter Omron secara aman dan efisien, sehingga memungkinkan beberapa pengguna untuk menggunakan perangkat tanpa risiko pencampuran data.
2. Mengembangkan mekanisme isolasi data melalui integrasi API *backend* guna mencegah pencampuran data pengukuran tekanan darah antar pengguna dalam skenario *multiuser*.
3. Mengevaluasi efektivitas sistem *multiuser* yang dikembangkan dari segi keamanan data, skalabilitas, dan kemudahan penggunaan untuk mendukung pemantauan hipertensi di lingkungan domestik atau klinik kecil.
4. Merancang mekanisme manajemen sesi berbasis *backend* untuk mengatasi konflik koneksi simultan, memastikan hanya satu pengguna aktif yang dapat mengakses perangkat pada satu waktu, serta mengembangkan algoritma untuk memprioritaskan atau mengantrekan akses pengguna guna meningkatkan efisiensi penggunaan tensimeter Omron dalam lingkungan dengan banyak pengguna.

I.4 Batasan Penelitian

Untuk menjaga agar pembahasan tetap terfokus dan terorganisir dengan baik, skripsi ini dilengkapi dengan batasan masalah yang bertujuan untuk menghindari perluasan topik yang berlebihan atau penyimpangan dari sasaran pokok. Adapun batasan-batasan dalam skripsi ini mencakup hal-hal berikut:

1. Penelitian dibatasi pada tensimeter Omron dengan konektivitas yang sudah dimodifikasi sebelumnya untuk mendukung *transfer* data tensimeter Omron.
2. *Dashboard mobile* dikembangkan hanya untuk *prototype* menggunakan *framework Expo* (React Native) tanpa dilakukan *build* aplikasi.

3. Penelitian tidak mencakup analisis kepatuhan terhadap regulasi medis seperti HIPAA atau GDPR, melainkan fokus pada aspek teknis autentikasi, isolasi data, dan manajemen sesi.
4. Pengujian efektivitas sistem terbatas pada aspek keamanan data (enkripsi AES dan isolasi data per pengguna), dan kemudahan penggunaan (*user experience* pada *dashboard*), dilakukan dalam lingkungan simulasi.

I.5 Manfaat Penelitian

Adapun manfaat yang diharapkan dari tugas akhir berikut, yaitu:

1. Bagi Universitas: Penelitian ini memberikan kontribusi bagi Universitas Telkom dengan memperkaya penelitian di bidang teknologi khususnya dalam pengembangan sistem *Internet of Things* (IoT) untuk kesehatan. Penelitian ini mendukung reputasi universitas sebagai pusat inovasi teknologi berbasis kebutuhan masyarakat, terutama melalui pengembangan solusi autentikasi multiuser untuk perangkat medis seperti tensimeter Omron.
2. Bagi Peneliti Lain: Penelitian ini bermanfaat bagi peneliti lain di bidang IoT, sistem kesehatan berbasis teknologi, dan pengembangan aplikasi *mobile*, dengan menyediakan kerangka kerja autentikasi *multiuser* berbasis *backend* untuk perangkat BLE seperti tensimeter Omron. Pendekatan isolasi data dan manajemen sesi yang dikembangkan dapat menjadi referensi untuk mengatasi tantangan privasi dan skalabilitas dalam sistem *Remote Patient Monitoring* (RPM). Penelitian ini juga menawarkan contoh implementasi *API backend* dan *dashboard mobile* menggunakan framework Expo, yang dapat menjadi acuan untuk pengembangan aplikasi serupa dalam konteks *multiuser*.
3. Bagi Masyarakat: Hasil penelitian ini bermanfaat bagi masyarakat, khususnya keluarga dengan anggota yang memiliki risiko hipertensi (seperti lansia atau individu dengan riwayat penyakit kardiovaskular) dan klinik kecil di Indonesia, dengan menyediakan sistem pemantauan tekanan darah yang mendukung penggunaan bersama oleh beberapa pengguna tanpa risiko pencampuran data.

4. Bagi Peneliti Selanjutnya: secara akademis penelitian ini memberikan landasan untuk penelitian lanjutan di bidang IoT kesehatan, khususnya dalam pengembangan sistem *multiuser* untuk perangkat medis berbasis BLE. Peneliti selanjutnya dapat menggunakan kerangka autentikasi dan isolasi data berbasis *backend* sebagai dasar untuk mengeksplorasi algoritma manajemen sesi yang lebih kompleks, seperti prioritas dinamis atau integrasi *machine learning* untuk analisis data kesehatan.

I.6 Sistematika Penulisan

Penelitian ini diuraikan dengan sistematika penulisan sebagai berikut:

BAB I PENDAHULUAN

Bab ini menyajikan latar belakang masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan dan asumsi penelitian, serta ruang lingkup penelitian. Selain itu, sistematika penulisan tugas akhir juga diuraikan untuk memberikan gambaran keseluruhan alur pembahasan

BAB II KAJIAN PUSTAKA

Bab ini membahas landasan teori, konsep-konsep dasar, dan kerangka pemikiran yang relevan dengan topik penelitian. Di dalamnya juga *direview* studi-studi sebelumnya, penelitian terkait, serta temuan-temuan empiris yang mendukung atau menjadi pembanding bagi penelitian ini, guna memperkuat fondasi analisis.

BAB III METODE PENYELESAIAN MASALAH

Bab ini menguraikan pendekatan penelitian yang dipilih, desain metode, teknik pengumpulan data, alat analisis, serta langkah-langkah prosedural yang diterapkan untuk mencapai tujuan penelitian secara sistematis dan dapat direplikasi.

BAB IV HASIL PENELITIAN

Bab ini menyajikan analisis mendalam terhadap data kebutuhan, persyaratan sistem, serta proses penyelesaian masalah. Termasuk di dalamnya adalah pemodelan algoritma, integrasi teknologi, dan evaluasi awal terhadap desain yang diusulkan untuk implementasi *Remote Patient Monitoring* (RPM).

BAB V VALIDASI HASIL DAN DISKUSI

Bab ini mendeskripsikan implementasi model atau sistem yang dirancang, termasuk pengujian, validasi, dan analisis hasil secara komprehensif. Dijelaskan pula langkah-langkah eksekusi, tantangan yang muncul selama proses, solusi yang diterapkan, serta implikasi dari temuan terhadap pengelolaan hipertensi dan gagal jantung.

BAB VI KESIMPULAN DAN SARAN

Bab ini merangkum keseluruhan hasil penelitian, menarik kesimpulan utama, serta membahas keterbatasan yang ada. Selain itu, diberikan saran praktis dan rekomendasi untuk pengembangan lebih lanjut, baik dari aspek teknologi maupun aplikasi di bidang kesehatan.

BAB II TINJAUAN PUSTAKA

Studi literatur ini akan menyajikan landasan teori tentang teknologi *Internet of Things* (IoT) dalam pemantauan kesehatan, khususnya penggunaan *Bluetooth Low Energy* (BLE) untuk transfer data pada perangkat medis seperti tensimeter Omron, serta konsep autentikasi *multiuser* dan isolasi data berbasis *backend* untuk mendukung sistem pemantauan tekanan darah yang aman dan efisien.

II.1 *Remote Patient Monitoring* (RPM)

Remote Patient Monitoring (RPM) merupakan metode perpindahan otomatis data pasien, seperti tekanan darah dan glukosa, tanpa menatap muka langsung. Sehingga, hal ini memungkinkan sistem digitalisasi kesehatan melalui integrasi perangkat IoT untuk pemantauan yang berkelanjutan (Debnath dkk., 2023). Dalam hal medis, RPM dapat meningkatkan pengelolaan penyakit kronis yang dapat menurunkan angka rawat inap, mengurangi biaya operasional medis, serta dapat memperluas akses layanan kesehatan melalui interaksi yang lebih sering antara pasien dan tenaga medis (Hailu dkk., 2024). Secara teknis, RPM memanfaatkan teknologi sensor biomedis seperti *bluetooth* untuk mengukur vital seperti detak jantung, saturasi oksigen, serta dapat mengukur suhu tubuh secara *real-time*. Data yang dihasilkan dapat ditransmisikan atau disalurkan oleh jaringan *bluetooth* ke server untuk kemudian diakses oleh dokter untuk melihat data kesehatan pasien melalui visualisasi grafik untuk analisis tren (Debnath dkk., 2023).

Meskipun demikian, adopsi RPM menghadapi beberapa tantangan seperti kesulitan staf medis dalam mengelola *volume* data yang besar (Hailu dkk., 2024), kebutuhan komunikasi data secara terpisah untuk mencegah pencampuran informasi pasien, konflik sinyal pada banyak perangkat nirkabel (Al-Shareeda dkk., 2023), serta keterbatasan efisiensi energi dan *bandwidth* protokol komunikasi (Abdulmalek dkk., 2022). Hal ini memerlukan strategi pengelolaan jaringan yang lebih optimal.

II.2 *Bluetooth Low Energy* (BLE)

Bluetooth merupakan teknologi komunikasi jarak pendek pada jaringan 2.4 GHz yang dirancang untuk konsumsi daya yang sangat rendah, menggunakan modul GFSK dan frekuensi *hopping* untuk komunikasi data dalam paket kecil (Liu dkk.,

2021). Teknologi ini ideal untuk perangkat IoT dengan sumber utama baterai, karena mampu bertahan bertahun-tahun, sementara *Bluetooth* 5.0 mampu meningkatkan kecepatan hingga 2 Mbps dan jangkauan hingga empat kali lipat dibandingkan model *bluetooth* sebelumnya (McLenna Australia & Onotu Fache, 2023). Selain itu, menurut Raj (2021) *Bluetooth Low Energy* (BLE) merupakan teknologi komunikasi nirkabel berdaya rendah yang mampu menghubungkan banyak perangkat seperti *smartphone*, *smartwatch*, dan *fitness tracker* secara bersamaan (Raj, 2021).

Bluetooth Low Energy (BLE) mengalami perubahan yang signifikan sejak diperkenalkannya *Bluetooth* 5.0 dengan fitur-fitur baru seperti *mesh networking* peningkatan keamanan yang mendukung integrasi luas dalam ekosistem IoT (Koulouras dkk., 2025). *Bluetooth* 5.0 memberikan peningkatan signifikan, termasuk jangkauan hingga empat kali lebih jauh, kecepatan dua kali lebih tinggi, dan kapasitas penyiaran data delapan kali lebih besar, disertai dukungan *mesh networking* untuk komunikasi *many-to-many* serta fitur keamanan yang ditingkatkan, sehingga mengoptimalkannya untuk aplikasi IoT (Koulouras dkk., 2025). Menurut Alhussan dkk. (2025) meskipun demikian, protokol BLE yang sederhana menimbulkan kerentanan keamanan dan privasi, seperti serangan *Man-in-the-Middle* (MITM) aktif yang memungkinkan manipulasi data fisiologis (misalnya detak jantung atau kadar oksigen), sehingga mengancam integritas data dan keselamatan pasien. Dalam konteks *Internet of Medical Things* (IoMT), BLE menawarkan kinerja yang seimbang dalam hal latensi rendah dan tidak adanya kehilangan data untuk transmisi *real-time* seperti pemantauan ECG, menjadikannya pilihan ideal untuk pemantauan pasien bergerak di lingkungan rumah sakit (Doe dkk., 2025).

Keunggulan *Bluetooth Low Energy* (BLE) terletak pada efisiensi energi melalui mekanisme *broadcast* data dengan paket kecil dan melakukan *scanning* dengan cepat, dukungan topologi *star* untuk *mesh* ribuan node, fitur keamanan AES-128, serta kemampuan *pairing* instan dengan perangkat seluler tanpa perangkat keras lainnya (Al-Shareeda dkk., 2023). Dalam RPM, *Bluetooth Low Energy* (BLE) berperan penting sebagai protokol standar pada perangkat *wearable* dan sensor biomedis seperti monitor detak jantung dan glukosa, menghubungkan data

fisiologis serta *real-time* ke *gateway* atau *smartphone*, kemudian diteruskan ke server untuk melakukan analisis, sehingga menjadi fondasi utama inovasi layanan kesehatan berbasis IoT (Liu dkk., 2021). Menurut Raj (2021) *Bluetooth Low Energy* (BLE) juga dapat mendukung aplikasi pada perangkat medis untuk mentransfer data pasien ke rumah sakit secara jarak jauh, memungkinkan inovasi di bidang kesehatan melalui konektivitas berdaya rendah

II.3 Pemanfaatan Internet of Things (IoT) untuk Kesehatan

IoT merupakan sebuah gelombang transformasi yang ditandai dengan jaringan perangkat yang terhubung dan mampu menghasilkan, mengumpulkan, serta bertukar data (Akoh Atadoga dkk., 2024). Dalam pemanfaatan *Internet of Things* (IoT) di dunia medis, atau bisa kita sebut *Internet of Medical Things* (IoMT), *Internet of Medical Things* (IoMT) merupakan aplikasi khusus IoT di sektor kesehatan berupa jaringan perangkat medis yang terhubung dengan monitor, sensor dan alat kesehatan pemantauan pasien jarak jauh yang memungkinkan pengumpulan serta pertukaran data kesehatan pasien secara *real-time* untuk diagnosis dan pengelolaan pasien agar lebih efisien (Akoh Atadoga dkk., 2024). Namun, implementasi ini menghadapi tantangan yang serius terkait privasi serta keamanan data yang memerlukan enkripsi yang kuat (Akoh Atadoga dkk., 2024)

Keterhubungan sistem IoMT sangat krusial untuk mengintegrasikan berbagai perangkat dan platform yang beragam, sehingga memungkinkan pandangan komprehensif terhadap kesehatan pasien melalui pertukaran data yang lancar. Salah satu aplikasi utama dari arsitektur IoMT ini adalah pemantauan tanda vital pasien secara *real-time*, di mana sensor-sensor biomedis yang terintegrasi pada perangkat *wearable* atau portabel memungkinkan pengukuran parameter fisiologis krusial seperti detak jantung, tekanan darah, saturasi oksigen, dan kadar glukosa darah secara *real-time* dan akurat (Islam et al., 2020).

II.4 Pemantauan Tanda Vital

Pemantauan tanda vital merupakan proses pengukuran dan pengamatan parameter dasar tubuh manusia, seperti tekanan darah, detak jantung, saturasi oksigen (SpO₂), suhu tubuh, dan laju pernapasan, yang berfungsi sebagai indikator utama kondisi kesehatan pasien (Hymczak dkk., 2021). Secara umum, batas normal untuk orang

dewasa sehat saat istirahat mencakup tekanan darah sistolik kurang dari 130 mmHg dan diastolik kurang dari 85 mmHg, detak jantung 60-100 kali per menit, laju pernapasan 12-20 kali per menit, suhu tubuh inti 36,5-37,5 °C, serta saturasi oksigen 95–100% (Shaffer dkk., 2020). Dalam era digital kesehatan, pemantauan tanda vital telah berevolusi dari metode manual dan periodik di fasilitas medis menjadi sistem kontinu dan jarak jauh (*remote monitoring*) berbasis *Internet of Medical Things* (IoMT), di mana sensor-sensor biomedis yang terintegrasi pada perangkat *wearable* atau portabel memungkinkan pengumpulan data secara *real-time* (Syarifudin dkk., 2020).

Studi kesehatan menunjukkan bahwa pemantauan tanda vital jarak jauh secara signifikan meningkatkan kualitas hasil perawatan pasien, misalnya dengan menurunkan tekanan darah sistolik hingga 16,7 mmHg pada pasien hipertensi stage 2 dan mengurangi frekuensi kunjungan darurat melalui intervensi dini berbasis data historis (Smith dkk., 2024). Integrasi dengan *dashboard mobile* memungkinkan visualisasi data dalam bentuk grafik dan alert otomatis jika parameter melebihi ambang batas normal seperti tekanan darah $\geq 140/90$ mmHg yang menandakan hipertensi sehingga mendukung pengelolaan pasien yang lebih proaktif (Chakraborty dkk., 2021). Namun, implementasi pemantauan tanda vital dalam lingkungan *multi-user* seperti rumah tangga dengan beberapa anggota keluarga atau fasilitas kesehatan dengan banyak pasien menghadirkan tantangan teknis, di mana penggunaan secara bersamaan pada perangkat *Bluetooth* pada frekuensi 2,4 GHz dapat menyebabkan konflik sinyal, dan penundaan data, yang berpotensi mengganggu kualitas data vital (Cheng dkk., 2025).

Dengan demikian, pengembangan sistem pemantauan tanda vital yang mendukung *multi-user* melalui konektivitas *Bluetooth* dan *dashboard mobile* menjadi solusi potensial untuk mengatasi keterbatasan tersebut sambil meningkatkan aksesibilitas layanan kesehatan (Dias dkk., 2023). Salah satu perangkat yang telah mengadopsi pendekatan ini dengan baik adalah tensimeter Omron, yang mengintegrasikan teknologi pengukuran tekanan darah akurat dengan fitur konektivitas nirkabel untuk mendukung pemantauan jarak jauh yang andal.

II.5 Tensimeter Omron

Tensimeter Omron adalah perangkat pemantau tekanan darah otomatis berbasis osilometri yang dirancang untuk penggunaan mandiri di rumah dengan kemudahan memasukkan lengan ke dalam silinder manset tanpa perlu memperhatikan posisi atau pembungkusan secara khusus (Takahashi dkk., 2024). Perangkat ini menampilkan hasil tekanan darah sistolik, diastolik, serta denyut nadi melalui layar digital, sekaligus mendeteksi tanda posisi lengan, denyut jantung tidak teratur, dan gerakan tubuh selama proses pengukuran (Darmawan et al., 2022). Pengukuran dilakukan selama pemakaian manset tensimeter Omron, dengan rentang tekanan 0–299 mmHg dan denyut nadi 40–180 bpm, sehingga memberikan hasil yang cepat dan nyaman bagi pengguna umum (Jena & Gupta, 2021).

Tensimeter Omron mendukung pengukuran pada lengan atas dengan rentang lingkaran manset yang luas untuk kenyamanan berbagai ukuran lengan dewasa, serta menawarkan fitur deteksi kesalahan posisi dan transfer data hasil pengukuran ke *smartphone* (Boima dkk., 2024). Kemampuan konektivitas nirkabel seperti Bluetooth pada tensimeter Omron oleh beberapa seri memfasilitasi pengelolaan data oleh beberapa pengguna sekaligus (Darmawan et al., 2022).

II.6 Manajemen Multiuser pada Bluetooth

Manajemen multiuser dalam jaringan *Bluetooth Low Energy* (BLE) mengacu pada mekanisme kompleks yang memungkinkan sebuah *central device* (seperti *gateway* atau *smartphone*) untuk mempertahankan koneksi stabil dengan beberapa *peripheral devices* (sensor atau tensimeter) secara simultan. Tantangan utama dalam skenario ini adalah keterbatasan alokasi waktu dan *bandwidth*, di mana peningkatan jumlah perangkat yang terhubung berbanding lurus dengan risiko tabrakan paket data (*packet collision*) dan kegagalan transmisi (Agnolotto dkk., 2020). Manajemen *multiuser* pada *Bluetooth Low Energy* (BLE) juga memungkinkan perangkat untuk menangani beberapa koneksi secara bersamaan dengan mengatur titik *anchor* untuk menghindari tumpang tindih sumber daya (Park dkk., 2024). Pendekatan ini meningkatkan efisiensi dengan menyesuaikan interval koneksi berdasarkan beban lalu lintas data, sehingga memastikan kualitas layanan yang stabil untuk aplikasi kesehatan seperti pemantauan tekanan darah

(Park dkk., 2024). Dalam implementasi sistem *multiuser* untuk tensimeter, BLE memanfaatkan skema penjadwalan fleksibel untuk mengalokasikan sumber daya secara adil di antara pengguna, mencegah penurunan performa akibat konflik koneksi (Park dkk., 2024). Integrasi dengan *dashboard mobile* memungkinkan akses data *real-time* dari beberapa pengguna tanpa mengganggu privasi, melalui pengaturan parameter *WinOffset* untuk mengoptimalkan alokasi waktu (Park dkk., 2024).

Penerapan *Role-Based Access Control* (RBAC) dalam manajemen *multiuser* BLE memastikan akses data kesehatan dibatasi berdasarkan peran pengguna, seperti dokter atau pasien, untuk melindungi privasi dalam sistem *tensimeter* (Suneetha & Singhal, 2024). Mekanisme ini menggabungkan *tagging* data dengan anonimisasi untuk mengklasifikasikan informasi sensitif, sehingga mendukung berbagi data *multiuser* tanpa risiko pelanggaran (T. Garg dkk., 2023). Dalam konteks konektivitas *Bluetooth*, penjadwalan slot waktu dengan prioritas paket memungkinkan transmisi *real-time* dari perangkat seperti *tensimeter* Omron ke *dashboard mobile*, mengurangi *jitter* dan kehilangan paket untuk beberapa pengguna (Agnoletto et al., 2020). Selain itu, skema alokasi sumber daya yang adil menggunakan *subrating* koneksi memungkinkan skalabilitas hingga puluhan pengguna, sambil menjaga efisiensi energi dan *throughput* data (Kim & Paek, 2025).

Implementasi teknis dari manajemen koneksi *multiuser* dan logika keamanan ini pada sisi pengguna akhir sangat bergantung pada keandalan perangkat lunak antarmuka yang digunakan. Kompleksitas penanganan banyak koneksi BLE secara asinkron, pengelolaan *queue* (antrian) data dari berbagai sensor, serta visualisasi data secara *real-time* menuntut lingkungan pengembangan aplikasi yang tangguh dan efisien (Kurniawan et al., 2021). Platform Android modern menyediakan API *Bluetooth* yang lengkap, namun pengelolaannya membutuhkan bahasa pemrograman yang mampu menangani *multiuser* dengan aman untuk mencegah *crash* aplikasi saat beban trafik data tinggi. Kebutuhan akan kode yang ringkas, aman dari *null-pointer exception*, dan memiliki performa tinggi inilah yang mendorong pemilihan teknologi pemrograman modern, yang akan dibahas lebih lanjut.

II.7 Pengembangan Aplikasi Mobile dengan Framework Expo

React Native adalah sebuah *Framework* berbasis JavaScript untuk membuat sebuah aplikasi berbasis mobile, baik itu android maupun iOS (Wibawa & Naufal, 2023). React Native sendiri mempunyai salah satu framework yaitu Expo. Expo adalah sebuah Framework untuk membangun cepat aplikasi dari React Native tanpa harus adanya IDE seperti Android Studio maupun Xcode, Expo bisa dijalankan langsung di handphone sendiri tanpa memiliki SDK (*Software Development Kit*) sendiri yang mempermudah developer yang ingin membuat sebuah aplikasi Android maupun iOS dengan cepat (Wibawa & Naufal, 2023). Menurut Wibawa & Naufal (2023) salah satu keunggulan fundamental *framework* ini adalah di dalam Expo sudah di sediakan sebuah SDK kemudian terdapat *QRCode* yang dapat discan untuk menjalankan aplikasi yang sedang dibangun.

Dalam konteks pengembangan sistem pemantauan kesehatan berbasis *Internet of Things*, kemampuan React Native dalam menangani pemrograman asinkron melalui mekanisme *async/await* dan *Promise* menjadi nilai tambah yang krusial, di mana Expo sebagai framework berbasis React Native turut mendukung pengelolaan operasi tersebut (Shafi dkk., 2024). Mekanisme *async/await* yang dimiliki React Native memungkinkan aplikasi untuk membaca data sensor tanpa memblokir UI, seperti pengambilan data detak jantung, saturasi oksigen, dan suhu tubuh dari perangkat IoT secara *real-time*, di mana proses pengambilan data berjalan di latar belakang sementara antarmuka pengguna tetap responsif (Yi dkk., 2022). Hal ini sejalan dengan temuan P. Garg dkk. (2023) yang menjelaskan bahwa React Native memisahkan eksekusi logika JavaScript pada JavaScript *thread* dan pengelolaan antarmuka pada UI *thread* (*native thread*), di mana komunikasi antar keduanya dilakukan melalui mekanisme *bridge* yang mengirimkan data dalam format JSON secara asinkron. Arsitektur dua-*thread* ini memungkinkan operasi pengambilan data sensor IoT berjalan tanpa menghambat *rendering* antarmuka pengguna, sehingga *frame rate* UI dapat dipertahankan secara konsisten. Dalam pengembangan aplikasi mobile menggunakan Expo adapun metode pengembangan yang harus diterapkan, untuk mencapai hasil yang sempurna.

II.8 Pengembangan Application Programming Interface (API) Berbasis Framework FastAPI

Dalam arsitektur sistem *Remote Patient Monitoring* (RPM) berbasis *Internet of Things* (IoT), keberadaan *backend* yang tangguh dan responsif memainkan peran sentral sebagai perantara (*middleware*) antara perangkat antarmuka pengguna (*mobile dashboard*) dan basis data terpusat (Mabotha dkk., 2025). *Backend* bertanggung jawab untuk mengolah logika bisnis, mengautentikasi pengguna, mengelola sesi koneksi, serta memastikan lalu lintas data kesehatan dikirimkan secara aman tanpa risiko kebayangan atau kebingungan data antar pengguna (*data mismatch*). Dalam menangani pemantauan tanda vital secara berkesinambungan diperlukan sebuah kerangka kerja (*framework*) *web service* yang memiliki latensi rendah dan *throughput* tinggi (Ivanov, 2026).

FastAPI merupakan salah satu *framework* web Python modern berkinerja tinggi yang dirancang khusus untuk membangun *Application Programming Interface* (API) berbasis standar OpenAPI dan JSON Schema dalam lingkungan IoT (Mabotha dkk., 2025). Berbeda dengan *framework* Python tradisional berbasis WSGI (*Web Server Gateway Interface*) seperti Flask atau Django yang bersifat *synchronous*, FastAPI dibangun di atas Starlette dan Pydantic yang memanfaatkan arsitektur *Asynchronous Server Gateway Interface* (ASGI). Hal ini memungkinkan FastAPI untuk menjalankan operasi I/O secara asinkron (*non-blocking I/O*) memanfaatkan sintaks ``async`` dan ``await`` bawaan Python (Mabotha dkk., 2025). Dalam skenario sistem pemantauan *multiuser*, kemampuan pemrosesan asinkron ini sangat krusial karena *backend* dapat menangani ratusan hingga ribuan permintaan koneksi dan transmisi data secara bersamaan dari berbagai perangkat tanpa menyebabkan kemacetan (*bottleneck*) pada server (Ivanov, 2026).

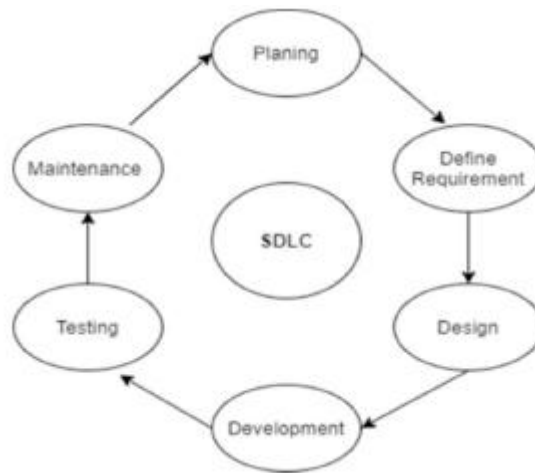
Keunggulan fundamental lainnya dari FastAPI terletak pada integrasinya dengan perpustakaan Pydantic untuk validasi data otomatis dan eksekusi tipe data (*type hinting*) secara *real-time* (Mabotha dkk., 2025). Setiap data masukan seperti nilai sistolik, diastolik, *pulse rate*, dan SpO2 akan divalidasi berdasarkan skema Pydantic yang telah ditentukan sebelum diproses lebih lanjut ke basis data. Jika data yang dikirimkan oleh perangkat *mobile* atau sensor tidak memenuhi spesifikasi (misalnya nilai sampel kosong atau di luar rentang fisiologis valid), FastAPI secara otomatis

menolak permintaan tersebut dan mengembalikan respon kesalahan berformat JSON secara terstruktur. Hal ini sangat penting dalam menjaga integritas data medis dan mencegah korupsi data pada basis data terpusat (Ivanov, 2026).

Selain itu, FastAPI menyediakan fitur keamanan terintegrasi berbasis OAuth2 dan JSON Web Token (JWT) yang sangat mendukung skenario autentikasi dan isolasi data *multiuser* (Mabotha dkk., 2025). Melalui mekanisme otorisasi berbasis token ini, setiap permintaan yang dikirim dari aplikasi *mobile* harus membawa *token* enkripsi yang memverifikasi identitas pengguna aktif. Dengan demikian, sistem *backend* dapat mengisolasi hak akses data secara ketat per ID pengguna (User ID), memastikan bahwa pengguna hanya dapat mengakses dan menyimpan data pengukuran miliknya sendiri, serta mendukung kontrol akses berbasis peran (*Role-Based Access Control* / RBAC) untuk membedakan hak akses antara pengguna (pasien) dan tenaga medis (Li dkk., 2022). Penggunaan FastAPI juga mempermudah proses integrasi dan pengujian sistem melalui fitur dokumentasi otomatis interaktif (Swagger UI dan ReDoc) yang di-*generate* secara langsung dari kode program, sehingga mempercepat kolaborasi pengembangan antara sisi *backend* dan *frontend mobile*. Dalam pengembangan aplikasi dan *backend* adapun metode pengembangan yang harus diterapkan, untuk mencapai hasil yang sempurna.

II.9 Pengembangan Sistem RPM Berbasis Mobile

Metode pengembangan perangkat lunak merupakan sebuah serangkaian langkah yang dirancang secara sistematis untuk memastikan keberhasilan dalam membangun sebuah sistem (ALazzawi et al., 2023). Menurut ALazzawi dkk. (2023) setiap tahapan metode pengembangan harus berdasarkan rencana yang terstruktur serta analisis terstruktur agar dapat memenuhi kebutuhan proyek dengan efisien, metode pengembangan yang terencana dan teruji dapat memungkinkan pengembangan perangkat lunak yang sempurna sebelum diterapkan ke model pengembangan spesifik seperti *Software Development Life Cycle* (SDLC). *Software Development Life Cycle* (SDLC) adalah proses pengembangan perangkat lunak yang sesuai dengan langkah-langkah rangkaian *designing, coding, testing, dan finalizing the product* (Alam dkk., 2022).



Gambar II-1 *Software Development Life Cycle (SDLC)*

Sumber: (Alam dkk., 2022)

Menurut Alam dkk. (2022) model-model SDLC dibagi menjadi 2 model berdasarkan faktor-faktor seperti dokumentasi, ukuran tim, orientasi, dan pendekatan seperti dijelaskan sebagai berikut:

II.9.1 **Lightweight Methodologies**

Menurut Montoya-Murillo dkk. (2025) *Lightweight Methodologies* adalah pendekatan pengembangan perangkat lunak yang lebih fleksibel, berorientasi hasil, dan adaptif terhadap perubahan, dengan keseimbangan antara prediktif dan adaptif untuk proyek skala kecil hingga menengah. Menurut Alam dkk. (2022) Ciri-ciri *Lightweight Methodologies* antara lain Dokumentasi lebih sedikit, Ukuran tim kecil Berorientasi pada orang, Pendekatan non-prediktif, Tidak kritis, dan contoh dari penerapan dari *Lightweight Methodologies* yaitu:

II.9.1.1 **Agile**

Agile adalah pendekatan pengembangan perangkat lunak yang dinamis dan iteratif, dengan siklus pendek (*sprint*) yang memungkinkan adaptasi berkelanjutan terhadap perubahan kebutuhan (Wadkar & Ghorpade, 2024). Kelebihan menggunakan Agile menurut Wadkar & Ghorpade (2024) adalah Sangat fleksibel dan adaptif terhadap perubahan dan kolaborasi erat dengan objek untuk hasil yang sesuai kebutuhan. Namun menurut Wadkar & Ghorpade (2024) adapun kekurangan dari model Agile ini adalah Kurang dapat diprediksi (*timeline*, biaya, *scope* sulit

diperkirakan awal), membutuhkan kolaborasi tim yang kuat dan komitmen tim, Risiko *scope creep* jika perubahan tidak dikelola dengan baik.

II.9.1.2 AZ Model

AZ Model adalah metodologi pengembangan perangkat lunak *intermediate* antara model *heavyweight* dan *lightweight*, model ini dirancang untuk mengatasi keterbatasan keduanya (Akbar dkk., 2017). Model ini dibagi menjadi tiga fase utama yaitu pertama *Design and Communication* (melibatkan customer untuk pengumpulan *requirement*, analisis, desain, *prototyping*, *unit testing*, dan *risk analysis* hingga desain disetujui), kedua *Development* (*coding*, *testing*, *deployment* dengan fokus pada *usability* dan SRS), Ketiga *Product Release* (pelepasan produk ke pasar dengan *time-boxing* dan manajemen proyek khusus sepanjang proses) (Akbar dkk., 2017). Selain itu, adapun kelebihan jika menggunakan AZ Model menurut Akbar dkk. (2017) yaitu membatasi *Work in Process* (WIP) untuk tugas terhitung dan terkontrol dan proses pengembangan terlihat, memudahkan penyesuaian alur kerja dan kontrol *overloading*. Namun, adapun kekurangan dalam menggunakan AZ Model menurut Akbar dkk. (2017) adalah kurang fleksibel dibanding Agile untuk perubahan berkelanjutan setelah fase desain, berpotensi lebih mahal dan memakan waktu dibanding Agile, berdasarkan perbandingan, memerlukan manajer proyek khusus per fase, yang bisa menambah *overhead* organisasi

II.9.2 Heavyweight Methodologies

Menurut Montoya-Murillo dkk. (2025) *Heavyweight Methodologies* adalah metodologi pengembangan perangkat lunak tradisional yang bersifat kaku dan berfokus pada pembangunan sistem prediktif dalam skala besar, tetapi kurang fleksibel dan adaptif terhadap perubahan lingkungan bisnis yang cepat. Menurut Alam dkk. (2022) Ciri-ciri *Heavyweight Methodologies* antara lain Dokumentasi yang rumit (SRS), Ukuran tim yang besar, Berorientasi pada alat, Pendekatan prediktif, Sangat kritis, dan contoh dari penerapan dari *Heavyweight Methodologies* yaitu:

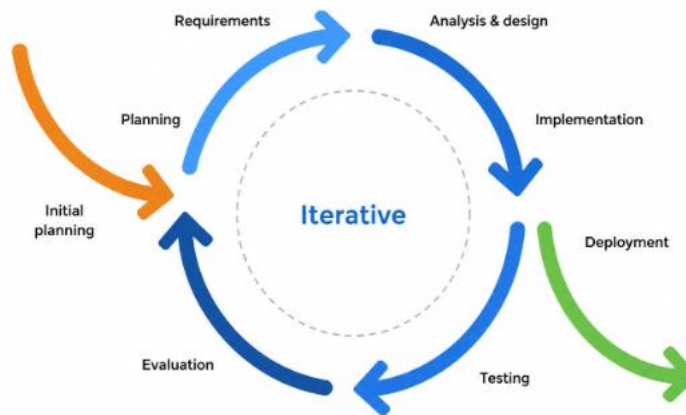
II.9.2.1 Waterfall

Waterfall adalah model yang mengikuti pendekatan linier dan sekuensial, di mana setiap fase (seperti analisis kebutuhan, desain, implementasi, pengujian, *deployment*, dan *maintenance*) diselesaikan sepenuhnya sebelum fase berikutnya dimulai (Wadkar & Ghorpade, 2024). Kelebihan dari menggunakan Waterfall menurut Wadkar & Ghorpade (2024) adalah sederhana dan mudah digunakan, terutama untuk proyek kecil dengan kebutuhan yang jelas, dan Mudah dikelola karena tahapan yang jelas dan milestone tertentu. Sedangkan adapun kekurangan dari model Waterfall ini, yaitu kaku dan sulit mengakomodasi perubahan setelah fase selesai, pengujian dilakukan terlambat, sehingga masalah terdeteksi di akhir proses dan mahal untuk diperbaiki, dan tidak cocok untuk proyek kompleks atau yang kebutuhannya sering berubah

II.9.2.2 Model V

Model V atau bisa disebut *Verification and Validation Model* adalah ekstensi dari Waterfall yang berbentuk "V", dengan penekanan pada hubungan paralel antara fase pengembangan (di sisi kiri) dan fase pengujian (di sisi kanan). Kelebihan dalam menggunakan model V ini menurut Wadkar & Ghorpade (2024) adalah deteksi cacat dini melalui pengujian di setiap tahap, pendekatan terstruktur dan jelas, mudah dikelola, dokumentasi kuat untuk referensi jangka panjang. Selain itu, kekurangan dalam menggunakan Model V ini menurut Wadkar & Ghorpade (2024) adalah kaku dan sulit mengakomodasi perubahan setelah fase selesai, terdapat biaya tinggi jika ada revisi fase awal, kurang cocok untuk proyek dengan kebutuhan yang tidak pasti atau sering berubah

II.9.2.3 *Iterative Incremental*



Gambar II-2 Iterative Incremental

Iterative Incremental adalah sebuah metode pengembangan perangkat lunak yang bersifat plan-driven, di mana proses pembangunan sistem dipecah menjadi beberapa siklus yang berulang (iterasi) sekaligus penambahan fitur yang dilakukan secara bertahap (inkremen) (Chowdhury dkk., 2020). Menurut Chowdhury dkk. (2020), kelemahan mendasar dari model lainnya seperti *Waterfall* tradisional terletak pada ketidakmampuannya mengakomodasi perubahan kebutuhan maupun masukan pengguna yang muncul di fase-fase awal pengembangan. Sebagai solusinya, model *Iterative Incremental* membagi produk perangkat lunak ke dalam unit-unit kecil (*increments*), di mana masing-masing unit tersebut tetap melalui seluruh tahapan SDLC secara utuh (Chowdhury dkk., 2020). Pada pendekatan ini, seluruh kebutuhan sistem terlebih dahulu diidentifikasi, kemudian dipilah dan disusun berdasarkan tingkat prioritasnya. Kebutuhan yang memiliki prioritas paling tinggi akan ditempatkan pada inkremen paling awal, sehingga hasil produk dapat terlihat lebih cepat (*early outcome*) dan tim pengembang bisa memperoleh umpan balik dari pengguna secara berkala guna menyempurnakan inkremen-inkremen selanjutnya (Chowdhury dkk., 2020).

Secara garis besar, Chowdhury dkk. (2020) menggambarkan alur kerja model *Iterative Incremental* yang dimulai dari tahap *Initial Planning* (perencanaan awal), yang kemudian dilanjutkan dengan siklus iteratif-inkremental yang mencakup enam tahapan pokok.

Tabel II-1 Tahap Pengembangan *Iterative Incremental*

No	Tahapan	Fokus Utama & Aktivitas	Deskripsi
1	<i>Requirements & Planning</i>	Identifikasi kebutuhan & perencanaan	Mengidentifikasi tujuan proyek, menentukan batasan sistem, serta membagi dan memprioritaskan fungsi-fungsi utama yang akan dikembangkan pada inkremen berjalan (Chowdhury dkk., 2020).
2	<i>Analysis & Design</i>	Analisis spesifikasi & perancangan arsitektur	Menganalisis spesifikasi kebutuhan dan merancang arsitektur sistem, basis data, serta antarmuka komponen perangkat lunak maupun perangkat keras (Chowdhury dkk., 2020).
3	<i>Implementation</i>	Realisasi rancangan & penulisan kode	Merealisasikan rancangan ke dalam kode program (<i>code writing</i>) dan memfungsikan modul atau subsistem sesuai

			spesifikasi inkremen (Chowdhury dkk., 2020).
4	<i>Testing</i>	Pengujian fungsional & teknis	Melakukan pengujian fungsionalitas dan kinerja teknis terhadap komponen yang telah dibangun untuk memastikan kesesuaian dengan spesifikasi kebutuhan (Chowdhury dkk., 2020).
5	<i>Evaluation</i>	Penilaian & pengumpulan umpan balik	Menilai hasil pengembangan dan mengumpulkan umpan balik dari pengguna maupun pemangku kepentingan sebagai dasar perbaikan pada iterasi selanjutnya (Chowdhury dkk., 2020).
6	<i>Deployment</i>	Peluncuran & penyebaran sistem	Mengintegrasikan dan menyebarkan versi sistem yang telah matang ke lingkungan operasional setelah

			seluruh siklus iterasi diselesaikan secara memadai (Chowdhury dkk., 2020).
--	--	--	--

Chowdhury dkk. (2020) menegaskan bahwa model *iterative incremental* menawarkan tingkat fleksibilitas serta kendali yang baik dalam mengelola kompleksitas dan dinamika yang muncul selama pengembangan sistem. Melalui pendekatan ini, fungsionalitas dapat ditambahkan secara bertahap sekaligus dievaluasi secara berkala pada setiap akhir siklus iterasi (Desai dkk., 2024; Kokol dkk., 2022). Shaheen dkk. (2025) menambahkan bahwa model iteratif memungkinkan dilakukannya peninjauan ulang terhadap prioritas subsistem setelah suatu inkremen sebelumnya selesai dikembangkan dan dievaluasi, sehingga evolusi perangkat lunak dapat berlangsung secara berkelanjutan serta adaptif terhadap perubahan kebutuhan yang terjadi.

Jika dibandingkan dengan model *waterfall*, pendekatan *iterative incremental* dinilai lebih mampu mengakomodasi perubahan dan ketidakpastian yang muncul di sepanjang proses rekayasa perangkat lunak (Chowdhury dkk., 2020; Thesing dkk., 2021). Sementara itu, jika dibandingkan dengan kerangka kerja *Agile* seperti *Scrum* yang bertumpu pada peran-peran formal dan rangkaian seremoni yang intensif, *Iterative Incremental* menawarkan struktur siklus evaluasi dan perbaikan yang lebih terukur, namun tetap fleksibel untuk diterapkan pada berbagai skala proyek tanpa mengharuskan penerapan ritual manajemen formal secara penuh (Sassa dkk., 2023; Shaheen dkk., 2025).

II.10 Pengujian Perangkat Lunak

II.10.1 User Acceptance Test (UAT)

User Acceptance Testing (UAT) merupakan tahapan pengujian perangkat lunak yang bertujuan untuk memastikan sistem yang telah dibangun sesuai dengan kebutuhan serta ekspektasi pengguna akhir (Wulandari Putri Bahmin dkk., 2025). Adapun Vanesha dkk. (2024) mendefinisikan UAT sebagai tahap pengujian akhir

dalam siklus pengembangan sistem, yang memungkinkan pengguna untuk mencoba dan berinteraksi secara langsung dengan sistem sebelum sistem tersebut resmi diimplementasikan atau diluncurkan.

Lebih lanjut, Aliyah dkk. (2024) menjelaskan bahwa evaluasi dalam UAT mencakup lima variabel pokok, yaitu fungsionalitas sistem, kinerja sistem, pengalaman pengguna serta tampilan antarmuka, efisiensi dan produktivitas, serta aspek keamanan dan keandalan sistem. Hasil dari pengujian ini biasanya dituangkan dalam bentuk skor interpretasi kuesioner yang menggambarkan sejauh mana pengguna dapat menerima sistem yang diuji. Skor tersebut selanjutnya diklasifikasikan ke dalam skala kelayakan tertentu, di mana kategori "sangat baik" diberikan apabila persentase hasil interpretasi berada pada rentang yang tinggi.

II.10.2 Blackbox Testing

Black Box Testing adalah metode pengujian perangkat lunak yang menitikberatkan pada pengujian fungsi-fungsi sistem tanpa perlu menelusuri kode program atau struktur internal aplikasi (Huda dkk., 2022). Dengan pendekatan ini, penguji cukup mengamati kesesuaian antara data masukan (*input*) yang diberikan dengan keluaran (*output*) yang dihasilkan, mengacu pada skenario pengujian yang sudah dirancang sebelumnya, tanpa perlu memahami alur logika pemrograman di balik sistem tersebut. Sementara itu, Abdillah dkk. (2023) mengungkapkan bahwa *Black Box Testing* adalah teknik pengujian fungsional yang disusun berdasarkan spesifikasi sistem, tanpa melibatkan pemahaman terhadap struktur internalnya, sehingga teknik ini sesuai digunakan untuk memastikan bahwa keluaran sistem telah selaras dengan kebutuhan fungsional yang telah ditetapkan sebelumnya.

Terdapat beragam teknik yang tercakup dalam pengujian *Black Box*, antara lain *Equivalence Partitioning*, *Boundary Value Analysis (Limit Testing)*, *Comparison Testing*, *Sample Testing*, *Robustness Testing*, *Behavior Testing*, *Requirement Testing*, *Performance Testing*, *Endurance Testing*, dan *Cause-Effect Relationship Testing* (Pangestika & Widiarti., 2025). Dari sekian teknik tersebut, *Equivalence Partitioning* menjadi salah satu yang paling sering diterapkan dalam penelitian bidang sistem informasi. Teknik ini dilakukan dengan menguji data masukan pada setiap *form* yang terdapat dalam sistem, di mana tiap menu *input* dikelompokkan

sesuai fungsinya, baik untuk kondisi data valid maupun tidak valid Pangestika & Widiyanti (2025). Pada praktiknya, setiap test case disusun secara terstruktur dengan mencantumkan uraian skenario pengujian, data *input* yang digunakan, hasil yang diharapkan, serta hasil yang benar-benar diperoleh, sebagai dasar penentuan status pengujian "berhasil" (*pass*) atau "gagal" (*fail*) (Huda dkk., 2022).

II.10.3 Load Testing

Load testing merupakan salah satu bentuk pengujian performa non-fungsional yang bertujuan mengukur responsivitas, keandalan, dan stabilitas sistem atau API saat menerima beban kerja tertentu (Hendayun dkk., 2023; Rath dkk., 2025). Pengujian ini dilakukan dengan mensimulasikan sejumlah pengguna *virtual* (*Virtual Users/VUs*) yang mengirimkan request secara bersamaan ke server *backend* dalam rentang waktu tertentu, guna mengidentifikasi batas kapasitas sistem serta menemukan potensi *bottleneck* seperti keterbatasan memori, *database locking*, atau lambatnya proses I/O sebelum sistem dirilis ke tahap produksi (Hendayun dkk., 2023; Jain dkk., 2022). Pada arsitektur *Remote Patient Monitoring* (RPM) berbasis IoT, pengujian ini penting untuk memastikan server *backend* tetap dapat menangani lonjakan transmisi data kesehatan dari banyak perangkat tanpa mengorbankan integritas data maupun kecepatan respons.

Efektivitas *load testing* umumnya dinilai melalui beberapa indikator kinerja utama (KPI), yaitu *response time* (rata-rata maupun persentil seperti p(95)), *throughput*, dan *error rate* (Hendayun dkk., 2023). Pengujian ini biasanya menggunakan *tools* berbasis kode seperti Grafana k6, yang memungkinkan perancangan skenario secara bertahap mulai dari *ramp-up*, *peak load*, hingga *ramp-down* serta penetapan *thresholds* untuk mengevaluasi apakah setiap endpoint API memenuhi standar *Quality of Service* (QoS) yang ditetapkan (Khamdani dkk., 2025).

II.11 State of the art

Penelitian terdahulu membahas tentang penelitian-penelitian sebelumnya yang relevan dengan topik yang akan diangkat. Adapun yang dijelaskan lebih lanjut pada Tabel II-1.

Tabel II-2 State of The Art

No	Peneliti (Tahun)	Metode/Teknologi	Fokus Penelitian	Kekuatan	Kekurangan	Pendekatan Evaluasi / Hasil Implementasi
1	Mabotha dkk. (2025)	Dockerized <i>dynamic RESTful</i> API berbasis FastAPI, Nginx <i>reverse proxy</i> , dan PostgreSQL untuk manajemen <i>endpoint</i> IoT	Pembuatan/pengelolaan <i>endpoint</i> dan skema data secara dinamis (<i>runtime</i>) untuk ekosistem IoT tanpa mengubah kode inti	<i>Endpoint</i> & skema baru dapat dibuat saat <i>runtime</i> ; <i>deployment</i> portabel via Docker; tervalidasi pada perangkat nyata (ESP8266 & Raspberry Pi)	<i>Error rate</i> 3,11% dan lonjakan <i>response time</i> (hingga 760 ms) saat beban tinggi akibat keterbatasan <i>resource</i> VPS (2 vCPU/2GB RAM)	<i>Functional testing</i> (<i>success rate</i> 100%) dan load testing dengan JMeter (<i>throughput</i> $\pm 9,83$ req/s, rata-rata <i>response time</i> 88,69 ms); studi kasus sensor ESP8266 dan Raspberry Pi

2	Shaheen dkk. (2025)	<i>Incremental</i> model + <i>iterative</i> model, dengan skor profitabilitas (1–10)	Prioritisasi subsistem awal pengembangan berdasarkan bobot keuntungan (<i>total profit</i>)	Menggabungkan 5 faktor (CCRB, IR, T, B, S) dan bobot <i>stakeholder</i> via matriks <i>importance-influence</i> , hasil lebih objektif	Baru cocok untuk sistem kecil; iterasi ulang menambah waktu; nilai masih subjektif	Simulasi voting & algoritma iteratif (Gambar 6); subsistem profit tertinggi konsisten terpilih sebagai titik awal, belum diuji pada proyek nyata
3	Boima dkk. (2024)	<i>Digital health interventions</i> (SMS, <i>mobile apps</i> , <i>phone calls</i>), WHO <i>classifications</i>	<i>Systematic review</i> dan <i>meta-analysis</i> tentang efektivitas intervensi digital untuk kontrol tekanan darah, perubahan perilaku gaya hidup, dan adherensi obat pada pasien hipertensi di LMICs (Judul: <i>Effectiveness of Digital</i>	Signifikan mengurangi SBP (MD -4.43 mmHg) dan meningkatkan BP <i>control</i> (OR 2.20); mendukung perubahan gaya hidup (rendah garam, aktivitas	Heterogenitas tinggi antar studi (I^2 hingga 97%); terbatas pada LMICs; tidak spesifik pada perangkat seperti Omron dengan <i>Bluetooth</i> atau <i>multiuser dashboard</i> .	<i>Meta-analysis</i> dari 22 RCTs (n=12,892): <i>subgroup analysis</i> berdasarkan jenis intervensi; evaluasi perubahan SBP, DBP, perilaku, dan adherensi menggunakan model <i>random-effect</i> . Kutipan: "Kemampuan konektivitas nirkabel seperti <i>Bluetooth</i>

			<i>Health Interventions for Hypertension Management in US Populations Experiencing Health Disparities).</i>	fisik) dan adherensi obat.		pada tensimeter Omron memfasilitasi pengelolaan data oleh beberapa pengguna sekaligus."
4	Akoh dkk. (2024)	IoT devices (wearables, remote monitoring, smart beds), data exchange protocols	<i>Systematic review tentang use cases IoT di healthcare, termasuk patient monitoring, supply chain management, dan telehealth; menyoroti manfaat seperti efisiensi dan perawatan personalisasi (Judul: Internet of Things (IoT) in Healthcare: A Systematic Review of</i>	Transformasi <i>healthcare</i> melalui <i>monitoring real-time</i> , pengambilan keputusan berbasis bukti, dan pengurangan biaya; contoh seperti <i>wearable trackers</i> untuk vital signs.	Tantangan privasi, keamanan, dan interoperabilitas; tidak fokus pada perangkat spesifik seperti tensimeter Bluetooth atau <i>multiuser</i> ; kurangnya detail implementasi prototipe.	Analisis dari jurnal, <i>proceedings</i> , dan <i>white papers</i> : identifikasi aplikasi kunci dan manfaat; rekomendasi untuk riset lebih lanjut tanpa hasil eksperimen langsung. Kutipan: "Implementasi ini menghadapi tantangan yang serius terkait privasi serta keamanan data yang memerlukan enkripsi yang kuat."

			<i>Use Cases and Benefits).</i>			
5	Hailu dkk. (2024)	Wawancara semi-struktural dengan 20 PCPs, analisis tematik berbasis NASSS framework	Tantangan (<i>data influx, workflow, literasi digital/kesehatan, profitabilitas</i>) dan fasilitator implementasi RPM di <i>primary care</i> untuk BP/glukosa (Judul: <i>Challenges and Facilitators in Implementing Remote Patient Monitoring Programs in Primary Care</i>).	Meningkatkan kualitas kepedulian kronis, <i>engagement</i> pasien, potensi akses <i>equitable</i> dengan <i>policy</i> changes.	Barrier adopsi: <i>handling data</i> berlebih, literasi pasien rendah, <i>workflow</i> tidak <i>manageable</i> .	Analisis interview; hasil saran <i>policy (reimbursement, training), staff roles</i> baru; berguna untuk mengatasi isu implementasi sistem <i>multiuser tensimeter</i> berbasis <i>Bluetooth</i> . Kutipan: "RPM memanfaatkan teknologi sensor biomedis seperti <i>bluetooth</i> untuk mengirim data pasien ke <i>dashboard mobile</i> ."
6	Park dkk. (2024)	Anchor point adjustment pada Bluetooth Low	Penjadwalan multi-koneksi BLE agar QoS tiap koneksi terjaga	Murni di link-layer (tidak perlu ubah host/HCI);	Hanya berlaku topologi single-hop star, belum	Eksperimen pada skenario same/increasing/decreasing traffic; dibandingkan

		Energy, diimplementasikan di controller Zephyr	tanpa melanggar spesifikasi Bluetooth	diuji pada perangkat nyata (Nordic nRF52840); throughput naik hingga 113% dibanding skema CSR/BCM	mendukung multi-hop; parameter ambang update konfirmasi masih perlu tuning lanjutan	dengan skema CSR-based, BCM-based, dan Zephyr default; BLEX unggul pada throughput dan QoS achievement ratio
7	Al-Shareeda dkk. (2023)	<i>Bluetooth Low Energy (BLE) untuk IoT, Review Literatur dengan Taksonomi (Parameter Adjustment dan Collision Avoidance Schemes)</i>	Kajian <i>review</i> mengenai BLE-IoT, termasuk sejarah <i>Bluetooth</i> , aplikasi seperti <i>beacon</i> dan <i>mobile payment</i> , tantangan keamanan serta privasi, dan isu terbuka seperti efisiensi energi dan keamanan di perangkat IoT (Judul:	Konsumsi daya ultra-rendah, pengembangan sederhana, jangkauan jaringan memadai, dan kecepatan transfer data cepat, membuatnya	Masalah keamanan dan privasi signifikan karena stack protokol sederhana, rentan terhadap serangan seperti <i>eavesdropping</i> , serta keterbatasan dalam modifikasi	Pendekatan evaluasi berbasis <i>review</i> literatur dengan taksonomi skema <i>advertiser/scanner</i> ; hasil menyoroti peluang untuk <i>dashboard mobile</i> di <i>healthcare</i> , tetapi menekankan kebutuhan peningkatan keamanan untuk sistem <i>multiuser</i> berbasis BLE. Kutipan:

			<i>Bluetooth Low Energy for Internet of Things: A Systematic Review</i>).	ideal untuk integrasi <i>multiuser</i> di sistem medis seperti tensimeter dengan konektivitas <i>Bluetooth</i> .	<i>driver radio</i> yang membatasi fleksibilitas implementasi.	" <i>Bluetooth Low Energy</i> (BLE) berperan penting sebagai protokol standar pada perangkat medis."
8	Debnath dkk. (2023)	IoT dengan sensor biomedis (suhu, <i>heart rate</i> , BP, SpO2, ECG), <i>microcontroller</i> , hybrid <i>cloud server</i> , GSM untuk alarm/SMS	Desain sistem <i>telemonitoring</i> pasien <i>real-time</i> di ICU/OR dengan update data setiap 45s ke <i>cloud</i> untuk akses dokter global dan <i>alert</i> darurat (Judul: <i>Design and Application of IOT-Based Real-Time</i>	<i>Monitoring real-time</i> efisien, akses <i>worldwide</i> , alarm otomatis mengurangi respons lambat, diuji sukses pada 10 pasien.	Ketergantungan koneksi internet/GSM, potensi isu privasi data, biaya implementasi tinggi untuk setup multimodal.	Pengujian pada pasien nyata >2000 siklus/hari; hasil memuaskan dengan visi ICU <i>remote</i> interaktif, relevan untuk <i>dashboard mobile multiuser</i> tensimeter Omron.

			<i>Patient Telemonitoring System</i>).			
9	Garg dkk. (2023)	Data <i>tagging</i> berdasarkan ICD-10 codes dengan <i>color codes</i> , ARX <i>tool</i> untuk <i>anonymization</i> , <i>role-based access control</i> (RBAC)	Sistem sharing data <i>health-care</i> yang <i>preserve privacy</i> melalui <i>tagging</i> dan <i>anonymization</i> , memastikan akses hanya oleh <i>authorized users</i> untuk <i>improve efficiency</i> dan <i>usability</i> data riset (Judul: <i>A Novel Approach of Privacy-Preserving Healthcare Data Sharing Framework</i>).	Klasifikasi data sensitif efektif, <i>maintain usability</i> untuk riset sambil <i>preserve privacy</i> ; <i>Integrasi tagging</i> , <i>anonymization</i> , dan RBAC <i>seamless</i> .	Proses <i>fabrication</i> rumit; Tantangan <i>integration</i> dengan sistem <i>existing</i> ; Potensi <i>over-anonymization</i> kurangi nilai data.	Analisis metrik <i>anonymized</i> data terhadap <i>attacks</i> ; Model matematis RBAC; Hasil tunjukkan <i>usability</i> tinggi untuk <i>multiuser health monitoring</i> seperti <i>dashboard</i> tensimeter Omron. Kutipan: "Penjadwalan slot waktu dengan prioritas paket real-time memungkinkan sistem untuk mengklasifikasikan informasi sensitif."
10	Hendayun dkk. (2023)	Load testing & stress testing dengan Apache	Analisis performa API layanan pembiayaan kendaraan (6 endpoint)	Root cause kegagalan (connection	Optimasi hanya pada level konfigurasi	3 kali pengujian tiap endpoint dengan 500 concurrent user, mengukur

		JMeter, dipadukan APM Dynatrace, metode blackbox testing	pada beban 500 concurrent users serta optimasi konfigurasi service/server	timeout) berhasil ditemukan lewat tracing Dynatrace; solusi konfigurasi (timeout & load balancing) terbukti menghilangkan failed request	service/web server, bukan pada arsitektur/kode aplikasi; skenario pengujian terbatas hingga 500 user	success rate, loading time, serta CPU/RAM; setelah tuning (connection-timeout 60s + load balancing) tidak ditemukan request gagal
11	Yi dkk. (2022)	IoT, Bluetooth Low Energy (BLE), React Native (cross-platform mobile), Spring Boot (backend), MySQL,	Pengembangan aplikasi mobile cross-platform berbasis React Native untuk sistem monitoring ECG personal, mencakup akuisisi data real-time melalui BLE, visualisasi sinyal ECG,	Satu codebase React Native berjalan di Android dan iOS; mendukung koneksi BLE real-time ke perangkat wearable;	Komunikasi antara mobile dan backend masih HTTP (belum HTTPS); tidak mendukung skenario multiuser data ECG tidak diisolasi per	Implementasi nyata aplikasi React Native yang diuji pada perangkat Android; data ECG dikirim ke backend setiap 10 detik; pesan dari advisor di-fetch setiap 2 detik; aplikasi terintegrasi penuh dengan server Spring Boot yang di-

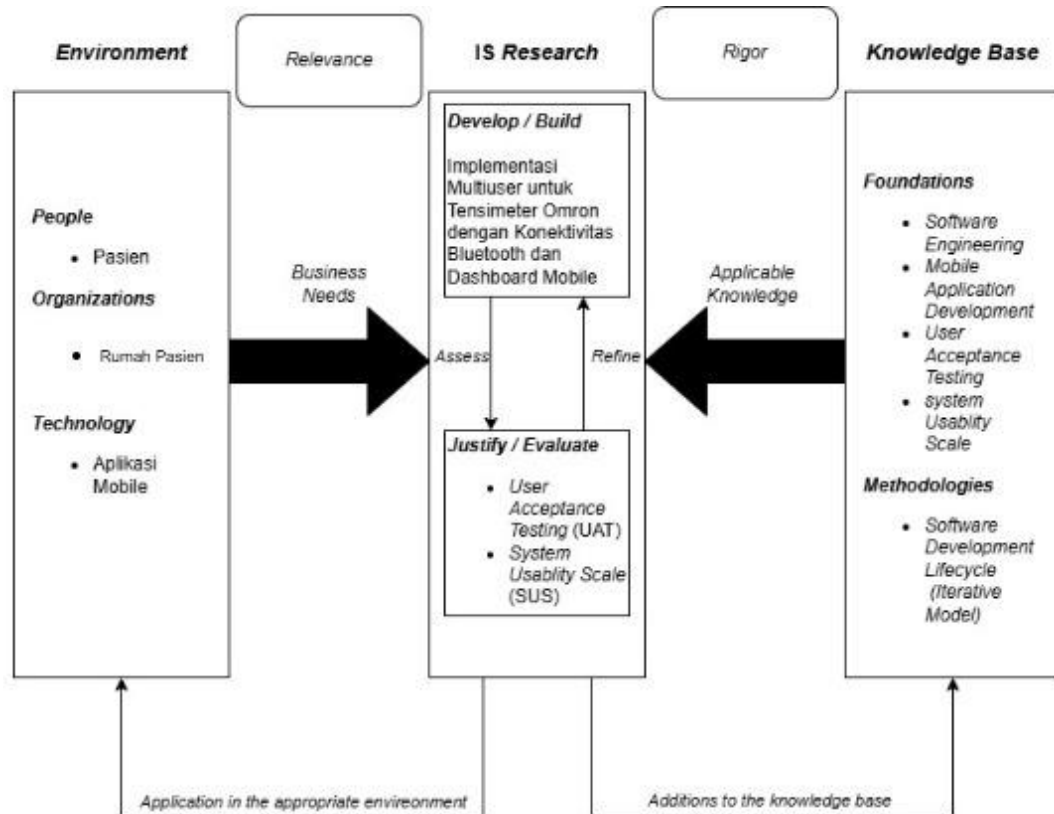
		MongoDB, Apache Kafka	penyimpanan lokal, dan komunikasi dengan health advisor melalui fitur messenger terintegrasi.	menampilkan sinyal ECG langsung dengan refresh rate 16 Hz; terdapat modul networking, akuisisi, visualisasi, manajemen data, dan messenger dalam satu aplikasi yang terintegrasi.	pengguna sehingga tidak cocok untuk penggunaan bersama dalam satu perangkat.	deploy pada Tencent Cloud dan AWS EC2. Relevan sebagai referensi arsitektur mobile BLE untuk sistem monitoring tanda vital, namun belum menangani isolasi data multiuser seperti yang dibutuhkan pada sistem tensimeter Omron yang dikembangkan dalam penelitian ini.
12	Liu dkk. (2021)	IoT, <i>Bluetooth Low Energy</i> (BLE), <i>Modulation</i> dan <i>Encoding</i> di PHY,	Kajian komprehensif BLE: modulasi, encoding, karakteristik jarak, latensi, dan konsumsi daya untuk	Integrasi BLE dengan IoT untuk perangkat berdaya rendah, mendukung	Jarak transmisi pendek, potensi kegagalan koneksi sesekali, skalabilitas	Evaluasi berbasis studi literatur & analisis historis BLE; hasil menunjukkan BLE cocok untuk implementasi multiuser

		<i>Cyclic Redundancy Check (CRC)</i>	mendukung pengembangan IoT	operasi jangka panjang dengan baterai kecil, efisiensi transmisi data real-time	terbatas untuk aplikasi healthcare global	pada perangkat medis (mis. tensimeter) dengan landasan konsep dashboard mobile berbasis BLE untuk transfer data pasien jarak jauh
--	--	--------------------------------------	----------------------------	---	---	---

Berdasarkan literatur yang terkumpul pada Tabel II-1, berbagai penelitian telah membahas sistem pemantauan kesehatan berbasis *Internet of Things* (IoT) dan konektivitas nirkabel untuk pemantauan pasien secara *real-time*. Sistem-sistem tersebut terbukti mampu meningkatkan efisiensi, akurasi, serta aksesibilitas data fisiologis bagi tenaga medis maupun pengguna. Namun, sebagian besar penelitian terdahulu masih berfokus pada skenario penggunaan individu (*single-user*) atau belum mengintegrasikan mekanisme autentikasi terpusat dan isolasi data antar pengguna secara aman. Oleh karena itu, penelitian ini dirancang untuk mengembangkan sistem pemantauan kesehatan berbasis *multi-user* yang terintegrasi dengan *dashboard mobile* dan *backend* guna menjamin keamanan, isolasi data per profil pengguna, serta keterberlanjutan alur kerja sistem secara optimal.

BAB III METODE PENYELESAIAN MASALAH

III.1 Metode Konseptual



Gambar III-1 Model Konseptual

Pada sisi *Environment*, digambarkan konteks nyata tempat penelitian ini berangkat, yang terdiri dari unsur *People* (pasien), *Organizations* (Rumah Pasien), serta *Technology* (aplikasi mobile) yang sudah ada. Dari lingkungan inilah muncul *Business Needs*, yaitu kebutuhan bisnis atau permasalahan nyata di lapangan, kebutuhan akan implementasi multiuser untuk tensimeter Omron dengan konektivitas *Bluetooth* dan *dashboard mobile*. Kebutuhan ini kemudian dinilai (*Assess*) dan menjadi dasar bagi tahap *Develop/Build*, yaitu proses membangun solusi (aplikasi/sistem) yang relevan dengan kebutuhan tersebut.

Pada *IS Research* menjadi inti dari model ini, yang terdiri dari dua siklus utama: *Develop/Build* (membangun sistem) dan *Justify/Evaluate* (menguji dan

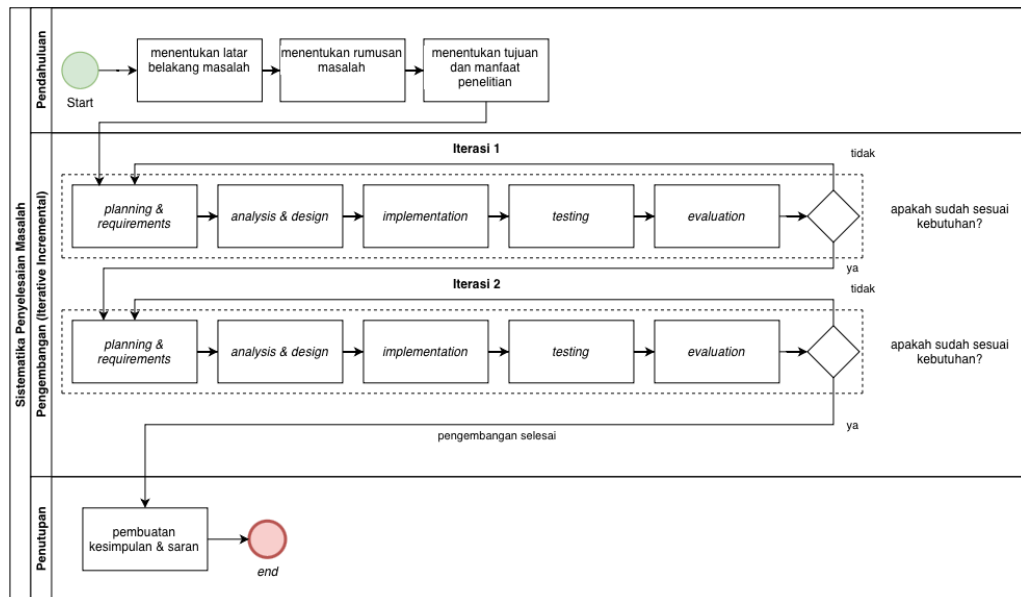
mengevaluasi sistem yang telah dibangun). Dalam kasus ini, evaluasi dilakukan melalui *User Acceptance Testing* (UAT) dan *System Usability Scale* (SUS), untuk memastikan bahwa sistem yang dikembangkan dapat diterima pengguna dan memiliki tingkat usabilitas yang baik. Hasil evaluasi ini kemudian digunakan untuk Refine atau menyempurnakan kembali desain sistem, menciptakan siklus iteratif antara pembangunan dan evaluasi.

Di sisi kanan, *Knowledge Base* menyediakan fondasi teoretis dan metodologis yang mendukung proses penelitian, mencakup *Foundations* seperti *Software Engineering*, *Mobile Application Development*, *User Acceptance Testing*, dan *System Usability Scale*, serta *Methodologies* seperti *Software Development Lifecycle* dengan pendekatan iteratif. Pengetahuan inilah yang memberikan *Applicable Knowledge*, yaitu landasan ilmiah yang digunakan untuk membangun dan mengevaluasi solusi pada tahap *IS Research*.

Terakhir, model ini ditutup dengan dua jalur umpan balik pada bagian bawah, yaitu *Application in the appropriate environment*, yang menunjukkan bahwa hasil penelitian (sistem yang dikembangkan) diterapkan kembali ke lingkungan aslinya untuk menyelesaikan masalah nyata, dan *Additions to the knowledge base*, yang menunjukkan bahwa temuan atau kontribusi baru dari penelitian ini turut memperkaya basis pengetahuan yang sudah ada. Dengan demikian, model ini menggambarkan siklus penelitian yang bersifat relevan secara praktis (*relevance*) sekaligus kuat secara ilmiah, di mana kebutuhan nyata di lapangan menjadi pemicu penelitian, dan hasilnya memberikan kontribusi baik bagi praktik maupun bagi ilmu pengetahuan.

III.2 Sistematika Penyelesaian Masalah

Sistematika penyelesaian langkah-langkah perencanaan masalah seperti yang terlihat pada gambar berikut:



Gambar III-2 Sistematika Penyelesaian Masalah

III.2.1 Identifikasi

Pada tahap identifikasi peneliti menentukan masalah berdasarkan dengan latar belakang penelitian yang telah dilakukan. Rumusan masalah yang ditetapkan akan menjelaskan tujuan dari Implementasi Sistem *Multiuser* Untuk Tensimeter Omron Dengan Konektivitas *Bluetooth* Dan *Dashboard Mobile* yang akan menghasilkan sebuah solusi. Solusi yang telah ditemukan, akan ditentukan batasan masalah dari penelitian

III.2.2 Tahap Pengembangan

Tahap pengembangan ini akan melalui 4 fase yang mengacu pada metode pengembangan aplikasi dengan menggunakan metode *Iterative Incremental* yaitu:

1. *Requirement and Planning*

Pada tahap *Requirement and Planning* ini dilakukan proses perencanaan serta analisis kebutuhan pengguna terhadap aplikasi yang akan dikembangkan. Tahap ini menggambarkan fitur dan fungsi yang dibutuhkan pada aplikasi sesuai dengan kebutuhan pengguna.

2. *Analysis and Design*.

Pada tahap *Analysis and Design* ini dilakukan proses perancangan sistem sesuai dengan kebutuhan pengguna yang telah ditentukan sebelumnya pada

tahap *Requirement and Planning*. Tahap ini akan menghasilkan sebuah artefak diagram UML.

3. *Implementation*

Pada tahap *Implementation* ini dilakukan proses implementasi sebuah program atau *code* berdasarkan rancangan desain yang telah kita buat pada tahap *Analysis and Design*.

4. *Evaluation*

Pada tahap *Evaluation* ini dilakukan evaluasi terhadap pengguna untuk aplikasi yang telah kita lakukan *Implementation* pada tahap sebelumnya untuk mengetahui kelebihan serta kekurangan yang terdapat pada aplikasi.

III.2.3 Tahap Kesimpulan dan Saran

Pada tahap ini, seluruh project yang telah kita rancang akan di evaluasi secara menyeluruh serta mencari solusi berdasarkan *feedback* yang kita dapatkan dari *user* pada tahap *Evaluation*, kemudian mengambil kesimpulan serta saran dari *project* yang sudah dikembangkan.

III.3 Alasan Pemilihan Metode

Pemilihan metode pengembangan sistem yang tepat merupakan langkah krusial untuk memastikan bahwa solusi yang dihasilkan mampu menjawab permasalahan secara efektif dan adaptif terhadap perubahan kebutuhan selama proses pengembangan. Sebagaimana telah diuraikan pada BAB II.1.8, terdapat beberapa pendekatan *System Development Life Cycle* (SDLC) yang dapat diterapkan dalam pengembangan sistem, mulai dari kategori *heavyweight methodologies* seperti Waterfall dan Model V, hingga pendekatan yang lebih adaptif seperti model Iteratif.

Tabel III-1 *Perbandingan Pendekatan Metodologi Penelitian dan Pengembangan Sistem RPM*

No	Metode	Keluaran Utama	Kekurangan	Keseuaian
1	<i>Waterfall</i>	Struktur jelas, dokumentasi lengkap, cocok untuk	Tidak fleksibel terhadap perubahan;	Tidak sesuai, karena sifat eksploratif

		kebutuhan yang sudah terdefinisi	kesalahan baru terdeteksi di akhir	integrasi BLE membutuhkan fleksibilitas
2	<i>Model V</i>	Deteksi cacat dini, pengujian paralel di setiap tahap	Masih sekuensial, biaya tinggi jika ada revisi, kurang cocok untuk kebutuhan tidak pasti	Tidak sesuai, karena Perubahan kebutuhan selama pengujian perangkat sulit diakomodasi
3	<i>Iteratif Incremental</i>	Fleksibel terhadap perubahan, umpan balik di setiap siklus, risiko teridentifikasi sejak awal	Memerlukan manajemen iterasi yang baik, potensi <i>scope creep</i>	Sesuai, karena mendukung siklus berulang yang dibutuhkan untuk integrasi BLE eksploratif.

Dalam perancangan Implementasi Sistem Multiuser Untuk Tensimeter Omron Dengan Konektivitas Bluetooth Dan *Dashboard Mobile*, peneliti menggunakan metode *Iterative Incremental* karena lebih sesuai dengan penelitian *project* dan berdasarkan perbandingan metode yang didapatkan pada Tabel III-1

III.4 Metode Pengumpulan data

Pengumpulan data dalam penelitian ini dilakukan melalui metode wawancara dengan responden yang relevan, yaitu staf klinik kesehatan dan pengguna yang akan menggunakan Tensimeter Omron. Wawancara dirancang untuk menggali informasi mendalam mengenai permasalahan yang dialami dalam pemantauan tekanan darah menggunakan tensimeter Omron, termasuk tantangan dalam manajemen data *multiuser*, konektivitas *Bluetooth*, dan integrasi dengan *dashboard mobile* setiap pasien. Data yang diperoleh dari wawancara akan dicatat serta

dilakukan analisis untuk mengidentifikasi pola permasalahan utama. Selanjutnya, data tersebut akan menjadi dasar untuk memastikan bahwa solusi yang dikembangkan sesuai dengan kebutuhan nyata pengguna dan dapat memberikan kontribusi efektif dalam mengatasi isu-isu yang dihadapi.

III.5 Metode Evaluasi

Pada penelitian ini secara spesifik, metode evaluasi mencakup dua pendekatan utama:

1. *Blackbox Testing*

Blackbox Testing merupakan sebuah metode untuk melakukan pengujian terhadap fungsional atau kegunaan sebuah aplikasi. *Blackbox Testing* cukup untuk meninjau *input* dan *output* sebuah sistem tersebut tanpa pengetahuan tentang internal programnya

2. *User Acceptance Testing (UAT)*

User Acceptance Testing (UAT) adalah tahap pengujian perangkat lunak yang dilakukan oleh pengguna sebelum sebuah sistem resmi digunakan secara operasional. Tujuan utama UAT adalah memastikan bahwa aplikasi atau sistem yang dibangun benar-benar memenuhi kebutuhan bisnis dan sesuai dengan spesifikasi yang telah disepakati di awal proyek.

3. *Load Test (Grafna K6)*

Load Test adalah jenis pengujian yang bertujuan untuk mengetahui batas kemampuan atau ketahanan sebuah sistem ketika diberikan beban kerja yang sangat tinggi, bahkan melebihi kapasitas normal yang biasa ditangani. Pengujian ini dilakukan dengan cara memberikan tekanan berupa jumlah pengguna yang sangat banyak, volume data yang besar, atau permintaan (*request*) yang terus-menerus dalam waktu singkat, untuk melihat bagaimana sistem merespons kondisi ekstrem tersebut menggunakan Grafna K6, salah satu tools *open-source* yang paling populer digunakan untuk melakukan pengujian performa (*performance testing*), termasuk *stress test*, *load test*, *spike test*, dan *soak test*..

BAB IV HASIL PENELITIAN

IV.1 Planning and Requirements

Dalam menyelesaikan masalah pada penelitian ini, penulis memulai dengan fase awal dari metodologi yang telah dirancang yaitu mengidentifikasi suatu masalah yang sesuai dengan kebutuhan pengguna. Pada tahap ini, penulis merumuskan identifikasi kebutuhan berdasarkan hasil wawancara yang telah dilakukan dengan pihak terkait. Fase ini bertujuan untuk menggali informasi wawasan dari berbagai sumber yang akan menjadi landasan untuk merancang implementasi sistem *multiuser* untuk tensimeter omron.

IV.1.1 Analisis Sistem Eksisting

Sistem eksisting merupakan aplikasi mobile berbasis React Native yang dirancang untuk memantau tekanan darah dan tanda-tanda vital pasien secara *real-time* melalui koneksi *Bluetooth Low Energy* (BLE) dengan perangkat tensimeter. Aplikasi ini dibangun menggunakan arsitektur berbasis navigasi tab dengan empat halaman utama, yaitu halaman beranda, halaman perangkat, halaman riwayat data, dan halaman profil pengguna. Pada halaman beranda, pengguna dapat melihat pembacaan tekanan darah terbaru, informasi *health insight*, tanda vital terkini, serta ringkasan data pengukuran. Halaman perangkat memfasilitasi proses koneksi dan pengelolaan perangkat tensimeter melalui BLE, sementara halaman riwayat data menyajikan rekam jejak pengukuran dalam bentuk grafik dan tabel.

Dari sisi autentikasi, sistem eksisting telah memiliki mekanisme login dan registrasi yang memanfaatkan *token* berbasis JWT (*JSON Web Token*). Namun, fitur autentikasi ini bersifat opsional dan tidak terintegrasi ke dalam alur navigasi utama aplikasi. Proses login dan registrasi tidak ditampilkan sebagai halaman mandiri yang muncul saat aplikasi pertama kali dibuka, melainkan diakses melalui jendela modal yang dipanggil dari halaman profil. Dengan demikian, pengguna dapat langsung mengakses seluruh fitur aplikasi termasuk halaman beranda, perangkat, dan riwayat data tanpa harus melakukan autentikasi terlebih dahulu. Selain itu, tidak terdapat fitur lupa *password* yang memungkinkan pengguna mengatur ulang kata sandi apabila lupa.

Sistem eksisting juga tidak menerapkan mekanisme proteksi rute (*route protection*) atau pengalihan otomatis (*redirect*). Artinya, tidak ada pembatasan akses berdasarkan status autentikasi pengguna. Seluruh halaman pada aplikasi dapat diakses secara bebas tanpa memerlukan proses login, sehingga menimbulkan potensi risiko keamanan terkait akses data kesehatan yang bersifat pribadi.

Pengelolaan data pengguna pada sistem eksisting dirancang untuk mendukung satu pengguna tunggal (*single-user*) per akun. Pada halaman profil, sistem menampilkan informasi akun secara statis berupa nama lengkap dan nomor telepon. Tidak terdapat mekanisme pemilihan profil yang memungkinkan satu akun mengelola data kesehatan untuk beberapa individu secara terpisah, misalnya anggota keluarga. Seluruh data pengukuran tekanan darah langsung terikat pada satu identitas pengguna tanpa adanya konsep *multi-profil*.

Dari aspek arsitektur perangkat lunak, pengelolaan *state* autentikasi pada sistem eksisting masih bersifat sederhana dengan menyimpan informasi *token* dan data pengguna ke penyimpanan lokal perangkat. Komunikasi dengan server *backend* dilakukan secara langsung di dalam logika pengelolaan state tanpa adanya abstraksi lapisan layanan (*service layer*) yang terpisah, sehingga logika pemanggilan API tersebar di berbagai bagian aplikasi. Konfigurasi aplikasi juga bersifat statis, hanya memuat satu alamat server yang ditulis secara tetap tanpa mekanisme penyimpanan konfigurasi secara dinamis atau parameter tambahan seperti pengaturan koneksi perangkat BLE.

Berdasarkan analisis terhadap sistem eksisting tersebut, teridentifikasi beberapa kebutuhan pengembangan lebih lanjut untuk mendukung skenario multiuser, keamanan akses, serta pengalaman pengguna yang lebih baik. Kebutuhan-kebutuhan ini dirangkum dalam tabel gap analisis pada sub bab berikutnya.

IV.1.2 Gap Analisis

Berdasarkan temuan tersebut telah dipaparkan pada subbab IV.1, dilakukan analisis gap untuk memetakan kesenjangan antara kondisi *existing* penggunaan tensimeter Omron dengan kebutuhan sistem yang akan dikembangkan. Analisis gap ini bertujuan untuk mempertegas urgensi pengembangan sistem serta menjadi dasar

perumusan solusi teknis yang akan diimplementasikan berdasarkan hasil aplikasi yang sebelumnya telah dibuat. Hasis analisis gap disajikan pada Tabel IV-1 Berikut

Tabel IV-1 Gap Analisis

No	Kebutuhan atau Tujuan	Kondisi Existing	Fulfillment			Solusi
			N	P	F	
1.	Halaman login dan register secara terpisah	Saat user membuka aplikasi, aplikasi tidak langsung menampilkan <i>login</i> atau <i>register</i> , melainkan dashboard utama.	v			Membuat halaman <i>Login</i> serta <i>Register</i> tampil langsung saat aplikasi terbuka. <i>Login</i> mendukung menggunakan <i>email</i> maupun <i>password</i>
2.	Halaman lupa <i>password</i>	Tidak ada halaman lupa <i>password</i> sama sekali	v			Membuat halaman lupa <i>password</i> dengan UI lengkap untuk melakukan <i>reset password</i> via <i>Email</i> /Nomor HP
3.	<i>Route Protection (Auth Guard)</i> otomatis	Tidak ada mekanisme redirect otomatis. <i>User</i> bisa langsung akses semua tab tanpa <i>login</i>	v			Membuat proteksi bahwa user harus melakukan <i>login/register</i> terlebih dahulu sebelum mengakses dashboard utama
4.	Halaman pemilihan profil (<i>Profile Selection</i>)	Tidak ada konsep <i>multiuser</i>	v			Membuat pembuatan <i>profile</i> setelah <i>user</i> melakukan <i>login</i> .

5.	Struktur <i>AuthProvider</i> yang mendukung multiuser	<i>AuthProvider</i> sudah menyimpan <i>userToken</i> dan user (<i>single-user</i>) dengan JWT, tetapi belum mendukung <i>multi-profil</i> .		v		Membuat <i>AuthProvider</i> baru dengan state yang mendukung implementasi multiuser
----	---	---	--	---	--	---

IV.1.3 Pengumpulan Data dan Wawancara

Wawancara dilakukan kepada seorang dokter yang memiliki pengalaman dalam pemantauan pasien hipertensi, dengan tujuan untuk memperoleh perspektif klinis mengenai kebutuhan sistem pemantauan tekanan darah pada pasien jarak jauh serta tantangan nyata yang dihadapi dalam praktik medis sehari-hari. Wawancara selanjutnya dilakukan kepada pasien penderita hipertensi, dengan tujuan untuk memperoleh perspektif dari pasien hipertensi dalam menjalani pemeriksaan tekanan darah sehari-hari serta tantangan nyata apa yang dihadapi pasien hipertensi sehari-hari dalam pemantauan pemeriksaan tekanan darah tersebut.

Sebelum wawancara, peneliti Menyusun panduan pertanyaan yang mencakup beberapa topik utama, yaitu:

1. Kebiasaan dan tantangan dalam pemantauan tekanan darah pasien secara rutin.
2. Kebutuhan sistem *Remote Patient Monitoring* (RPM) berbasis teknologi.
3. Kebutuhan data profil pasien yang diperlukan untuk mendukung diagnosis
4. Aspek keamanan dan privasi data keehatan yang menjadi perhatian dalam konteks penggunaan Bersama

Wawancara dilakukan secara daring dan hasilnya dicatat serta dianalisis untuk mengidentifikasi pola kebutuhan yang akan menjadi dasar perancangan sistem.

Wawancara pertama dilakukan kepada seorang dokter yang berpengalaman dalam pemantauan pasien hipertensi. Dari hasil wawancara dengan ahli, diperoleh beberapa temuan utama. Pertama, dokter menyatakan bahwa RPM sangat

dibutuhkan untuk memantau kondisi pasien hipertensi secara berkala tanpa harus datang ke fasilitas kesehatan, karena data historis tekanan darah yang tersimpan secara digital sangat membantu dalam menganalisis tren kondisi pasien dan menentukan langkah medis yang lebih tepat sasaran. Kedua, ahli menekankan pentingnya sistem yang mampu memisahkan data pengukuran antar individu secara spesifik, mengingat dalam satu rumah tangga sering terdapat lebih dari satu anggota keluarga yang perlu dipantau menggunakan satu perangkat yang sama. Ketiga, terkait data profil yang diperlukan secara klinis, ahli mengidentifikasi bahwa sistem minimal harus menyimpan Nama, Usia, Jenis Kelamin, Tinggi Badan, dan Berat Badan sebagai konteks untuk menginterpretasikan hasil pengukuran secara lebih akurat. Keempat, mengingat sensitivitas data kesehatan, ahli menekankan perlunya mekanisme autentikasi berbasis password untuk menjamin keamanan dan privasi data setiap pengguna.

Wawancara kedua dilakukan kepada pasien penderita hipertensi selaku pengguna akhir sistem yang akan dikembangkan. Dari hasil wawancara dengan pengguna, diperoleh beberapa temuan penting. Pertama, pengguna menyatakan kesulitan ketika harus bergantian menggunakan satu perangkat tensimeter dengan anggota keluarga lain, terutama karena tidak ada cara mudah untuk memisahkan data hasil pengukuran masing-masing orang untuk sekarang data yang telah di periksa hanya dicatat secara manual. Kedua, pengguna mengungkapkan bahwa mereka tidak pernah menggunakan bahkan tidak mengetahui keberadaan aplikasi resmi OMRON Connect, sehingga selama ini data pengukuran tidak pernah tersimpan secara digital dan hanya dicatat secara manual atau tidak dicatat sama sekali. Ketiga, pengguna menyatakan kekhawatiran bahwa data tekanan darahnya dapat tercampur dengan data anggota keluarga lain atau bahkan hilang, yang dinilai dapat membingungkan dan berpotensi berbahaya apabila digunakan sebagai acuan medis. Keempat, pengguna mengungkapkan kebutuhan untuk dapat melihat riwayat data pengukuran di smartphone secara mudah, sehingga dapat dipantau kapan saja dan dapat dibagikan kepada dokter saat konsultasi. Kelima, pengguna menyatakan kebutuhan terhadap fitur notifikasi atau peringatan apabila hasil pengukuran tekanan darah menunjukkan nilai yang melebihi batas normal, agar dapat segera mengambil tindakan. Keenam, pengguna secara konsisten menekankan bahwa antarmuka

aplikasi harus sederhana, dan tidak memerlukan pelatihan khusus untuk dapat digunakan, mengingat sebagian besar pengguna adalah kelompok usia dewasa hingga lansia yang tidak terbiasa dengan aplikasi teknologi yang kompleks.

IV.1.4 Analisis Kebutuhan Sistem

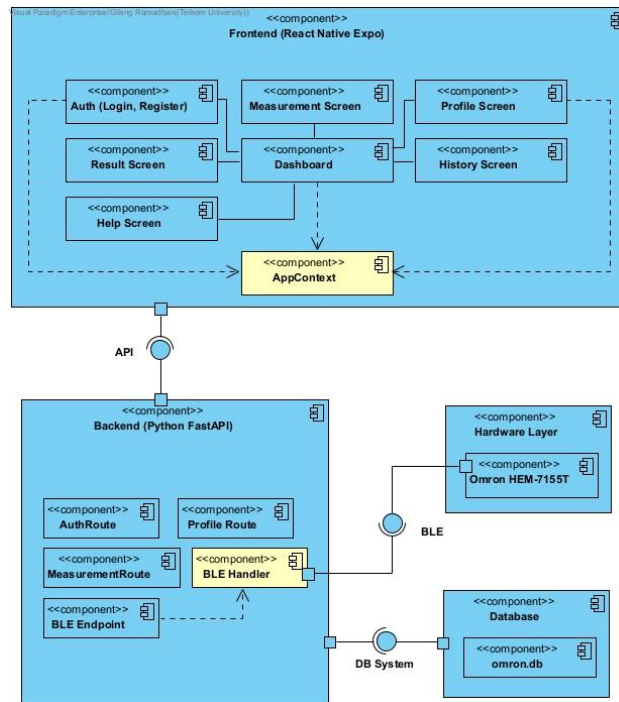
Berdasarkan gap analisis tersebut, peneliti dapat menyusun table fungsionalitas sebagai berikut:

Tabel IV-2 Kebutuhan Fungsional

No	Kebutuhan Fungsional	Lapisan	Prioritas
KF-1	Sistem dapat membaca data tekanan darah (SYS, DIA, BPM, IHB, MOV) serta Prediksi Gula Darah berdasarkan <i>profile</i> yang telah di pilih	Hardware/BLE	Tinggi
KF-2	Sistem dapat membedakan data <i>User 1</i> dan <i>User 2</i> pada alat	Hardware/BLE	Tinggi
KF-3	<i>User</i> dapat mendaftar akun (nama, email, no. HP, password)	Backend	Tinggi
KF-4	<i>User</i> dapat <i>login</i> dengan <i>email</i> + password	Backend	Tinggi
KF-5	User dapat membuat profil anggota keluarga (nama, umur, <i>gender</i> , TB, BB)	Backend	Tinggi
KF-6	User dapat melihat daftar profil yang dimiliki	Backend	Tinggi
KF-7	User dapat menghapus profil	Backend	Sedang
KF-8	Data pengukuran tersimpan dan terasosiasi ke profil tertentu	Backend	Tinggi
KF-9	Dapat mengambil riwayat pengukuran per profil	Backend	Tinggi
KF-10	Dapat mengambil pengukuran terbaru per profil	Backend	Tinggi
KF-11	Antarmuka login dan registrasi yang intuitif	Mobile	Tinggi
KF-12	<i>Data user</i> setiap profile terjaga dan terenkripsi dengan aman	Mobile	Tinggi

IV.1.5 Arsitektur Model Konseptual

Sistem Monitoring Tekanan Darah Multiuser terdiri atas empat lapisan utama, yaitu Frontend, Backend, Database, dan Hardware Layer, yang saling berinteraksi untuk menjalankan seluruh fungsionalitas sistem. Keempat lapisan ini dirancang untuk memisahkan tanggung jawab antara antarmuka pengguna, logika bisnis, penyimpanan data, dan komunikasi perangkat keras secara modular



Gambar IV-1 Component Diagram

1. Frontend (React Native Expo)

Frontend merupakan bagian sistem yang berhubungan langsung dengan pengguna dan bertugas menyajikan antarmuka visual untuk seluruh fitur monitoring tekanan darah. Lapisan ini dibangun menggunakan React Native Expo sehingga dapat berjalan pada platform *mobile*. Di dalam *Frontend* terdapat beberapa komponen utama yang masing-masing memiliki peran spesifik dalam mendukung alur kerja pengguna.

Auth Screens menangani proses autentikasi, yaitu Login dan Register. Melalui komponen ini pengguna dapat mendaftarkan akun baru menggunakan nama, email, nomor HP, dan password, serta melakukan login untuk mengakses fitur-fitur utama aplikasi. Setelah berhasil login, pengguna diarahkan ke *Profile Screens* yang berfungsi sebagai pengelola

profil anggota keluarga. Karena sistem ini mendukung multiuser, satu akun dapat memiliki beberapa profil seperti Ayah, Ibu, atau Anak, masing-masing dengan data usia, jenis kelamin, tinggi badan, dan berat badan yang berbeda.

Dashboard Home merupakan halaman utama yang menampilkan ringkasan status pengguna, profil aktif, serta perangkat BLE yang sedang terhubung. Dari Dashboard, pengguna dapat mengakses Measure Screen untuk melakukan proses pengambilan data pengukuran tekanan darah dari perangkat Omron melalui Bluetooth Low Energy, termasuk melakukan scan device dan mengambil data tensi terbaru secara otomatis. Setelah pengukuran selesai, Result Screen menampilkan hasil pengukuran terakhir yang mencakup nilai sistolik (SYS), diastolik (DIA), denyut jantung (BPM), indikator detak jantung tidak teratur (IHB), dan indikator pergerakan tubuh (MOV) secara visual dan mudah dipahami.

History Screen menyediakan riwayat seluruh pengukuran per profil dalam bentuk tabel dan grafik tren tekanan darah, sehingga pengguna dapat memantau perkembangan kesehatan dari waktu ke waktu. Help Screen berisi panduan batas tanda vital.

Di bagian internal Frontend, terdapat *AppContext* yang berperan sebagai pengelola state global aplikasi seperti *session login*, profil yang sedang aktif, dan *device* BLE yang dipilih. Seluruh komunikasi antara *Frontend* dan *Backend* dilakukan melalui *API Service*, yaitu *HTTP client* yang mengirimkan request ke Backend melalui protokol RESTful API dengan autentikasi *JWT Token* untuk menjamin keamanan pertukaran data.

2. Backend (Python FastAPI)

Backend berperan sebagai pusat logika bisnis yang memproses seluruh permintaan dari Frontend. Lapisan ini dibangun menggunakan Python dengan framework FastAPI yang mendukung pemrosesan asynchronous dan dokumentasi API secara otomatis. Di dalam Backend terdapat beberapa komponen yang terbagi menjadi tiga kelompok, yaitu Router, Service pendukung, dan BLE Handler.

Auth Router menangani *endpoint* `/api/auth` yang meliputi proses registrasi dan login pengguna. Pada saat registrasi, sistem melakukan validasi untuk memastikan bahwa email dan nomor HP belum terdaftar sebelumnya, kemudian menyimpan *password* dalam bentuk *hash* menggunakan algoritma *bcrypt*. Pada saat login, sistem memverifikasi kredensial pengguna dan menghasilkan *JWT token* yang berlaku selama tujuh hari untuk autentikasi request selanjutnya. Seluruh proses autentikasi ini didukung oleh komponen *JWT Auth* yang menggunakan *library* *python-jose* untuk pembuatan token dan *passlib* untuk *hashing* password.

Profiles Router menangani *endpoint* `/api/profiles` yang mengelola operasi CRUD pada data profil anggota keluarga. Pengguna dapat menambahkan profil baru dengan menyertakan data nama, usia, jenis kelamin, tinggi badan, berat badan, dan foto profil. Foto yang diunggah disimpan di direktori `uploads` pada server dan dapat diakses melalui URL statis. Selain itu, pengguna juga dapat melihat daftar seluruh profil yang dimiliki serta menghapus profil yang tidak lagi diperlukan.

Measurements Router menangani *endpoint* `/api/measurements` yang bertanggung jawab untuk menyimpan dan mengambil data hasil pengukuran tekanan darah. Setiap data pengukuran dikaitkan dengan user dan profil tertentu, sehingga riwayat pengukuran dapat dikelola secara terpisah untuk setiap anggota keluarga. Router ini menyediakan *endpoint* untuk menyimpan pengukuran baru, mengambil seluruh riwayat pengukuran per profil, serta mengambil data pengukuran terbaru yang ditampilkan pada Dashboard.

BLE Endpoints menangani *endpoint* `/scan` untuk memindai perangkat *Bluetooth Low Energy* di sekitar dan *endpoint* `/latest-bp-records` untuk membaca data pengukuran terbaru langsung dari perangkat Omron. Kedua *endpoint* ini bergantung pada komponen *BLE Handler* yang dibangun menggunakan *library* *Bleak* untuk berkomunikasi dengan perangkat melalui protokol BLE GATT. *BLE Handler* juga bergantung pada *Device Driver Loader* yang bertugas memuat driver spesifik untuk setiap model

perangkat Omron dari direktori *deviceSpecific*, sehingga sistem dapat mendukung berbagai jenis tensimeter Omron.

Seluruh Router bergantung pada dua komponen pendukung utama, yaitu Schemas dan Models. Schemas menggunakan Pydantic untuk mendefinisikan struktur validasi data *request* dan *response*, memastikan bahwa setiap data yang masuk dan keluar dari sistem memiliki format yang benar. Models menggunakan SQLAlchemy ORM untuk mendefinisikan struktur tabel database dan mengelola interaksi dengan lapisan Database.

3. Database (SQLite)

Database menjadi pusat penyimpanan seluruh informasi penting dalam sistem. Sistem database yang digunakan adalah SQLite karena ringan, tidak memerlukan server terpisah, dan cocok untuk skala aplikasi monitoring kesehatan personal. Akses ke database dilakukan melalui SQLAlchemy ORM yang menyediakan abstraksi objek-relasional sehingga operasi database dapat dilakukan menggunakan model Python tanpa perlu menulis *query* SQL secara langsung.

Struktur database terdiri dari tiga tabel utama. Tabel *users* menyimpan data akun pengguna yang meliputi id, nama, email, dan password yang telah di-hash. Tabel *profiles* menyimpan data profil anggota keluarga yang terkait dengan satu user, mencakup id, id_user sebagai foreign key, nama profil, usia, jenis kelamin, tinggi badan, berat badan, dan path foto profil. Tabel *measurements* menyimpan data hasil pengukuran tekanan darah yang terkait dengan user dan profil tertentu, meliputi id, id_user, id_profile, nilai sistolik (SYS), diastolik (DIA), denyut jantung (BPM), indikator detak jantung tidak teratur (IHB), indikator pergerakan tubuh (MOV), serta timestamp pengukuran. Ketiga tabel ini saling berelasi melalui foreign key, di mana satu user dapat memiliki banyak profil dan setiap profil dapat memiliki banyak data pengukuran.

4. Hardware Layer

Hardware Layer terdiri dari perangkat tensimeter *digital* Omron HEM-7155T yang berfungsi sebagai alat pengukur tekanan darah. Perangkat ini berkomunikasi dengan *Backend* melalui protokol *Bluetooth Low Energy*

(BLE) menggunakan GATT (*Generic Attribute Profile*) dengan UUID service khusus Omron yaitu `ecbe3980-c9a2-11e1-b1bd-0002a5d5c51b`. Proses komunikasi meliputi scanning perangkat, koneksi, pairing, pembacaan data EEPROM, dan sinkronisasi waktu.

Omron HEM-7155T memiliki fitur *User Switch* yang mendukung dua slot penyimpanan pengguna (Slot 1 dan Slot 2), sehingga perangkat ini secara native mendukung konsep multiuser yang sejalan dengan arsitektur sistem *backend*. Ketika Backend membaca data dari perangkat, parameter `user_index` digunakan untuk menentukan slot pengguna mana yang akan dibaca, memungkinkan pengambilan data pengukuran yang terpisah untuk masing-masing pengguna perangkat.

IV.2 Analysis and Design

Arsitektur backend sistem implementasi multiuser pada aplikasi yang telah dibuat bernama Rumah Berpulang yang dirancang menggunakan *framework* FastAPI berbasis Python, yang dipilih karena dua keunggulan utama: dukungan penuh terhadap operasi asinkron (*async/await*) yang esensial untuk komunikasi BLE yang bersifat *non-blocking*, serta kemampuan *auto-generate* dokumentasi API interaktif melalui Swagger UI yang mempercepat proses pengujian dan validasi endpoint. Library komunikasi BLE menggunakan Bleak yang bersifat *cross-platform* dan kompatibel dengan paradigma asinkron FastAPI.

Desain endpoint REST API terbagi menjadi dua kelompok utama. Kelompok pertama adalah tiga endpoint komunikasi BLE: GET `/scan` untuk memindai perangkat BLE di sekitar, POST `/connect-and-read` untuk membaca seluruh data pengukuran dari perangkat, dan POST `/latest-bp-records` untuk membaca data pengukuran terbaru saja. Kelompok kedua adalah delapan endpoint multiuser yang terbagi dalam tiga domain fungsional: autentikasi (POST `/api/auth/register` dan POST `/api/auth/login`), manajemen profil (POST `/api/profiles`, GET `/api/profiles/{id_user}`, DELETE `/api/profiles/{id_profile}`), dan manajemen pengukuran (POST `/api/measurements`, GET `/api/measurements/{id_profile}`, GET `/api/measurements/{id_profile}/latest`). Seluruh endpoint profil dan pengukuran

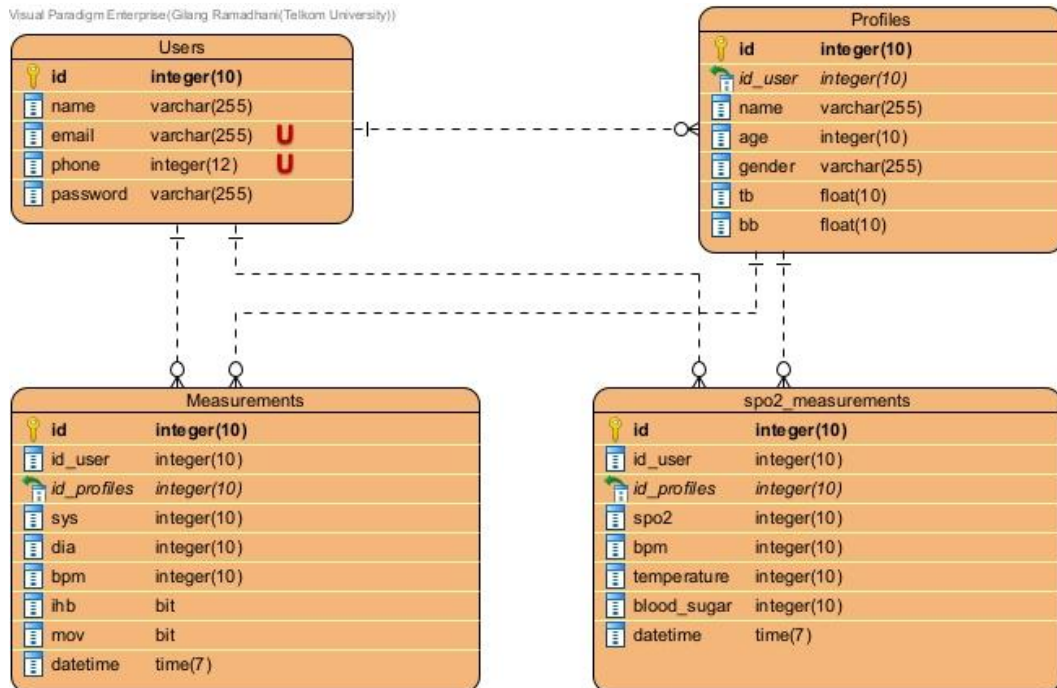
diamankan menggunakan JWT (JSON Web Token) untuk memastikan hanya pengguna yang terautentikasi yang dapat mengakses data.

Untuk endpoint koneksi BLE, dirancang model input `ConnectAndReadInput` dengan parameter `mac_address`, `device_name`, `new_records_only`, `sync_time`, `pairing`, dan `user_index`. Arsitektur *driver loader pattern* juga diterapkan, di mana setiap model perangkat Omron memiliki file driver tersendiri dalam folder `deviceSpecific/`, memungkinkan penambahan dukungan perangkat baru tanpa mengubah kode inti sistem.

Tabel IV-3 Endpoint dan Fungsi

Method	Endpoint	Fungsi	Auth
GET	/scan	Memindai perangkat BLE	-
POST	/connect-and-read	Baca semua data dari alat	-
POST	/latest-bp-records	Baca data terbaru saja	-
POST	/api/auth/register	Daftar akun baru	-
POST	/api/auth/login	Login, return JWT	-
POST	/api/profiles	Buat profil baru (+ upload foto)	JWT
GET	/api/profiles/{id_user}	Daftar profil per user	JWT
DELETE	/api/profiles/{id_profile}	Hapus profil	JWT
POST	/api/measurements	Simpan hasil pengukuran	JWT
GET	/api/measurements/{id_profile}	Riwayat per profil	JWT
GET	/api/measurements/{id_profile}/latest	Data terbaru per profil	JWT

IV.2.1 Entity Relationship Diagram (ERD)



Gambar IV-2 Entity Relationship Diagram

Perancangan basis data pada sistem ini menggunakan pendekatan relasional yang diwujudkan dalam bentuk *Entity Relationship Diagram* (ERD). Basis data terdiri dari empat entitas utama, yaitu *Users*, *Profiles*, *Measurements*, dan *spo2_measurements*, yang saling berelasi satu sama lain untuk mendukung fungsionalitas sistem pemantauan kesehatan secara multiuser.

Entitas *Users* merupakan entitas utama yang menyimpan data akun pengguna, mencakup atribut nama, email, nomor telepon, dan kata sandi. Entitas ini berelasi *one-to-many* (1:N) dengan entitas *Profiles*, yang berarti satu akun pengguna dapat memiliki lebih dari satu profil. Hal ini dirancang untuk mengakomodasi penggunaan oleh beberapa anggota keluarga dalam satu akun yang sama, di mana setiap profil menyimpan data pribadi seperti usia, jenis kelamin, tinggi badan, dan berat badan.

Entitas *Measurements* berfungsi untuk menyimpan data hasil pengukuran tekanan darah yang diperoleh dari perangkat tensimeter Omron melalui koneksi Bluetooth Low Energy (BLE). Entitas ini memiliki dua *foreign key*, yaitu *id_user* yang mengacu pada entitas *Users* dan *id_profiles* yang mengacu pada entitas *Profiles*,

sehingga membentuk relasi *one-to-many* dengan kedua entitas tersebut. Keberadaan *id_user* secara langsung pada tabel ini merupakan bentuk denormalisasi yang disengaja guna meningkatkan performa query dan mempermudah validasi otorisasi pada setiap permintaan data.

Entitas *spo2_measurements* berfungsi untuk menyimpan data hasil pengukuran yang diperoleh dari sensor MAX30102 melalui perangkat ESP32, meliputi kadar oksigen darah (*SpO2*), detak jantung (*heart rate*), dan suhu tubuh. Sama seperti entitas *Measurements*, entitas ini juga berelasi *one-to-many* dengan entitas *Users* dan *Profiles* melalui *foreign key* *id_user* dan *id_profiles*. Kedua entitas pengukuran ini tidak memiliki relasi langsung satu sama lain, melainkan bersifat sibling yang terhubung secara tidak langsung melalui entitas *Profiles*. Desain ini dipilih karena pengukuran dari tensimeter dan sensor MAX30102 dilakukan secara independen dan tidak harus dilakukan pada waktu yang bersamaan.

IV.2.2 Use Case Scenario

IV.2.2.1 Login

Tabel IV-4 Use Case Scenario Login

Nama Use Case	<i>Login</i>
ID Use Case	UC-01
Aktor	Pasien
Include	-
Pre-Condition	Pasien pertama kali mengakses aplikasi
Post-Condition	Pasien dapat melihat fitur manajemen profile
Alur Peristiwa	1. Pengguna mengakses aplikasi 2. Sistem menampilkan halaman login 3. Pengguna memasukkan <i>email/nomor hp</i> dan password 4. Pengguna menekan button login

	5. Sistem menampilkan tampilan manajemen profile
False-Condition	1. Pengguna mengakses aplikasi 2. Sistem menampilkan halaman login 3. Pengguna memasukkan <i>email/nomor</i> hp dan password yang salah 4. Pengguna menekan <i>button login</i> 5. Sistem menampilkan <i>error message</i>

IV.2.2.2 Register

Tabel IV-5 Use Case Scenario Register

Nama Use Case	<i>Register</i>
ID Use Case	UC-02
Aktor	Pasien
<i>Include</i>	-
<i>Pre-Condition</i>	Pasien pertama kali mengakses aplikasi dan belum mempunyai akun
<i>Post-Condition</i>	Pasien dapat melihat fitur login
Alur Peristiwa	1. Pengguna mengakses aplikasi 2. Sistem menampilkan halaman login 3. Pengguna memasukkan data yang dibutuhkan seperti nama, email, Nomor HP, Password 4. Pengguna menekan button Daftar 5. Sistem menampilkan tampilan manajemen profile

<i>False-Condition</i>	1. Pengguna mengakses aplikasi 2. Sistem menampilkan halaman login 3. Pengguna memasukan data yang dibutuhkan seperti nama, email, Nomor HP, Password dengan salah 4. Pengguna menekan button Daftar 5. Sistem menampilkan error message
------------------------	--

IV.2.2.3 Manajemen Profile

Tabel IV-6 Use Case Scenario Manajemen Profile

Nama <i>Use Case</i>	Manajemen Profile
ID <i>Use Case</i>	UC-03
Aktor	Pasien
<i>Include</i>	-
<i>Pre-Condition</i>	Pasien sudah melakukan login
<i>Post-Condition</i>	Pasien dapat melihat fitur manajemen profile
Alur Peristiwa	1. Pengguna menekan button “Tambah Profil” 2. Pengguna mengisi input nama <i>profile</i>, umur, gender, tinggi badan, berat badan. 3. Pengguna berhasil membuat <i>profile</i> 4. Pengguna bisa melakukan <i>Update</i> dan <i>Delete profile</i>
<i>False-Condition</i>	1. Pengguna menekan button “Tambah Profil”

	- Pengguna tidak mengisi <i>input</i> nama profile, umur, gender, tinggi badan, berat badan. - Sistem menampilkan <i>error message</i>
--	---

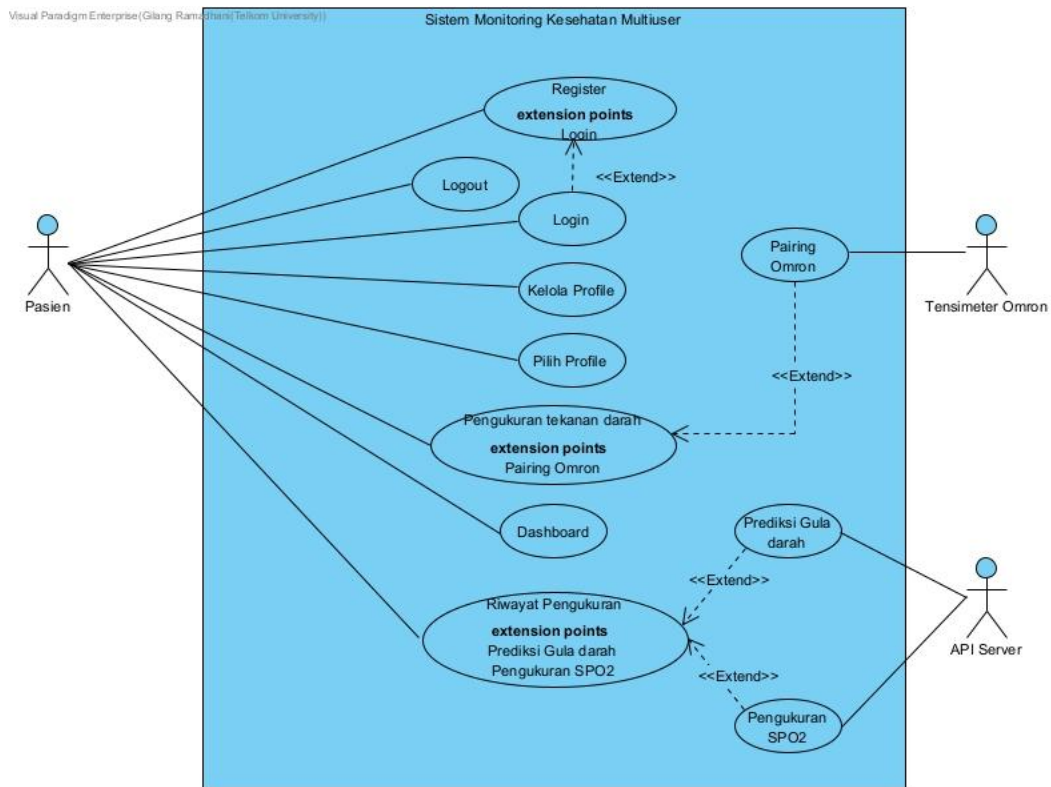
IV.2.2.4 Measurement Flow

Tabel IV-7 Use Case Scenario Measurement Flow

Nama Use Case	Measurement Flow
ID Use Case	UC-04
Aktor	Pasien
Include	-
Pre-Condition	Pasien sudah melakukan <i>login</i> dan mempunyai <i>profile</i>
Post-Condition	Pasien dapat melihat <i>dashboard</i>
Alur Peristiwa	- Pengguna melakukan pengukuran secara manual menggunakan tensimeter Omron - Pengguna menekan button “Ambil data” - Sistem menampilkan data tensimeter, kemudian tekan button “Lanjut Prediksi Gula Darah” - Pengguna menempelkan jari ke sensor max30102 pada ESP32-S3 - Pengguna bisa melihat data sys, dia, bpm, spo2, dan prediksi gula darah

<i>False-Condition</i>	1. Pengguna melakukan pengukuran secara manual menggunakan tensimeter Omron 2. Pengguna menekan button “Ambil data” 3. Sistem menampilkan data tensimeter, kemudian tekan button “Lanjut Prediksi Gula Darah” 4. Pengguna menempelkan jari ke sensor max30102 pada ESP32-S3 5. Saat loading, pengguna menekan button “Batalkan Pengukuran” 6. Pengguna akan kehilangan data yang di ukur (sys, dia, bpm, spo2, dan prediksi gula darah)
------------------------	--

IV.2.3 Use Case Diagram



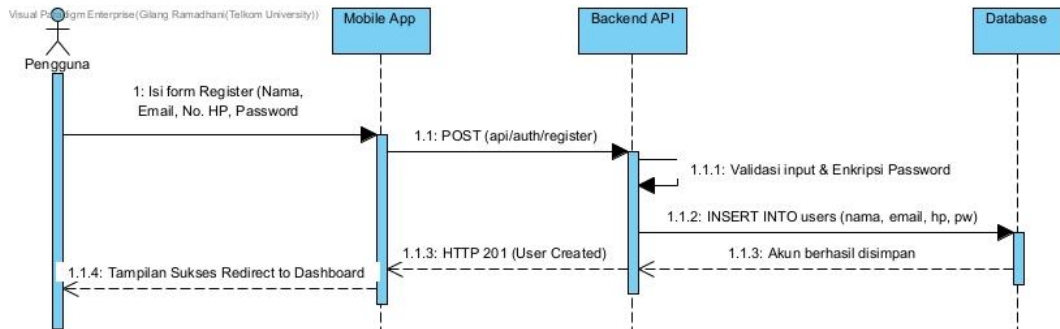
Gambar IV-3 Use Case Diagram

IV.2.4 Sequence Diagram

Sequence diagram dirancang untuk memodelkan alur interaksi antar komponen sistem secara kronologis pada setiap fitur utama Rumah Berpulang. Diagram ini menggambarkan bagaimana pesan atau permintaan (*request*) mengalir dari satu komponen ke komponen lainnya mulai dari antarmuka pengguna (*Mobile App*), backend server (FastAPI), hingga perangkat keras (Omron HEM-7155T via BLE) dan basis data. Dengan *sequence diagram*, urutan proses, titik keputusan, serta respons yang dikembalikan pada setiap langkah dapat dipahami secara visual dan sistematis.

Berikut adalah *sequence diagram* untuk empat alur fitur utama sistem multiuser pada aplikasi Rumah Berpulang yang sedang dikembangkan:

IV.2.4.1 Sequence Diagram Register Akun



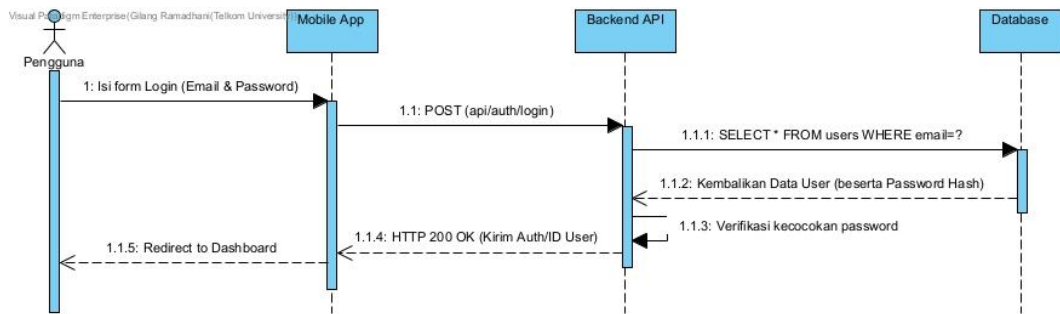
Gambar IV-4 Sequence Diagram Register

Pada sub bab ini, *Sequence Diagram* mengilustrasikan alur interaksi sistem saat pengguna baru melakukan pembuatan akun. Proses dimulai ketika Pengguna mengisi form registrasi pada *Mobile App* dengan memasukkan data diri berupa Nama, Email, Nomor HP, dan *Password*. Setelah data disubmit, *Mobile App* mengirimkan *request* POST ke *endpoint* `api/auth/register` pada *Backend API*.

Menerima data tersebut, *Backend API* secara internal melakukan validasi terhadap input pengguna dan melakukan enkripsi (*hashing*) pada *password* untuk menjaga keamanan data. Setelah divalidasi, *Backend API* mengeksekusi perintah `INSERT INTO users` untuk menyimpan data nama, email, nomor HP, dan password pengguna ke dalam Database.

Ketika Database memberikan konfirmasi bahwa akun telah berhasil disimpan, *Backend API* merespons *Mobile App* dengan status `HTTP 201 (User Created)`. Proses registrasi ini diakhiri dengan *Mobile App* yang menampilkan antarmuka sukses dan secara otomatis mengarahkan pengguna (*redirect*) ke halaman *Dashboard*.

IV.2.4.2 Sequence Diagram Login



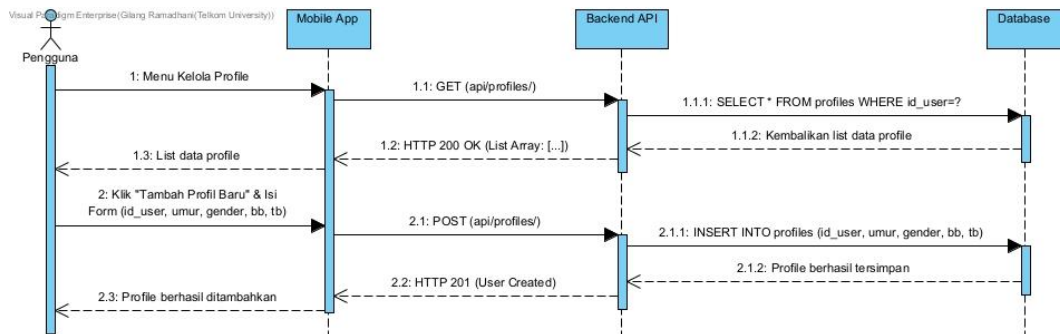
Gambar IV-5 Sequence Diagram Login

Pada sub bab ini, *Sequence Diagram* menjelaskan alur proses autentikasi pengguna untuk masuk (*login*) ke dalam sistem. Proses diawali oleh Pengguna yang memasukkan kredensial berupa *Email* dan *Password* ke dalam *form login* di *Mobile App*. Data ini kemudian dikirimkan oleh Mobile App melalui request POST ke endpoint *api/auth/login* menuju *Backend API*.

Backend API selanjutnya melakukan pencarian ke Database dengan perintah `SELECT * FROM users WHERE email=?` untuk mengambil data akun berdasarkan *email* yang dikirimkan. Database kemudian mengembalikan sekumpulan data pengguna tersebut, termasuk password hash yang tersimpan, kembali ke *Backend API*.

Secara internal, *Backend API* melakukan verifikasi untuk mencocokkan apakah *password* yang diinputkan pengguna sesuai dengan *password hash* dari database. Apabila verifikasi berhasil, *Backend API* akan membalas *Mobile App* dengan respons HTTP 200 OK yang juga menyertakan token autentikasi atau ID *User*. Sebagai langkah terakhir, *Mobile App* memproses respons tersebut dan mengarahkan pengguna (*redirect*) ke halaman *Dashboard*, yang menandakan bahwa pengguna telah berhasil masuk ke dalam sistem.

IV.2.4.3 Sequence Diagram Manajemen Profile

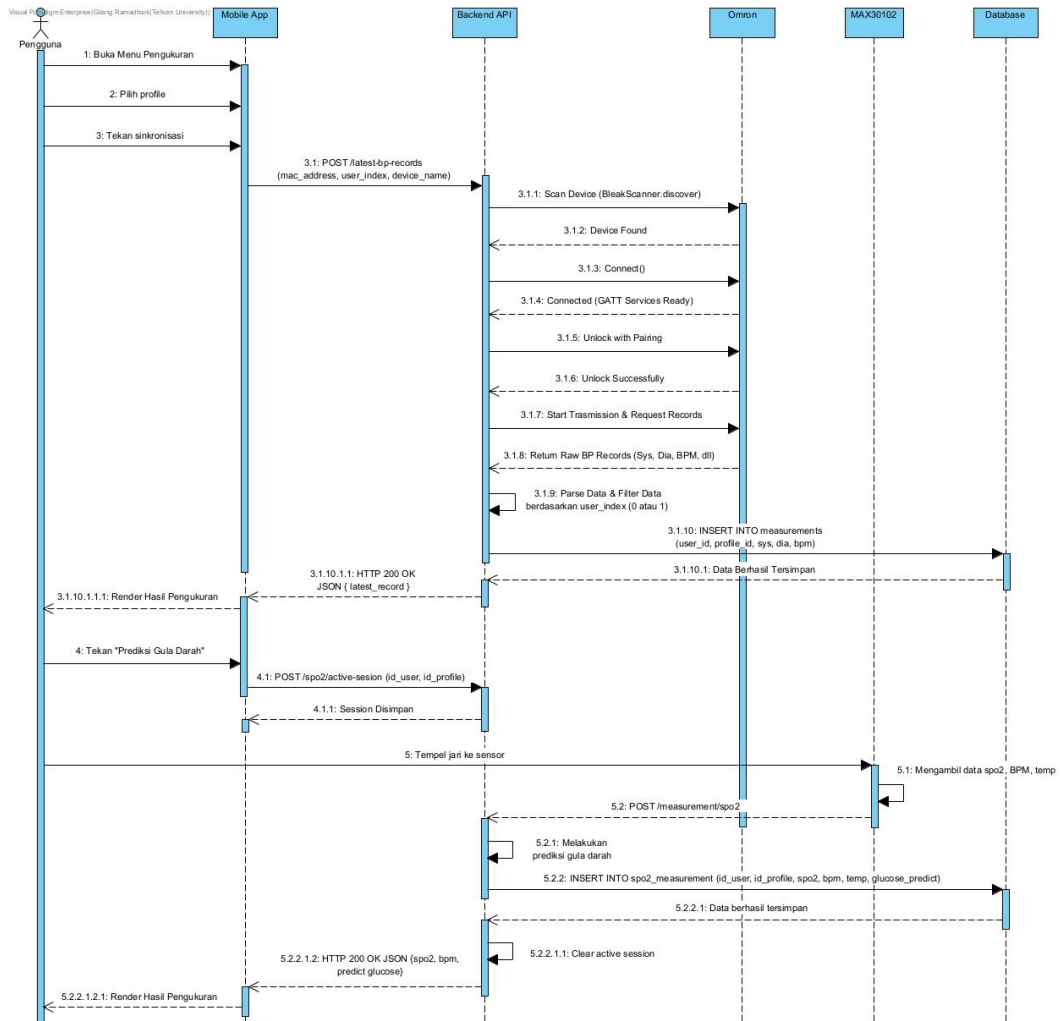


Gambar IV-6 Sequence Diagram Manajemen Profile

Pada sub bab ini, *Sequence Diagram* mengilustrasikan dua alur utama dalam fitur manajemen profil pengguna, yaitu menampilkan daftar profil dan menambahkan profil baru.

1. Menampilkan Profil: Proses dimulai saat Pengguna membuka menu "Kelola Profile" di Mobile App. Aplikasi kemudian melakukan request GET ke endpoint `api/profiles/` pada Backend API. Backend API akan mengambil data profil dari Database menggunakan kueri `SELECT * FROM profiles WHERE id_user=?`. Setelah Database mengembalikan data tersebut, Backend API merespons Mobile App dengan status HTTP 200 OK beserta *array* berisi daftar profil. Mobile App lalu menampilkan daftar tersebut kepada pengguna.
2. Menambah Profil Baru: Jika Pengguna memilih "Tambah Profil Baru" dan mengisi form data fisik (seperti `id_user`, umur, gender, berat badan, tinggi badan), Mobile App akan mengirim request POST ke `api/profiles/` menuju Backend API. Backend API menyimpan data tersebut ke Database melalui perintah `INSERT INTO profiles`. Setelah berhasil disimpan, Backend API mengirimkan respons HTTP 201 (*User Created*) ke Mobile App, yang kemudian menampilkan pesan bahwa profil telah berhasil ditambahkan.

IV.2.4.4 Sequence Diagram Measurement Flow



Gambar IV-7 Sequence Diagram Measurement Flow

Pada sub bab ini, *Sequence Diagram* mendeskripsikan alur proses sinkronisasi pengukuran tekanan darah secara *end-to-end* antara pengguna, aplikasi, *backend*, perangkat tensimeter (Omron), dan database. Proses diawali dengan interaksi pengguna pada *Mobile App* yaitu membuka menu pengukuran, memilih profil pengguna, dan menekan tombol sinkronisasi. *Mobile App* lalu mengirim *request* POST ke endpoint `/latest-bp-records` dengan membawa *payload* seperti `mac_address`, `user_index`, dan `device_name` ke *Backend API*.

Selanjutnya, *Backend API* memprakarsai komunikasi *Bluetooth* dengan melakukan proses pencarian perangkat (`BleakScanner.discover`) hingga tensimeter Omron

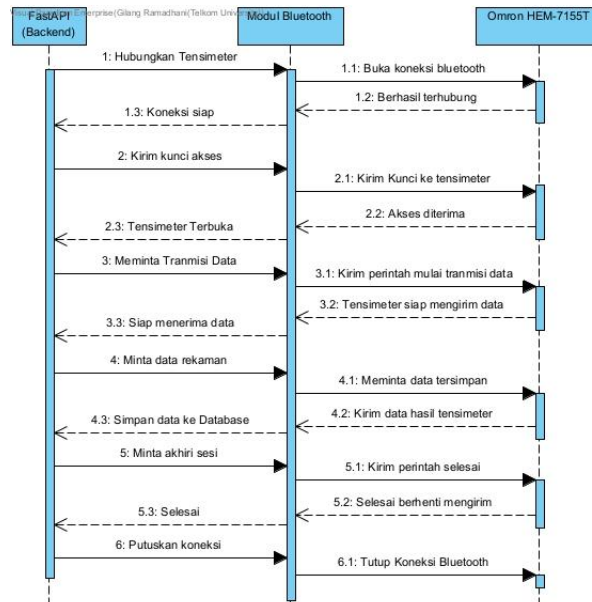
ditemukan dan koneksi berhasil dibangun (layanan GATT siap). Setelah koneksi terbentuk, *Backend API* melakukan *pairing* (membuka kunci akses), kemudian mengirim perintah untuk meminta transmisi data rekaman tekanan darah. Data mentah (*raw data*) yang dikembalikan oleh tensimeter (berisi Sys, Dia, BPM, dll) ditangkap oleh *Backend API* untuk melalui tahap parsing dan penyaringan data (*filtering*) berdasarkan parameter *user_index*. Data yang telah diolah kemudian disimpan ke dalam Database (`INSERT INTO measurements`). Apabila penyimpanan berhasil, *Backend API* merespons *Mobile App* dengan status HTTP 200 OK beserta data rekaman terbaru dalam format JSON, dan pada langkah terakhir *Mobile App* menampilkan (*render*) hasil pengukuran tersebut kepada Pengguna.

Setelah hasil pengukuran tekanan darah ditampilkan, Pengguna kemudian diwajibkan untuk menekan tombol "Prediksi Gula Darah" untuk melanjutkan ke tahap pengukuran berikutnya. *Mobile App* merespons aksi tersebut dengan mengirim *request POST* ke endpoint `/spo2/active-session` yang membawa parameter *id_user* dan *id_profile* ke *Backend API*, yang kemudian menyimpan sesi aktif tersebut dan mengonfirmasikannya kembali ke *Mobile App*. Selanjutnya, Pengguna diarahkan untuk menempelkan jari pada sensor yang terhubung dengan perangkat Omron, di mana perangkat tersebut secara otomatis mengambil data SpO2 dan BPM dari jari pengguna.

Data yang terekam oleh sensor kemudian dikirimkan melalui *request POST* ke endpoint `/measurement/spo2` menuju *Backend API*. *Backend API* selanjutnya melakukan proses prediksi gula darah berdasarkan data SpO2, BPM, dan suhu yang diterima, menggunakan model atau algoritma prediksi yang telah diimplementasikan. Hasil pengolahan tersebut, yang mencakup *id_user*, *id_profile*, *spo2*, *bpm*, *temp*, dan *glucose_predict*, kemudian disimpan ke dalam Database melalui perintah `INSERT INTO spo2_measurement`. Setelah data berhasil tersimpan, *Backend API* menghapus sesi aktif (*clear active session*) sebagai penanda bahwa proses pengukuran telah selesai, lalu mengirimkan respons HTTP 200 OK beserta data hasil prediksi (*spo2*, *bpm*, dan *predict glucose*) dalam format JSON kepada *Mobile App*. Pada tahap akhir, *Mobile App* menampilkan (*render*) hasil prediksi gula darah tersebut kepada Pengguna, sehingga keseluruhan alur

pengukuran tekanan darah hingga prediksi gula darah dapat diselesaikan secara *end-to-end* tanpa intervensi manual lebih lanjut dari sisi Pengguna.

IV.2.4.5 Sequence Diagram Alur Komunikasi BLE



Gambar IV-8 Sequence Diagram Alur Komunikasi BLE

Pada sub bab ini, *Sequence Diagram* secara spesifik menekankan alur komunikasi *Bluetooth Low Energy* (BLE) yang lebih teknis antara modul FastAPI di sisi *Backend*, Modul *Bluetooth*, dan perangkat tensimeter Omron HEM-7155T. Komunikasi berjalan melalui tahapan interaksi sekuensial yang saling berbalas instruksi

Pertama, *Backend* memerintahkan modul Bluetooth untuk membuka koneksi fisik dengan perangkat tensimeter Omron hingga didapatkan konfirmasi koneksi telah siap. Selanjutnya, *Backend* memulai tahapan autentikasi dengan mengirimkan "kunci akses" kepada tensimeter agar interaksi pertukaran data dilegitimasi (akses diterima). Setelah hak akses diberikan, *Backend* mengirimkan perintah agar tensimeter bersiap untuk mengirimkan data transmisi (*request start transmission*).

Backend kemudian melakukan permintaan data secara spesifik untuk mengambil rekaman memori yang tersimpan. Data hasil pengukuran tersebut dikirimkan oleh tensimeter kembali menuju *Backend* lalu diteruskan agar kelak disimpan ke database. Setelah semua pengambilan data beres, *Backend* mengirimkan instruksi

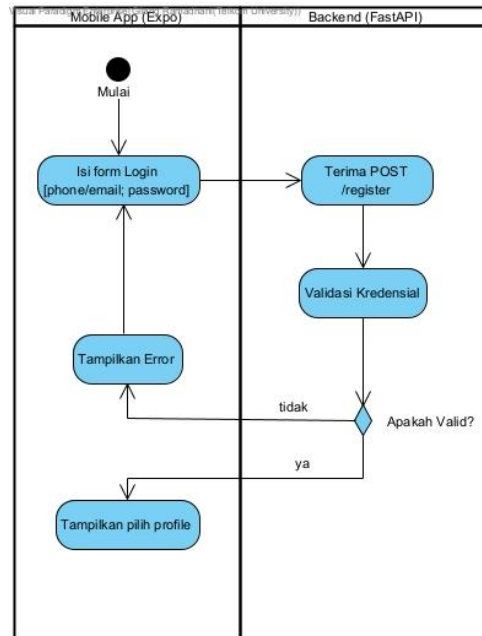
untuk mengakhiri sesi pengiriman data dengan tensimeter. Terakhir, setelah konfirmasi sesi transmisi selesai diterima, *Backend* memutus koneksi (*disconnect/close connection*) *Bluetooth* dengan tensimeter Omron untuk menyudahi proses komunikasi secara utuh.

IV.2.5 Activity Diagram

Activity diagram merupakan salah satu jenis diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan alur kerja (*workflow*) atau urutan aktivitas dari suatu proses bisnis maupun proses sistem secara berurutan, mulai dari titik awal (*initial node*) hingga titik akhir (*final node*). Diagram ini menampilkan berbagai elemen seperti aktivitas (*action*), titik keputusan (*decision node*) yang merepresentasikan percabangan logika berdasarkan suatu kondisi, serta alur kontrol yang menunjukkan urutan eksekusi antaraktivitas. Dalam konteks pengembangan sistem, *activity diagram* sangat bermanfaat untuk memvisualisasikan interaksi antaraktor maupun antarkomponen sistem, termasuk proses pengambilan keputusan dan penanganan kondisi gagal (*error handling*), sehingga dapat memberikan pemahaman yang lebih jelas mengenai bagaimana suatu sistem bekerja secara menyeluruh sebelum diimplementasikan ke dalam kode program.

Untuk memberikan gambaran yang lebih rinci dan mudah dipahami mengenai alur kerja sistem, *activity diagram* pada aplikasi Rumah Berpulang disajikan secara terpisah berdasarkan masing-masing fitur utama, yaitu fitur autentikasi (*login*), fitur pemilihan profil pengguna, dan fitur pengukuran tekanan darah. Pemecahan diagram per fitur ini dilakukan untuk mempermudah pembacaan alur proses yang melibatkan interaksi kompleks antara tiga aktor utama, yaitu *Mobile App* berbasis Expo, *Backend* berbasis FastAPI, dan perangkat tensimeter Omron HEM-7155T, yang saling berkomunikasi melalui protokol *Bluetooth Low Energy* (BLE) dan REST API. Dengan pendekatan ini, setiap diagram dapat menggambarkan secara lebih spesifik urutan aktivitas, titik pengambilan keputusan (*decision point*), serta kondisi keberhasilan maupun kegagalan (*error handling*) pada tiap tahapan proses, sehingga memudahkan pembaca dalam memahami logika sistem secara bertahap sebelum melihat keterkaitan keseluruhan alur pada sistem secara utuh.

IV.2.5.1 Activity Diagram Register

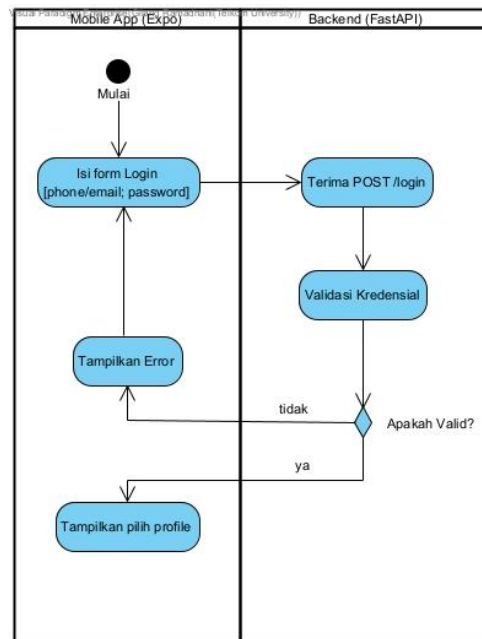


Gambar IV-9 Activity Diagram Register

Gambar IV-9 menunjukkan *activity diagram* untuk proses registrasi pengguna (*register*) pada aplikasi Rumah Berpulang, yang digambarkan menggunakan dua *swimlane* untuk merepresentasikan interaksi antara sisi klien (*Mobile App Expo*) dan sisi server (*Backend FastAPI*). Proses dimulai ketika pengguna mengisi *form* registrasi dengan memasukkan data berupa nomor telepon atau alamat *email* beserta kata sandi. Setelah data dimasukkan, aplikasi *mobile* mengirimkan permintaan melalui HTTP POST ke *endpoint* /register pada backend. Pada sisi server, sistem menerima permintaan tersebut kemudian melakukan proses validasi kredensial untuk memastikan data yang dimasukkan memenuhi ketentuan yang berlaku, misalnya format *email*/nomor telepon yang benar serta memastikan akun belum pernah terdaftar sebelumnya. Hasil validasi tersebut kemudian dievaluasi melalui decision node "Apakah Valid?". Apabila data yang dimasukkan tidak valid, sistem akan mengembalikan respons kegagalan ke aplikasi *mobile*, yang selanjutnya menampilkan pesan *error* kepada pengguna dan mengarahkan pengguna kembali ke form registrasi untuk melakukan perbaikan data. Sebaliknya, apabila data yang dimasukkan valid, sistem akan menyimpan data akun baru dan melanjutkan alur proses ke tampilan pemilihan profil (pilih profile), yang memungkinkan pengguna untuk membuat atau memilih profil yang akan digunakan dalam aplikasi. *Activity*

diagram ini menggambarkan alur kendali (*control flow*) antara proses *input* data registrasi, komunikasi *client-server*, serta percabangan logika berdasarkan hasil validasi, sehingga memberikan gambaran menyeluruh mengenai mekanisme pendaftaran akun baru pada aplikasi Rumah Berpulang.

IV.2.5.2 Activity Diagram Login

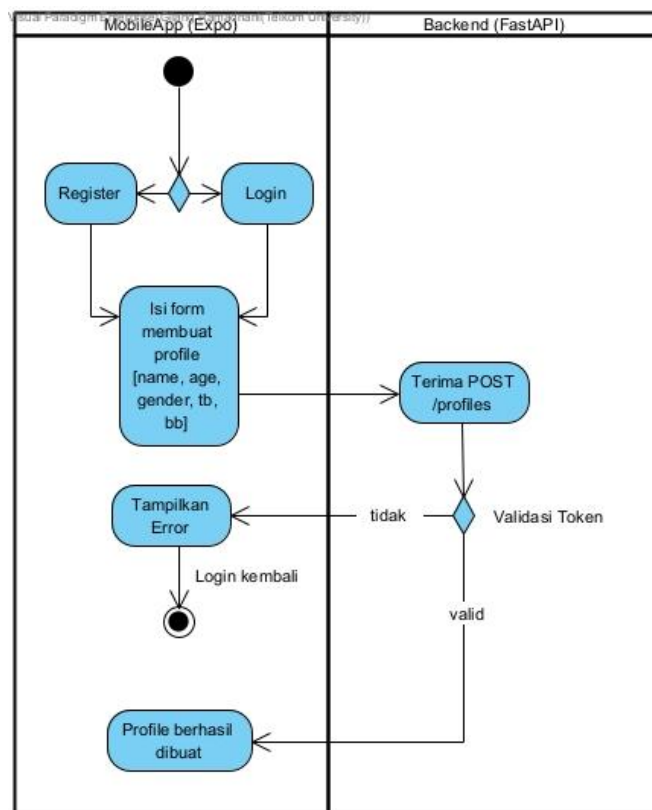


Gambar IV-10 Activity Diagram Login

Gambar IV-10 menunjukkan *activity diagram* untuk proses autentikasi pengguna (login) pada aplikasi Rumah Berpulang, yang digambarkan menggunakan dua swimlane untuk merepresentasikan interaksi antara sisi klien (*Mobile App Expo*) dan sisi server (*Backend FastAPI*). Proses dimulai ketika pengguna membuka aplikasi dan diarahkan ke *form login* untuk mengisi data berupa nomor telepon atau alamat email beserta kata sandi. Setelah data dimasukkan, aplikasi *mobile* mengirimkan permintaan melalui HTTP POST ke endpoint `/login` pada *backend*. Pada sisi *server*, sistem menerima permintaan tersebut kemudian melakukan proses validasi kredensial dengan mencocokkan data yang dikirimkan terhadap data pengguna yang tersimpan pada basis data. Hasil validasi tersebut kemudian dievaluasi melalui decision node "Apakah Valid?". Apabila kredensial yang dimasukkan tidak valid, sistem akan mengembalikan respons kegagalan ke aplikasi *mobile*, yang selanjutnya menampilkan pesan *error* kepada pengguna dan

mengarahkan pengguna kembali ke *form login* untuk melakukan percobaan ulang. Sebaliknya, apabila kredensial yang dimasukkan valid, sistem akan melanjutkan alur proses ke tampilan pemilihan profil (pilih *profile*), yang memungkinkan pengguna untuk memilih profil pengguna yang akan digunakan dalam sesi pemantauan tekanan darah selanjutnya. *Activity diagram* ini menggambarkan alur kendali (*control flow*) yang jelas antara proses input pengguna, komunikasi *client-server*, serta percabangan logika berdasarkan hasil validasi, sehingga memberikan gambaran menyeluruh mengenai mekanisme autentikasi yang diterapkan pada aplikasi Rumah Berpulang.

IV.2.5.3 Activity Diagram Manajemen Profile

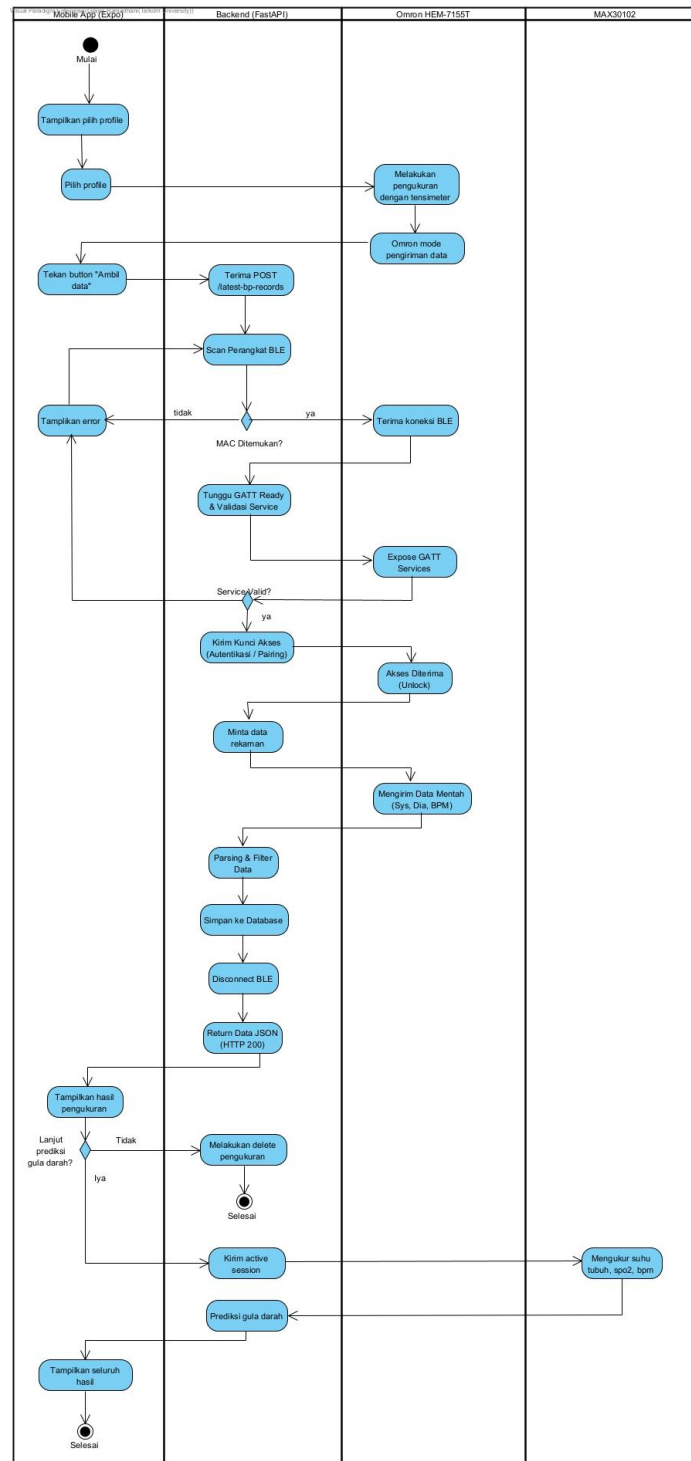


Gambar IV-11 Activity Manajemen Profile

Gambar IV-11 menunjukkan *activity diagram* untuk proses pembuatan profil (*create profile*) pada aplikasi Rumah Berpulang, yang digambarkan menggunakan dua *swimlane* untuk merepresentasikan interaksi antara sisi klien (*Mobile App Expo*) dan sisi server (*Backend FastAPI*). Proses ini dapat diakses melalui dua jalur, yaitu setelah pengguna melakukan registrasi (*Register*) atau setelah pengguna

melakukan login (*Login*), yang keduanya berhak melanjutkan pada *activity* selanjutnya yaitu pengisian form pembuatan profil. Pada tahap ini, pengguna memasukkan data profil berupa nama, usia (*age*), jenis kelamin (*gender*), tinggi badan (*tb*), dan berat badan (*bb*). Setelah data lengkap diisi, aplikasi mobile mengirimkan permintaan melalui HTTP POST ke *endpoint /profiles* pada *backend*. Pada sisi server, sistem menerima permintaan tersebut kemudian melakukan validasi *token* untuk memastikan bahwa permintaan berasal dari pengguna yang telah terautentikasi secara sah. Apabila *token* tidak valid, sistem akan mengembalikan respons kegagalan ke aplikasi *mobile*, yang selanjutnya menampilkan pesan *error* kepada pengguna dan mengarahkan pengguna untuk melakukan login kembali, yang menandai berakhirnya alur proses pada percabangan tersebut (*final node*). Sebaliknya, apabila token dinyatakan valid, sistem akan melanjutkan proses penyimpanan data profil, dan aplikasi mobile akan menampilkan notifikasi bahwa profil berhasil dibuat (*Profile* berhasil dibuat). Activity diagram ini menggambarkan alur kendali yang mengintegrasikan dua titik masuk (*entry point*) dari proses registrasi maupun login menuju satu proses bersama, yaitu pembuatan profil pengguna, sekaligus menunjukkan mekanisme validasi token sebagai lapisan keamanan tambahan sebelum data profil disimpan pada aplikasi Rumah Berpulang.

IV.2.5.4 Activity Diagram Measurement Flow



Gambar IV-12 Activity Measurement Flow

Gambar IV-12 menunjukkan activity diagram untuk proses pengukuran tekanan darah pada aplikasi Rumah Berpulang, yang digambarkan menggunakan empat *swimlane* untuk merepresentasikan interaksi antara sisi klien (Mobile App Expo),

sisi server (*Backend* FastAPI), perangkat tensimeter Omron HEM-7155T, dan sensor MAX30102 dari ESP32-S3. Proses dimulai ketika pengguna memilih profil (pilih profile) yang akan digunakan untuk sesi pengukuran, sementara secara bersamaan pengguna melakukan pengukuran tekanan darah secara fisik menggunakan tensimeter, yang kemudian beralih ke mode pengiriman data (data transmission mode) pada perangkat Omron. Selanjutnya, pengguna menekan tombol "Ambil data" pada aplikasi *mobile*, yang mengirimkan permintaan HTTP POST ke *endpoint* /latest-bp-records pada *backend*. *Backend* kemudian melakukan proses pemindaian perangkat (scan perangkat BLE) untuk menemukan alamat MAC dari perangkat Omron yang terhubung. Proses ini dievaluasi melalui decision node "MAC Ditemukan?"; apabila perangkat tidak ditemukan, aplikasi *mobile* akan menampilkan pesan *error* dan proses pemindaian akan diulang kembali, sedangkan apabila perangkat berhasil ditemukan, koneksi BLE akan diterima oleh perangkat Omron.

Setelah koneksi BLE berhasil terbentuk, *backend* melakukan proses menunggu kesiapan GATT (GATT Ready) sekaligus validasi service, yang pada sisi perangkat Omron ditandai dengan proses expose GATT services. Hasil validasi tersebut dievaluasi melalui decision node "Service Valid?"; apabila service tidak valid, aplikasi *mobile* akan menampilkan pesan error dan proses akan diulang, sedangkan apabila service dinyatakan valid, *backend* akan mengirimkan kunci akses (otentikasi/pairing) yang kemudian direspons oleh perangkat Omron melalui proses *unlock* (akses diterima). Selanjutnya, *backend* meminta data rekaman pengukuran kepada perangkat Omron, yang kemudian mengirimkan data mentah berupa nilai sistol (sys), diastol (dia), dan detak jantung per menit (BPM). Data mentah tersebut kemudian diproses melalui tahapan *parsing* dan *filtering* pada sisi *backend*, disimpan ke dalam basis data, diikuti dengan pemutusan koneksi BLE (*disconnect*), dan pengembalian data dalam format JSON melalui response HTTP 200.

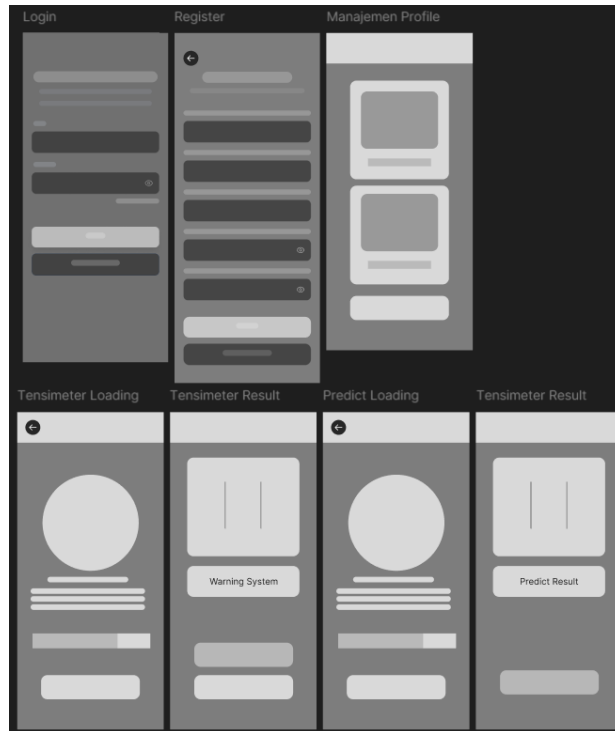
Setelah data diterima, aplikasi *mobile* menampilkan hasil pengukuran (tampilkan hasil pengukuran) kepada pengguna, yang kemudian dihadapkan pada decision node "Lanjut prediksi gula darah?". Apabila pengguna memilih "Tidak", sistem akan melakukan penghapusan data pengukuran (*delete* pengukuran) dan proses

berakhir pada *final node*. Sebaliknya, apabila pengguna memilih "Iya", aplikasi mobile akan mengirimkan status sesi aktif (kirim *active session*) ke backend, yang diteruskan ke sensor MAX30102 untuk melakukan pengukuran suhu tubuh, SpO2, dan BPM. Hasil pengukuran dari sensor tersebut dikirimkan kembali ke *backend* untuk diproses pada tahap prediksi gula darah (prediksi gula darah). Selanjutnya, *backend* mengirimkan hasil prediksi tersebut ke aplikasi mobile untuk ditampilkan sebagai keseluruhan hasil pengukuran (tampilkan seluruh hasil), yang menandai berakhirnya alur proses (*final node*).

Activity diagram ini menggambarkan kompleksitas komunikasi antara aplikasi *mobile*, *backend*, serta dua perangkat IoT (Omron HEM-7155T dan MAX30102) melalui protokol *Bluetooth Low Energy* (BLE), termasuk mekanisme penanganan kegagalan koneksi pada setiap tahapan validasi serta alur lanjutan opsional menuju prediksi gula darah berbasis data fisiologis tambahan.

IV.2.6 Wireframe

Wireframe merupakan representasi visual awal dari antarmuka aplikasi yang menggambarkan tata letak elemen-elemen dasar pada setiap halaman tanpa memperhatikan detail visual seperti warna, tipografi, maupun aset grafis. Pada tahap ini, *wireframe* berfungsi sebagai kerangka struktural yang menentukan posisi komponen-komponen utama seperti tombol, formulir *input*, navigasi, dan area konten, sehingga alur interaksi pengguna dapat divalidasi sejak dini sebelum memasuki tahap desain yang lebih detail. Dalam penelitian ini, *wireframe* dikembangkan berdasarkan kebutuhan fungsional yang telah diidentifikasi pada tahap analisis kebutuhan serta mengacu pada alur interaksi yang telah dimodelkan melalui *sequence diagram*, mencakup halaman autentikasi, halaman dasbor pemantauan tekanan darah, halaman manajemen pengguna, serta halaman riwayat pengukuran. Dengan adanya *wireframe*, perancang dapat memfokuskan perhatian pada aspek fungsionalitas dan kemudahan navigasi tanpa terdistraksi oleh elemen estetika, sehingga revisi struktural dapat dilakukan secara lebih efisien sebelum proses pengembangan antarmuka lebih lanjut dimulai.

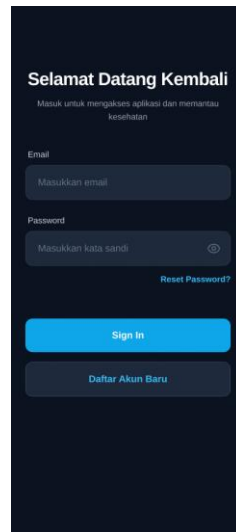


Gambar IV-13 *Wireframe*

IV.2.7 *Mockup High-fidelity*

Mockup merupakan tahap lanjutan dari wireframe yang menyajikan rancangan antarmuka dalam bentuk visual statis yang lebih detail dan mendekati tampilan aplikasi sesungguhnya, di mana elemen-elemen visual seperti warna, tipografi, ikon, ilustrasi, serta komposisi tata letak telah diterapkan secara menyeluruh pada setiap halaman. Berbeda dengan *wireframe* yang hanya menampilkan kerangka struktural, *mockup* memberikan gambaran yang lebih nyata mengenai bagaimana aplikasi akan terlihat oleh pengguna akhir, sehingga dapat digunakan sebagai media komunikasi yang efektif antara perancang dan pemangku kepentingan dalam proses evaluasi desain. Dalam penelitian ini, *mockup* dikembangkan menggunakan perangkat desain Figma dengan menerapkan prinsip-prinsip desain yang konsisten pada seluruh halaman aplikasi, mencakup pemilihan skema warna yang sesuai dengan konteks kesehatan, hierarki informasi yang jelas pada tampilan data tekanan darah, serta tata letak yang intuitif dan ramah pengguna bagi berbagai kalangan usia mengingat sasaran utama aplikasi adalah pasien hipertensi.

IV.2.7.1 Halaman Login



Gambar IV-14 High-Fidelity Login

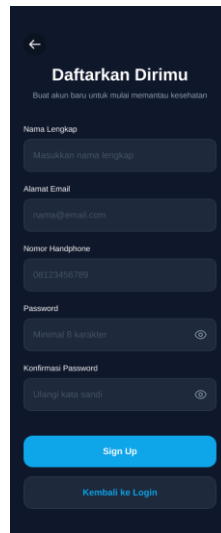
Ketika pengguna pertama kali membuka aplikasi, sistem akan langsung menampilkan halaman Login sebagai titik masuk utama. Di bagian atas halaman, terdapat teks sambutan besar "Selamat Datang Kembali" berwarna putih tebal, diikuti subjudul "Masuk untuk mengakses aplikasi dan memantau kesehatan" berwarna abu-abu muda yang memberikan konteks personal bahwa aplikasi ini berkaitan dengan pemantauan kesehatan pengguna.

Di bawahnya, terdapat dua kolom input dengan gaya kartu berwarna slate gelap (lebih terang dari *background*) bersudut tumpul: kolom "Email" dengan label di atasnya serta *placeholder* "Masukkan email" sebagai panduan format masukan, dan kolom "Password" yang dilengkapi *placeholder* "Masukkan kata sandi" beserta ikon mata di sisi kanan sebagai *toggle visibilitas* untuk meminimalkan risiko kesalahan pengetikan kata sandi. Di bawah kolom *input*, tersedia tautan "Reset Password?" berwarna biru cerah yang rata kanan, memudahkan pengguna mengatur ulang kata sandi jika terlupa.

Selanjutnya, terdapat dua tombol aksi utama yang tersusun secara vertikal dengan bentuk pill (sudut membulat penuh): tombol "Sign In" berwarna biru cerah (*cyan*) solid sebagai aksi primer untuk masuk ke halaman utama aplikasi, serta tombol "Daftar Akun Baru" dengan gaya *outline* berlatar gelap dan teks biru sebagai aksi

sekunder yang mengarahkan pengguna tanpa akun menuju halaman Registrasi. Secara keseluruhan, hierarki visual halaman ini dirancang agar pengguna dapat memahami alur penggunaan secara alami dari atas sampai bawah dari sambutan, pengisian data, hingga aksi akhir tanpa kebingungan dalam navigasi.

IV.2.7.2 Halaman Register

The image shows a mobile application registration screen with a dark blue background. At the top left is a back arrow icon. The title 'Daftarkan Dirimu' is centered in white. Below it is a subtitle 'Buat akun baru untuk mulai memantau kesehatan' in a lighter gray. The form consists of five input fields, each with a label and a placeholder: 'Nama Lengkap' (placeholder: 'Masukkan nama lengkap'), 'Alamat Email' (placeholder: 'nama@email.com'), 'Nomor Handphone' (placeholder: '08123456789'), 'Password' (placeholder: 'Minimal 8 karakter'), and 'Konfirmasi Password' (placeholder: 'Ulangi kata sandi'). Each field has a toggle icon on the right. At the bottom are two buttons: a blue 'Sign Up' button and a gray 'Kembali ke Login' button.

Gambar IV-15 High-Fidelity Register

Ketika pengguna memilih untuk membuat akun baru, sistem akan menampilkan halaman Registrasi dengan tema *dark mode* yang konsisten dengan halaman *Login*, berlatar biru gelap pekat. Di bagian atas halaman terdapat tombol back berbentuk lingkaran dengan ikon panah kiri untuk kembali ke halaman sebelumnya, diikuti judul besar "Daftarkan Dirimu" berwarna putih tebal serta subjudul "Buat akun baru untuk mulai memantau kesehatan" berwarna abu-abu muda yang menjelaskan tujuan halaman ini secara personal.

Di bawahnya, terdapat form registrasi dengan lima kolom input bergaya kartu slate gelap bersudut tumpul, tersusun vertikal dari atas ke bawah: Nama Lengkap dengan placeholder "Masukkan nama lengkap", Alamat *Email* dengan *placeholder format email* "nama@email.com", Nomor Handphone dengan *placeholder format nomor* "08123456789", Password dengan placeholder "Minimal 8 karakter" beserta ikon mata di sisi kanan sebagai *toggle visibilitas*, dan Konfirmasi *Password* dengan *placeholder* "Ulangi kata sandi" yang juga dilengkapi ikon mata serupa untuk

memastikan kata sandi yang dimasukkan konsisten dan meminimalkan kesalahan pengetikan.

Selanjutnya, terdapat dua tombol aksi utama berbentuk pill yang tersusun vertikal: tombol "*Sign Up*" berwarna biru cerah (*cyan*) solid sebagai aksi primer untuk menyelesaikan proses pendaftaran dan membuat akun baru, serta tombol "Kembali ke Login" dengan gaya *outline* berlatar gelap dan teks biru sebagai aksi sekunder bagi pengguna yang sudah memiliki akun dan ingin kembali ke halaman *Login*. Secara keseluruhan, hierarki visual halaman ini dirancang konsisten dengan halaman *Login*, memandu pengguna mengisi data secara berurutan dari informasi dasar hingga keamanan akun tanpa kebingungan dalam navigasi.

IV.2.7.3 Lupa Password

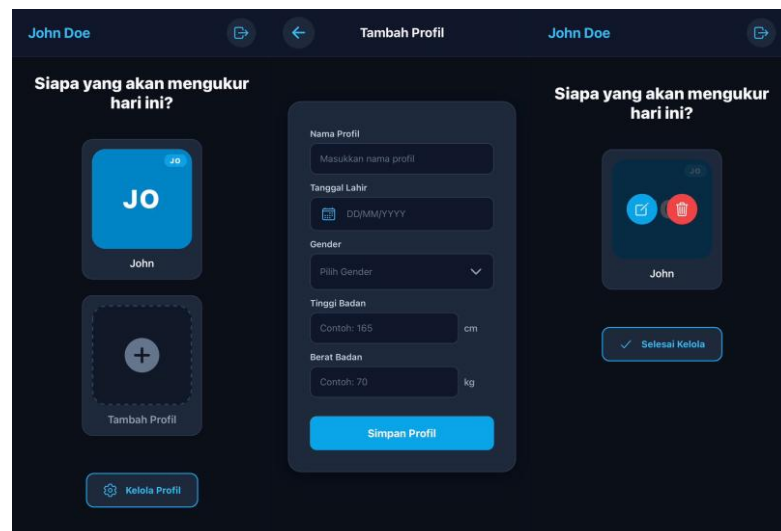


Gambar IV-16 High-Fidelity Lupa Password

Halaman *Lupa Password* dapat diakses pengguna melalui tautan "Lupa Kata Sandi?" pada halaman *Login*. Di bagian atas terdapat header navigasi dengan tombol panah kembali di sisi kiri, judul "*Lupa Password*" di tengah, serta ikon tanda tanya di sisi kanan sebagai akses bantuan. Di bawahnya, ikon gembok dengan simbol pemulihan memberikan representasi visual yang sesuai dengan konteks halaman, diikuti judul "Lupa Kata Sandi?" berukuran besar dan tebal, serta teks panduan "Jangan khawatir. Masukkan Email atau Nomor HP Anda di bawah ini untuk mendapatkan kode pemulihan" yang bersifat memandu pengguna agar tidak panik saat lupa kata sandi.

Di dalam kartu formulir, terdapat blok informasi bertanda garis biru di sisi kiri dengan label "Email atau Nomor HP" beserta keterangan "Pastikan data yang dimasukkan sudah benar dan aktif" sebagai pengingat sebelum mengisi data. Di bawahnya tersedia dua kolom input kolom "Masukan *Email*" dan kolom "Masukan password baru" masing-masing dilengkapi ikon di sisi kiri sebagai petunjuk visual. Pengguna kemudian dapat menekan tombol "Ubah *Password*" berwarna biru tua dengan ikon kirim untuk memproses pemulihan akun. Di luar kartu formulir, tersedia teks "Sudah ingat kata sandi Anda?" dengan tombol "Kembali ke Login" bergaris tepi yang memudahkan pengguna kembali ke halaman Login apabila sudah mengingat kata sandinya.

IV.2.7.4 Manajemen Profile



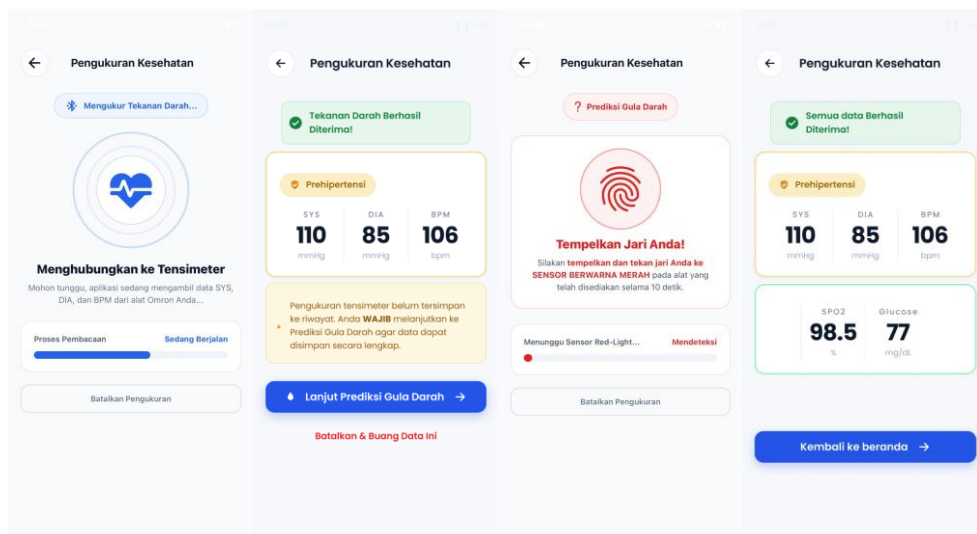
Gambar IV-17 High-Fidelity Manajemen Profile

Halaman Manajemen Profile ditampilkan setelah pengguna berhasil login, berfungsi sebagai halaman pemilihan pengguna yang akan melakukan pengukuran tekanan darah. Di bagian atas terdapat header dengan nama aplikasi "John Doe" di sisi kiri dan ikon manajemen profil di sisi kanan, diikuti judul utama "Siapa yang akan mengukur hari ini?" berukuran besar dan tebal yang langsung mengomunikasikan tujuan halaman secara jelas kepada pengguna.

Di bagian tengah halaman, profil yang sudah terdaftar ditampilkan dalam bentuk kartu bergambar dengan foto pengguna berbentuk persegi bertepi membulat, disertai nama "John" di bawahnya dalam format tebal. Di sampingnya, terdapat

kartu "Tambah Profil" dengan bingkai putus-putus dan ikon tambah di tengah, yang mengundang pengguna untuk menambahkan anggota keluarga baru sebagai profil terpisah mencerminkan fitur multiuser aplikasi ini. Di bagian bawah halaman, tersedia tombol "Kelola Profil" bergaris tepi dengan ikon pengaturan di sisi kiri, yang memungkinkan pengguna untuk mengedit, menghapus, atau mengatur profil yang sudah ada tanpa harus masuk ke pengaturan utama aplikasi.

IV.2.7.5 Measurement Flow



Gambar IV-18 High-Fidelity Measurement Flow

Saat pengguna klik *button* ambil data pada dashboard, *Measurement Flow* dimulai, halaman pertama menampilkan status "Menghubungkan ke Tensimeter..." dengan ikon denyut jantung berwarna biru yang dianimasikan di dalam lingkaran, memberikan indikasi visual bahwa aplikasi sedang melakukan pairing melalui Bluetooth dengan perangkat Omron. Di bawah ikon, terdapat teks instruksi "Mohon tunggu, aplikasi sedang mengambil data SYS, DIA, dan BPM dari alat Omron Anda" berwarna abu-abu muda yang menjelaskan proses secara transparan kepada pengguna, dilengkapi progress bar biru bertuliskan "Sedang Berjalan" agar pengguna dapat memantau kemajuan proses pembacaan data. Tombol "Batalan Pengukuran" berbentuk pill dengan gaya *outline* diletakkan di bagian bawah sebagai opsi keluar jika pengguna ingin menghentikan proses.

Setelah data berhasil terbaca, sistem akan menampilkan halaman kedua dengan notifikasi hijau "Tekanan Darah Berhasil Diterima!" di bagian atas sebagai

konfirmasi keberhasilan pengukuran. Di bawahnya, terdapat kartu berwarna kuning muda dengan label status "Prehipertensi" serta tiga nilai hasil pengukuran yang ditampilkan dalam format besar dan jelas, yaitu SYS 110 mmHg, DIA 85 mmHg, dan BPM 106 bpm, sehingga pengguna dapat langsung membaca kondisi kesehatannya tanpa perlu interpretasi tambahan. Sebuah kotak peringatan kuning muda di bawah kartu menegaskan bahwa "Pengukuran tensimeter belum tersimpan ke riwayat" dan pengguna "WAJIB" melanjutkan ke Prediksi Gula Darah agar data dapat disimpan secara lengkap, mendorong pengguna untuk menyelesaikan seluruh rangkaian pengukuran melalui tombol solid biru "Lanjut Prediksi Gula Darah" di bagian bawah, dengan opsi "Batalkan & Buang Data ini" berwarna merah sebagai jalan keluar bagi pengguna yang tidak ingin melanjutkan.

Pada tahap ketiga, halaman "Prediksi Gula Darah" menampilkan ikon sidik jari besar berwarna merah muda di tengah layar disertai instruksi tegas "Tempelkan Jari Anda!" dan penjelasan pendukung agar pengguna menempelkan serta menekan jari pada sensor berwarna merah di alat selama 10 detik. Status real-time "Menunggu Sensor *Red-Light*... Mendeteksi" beserta indikator titik merah berkedip memberikan umpan balik langsung bahwa sistem sedang aktif membaca sensor, sehingga pengguna memahami bahwa jari harus tetap menempel hingga proses selesai.

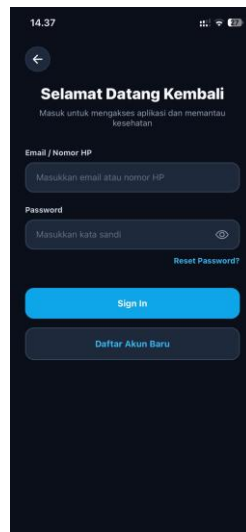
Halaman terakhir menampilkan notifikasi hijau "Semua data Berhasil Diterima!" sebagai penanda bahwa seluruh proses pengukuran telah selesai. Seluruh hasil ditampilkan dalam dua kartu ringkasan, yaitu kartu kuning muda berisi status "Prehipertensi" dengan nilai SYS, DIA, dan BPM, serta kartu hijau muda yang menampilkan hasil tambahan berupa SPO2 98.5% dan Glucose 77 mg/dL. Penyusunan seluruh data pengukuran dalam satu halaman ringkasan ini bertujuan agar pengguna dapat melihat kondisi kesehatannya secara menyeluruh dalam satu pandangan, sebelum akhirnya menekan tombol pill biru solid "Kembali ke beranda" untuk mengakhiri alur pengukuran dan kembali ke halaman utama aplikasi.

IV.3 Tahap Implementasi

Pada tahap implementasi, pengembangan aplikasi akan dilakukan sesuai dengan perencanaan dan desain dari tahap sebelumnya. Fitur-fitur yang telah dikembangkan sebagai berikut:

IV.3.1 Login

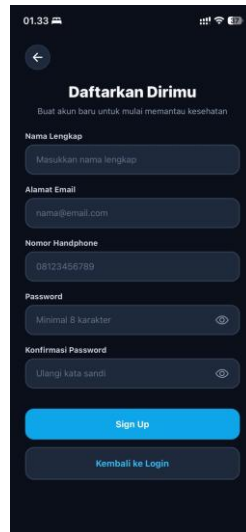
Halaman login yang sebelumnya bisa melihat langsung dashboard, sekarang diharuskan untuk login terlebih dahulu serta menambahkan *inputfield* untuk pengguna dapat login menggunakan email/nomor telepon serta mempertahankan *inputfield password*



Gambar IV-19 Implementasi Login

IV.3.2 Register

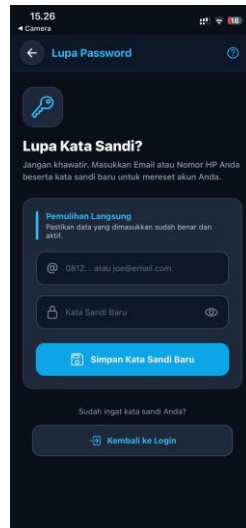
Halaman Register mengikuti halaman login. Jika pengguna belum memiliki akun bisa melakukan Register terlebih dahulu untuk mengakses halaman dashboard. Pada halaman Register terdapat *inputfield* baru yaitu nomor telepon dan tetap mempertahankan *inputfield* nama, *email*, dan *password*.



Gambar IV-20 Implementasi Register

IV.3.3 Lupa *Password*

Halaman *Lupa Password* merupakan halaman untuk mengubah password pengguna di setiap akunnya. Pada halaman ini pengguna dapat hanya mengisi kolom Nomor telepon/*Email* dan *password* barunya.

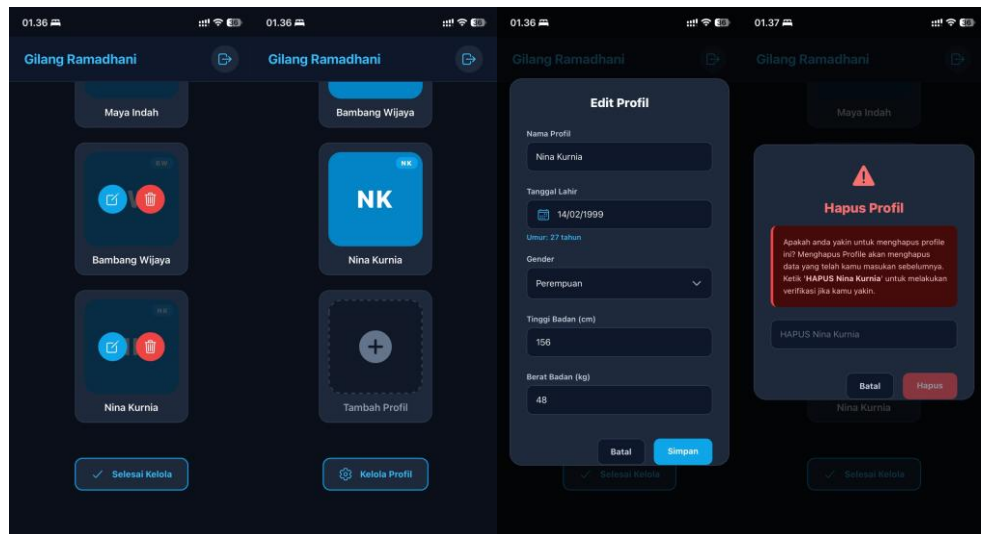


Gambar IV-21 Implementasi Lupa Password

IV.3.4 Manajemen *Profile*

Halaman Manajemen *Profile* merupakan halaman untuk mengelola *Profile* pada setiap akun yang telah dibuat. Pada halaman Manajemen *Profile*, pengguna dapat

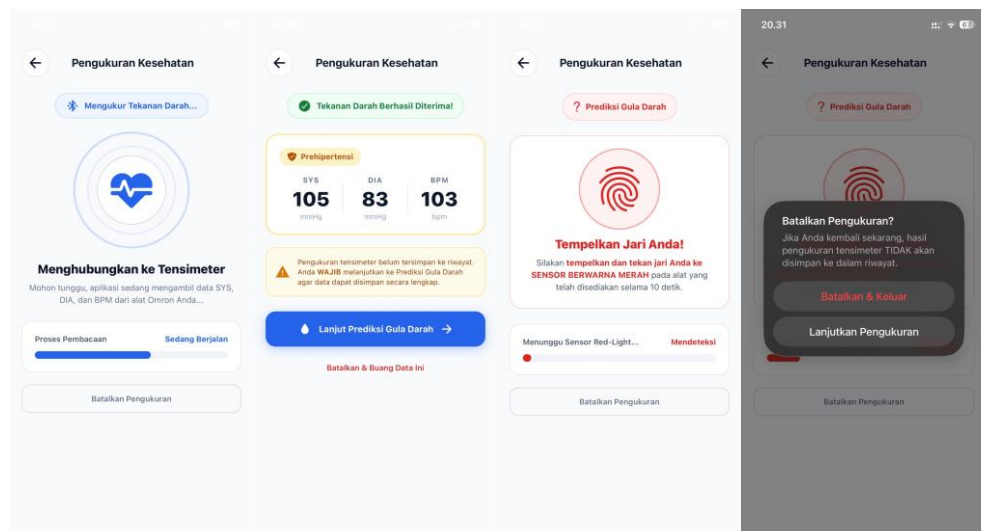
menambah, menghapus maupun *edit profile*. Pada Pembuatan *profile*, terdapat *inputfield* Nama, Tanggal Lahir, Berat Badan, dan Tinggi Badan.



Gambar IV-22 Implementasi Manajemen Profile

IV.3.5 Measurement Flow

Halaman *Measurement Flow* merupakan halaman untuk mengukur gula darah pasien. Sebelum melakukan pengukuran gula darah, pengguna diharuskan untuk mengukur sys, dia, dan bpm terlebih dahulu dan harus melanjutkannya hingga prediksi gula darah.



Gambar IV-23 Implementasi Prediksi Gula Darah

IV.4 Pengujian

Tahap pengujian merupakan tahap validasi krusial untuk memastikan bahwa seluruh sistem berjalan dengan baik pada lapisan komunikasi perangkat keras BLE, logika bisnis *server backend*, hingga antarmuka *mobile* bekerja sesuai dengan spesifikasi kebutuhan yang dirumuskan pada tahap analisis. Pengujian sistem ini dibagi menjadi dua metode pendekatan utama, yaitu *User Acceptance Testing* (UAT) beserta *Blackbox Testing* untuk mengevaluasi kelayakan *User Experience* Pengguna.

Tabel IV-8 Temuan Pengujian

No	Temuan Fase 1	Perbaikan Fase 2	Status
1.	Pengguna berharap bisa <i>login</i> menggunakan nomor telepon dikarenakan terkadang lupa dengan <i>email</i> dan <i>email</i> terlalu panjang.	Melakukan penambahan opsi <i>login</i> menggunakan nomor telepon	Berhasil
2.	Pengguna berharap ada <i>pop-up</i> konfirmasi sebelum <i>profile</i> benar-benar dihapus agar tidak salah pencet	Melakukan penambahan fitur <i>pop-up</i> untuk konfirmasi pengguna dengan harus menuliskan "HAPUS (nama_profile)"	Berhasil
3.	API mengalami <i>bottleneck</i> pada pengujian <i>load testing</i>	Melakukan <i>indexing</i> , <i>In-Memory Caching</i> , <i>Tuning SQLite PRAGMA & Connection Pool</i> , <i>Asynchronous Bcrypt Offloading</i> , <i>Asynchronous Non-Blocking Event Broadcasting</i> , <i>Direct In-Process ML Model Execution</i> , dan <i>Payload</i>	Berhasil

		<i>Compression</i>	GZip	
		<i>Middleware</i>		

IV.4.1 Blackbox Testing

Tabel IV-9 Blackbox Testing Fase 1

No	Fitur	Deskripsi	Hasil Diharapkan	Hasil Diterima	Status
1.	<i>Login</i>	<i>Input Email dan Password</i>	Berhasil <i>Login</i> dan menampilkan <i>page Manajemen Profile</i>	Berhasil login dan menampilkan Manajemen Profile	<i>Pass</i>
		<i>Input Email dan Password null</i>	Gagal <i>Login</i> dan menampilkan <i>error message</i>	Sistem meminta input ulang <i>password</i>	<i>Pass</i>
2.	Register	<i>Input Nama, email, Nomor Telepon, Password, dan Password Confirmation</i>	Berhasil <i>Register</i> dan menampilkan halaman <i>login</i>	Berhasil <i>Register</i> dan menampilkan halaman <i>login</i>	<i>Pass</i>
		Input Nama, email, Nomor Telepon, Password, dan Password Confirmation <i>null</i>	Gagal Register dan menampilkan <i>error message</i>	Sistem meminta memasukan kembali <i>inputfield</i> yang kosong	<i>Pass</i>

3.	Manajemen Profile	Menambahkan profile dengan <i>inputfield</i> Nama, Tanggal Lahir, Berat Badan, dan Tinggi Badan	Sistem menampilkan <i>profile</i> yang telah dibuat	Berhasil membuat <i>profile</i> dan melihat <i>profile</i> yang telah dibuat	<i>Pass</i>
		Menambahkan <i>profile</i> dengan <i>inputfield</i> Nama, Tanggal Lahir, Berat Badan, dan Tinggi Badan dalam bentuk String	Gagal menambahkan <i>profile</i> dan menampilkan <i>error message</i>	Sistem meminta memasukan kembali <i>inputfield</i> yang salah tipe datanya	<i>Pass</i>
		Edit <i>profile</i> dengan Edit Nama	Sistem menampilkan <i>profile</i> yang telah diedit	Sistem berhasil menampilkan <i>profile</i> yang telah diedit	<i>Pass</i>
		Edit <i>profile</i> dengan Edit Tinggi Badan dengan String	Gagal menambahkan <i>profile</i> dan menampilkan <i>error message</i>	Sistem meminta memasukan kembali <i>inputfield</i> yang salah tipe datanya	<i>Pass</i>
		Hapus <i>Profile</i>	Sistem langsung	Sistem berhasil	<i>Pass</i>

			menghapus profile	menghapus <i>profile</i>	
--	--	--	----------------------	-----------------------------	--

Tabel IV-10 *Blackbox Testing Fase 2*

No	Fitur	Deskripsi	Hasil Diharapkan	Hasil Diterima	Status
1.	<i>Login</i>	<i>Input Email dan Password</i>	Berhasil <i>Login</i> dan menampilkan <i>page Manajemen Profile</i>	Berhasil login dan menampilkan Manajemen Profile	<i>Pass</i>
		<i>Input Nomor Telepon dan Password</i>	Berhasil <i>Login</i> dan menampilkan <i>page Manajemen Profile</i>	Berhasil <i>login</i> dan menampilkan Manajemen <i>Profile</i>	<i>Pass</i>
		<i>Input Email dan Password null</i>	Gagal <i>Login</i> dan menampilkan <i>error message</i>	Sistem meminta input ulang <i>password</i>	<i>Pass</i>
2.	Register	<i>Input Nama, email, Nomor Telepon, Password, dan Password Confirmation</i>	Berhasil <i>Register</i> dan menampilkan halaman <i>login</i>	Berhasil <i>Register</i> dan menampilkan halaman <i>login</i>	<i>Pass</i>
		Input Nama, email, Nomor Telepon, Password, dan	Gagal Register dan menampilkan <i>error message</i>	Sistem meminta memasukan kembali	<i>Pass</i>

		Password Confirmation <i>null</i>		<i>inputfield</i> yang kosong	
3.	Manajemen Profile	Menambahkan <i>profile</i> dengan <i>inputfield</i> Nama, Tanggal Lahir, Berat Badan, dan Tinggi Badan	Sistem menampilkan <i>profile</i> yang telah dibuat	Berhasil membuat <i>profile</i> dan melihat <i>profile</i> yang telah dibuat	<i>Pass</i>
		Menambahkan <i>profile</i> dengan <i>inputfield</i> Nama, Tanggal Lahir, Berat Badan, dan Tinggi Badan dalam bentuk String	Gagal menambahkan <i>profile</i> dan menampilkan <i>error message</i>	Sistem meminta memasukan kembali <i>inputfield</i> yang salah tipe datanya	<i>Pass</i>
		Edit <i>profile</i> dengan Edit Nama	Sistem menampilkan <i>profile</i> yang telah diedit	Sistem berhasil menampilkan <i>profile</i> yang telah diedit	<i>Pass</i>
		Edit <i>profile</i> dengan Edit Tinggi Badan dengan String	Gagal menambahkan <i>profile</i> dan menampilkan <i>error message</i>	Sistem meminta memasukan kembali <i>inputfield</i> yang salah tipe datanya	<i>Pass</i>

		Hapus <i>Profile</i>	Sistem menampilkan <i>pop-up</i> “HAPUS {nama user}” dan di isi sesuai <i>pop-up</i> konfirmasi, kemudian sistem berhasil menghapus <i>profile</i>	Sistem berhasil menghapus <i>profile</i>	<i>Pass</i>
		Hapus <i>Profile</i>	Sistem menampilkan <i>pop-up</i> “HAPUS {nama user}” dan tidak di isi sesuai <i>pop-up</i> konfirmasi, kemudian sistem menampilkan <i>error message</i> untuk mengisi ulang	Sistem menampilkan <i>error message</i> untuk mengisi ulang	<i>Pass</i>

IV.4.2 User Acceptance Test (UAT)

Tabel IV-11 User Acceptance Test Scenario Fase 1

No	Deskripsi	Hasil Diharapkan	Hasil Aktual	Status	Catatan Pengguna
----	-----------	------------------	--------------	--------	------------------

1.	Skenario <i>Login</i> menggunakan <i>input Email</i> dan <i>Password</i>	Berhasil <i>Login</i> dan menampilkan <i>page</i> Manajemen <i>Profile</i>	Berhasil <i>login</i> dan menampilkan Manajemen <i>Profile</i>	<i>Pass</i>	Pengguna berharap bisa login menggunakan nomor telepon dikarenakan terkadang lupa dengan email dan email terlalu panjang.
2.	Skenario mengisi form register	Berhasil Register dan menampilkan halaman login	Berhasil Register dan menampilkan halaman login	<i>Pass</i>	-
3.	Menambahkan <i>profile</i> dengan <i>form</i> yang telah tersedia	Sistem menampilkan <i>profile</i> yang telah dibuat	Berhasil membuat <i>profile</i> dan melihat <i>profile</i> yang telah dibuat	<i>Pass</i>	-
4.	Edit <i>profile</i> dengan Edit Nama	Sistem menampilkan <i>profile</i> yang telah diedit	Sistem berhasil menampilkan <i>profile</i> yang telah diedit	<i>Pass</i>	-
5.	Hapus <i>Profile</i>	Sistem berhasil menghapus <i>profile</i>	Sesuai	<i>Pass</i>	Pengguna berharap ada <i>pop-up</i> konfirmasi sebelum <i>profile</i> benar-

					benar dihapus agar tidak salah pencet
--	--	--	--	--	---

Tabel IV-12 *User Acceptance Test Scenario* Fase 2

No	Deskripsi	Hasil Diharapkan	Hasil Aktual	Status	Catatan Pengguna
1.	Skenario <i>Login</i> menggunakan <i>input Email</i> dan <i>Password</i>	Berhasil <i>Login</i> dan menampilkan <i>page Manajemen Profile</i>	Berhasil <i>login</i> dan menampilkan <i>Manajemen Profile</i>	<i>Pass</i>	-
2.	Skenario <i>Login</i> menggunakan <i>input Nomor Telepon</i> dan <i>Password</i>	Berhasil <i>Login</i> dan menampilkan <i>page Manajemen Profile</i>	Berhasil <i>login</i> dan menampilkan <i>Manajemen Profile</i>	<i>Pass</i>	-
3.	Skenario mengisi form register	Berhasil Register dan menampilkan halaman login	Berhasil Register dan menampilkan halaman login	<i>Pass</i>	-
4.	Menambahkan <i>profile</i> dengan <i>form</i> yang telah tersedia	Sistem menampilkan <i>profile</i> yang telah dibuat	Berhasil membuat <i>profile</i> dan melihat <i>profile</i> yang telah dibuat	<i>Pass</i>	-
5.	Edit <i>profile</i> dengan Edit Nama	Sistem menampilkan	Sistem berhasil menampilkan	<i>Pass</i>	-

		<i>profile</i> yang telah diedit	<i>profile</i> yang telah diedit		
6.	Hapus <i>Profile</i>	Sistem menampilkan <i>pop-up</i> konfirmasi untuk pengguna menulis “HAPUS {nama_profile}”	Berhasil menghapus profile	<i>Pass</i>	-

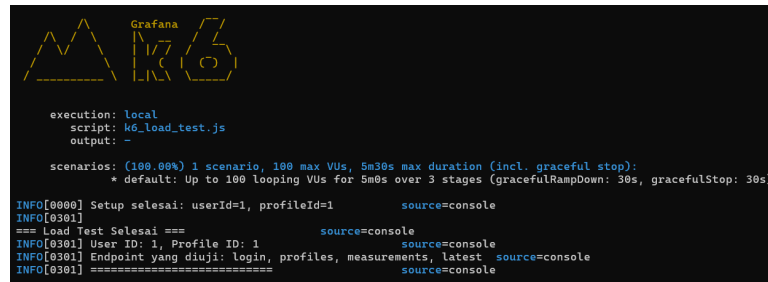
Tabel IV-13 Hasil Penilaian User Acceptance Test (UAT)

No	Pertanyaan	Nilai				
		1	2	3	4	5
1.	Apakah sistem dapat melakukan pencarian, penghubungan (<i>pairing</i>) <i>Bluetooth</i> dengan Tensimeter Omron, dan menyimpan hasil pengukuran (Sistol, Diastol, BPM) secara akurat ke profil pengguna yang dipilih?	0	0	0	2	3
2.	Apakah alur pengukuran berjalan secara berurutan tanpa kendala, mulai dari pengukuran tekanan darah hingga pengolahan data sensor SpO2/BPM untuk menghasilkan prediksi gula darah?	0	0	0	3	2
3.	Seberapa cepat dan stabil proses <i>scanning</i> serta transfer data antara aplikasi mobile dengan perangkat Tensimeter Omron saat tombol ‘Ambil data’	0	0	0	1	4
4.	Apakah aplikasi terasa cepat dan tidak mengalami <i>lag</i> atau penundaan (<i>delay</i>) yang mengganggu saat menampilkan hasil prediksi gula darah dan memperbarui <i>dashboard</i> ?	0	0	0	0	5
5.	Apakah tampilan antarmuka aplikasi (termasuk skema warna <i>Dark Mode</i> , grafik/kartu data tekanan darah, dan status koneksi) terlihat jelas, rapi, dan mudah dipahami?	0	0	1	2	2
6.	Apakah alur pembuatan akun, pemilihan profil pengguna (<i>multiuser</i>), hingga langkah-langkah pengukuran terasa intuitif dan mudah diikuti oleh pengguna baru?	0	0	2	0	3

7.	Seberapa efektif fitur <i>Multi-Profile</i> dalam membantu Anda mengelola dan memantau riwayat kesehatan beberapa anggota keluarga secara terpusat dalam satu aplikasi?	0	0	0	1	4
8.	Apakah otomatisasi penarikan data medis via Bluetooth ini secara signifikan mempermudah dan mempercepat proses pemantauan dibanding harus mencatat hasil tekanan darah secara manual?	0	0	0	0	5
9.	Apakah sistem mampu menangani kondisi kegagalan dengan baik (misalnya ketika Bluetooth terputus mendadak atau Tensimeter tidak ditemukan) dengan memberikan notifikasi/pesan peringatan yang jelas?	0	0	0	2	3
10.	Apakah Anda merasa aman dengan mekanisme perlindungan akun (seperti autentikasi <i>login</i> , enkripsi kata sandi) serta ketersediaan <i>pop-up</i> konfirmasi sebelum melakukan tindakan krusial (seperti menghapus profil)	0	0	0	0	5

IV.4.3 Load Testing

Pengujian beban (*load testing*) dilakukan menggunakan *tool* Grafana k6 dengan eksekusi secara lokal menggunakan script `k6_load_test.js` yang telah di buat. Skenario pengujian dikonfigurasi dengan maksimal 100 *Virtual Users* (VUs) dan durasi total 5 menit 30 detik (termasuk graceful stop). Proses *ramp-up* dilakukan melalui 3 tahap: tahap pertama menaikkan jumlah VU secara bertahap hingga 50 VU selama 1 menit, tahap kedua menaikkan hingga puncak 100 VU selama 3 menit, dan tahap ketiga menurunkan kembali ke 0 VU selama 1 menit. Parameter `gracefulRampDown` dan `gracefulStop` masing-masing diatur selama 30 detik untuk memastikan iterasi yang sedang berjalan dapat diselesaikan dengan baik sebelum pengujian berakhir. Fase setup berhasil dijalankan dengan membuat user uji (User ID: 1) dan profil uji (Profile ID: 1), kemudian menguji lima endpoint utama API yaitu *login*, *profiles*, *measurements*, dan *latest*.



```

Grafana
K6

execution: local
script: k6_load_test.js
output: -

scenarios: (100.00%) 1 scenario, 100 max VUs, 5m30s max duration (incl. graceful stop):
  * default: Up to 100 looping VUs for 5m0s over 3 stages (gracefulRampDown: 30s, gracefulStop: 30s)

INFO[0000] Setup selesai: userId=1, profileId=1          source=console
INFO[0301] == Load Test Selesai ==                      source=console
INFO[0301] User ID: 1, Profile ID: 1                    source=console
INFO[0301] Endpoint yang diuji: login, profiles, measurements, latest source=console
INFO[0301] =====                                     source=console
```

Gambar IV-24 Load Testing K6

Seluruh *threshold* (batas ambang performa) yang telah ditetapkan berhasil dipenuhi (ditandai dengan simbol ✓ hijau), yang menunjukkan bahwa sistem mampu beroperasi dalam batas performa yang diharapkan di bawah beban 100 VU. Secara rinci, *error_rate* tercatat sebesar 0,00%, jauh di bawah batas maksimal 5%, yang berarti tidak ada satu pun *request* yang mengalami kegagalan selama pengujian berlangsung. Untuk metrik durasi per-*endpoint*, *endpoint login* mencatat p(95) sebesar 205,14 ms (batas < 300 ms), *endpoint get profiles* sebesar 351,13 ms (batas < 400 ms), *endpoint get measurements* sebesar 347,03 ms (batas < 400 ms), *endpoint get latest measurement* sebesar 328,44 ms (batas < 400 ms), dan *endpoint save measurement* sebesar 380,25 ms (batas < 400 ms). Metrik keseluruhan *http_req_duration* juga memenuhi kedua batas yang ditetapkan dengan p(95) sebesar 333,38 ms (batas < 500 ms) dan rata-rata sebesar 113,59 ms (batas < 200 ms). Sementara itu, *http_req_failed* tercatat 0,00% (batas < 5%). Hasil ini menunjukkan bahwa seluruh *endpoint* API mampu merespons dengan waktu yang konsisten dan berada dalam batas toleransi yang diterima, bahkan ketika menerima beban konkuren dari 100 pengguna *virtual* secara simultan.

```
■ THRESHOLDS

error_rate
✓ 'rate<0.05' rate=0.00%

get_latest_measurement_duration
✓ 'p(95)<400' p(95)=328.44ms

get_measurements_duration
✓ 'p(95)<400' p(95)=347.03ms

get_profiles_duration
✓ 'p(95)<400' p(95)=351.13ms

http_req_duration
✓ 'p(95)<500' p(95)=333.38ms
✓ 'avg<200' avg=113.59ms

http_req_failed
✓ 'rate<0.05' rate=0.00%

login_duration
✓ 'p(95)<300' p(95)=205.14ms

save_measurement_duration
✓ 'p(95)<400' p(95)=380.25ms
```

Gambar IV-25 Threshold Load Testing

Pengujian *load testing* ini berhasil menjalankan total 111.000 *checks* dengan throughput 368,66 checks/detik, mencapai tingkat keberhasilan 100% tanpa satupun kegagalan (0% error rate dari 55.500 request). Seluruh validasi fungsional lulus, mencakup verifikasi status HTTP 200, keberadaan *token* pada *response login*, *format array* pada *response profiles* dan *measurements*, *field id* pada *response save measurement*, serta *field sys* pada *response latest measurement*. Dari sisi *custom metrics*, *endpoint login* tercatat paling responsif (rata-rata 73,12 ms, median 52,86 ms) dengan 11.100 login berhasil pada *throughput* 36,87 login/detik, sedangkan *endpoint save measurement* menjadi yang paling lambat (rata-rata 134,32 ms, p(95) 380,25 ms) namun tetap berada dalam batas *threshold* yang ditetapkan. Tiga *endpoint* lainnya *get latest measurement*, *get measurements*, dan *get profiles* menunjukkan performa yang relatif setara dengan rata-rata masing-masing 115,83 ms, 122,02 ms, dan 122,67 ms.

Secara keseluruhan, *http_req_duration* mencatat rata-rata 113,59 ms, median 79,38 ms, dan maksimum 844,66 ms, dengan seluruh *response* (0 dari 55.504) tergolong *expected response* tanpa adanya *error server* maupun *client*, serta total throughput HTTP mencapai 184,34 *request/detik*. Pada aspek eksekusi, setiap iterasi lengkap yang mencakup kelima *endpoint* membutuhkan rata-rata 1,62 detik (median 1,51 detik, maksimum 3,52 detik), dengan total 11.100 iterasi berhasil dijalankan menggunakan skala VU bertahap dari 1 hingga maksimum 100 VU sesuai konfigurasi stages. Dari sisi jaringan, *volume data* yang diterima sebesar 86 MB

(287 kB/detik) dan data terkirim sebesar 16 MB (51 kB/detik) menunjukkan pola lalu lintas yang wajar untuk skenario pengujian dengan 100 pengguna konkuren selama durasi 5 menit, mengindikasikan sistem mampu menangani beban tersebut secara stabil dan andal.

```

TOTAL RESULTS
checks_total.....: 111000 368.662128/s
checks_succeeded....: 100.00% 111000 out of 111000
checks_failed.....: 0.00% 0 out of 111000

✓ login: status 200
✓ login: has token
✓ profiles: status 200
✓ profiles: is array
✓ save meas: status 200
✓ save meas: has id
✓ get meas: status 200
✓ get meas: is array
✓ latest meas: status 200
✓ latest meas: has sys

CUSTOM
error_rate.....: 0.00% 0 out of 55500
get_latest_measurement_duration....: avg=115.83ms min=532µs med=83.39ms max=799.5ms p(90)=265.11ms p(95)=328.44ms
get_measurements_duration.....: avg=122.02ms min=515.9µs med=88.92ms max=822.43ms p(90)=278.86ms p(95)=347.03ms
get_profiles_duration.....: avg=122.67ms min=519.1µs med=86.09ms max=812.74ms p(90)=284.33ms p(95)=351.13ms
login_duration.....: avg=73.12ms min=524.4µs med=51.86ms max=535.76ms p(90)=164.37ms p(95)=208.14ms
save_measurement_duration.....: avg=134.32ms min=601.1µs med=97.95ms max=844.66ms p(90)=302.37ms p(95)=380.25ms
successful_logins.....: 11100 36.866213/s

HTTP
http_req_duration.....: avg=113.59ms min=515.9µs med=79.38ms max=844.66ms p(90)=265.02ms p(95)=333.38ms
{ expected_response:true }.....: avg=113.59ms min=515.9µs med=79.38ms max=844.66ms p(90)=265.02ms p(95)=333.38ms
http_req_failed.....: 0.00% 0 out of 55504
http_reqs.....: 55504 104.344349/s

EXECUTION
iteration_duration.....: avg=1.62s min=735.52ms med=1.51s max=3.52s p(90)=2.35s p(95)=2.58s
iterations.....: 11100 36.866213/s
vus.....: 1 min=1 max=100
vus_max.....: 100 min=100 max=100

NETWORK
data_received.....: 86 MB 287 kB/s
data_sent.....: 16 MB 51 kB/s

```

Gambar IV-26 *Result Load Testing*

BAB V VALIDASI HASIL DAN DISKUSI

V.1 Evaluasi Penelitian

Evaluasi penelitian dilakukan untuk memastikan bahwa sistem yang dikembangkan telah memenuhi standar kelayakan dari berbagai aspek, sebagaimana metode evaluasi yang telah ditetapkan pada Bab III. Pada penelitian ini, evaluasi dilaksanakan melalui tiga metode pengujian yang saling melengkapi, yaitu *Blackbox Testing* untuk memvalidasi kebenaran fungsional sistem, *User Acceptance Testing* (UAT) untuk mengukur tingkat penerimaan pengguna terhadap sistem, serta *Load Testing* untuk mengevaluasi keandalan dan performa sistem dalam menangani beban konkuren dari banyak pengguna secara bersamaan. Ketiga metode ini dipilih karena mampu memberikan gambaran evaluasi yang komprehensif terhadap sistem *multiuser* pemantauan tekanan darah Rumah Berpulang, baik dari sisi teknis maupun dari sisi pengalaman pengguna akhir.

Pada pengujian *Blackbox Testing*, seluruh 11 skenario pengujian yang mencakup fitur *Login* (3 skenario), *Register* (2 skenario), dan *Manajemen Profile* (6 skenario) menghasilkan status *Pass*. Hasil ini mengonfirmasi bahwa seluruh fungsionalitas sistem telah berjalan sesuai dengan spesifikasi kebutuhan fungsional yang ditetapkan, baik pada kondisi *input* valid (*normal flow*) maupun kondisi *input* *invalid* (*error handling*). Sistem terbukti mampu menangani validasi kredensial, pengelolaan data profil melalui operasi CRUD, serta menampilkan pesan kesalahan yang informatif ketika pengguna memberikan input yang tidak sesuai. Dengan tingkat kelulusan 100% pada seluruh skenario, hasil *Blackbox Testing* dapat dikategorikan baik, yang menandakan bahwa integrasi antara lapisan logika bisnis *backend* dan antarmuka *mobile* telah berfungsi sebagaimana yang dirancang.

Pada pengujian UAT, seluruh lima skenario yang diuji langsung oleh pengguna sasaran menghasilkan status *Pass*, yang mengindikasikan bahwa sistem Rumah Berpulang telah memenuhi ekspektasi pengguna dalam konteks penggunaan nyata. Selain pengujian skenario, dilakukan pula pengukuran tingkat kepuasan pengguna melalui kuesioner berskala Likert (1–5) yang terdiri dari 10 pertanyaan dan diisi

oleh 5 responden. Kuesioner tersebut mencakup aspek *pairing Bluetooth* dengan Tensimeter Omron, kecepatan dan stabilitas alur pengukuran, tampilan antarmuka, kemudahan alur *multiuser*, penanganan kegagalan koneksi, hingga keamanan akun. Berdasarkan hasil rekapitulasi, diperoleh total skor sebesar 233 dari skor maksimal 250, sehingga menghasilkan indeks persentase sebesar 93,2%. Indeks tersebut termasuk dalam kategori "Sangat Baik" pada interpretasi skala Likert.

Skor tertinggi diperoleh pada pertanyaan terkait kecepatan respons aplikasi, otomatisasi penarikan data medis via *Bluetooth*, dan keamanan akun, dengan masing-masing mencapai 100%. Sementara itu, skor terendah terdapat pada aspek kejelasan tampilan antarmuka dan kemudahan alur pembuatan akun/profil, dengan masing-masing sebesar 84%. Meskipun demikian, kedua aspek tersebut tetap berada pada kategori baik hingga sangat baik. Selain keberhasilan fungsional dan tingkat kepuasan yang tinggi tersebut, UAT pada Fase 1 menghasilkan dua temuan utama sebagaimana dirangkum pada Tabel 4.8.

Temuan pertama adalah harapan pengguna untuk dapat *login* menggunakan nomor telepon, mengingat pengguna terkadang lupa dengan alamat email yang digunakan dan merasa format email terlalu panjang untuk diingat atau diketik berulang kali. Temuan kedua adalah kebutuhan akan *pop-up* konfirmasi sebelum profil benar-benar dihapus, guna mencegah kesalahan penghapusan data akibat kesalahan sentuh (*misclick*) pada antarmuka *mobile*. Kedua umpan balik ini kemudian ditindaklanjuti pada Fase 2 pengembangan, yaitu dengan menambahkan opsi *login* menggunakan nomor telepon serta menambahkan fitur *pop-up* konfirmasi yang mewajibkan pengguna mengetikkan frasa "HAPUS (nama_profile)" sebelum proses penghapusan profil dieksekusi.

Mekanisme konfirmasi berbasis pengetikan ini dipilih karena memberikan lapisan verifikasi yang lebih eksplisit dibandingkan sekadar tombol konfirmasi biasa, sehingga risiko penghapusan data yang tidak disengaja dapat diminimalkan. Setelah perbaikan diterapkan, kedua item tersebut dinyatakan Berhasil pada evaluasi ulang. Dengan tingkat kelulusan 100% pada pengujian awal, indeks kepuasan pengguna sebesar 93,2% (kategori Sangat Baik), serta seluruh *feedback* yang telah ditindaklanjuti dan diverifikasi berhasil, hasil UAT dapat dikategorikan baik. Hal

ini menunjukkan bahwa sistem memiliki tingkat penerimaan pengguna yang tinggi, alur interaksi yang intuitif, serta proses pengembangan iteratif yang responsif terhadap kebutuhan pengguna.

Pada pengujian *Load Testing* menggunakan Grafana k6 dengan simulasi 100 *Virtual Users* secara bersamaan selama 5 menit, ditemukan *bottleneck* pada API sebagaimana tercatat pada temuan Fase 1 (Tabel 4.7). Untuk mengatasi *bottleneck* tersebut, dilakukan serangkaian perbaikan pada Fase 2 yang mencakup tujuh strategi optimasi, yaitu: (1) *indexing* pada basis data untuk mempercepat proses pencarian data; (2) *in-memory caching* untuk mengurangi beban *query* berulang ke basis data; (3) *tuning* SQLite PRAGMA dan *connection pool* untuk mengoptimalkan pengelolaan koneksi basis data; (4) *asynchronous bcrypt offloading* agar proses hashing kata sandi tidak memblokir *thread* utama; (5) *asynchronous non-blocking event broadcasting* untuk menjaga *responsivitas* sistem saat mendistribusikan *event* secara *real-time*; (6) *direct in-process ML model execution* untuk mengeliminasi *overhead* komunikasi antar-proses dalam eksekusi model *machine learning*; serta (7) *payload compression* menggunakan GZip *middleware* untuk memperkecil ukuran data yang dipertukarkan antara server dan klien. Setelah ketujuh strategi optimasi tersebut diimplementasikan, seluruh 8 *threshold* performa yang ditetapkan berhasil dipenuhi. Tingkat keberhasilan *checks* mencapai 100% (111.000 dari 111.000), tingkat kegagalan request sebesar 0,00% dari total 55.504 *request*, dan rata-rata *response time* keseluruhan sebesar 113,59 ms yang berada jauh di bawah batas *threshold* 200 ms. Endpoint login menjadi yang paling responsif dengan rata-rata 73,12 ms, sementara *endpoint save measurement* menjadi yang paling lambat dengan rata-rata 134,32 ms namun masih dalam batas toleransi yang ditetapkan. Hasil ini membuktikan bahwa strategi optimasi yang diterapkan, meliputi *indexing*, *in-memory caching*, *tuning* SQLite PRAGMA & *connection pool*, *asynchronous bcrypt offloading*, *asynchronous non-blocking event broadcasting*, *direct in-process ML model execution*, dan GZip *compression middleware*, berhasil secara sinergis mengatasi *bottleneck* yang teridentifikasi pada Fase 1 dan menjaga performa sistem pada tingkat yang optimal. Dengan seluruh *threshold* terpenuhi dan tanpa adanya kegagalan, hasil *Load Testing* dapat

dikategorikan baik. Ringkasan hasil dari ke tiga metode evaluasi disajikan pada Tabel V-1 .

Tabel V-1 Hasil Evaluasi

No	Metode Pengujian	Total Skenario/Metrik	Hasil
1.	<i>Blackbox Testing</i>	11 Skenario	11 Pass, 0 Fail
2.	<i>User Acceptance Test</i>	5 Skenario	5 Pass, 0 Fail
		10 Pertanyaan	233/250 x 100% = 93,2%
3.	<i>Load Testing</i>	8 <i>Threshold</i>	100% (0% <i>error rate</i>)

Berdasarkan hasil evaluasi dari ketiga metode pengujian tersebut disajikan pada Tabel V-1. Berdasarkan tabel tersebut, *Blackbox Testing* menghasilkan 11 Pass dari 11 skenario pengujian (tingkat kelulusan 100%), UAT menghasilkan 5 Pass dari 5 skenario pengujian disertai indeks kepuasan pengguna sebesar 93,2% (kategori Sangat Baik) dari 10 pertanyaan kuesioner, dan *Load Testing* berhasil memenuhi seluruh 8 *threshold* performa dengan tingkat keberhasilan 100% tanpa satu pun *error* (0% *error rate*). Ketiga metode evaluasi tersebut secara komprehensif memvalidasi bahwa sistem Rumah Berpulang telah memenuhi kriteria kelayakan dari tiga aspek utama, yaitu: kebenaran fungsional yang dibuktikan melalui *Blackbox Testing* dengan seluruh alur sistem berjalan sesuai spesifikasi kebutuhan; penerimaan pengguna yang dibuktikan melalui UAT dengan seluruh skenario penggunaan nyata berhasil tanpa hambatan, indeks kepuasan yang tinggi, serta *feedback* pengguna yang telah ditindaklanjuti dan diverifikasi berhasil; dan keandalan performa di bawah beban konkuren yang dibuktikan melalui *Load Testing* dengan sistem mampu menangani 100 pengguna virtual secara simultan tanpa kegagalan maupun degradasi performa. Dengan demikian, dapat disimpulkan bahwa sistem *backend* dan antarmuka *mobile* telah terintegrasi dengan baik dan siap untuk mendukung skenario operasional pada lingkungan *production* sebagai sistem *multiuser* pemantauan tekanan darah.

V.2 Diskusi

Temuan dari *Load Testing* menunjukkan bahwa rata-rata *response time* sistem sebesar 68,68 ms tergolong sangat baik jika dibandingkan dengan standar performa yang umumnya diterima untuk aplikasi web interaktif, yaitu di bawah 200 ms untuk operasi ringan dan di bawah 500 ms untuk operasi berat. Performa ini didukung oleh strategi optimasi berlapis yang mencakup *asynchronous processing* melalui *asyncio.run_in_executor*, *multi-layer in-memory caching* dengan mekanisme *invalidasi*, *database indexing* yang diselaraskan dengan pola *query* aktual, serta *PRAGMA tuning* pada SQLite. Kombinasi strategi ini secara sinergis mencegah bottleneck pada setiap titik kritis arsitektur, mulai dari komputasi bcrypt pada autentikasi hingga operasi baca-tulis pada *database*.

Jika dikaitkan dengan penelitian terdahulu, sistem yang dikembangkan dalam penelitian ini menawarkan solusi yang lebih spesifik dibandingkan dengan kajian umum IoT *healthcare* oleh Thottempudi dkk. (2025) yang masih bersifat tinjauan literatur tanpa implementasi sistem nyata. Sistem Rumah Berpulang telah berhasil mengimplementasikan dan memvalidasi arsitektur *multiuser* secara *end-to-end*, dari autentikasi hingga visualisasi data pada *dashboard mobile*. Dibandingkan dengan sistem telemonitoring oleh Debnath dkk. (2023) yang bergantung pada koneksi internet dan GSM, sistem ini memanfaatkan konektivitas BLE yang lebih efisien secara energi dan tidak memerlukan biaya operasional jaringan seluler, meskipun dengan *trade-off* berupa keterbatasan jangkauan komunikasi. Selain itu, penerapan mekanisme isolasi data berbasis profil pada penelitian ini menjawab tantangan privasi data yang diidentifikasi oleh Akoh Atadoga dkk. (2024) dan Garg dkk. (2023) sebagai salah satu isu kritis dalam implementasi IoT kesehatan.

Dari perspektif konektivitas BLE, temuan penelitian ini sejalan dengan analisis Al-Shareeda dkk. (2023) yang mengidentifikasi keunggulan BLE dalam hal konsumsi daya ultra-rendah dan kecepatan transfer data yang memadai untuk perangkat medis. Namun, tantangan keamanan yang disoroti oleh Alhussan dkk. (2025) mengenai kerentanan serangan *Man-in-the-Middle* pada protokol BLE telah dimitigasi dalam penelitian ini melalui implementasi autentikasi kunci akses pada

proses *pairing* antara *backend* dan tensimeter Omron, meskipun belum mencapai tingkat enkripsi *end-to-end* yang ideal.

Meskipun hasil evaluasi menunjukkan performa yang baik pada seluruh aspek pengujian, terdapat beberapa keterbatasan yang perlu diperhatikan. Pertama, pengujian *Load Testing* dilakukan dengan simulasi *Virtual Users* yang mengakses *endpoint* API secara langsung tanpa menyertakan latensi komunikasi BLE dengan perangkat tensimeter fisik, sehingga *response time* pada skenario penggunaan nyata yang melibatkan proses *scanning*, *pairing*, dan transfer data BLE akan lebih tinggi dibandingkan hasil pengujian. Kedua, pengujian UAT dilakukan dengan jumlah responden yang terbatas dalam lingkungan simulasi dengan dua hingga tiga perangkat Android, sehingga generalisasi terhadap populasi pengguna yang lebih luas memerlukan pengujian lanjutan dengan sampel yang lebih representatif. Ketiga, penelitian ini tidak mencakup analisis kepatuhan terhadap regulasi data kesehatan seperti HIPAA atau GDPR, sebagaimana telah dinyatakan pada batasan penelitian, sehingga aspek keamanan yang dievaluasi terbatas pada mekanisme autentikasi dan isolasi data tanpa melibatkan audit keamanan secara menyeluruh.

Berdasarkan temuan dan keterbatasan yang telah diidentifikasi, terdapat beberapa rekomendasi untuk pengembangan selanjutnya. Pertama, pengujian performa perlu diperluas dengan menyertakan simulasi *end-to-end* yang mencakup proses komunikasi BLE secara penuh untuk mendapatkan gambaran *response time* yang lebih realistis. Kedua, pengujian UAT sebaiknya dilakukan dengan melibatkan responden yang lebih beragam, termasuk pengguna lansia dan tenaga kesehatan di klinik kecil, untuk memvalidasi kemudahan penggunaan pada berbagai profil pengguna sasaran. Ketiga, integrasi mekanisme enkripsi *end-to-end* pada jalur komunikasi BLE perlu dipertimbangkan untuk meningkatkan keamanan transmisi data kesehatan, mengingat sensitivitas data yang ditransmisikan antara perangkat tensimeter dan *backend* server.

V.3 Dampak Hasil Penelitian

Penelitian ini tidak hanya menghasilkan sebuah prototipe sistem, melainkan memberikan kontribusi yang dapat dimanfaatkan secara lebih luas baik dari aspek

akademis, praktis, maupun teknologis dalam bidang Internet of Things (IoT) untuk kesehatan.

Dari aspek akademis, penelitian ini memberikan bukti empiris bahwa pendekatan autentikasi *multiuser* berbasis *backend* dapat diterapkan secara efektif pada perangkat medis berbasis *Bluetooth Low Energy* (BLE) yang pada dasarnya tidak dirancang untuk mendukung pemisahan data antar pengguna. Temuan ini memperkaya literatur di bidang *Remote Patient Monitoring* (RPM), khususnya dalam mengatasi kesenjangan yang diidentifikasi oleh Hailu dkk. (2024) dan Akoh Atadoga dkk. (2024) mengenai tantangan privasi dan isolasi data pada sistem IoT kesehatan. Kerangka arsitektur yang memisahkan lapisan autentikasi, manajemen profil, dan penyimpanan data pengukuran secara hierarkis (*user* → *profile* → *measurement*) dapat menjadi referensi bagi peneliti selanjutnya yang mengembangkan sistem serupa pada perangkat medis BLE lainnya, seperti *pulse oximeter*, *glucometer*, atau monitor EKG portabel. Selain itu, strategi optimasi performa berlapis yang terdokumentasi dalam penelitian ini meliputi *asynchronous processing*, *multi-layer caching*, *database indexing*, dan *PRAGMA tuning* menyediakan panduan teknis yang dapat direplikasi untuk meningkatkan skalabilitas sistem *backend* berbasis Python pada konteks IoT kesehatan.

Dari aspek praktis, sistem Rumah Berpulang menawarkan solusi nyata bagi permasalahan yang selama ini dihadapi oleh keluarga dengan beberapa anggota yang memerlukan pemantauan tekanan darah secara rutin menggunakan satu perangkat tensimeter yang sama. Hasil wawancara dengan dokter pada tahap analisis kebutuhan menegaskan bahwa kemampuan memisahkan data antar individu merupakan kebutuhan klinis yang esensial untuk menginterpretasikan tren tekanan darah secara akurat. Dengan tervalidasinya sistem melalui tiga metode pengujian yang menunjukkan tingkat keberhasilan 100%, penelitian ini membuktikan bahwa konsep *multiuser* pada perangkat medis personal dapat diimplementasikan tanpa memerlukan modifikasi pada *firmware* perangkat itu sendiri, melainkan cukup melalui pengelolaan pada lapisan *backend* dan aplikasi *mobile*. Pendekatan ini berpotensi menurunkan hambatan adopsi teknologi RPM di lingkungan domestik, khususnya bagi keluarga dengan anggota lansia atau individu

dengan riwayat penyakit kardiovaskular yang memerlukan pemantauan berkelanjutan.

Dari aspek teknologis, penelitian ini mendemonstrasikan bahwa kombinasi teknologi *open-source* FastAPI sebagai backend, SQLite sebagai basis data, React Native (Expo) sebagai *framework mobile*, dan k6 sebagai tool pengujian performa mampu menghasilkan sistem IoT kesehatan yang andal dengan performa rata-rata *response time* 68,68 ms dan 0% *error rate* pada beban 100 pengguna konkuren. Temuan ini memberikan *insight* bahwa pengembangan sistem RPM tidak selalu memerlukan infrastruktur yang kompleks dan mahal; dengan arsitektur yang dirancang secara tepat dan strategi optimasi yang terukur, teknologi yang dapat diakses secara luas pun mampu memenuhi standar performa yang dibutuhkan untuk aplikasi kesehatan. Kontribusi ini relevan bagi ekosistem pengembangan teknologi kesehatan di Indonesia, di mana keterjangkauan dan aksesibilitas teknologi menjadi faktor penting dalam memperluas cakupan layanan pemantauan kesehatan jarak jauh kepada masyarakat yang lebih luas.

BAB VI KESIMPULAN DAN SARAN

Pada bab ini, akan disajikan kesimpulan yang menjawab rumusan masalah berdasarkan temuan-temuan utama yang telah diuraikan sebelumnya. Selain itu, bab ini juga akan memberikan sejumlah saran konstruktif yang ditujukan untuk pengembangan sistem lebih lanjut serta sebagai acuan bagi penelitian di masa mendatang.

VI.1 Kesimpulan

Kesimpulan dari penelitian ini adalah:

1. Sistem autentikasi *multiuser* berbasis *backend* telah berhasil dirancang dan diimplementasikan menggunakan arsitektur hierarkis (*user* → *profile* → *measurement*) dengan autentikasi OAuth2/JWT pada FastAPI. Sistem ini memungkinkan beberapa pengguna menggunakan satu tensimeter Omron HEM-7155T secara bergantian melalui manajemen sesi BLE di sisi *backend* tanpa modifikasi *firmware* perangkat. Hasil *Blackbox Testing* menunjukkan tingkat kelulusan 100% dari 11 skenario pengujian, yang mengonfirmasi seluruh fungsionalitas autentikasi dan manajemen pengguna berjalan sesuai spesifikasi.
2. Isolasi data antar pengguna telah berhasil diimplementasikan melalui mekanisme kontrol akses berbasis *token* yang memastikan setiap pengguna hanya dapat mengakses data pengukuran miliknya sendiri berdasarkan *id_user* dan *id_profile*. Validasi data otomatis menggunakan Pydantic pada FastAPI turut mencegah potensi pencampuran data pengukuran tekanan darah dalam skenario *multiuser*.
3. Efektivitas tampilan sistem dalam hal kemudahan penggunaan telah dievaluasi melalui *User Acceptance Testing* (UAT) dengan 5 responden, menghasilkan indeks kepuasan sebesar 93,2% (kategori "Sangat Baik") dari kuesioner berskala Likert. Skor tertinggi (100%) diperoleh pada aspek kecepatan respons, otomatisasi penarikan data via *Bluetooth*, dan keamanan akun. Umpan balik pengguna berupa penambahan opsi *login* nomor telepon

dan pop-up konfirmasi penghapusan profil juga telah berhasil ditindaklanjuti pada Fase 2 pengembangan.

4. Arsitektur terpisah untuk pengolahan dan distribusi data real-time telah berhasil dirancang menggunakan FastAPI (ASGI), SQLite, dan React Native (Expo). Validasi melalui *Load Testing* k6 dengan 100 *Virtual Users* menunjukkan seluruh 8 threshold performa terpenuhi, dengan *error rate* 0,00%, rata-rata *response time* 113,59 ms (di bawah batas 200 ms), dan tingkat keberhasilan checks 100% setelah penerapan tujuh strategi optimasi pada Fase 2.

VI.2 Saran

Berdasarkan hasil penelitian dan keterbatasan yang telah diidentifikasi, berikut adalah saran untuk pengembangan dan penelitian selanjutnya:

1. Memperluas skenario pengujian performa secara *end-to-end* yang mencakup latensi komunikasi BLE (proses *scanning*, *pairing*, dan transfer data) dengan perangkat tensimeter fisik, sehingga menghasilkan gambaran *response time* yang lebih realistis.
2. Melibatkan responden UAT yang lebih beragam dan dalam jumlah yang lebih besar, termasuk pengguna lansia dan tenaga kesehatan di klinik kecil, guna meningkatkan generalisasi hasil pengujian kemudahan penggunaan.
3. Mengimplementasikan mekanisme enkripsi *end-to-end* pada jalur komunikasi BLE untuk meningkatkan keamanan transmisi data kesehatan antara perangkat tensimeter dan *backend server*.
4. Melakukan analisis kepatuhan terhadap regulasi data kesehatan melalui audit keamanan secara menyeluruh, khususnya jika sistem akan diimplementasikan pada skala yang lebih luas.
5. Mengeksplorasi kompatibilitas arsitektur *multiuser* yang dikembangkan untuk perangkat medis BLE lainnya, seperti *pulse oximeter*, *glucometer*, atau monitor EKG portabel, guna memperluas cakupan pemantauan kesehatan jarak jauh.

DAFTAR PUSTAKA

- Abdillah, M. T., Kurniastuti, I., Susanto, A., Yudianto, F., Studi, P., Informasi, S., Bisnis, E., & Digital, D. T. (2023). Implementasi Black box Testing dan Usability Testing pada Website Sekolah MI Miftahul Ulum Warugunung Surabaya. *Jurnal Ilmu Komputer dan Desain Komunikasi Visual*, 8(1).
- Abdulmalek, S., Nasir, A., Jabbar, W. A., Almhaya, M. A. M., Bairagi, A. K., Khan, M. A. M., & Kee, S. H. (2022). IoT-Based Healthcare-Monitoring System towards Improving Quality of Life: A Review. Dalam *Healthcare (Switzerland)* (Vol. 10, Nomor 10). MDPI. <https://doi.org/10.3390/healthcare10101993>
- Agnoletto, D., Jonsson, M., & de Freitas, E. P. (2020). Time Slot Transmission Scheme with Packet Prioritization for Bluetooth Low Energy Devices used in Real-Time Applications. *International Journal of Wireless Information Networks*, 27(4), 518–534. <https://doi.org/10.1007/s10776-020-00494-4>
- Akbar, M. A., Sang, J., Khan, A. A., Fazal-E-Amin, Nasrullah, Shafiq, M., Hussain, S., Hu, H., Elahi, M., & Xiang, H. (2017). Improving the quality of software development process by introducing a new methodology-Az-model. *IEEE Access*, 6, 4811–4823. <https://doi.org/10.1109/ACCESS.2017.2787981>
- Akoh Atadoga, Toritsemogba Tosanbami Omaghomi, Oluwafunmi Adijat Elufioye, Ifeoma Pamela Odilibe, Andrew Ifesinachi Daraojimba, & Oluwaseyi Rita Owolabi. (2024). Internet of Things (IoT) in healthcare: A systematic review of use cases and benefits. *International Journal of Science and Research Archive*, 11(1), 1511–1517. <https://doi.org/10.30574/ijrsra.2024.11.1.0243>
- Alam, I., Sarwar, N., & Noreen, I. (2022). Statistical analysis of software development models by six-pointed star framework. *PLoS ONE*, 17(4 April). <https://doi.org/10.1371/journal.pone.0264420>

- ALazzawi, A., Yas, Q. M., & Rahmatullah, B. (2023). A Comprehensive Review of Software Development Life Cycle methodologies: Pros, Cons, and Future Directions. *Iraqi Journal for Computer Science and Mathematics*, 4(4), 173–190. <https://doi.org/10.52866/ijcsm.2023.04.04.014>
- Alhussan, M., Boem, F., Ghoreishizadeh, S. S., & Mandalari, A. M. (2025). Hacking Health: Unveiling Vulnerabilities in BLE-Enabled Wearable Sensor Nodes. *Proceedings - IEEE International Symposium on Circuits and Systems*. <https://doi.org/10.1109/ISCAS56072.2025.11043235>
- Aliyah Aliyah, Nahrin Hartono, & Asrul Azhari Muin. (2024). Penggunaan User Acceptance Testing (UAT) Pada Pengujian Sistem Informasi Pengelolaan Keuangan Dan Inventaris Barang. *Switch : Jurnal Sains dan Teknologi Informasi*, 3(1), 84–100. <https://doi.org/10.62951/switch.v3i1.330>
- Al-Shareeda, M. A., Saare, M. A., Manickam, S., & Karuppayah, S. (2023). Bluetooth low energy for internet of things: review, challenges, and open issues. *Indonesian Journal of Electrical Engineering and Computer Science*, 31(2), 1182–1189. <https://doi.org/10.11591/ijeecs.v31.i2.pp1182-1189>
- Aniko, A. R., Kusumasari, T. F., & Suakanto, S. (2024). Application of User Experience Method to Determine User Requirements in Remote Patient Monitoring Systems: Systematic Literature Review. *11th International Conference on ICT for Smart Society: Integrating Data and Artificial Intelligence for a Resilient and Sustainable Future Living, ICISS 2024 - Proceeding*. <https://doi.org/10.1109/ICISS62896.2024.10751221>
- Boima, V., Doku, A., Agyekum, F., Tuglo, L. S., & Agyemang, C. (2024). *Effectiveness of digital health interventions on blood pressure control, lifestyle behaviours and adherence to medication in patients with hypertension in low-income and middle-income countries: a systematic review and meta-analysis of randomised controlled trials*. www.thelancet.com
- Chakraborty, D. S., Lahiry, S., & Choudhury, S. (2021). Hypertension Clinical Practice Guidelines (ISH, 2020): What Is New? *Medical Principles and Practice*, 30(6), 579–584. <https://doi.org/10.1159/000518812>

- Cheng, S. A., Tan, S. I., Goh, S. L. E., & Ko, S. Q. (2025). The Value of Remote Vital Signs Monitoring in Detecting Clinical Deterioration in Patients in Hospital at Home Programs or Postacute Medical Patients in the Community: Systematic Review. *Journal of Medical Internet Research*, 27. <https://doi.org/10.2196/64753>
- Chowdhury, A. E., Bhowmik, A., Hasan, H., & Rahim, M. S. (2020). Analysis of the Veracities of Industry Used Software Development Life Cycle Methodologies. *AIUB Journal of Science and Engineering (AJSE)*, 16(2). <https://doi.org/10.53799/ajse.v16i2.71>
- Debnath, P., Mahmud, A., Hossain, A. K., & Rahman, S. M. I. (2023). Design and Application of IOT-Based Real-Time Patient Telemonitoring System Using Biomedical Sensor Network. *SN Computer Science*, 4(2). <https://doi.org/10.1007/s42979-022-01516-z>
- Desai, M., Tardif-Douglin, M., Miller, I., Blitzer, S., Gardner, D. L., Thompson, T., Edmondson, L., & Levine, D. M. (2024). Implementation of Agile in healthcare: methodology for a multisite home hospital accelerator. *BMJ Open Quality*, 13(2), e002764. <https://doi.org/10.1136/bmjoq-2024-002764>
- Dias, M., Silva, L., Folgado, D., Nunes, M. L., Cepeda, C., Cheetham, M., & Gamboa, H. (2023). Cardiovascular load assessment in the workplace: A systematic review. Dalam *International Journal of Industrial Ergonomics* (Vol. 96). Elsevier B.V. <https://doi.org/10.1016/j.ergon.2023.103476>
- Doe, N., Scharoba, S., Reichenbach, M., & Herglotz, C. (2025). *Comparison of Bluetooth Low Energy (BLE), Wi-Fi, Serial and 5G in IoMT*. 859–866. <https://doi.org/10.5220/0013321500003911>
- Fujiwara, T., Sheppard, J. P., Hoshide, S., Kario, K., & McManus, R. J. (2023). Medical Telemonitoring for the Management of Hypertension in Older Patients in Japan. Dalam *International Journal of Environmental Research and Public Health* (Vol. 20, Nomor 3). MDPI. <https://doi.org/10.3390/ijerph20032227>

- Garg, P., Yadav, B., Gupta, S., & Gupta, B. (2023). Performance Analysis and Optimization of Cross Platform Application Development Using React Native. *Lecture Notes in Networks and Systems*, 615 LNNS, 559–567. https://doi.org/10.1007/978-981-19-9304-6_51
- Garg, T., Kagalwalla, N., Puthran, S., Churi, P., & Pawar, A. (2023). A novel approach of privacy-preserving data sharing system through data-tagging with role-based access control. *World Journal of Engineering*, 20(1), 12–28. <https://doi.org/10.1108/WJE-04-2021-0218>
- Hailu, R., Sousa, J., Tang, M., Mehrotra, A., & Uscher-Pines, L. (2024). Challenges and Facilitators in Implementing Remote Patient Monitoring Programs in Primary Care. *Journal of General Internal Medicine*, 39(13), 2471–2477. <https://doi.org/10.1007/s11606-023-08557-x>
- Hendayun, M., Ginanjar, A., & Ihsan, Y. (2023). ANALYSIS OF APPLICATION PERFORMANCE TESTING USING LOAD TESTING AND STRESS TESTING METHODS IN API SERVICE. *JURNAL SISFOTEK GLOBAL*, 13(1), 28. <https://doi.org/10.38101/sisfotek.v13i1.2656>
- Ho, K. L., Karwi, Q. G., Connolly, D., Pherwani, S., Ketema, E. B., Ussher, J. R., & Lopaschuk, G. D. (t.t.). *Metabolic, structural and biochemical changes in diabetes and the development of heart failure*. <https://doi.org/10.1007/s00125-021-05637-7>/Published
- Huda, M. N., Burhan, M., Satibi, A., Pradita, H. A., Saifudin, A., & Kusyadi, I. (2022). Implementasi Black Box Testing pada Aplikasi Sistem Kasir dengan Menggunakan Teknik Equivalence Partitions. *Jurnal Teknologi Sistem Informasi dan Aplikasi*, 5(2), 120. <https://doi.org/10.32493/jtsi.v5i2.17645>
- Hymczak, H., Gołab, A., Mendrala, K., Plicner, D., Darocha, T., Podsiadło, P., Hudziak, D., Gocoł, R., & Kosiński, S. (2021). Core temperature measurement—principles of correct measurement, problems, and complications. *International Journal of Environmental Research and Public Health*, 18(20). <https://doi.org/10.3390/ijerph182010606>

- Islam, M. M., Rahaman, A., & Islam, M. R. (2020). Development of Smart Healthcare Monitoring System in IoT Environment. *SN Computer Science*, 1(3). <https://doi.org/10.1007/s42979-020-00195-y>
- Ivanov, R. (2026). Scalable IoT-Based Architecture for Continuous Monitoring of Patients at Home: Design and Technical Validation. *Computers*, 15(3). <https://doi.org/10.3390/computers15030144>
- Jena, S., & Gupta, A. (2021). Review on pressure sensors: a perspective from mechanical to micro-electro-mechanical systems. Dalam *Sensor Review* (Vol. 41, Nomor 3, hlm. 320–329). Emerald Group Holdings Ltd. <https://doi.org/10.1108/SR-03-2021-0106>
- Khamdani, A. R., Muslikh, A. R., & Affandi, A. S. (2025). COMPARATIVE ANALYSIS OF PERFORMANCE AND EFFICIENCY OF LOAD BALANCING ALGORITHMS ON INGRESS CONTROLLER. *Jurnal Teknik Informatika (Jutif)*, 6(1), 453–468. <https://doi.org/10.52436/1.jutif.2025.6.1.4040>
- Kim, M., & Paek, J. (2025). Multi-Connection Scheduling with Fair Resource Management for Scalable Bluetooth Low Energy Networks. *IEEE Internet of Things Journal*. <https://doi.org/10.1109/JIOT.2025.3639570>
- Kokol, P., Blažun Vošner, H., Kokol, M., & Završnik, J. (2022). Role of Agile in Digital Public Health Transformation. *Frontiers in Public Health*, 10. <https://doi.org/10.3389/fpubh.2022.899874>
- Koulouras, G., Katsoulis, S., & Zantalis, F. (2025). Evolution of Bluetooth Technology: BLE in the IoT Ecosystem. Dalam *Sensors* (Vol. 25, Nomor 4). Multidisciplinary Digital Publishing Institute (MDPI). <https://doi.org/10.3390/s25040996>
- Li, Y., Du, Z., Fu, Y., & Liu, L. (2022). Role-Based Access Control Model for Inter-System Cross-Domain in Multi-Domain Environment. *Applied Sciences (Switzerland)*, 12(24). <https://doi.org/10.3390/app122413036>

- Liu, C., Zhang, Y., & Zhou, H. (2021). A comprehensive study of bluetooth low energy. *Journal of Physics: Conference Series*, 2093(1). <https://doi.org/10.1088/1742-6596/2093/1/012021>
- Mabotha, E., Mabunda, N. E., & Ali, A. (2025). A Dockerized Approach to Dynamic Endpoint Management for RESTful Application Programming Interfaces in Internet of Things Ecosystems. *Sensors*, 25(10). <https://doi.org/10.3390/s25102993>
- Mackwood, M., Pashchenko, O., Leggett, C., Fontanet, C., Skinner, J., & Fisher, E. (2024). Telehealth Trends and Hypertension Management Among Rural and Medicaid Patients After COVID-19. *Telemedicine and e-Health*, 30(6), e1677–e1688. <https://doi.org/10.1089/tmj.2023.0628>
- Mclenna Australia, E., & Onotu Fache, V. (2023). *MEDITERRANEAN PUBLICATION AND RESEARCH INTERNATIONAL EDITOR-IN-CHIEF BLUETOOTH LOW ENERGY: A COMPREHENSIVE TECHNOLOGY ANALYSIS AND ROLE IN INTERNET OF THINGS APPLICATIONS MPRI-JNSAR EDITOR-IN-CHIEF*.
- Montoya-Murillo, D. A., Galvan-Cruz, S., Mora, M., & Andrade, E. L. M. (2025). A Comprehensive Review of the Main Heavyweight and Lightweight SDLC for Big Data Analytics Systems (BDAS). *IEEE Access*, 13, 101328–101367. <https://doi.org/10.1109/ACCESS.2025.3577970>
- Pangestika, A., & Widiarti, L. W. (2025). IMPLEMENTASI PENGUJIAN MENGGUNAKAN TEKNIK EQUIVALENTS PARTITIONS PADA APLIKASI TAPERA DIGITAL SERVICES IMPLEMENTATION OF TESTING USING EQUIVALENT PARTITIONS TECHNIQUE ON TAPERA DIGITAL SERVICES. Dalam *Jurnal Sistem Informasi Dan Bisnis Cerdas* (Vol. 18, Nomor 1).
- Park, E., Kim, H.-S., & Bahk, S. (2024). BLEX: Flexible Multi-Connection Scheduling for Bluetooth Low Energy. *ACM International Conference Proceeding Series*, 349–354. <https://doi.org/10.1145/1122445.1122456>

- Raj, U. (2021). Bluetooth Low Energy: A Comprehensive Wireless Technology. *International Journal of Advance Research and Innovation*, 9(3), 64–69. <https://doi.org/10.51976/ijari.932110>
- Rath, S. P., Jain, N. K., Tomer, G., & Singh, A. K. (2025). A systematic literature review of agile software development projects. Dalam *Information and Software Technology* (Vol. 182). Elsevier B.V. <https://doi.org/10.1016/j.infsof.2025.107727>
- Sassa, A. C., Almeida, I. A. de, Pereira, T. N. F., & Oliveira, M. S. de. (2023). Scrum: A Systematic Literature Review. *International Journal of Advanced Computer Science and Applications*, 14(4). <https://doi.org/10.14569/IJACSA.2023.0140420>
- Shaffer, F., Meehan, Z. M., & Zerr, C. L. (2020). A Critical Review of Ultra-Short-Term Heart Rate Variability Norms Research. Dalam *Frontiers in Neuroscience* (Vol. 14). Frontiers Media S.A. <https://doi.org/10.3389/fnins.2020.594880>
- Shafí, I., Din, S., Farooq, S., de la Torre Díez, I., Breñosa, J., Martínez Espinosa, J. C., & Ashraf, I. (2024). Design and development of patient health tracking, monitoring and big data storage using Internet of Things and real time cloud computing. *PLoS ONE*, 19(3 March). <https://doi.org/10.1371/journal.pone.0298582>
- Shaheen, A., Alzyadat, W., Alhroob, A., & Asfour, A. N. (2025). Incremental prioritization using an iterative model for small-scale systems. *International Journal of Informatics and Communication Technology (IJ-ICT)*, 14(2), 565–574. <https://doi.org/10.11591/ijict.v99i1.pp565-574>
- Smith, W., Colbert, B. M., Namouz, T., Caven, D., Ewing, J. A., & Albano, A. W. (2024). *Remote Patient Monitoring Is Associated with Improved Outcomes in Hypertension: A Large, Retrospective, Cohort Analysis*. <https://doi.org/10.3390/healthcare>
- Suneetha, T., & Singhal, A. (2024). *INTERNATIONAL JOURNAL OF TRENDS IN EMERGING RESEARCH AND DEVELOPMENT* Securing patient data: An

empirical evaluation of role-based access control in healthcare environments
(Vol. 2). <https://researchtrendsjournal.com>

Syaifudin, A., Rusmana, I., & Aliyu, A. (2020). SISTEM PEMANTAUAN TANDA VITAL MANUSIA. Dalam *JMTE* (Vol. 01, Nomor 01).

Takahashi, H., Yamashita, S., & Yakura, N. (2024). Validating the Accuracy of Omron HEM-1026 (HCR-1901T2/HCR-1902T2): Blood Pressure Monitoring Device at Home, According to the Universal Protocol. *Vascular Health and Risk Management*, 20, 333–339. <https://doi.org/10.2147/VHRM.S468382>

Vanesha, N. A., Rizky, R., & Purwanto, A. (2024). Comparison Between Usability and User Acceptance Testing on Educational Game Assessment. *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, 13(2), 210–215. <https://doi.org/10.32736/sisfokom.v13i2.2099>

Wadkar, V., & Ghorpade, S. (2024). *A Comparative Analysis of Software Life Cycle Models: Phases, Activities, and Order of Execution*.

Wibawa, Y. E., & Naufal, M. (2023). PEMBANGUNAN PERANGKAT LUNAK KOMUNITAS MEDIA MUSIK DENGAN FRAMEWORK REACT NATIVE DAN FIREBASE BERBASIS ANDROID DAN IOS. *Jurnal SISKOM-KB (Sistem Komputer dan Kecerdasan Buatan)*, 6(2), 160–170. <https://doi.org/10.47970/siskom-kb.v6i2.515>

Wulandari Putri Bahmin, A. I., Muhammad Rizal H, Nur Cahaya Indah, & Audi Salsabila B. (2025). Evaluasi Pengujian Penerimaan Pengguna (User Acceptance Testing) pada Sistem Informasi Akademik Universitas Teknologi AKBA Makassar. *Inventor: Jurnal Inovasi dan Tren Pendidikan Teknologi Informasi*, 3(2), 50–59. <https://doi.org/10.37630/inventor.v3i2.2525>

Yi, W.-Y., Liu, P.-F., Lo, S.-L., Chan, Y.-F., Zhou, Y., Leung, Y., Woo, K.-S., Lee, A. P.-W., Chen, J.-M., & Leung, K.-S. (2022). *Towards Personalized Healthcare in Cardiac Population: The Development of a Wearable ECG Monitoring System, an ECG Lossy Compression Schema, and a ResNet-Based AF Detector*. <http://arxiv.org/abs/2207.05138>

Yulius Darmawan, A., Lestariningsih, D., Angka, P. R., Agustine, L., & Katolik
Widya Mandala Surabaya, U. (2022). PERANCANGAN TENSIMETER
DIGITAL DAN PENGIRIMAN DATA KE MONITORING PUSAT. Dalam
Scientific Journal Widya Teknik (Vol. 21, Nomor 2).

LAMPIRAN

LAMPIRAN ADOKUMENTASI WAWANCARA



Lampiran A-1 Dokter Spesialis Jantung



Lampiran A-2 Kepala IGD

LAMPIRAN B DOKUMENTASI USER ACCEPTANCE TEST



Lampiran B-1 UAT User 1



Lampiran B-2 UAT User 2



Lampiran B-3 UAT User 3



Lampiran B-4 UAT User 4

LAMPIRAN C LINK GITHUB

<https://github.com/TA-Remote-Patient-Monitoring/Flowbeat-RN-Expo.git>

<https://github.com/TA-Remote-Patient-Monitoring/Tensimeter-and-MAX30102.git>