

**RANCANG BANGUN ALAT KESEHATAN PENGUKUR DAN
SISTEM MONITORING DENYUT NADI, SUHU, DAN
TEKANAN DARAH BERBASIS *INTERNET OF THINGS* (IOT)**

PROYEK AKHIR

Laporan akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat kelulusan
Sarjana Terapan Politeknik Manufaktur Negeri Bangka Belitung



Disusun Oleh:

Hardi Perwira NIM: 1052241

Merizta Aulia NIM: 1052246

POLITEKNIK MANUFAKTUR NEGERI

BANGKA BELITUNG

TAHUN 2025

LEMBAR PENGESAHAN

JUDUL PROYEK AKHIR

RANCANG BANGUN ALAT KESEHATAN PENGUKUR DAN SISTEM MONITORING DENYUT NADI, SUHU, DAN TEKANAN DARAH BERBASIS *INTERNET OF THINGS* (IOT)

Oleh:

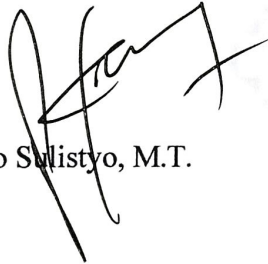
Hardi Perwira NIM: 1052241

Merizta Aulia NIM: 1052246

Laporan akhir ini telah disetujui dan disahkan sebagai salah satu syarat kelulusan
Program Sarjana Terapan Politeknik Manufaktur Negeri Bangka Belitung

Menyetujui,

Pembimbing 1



Eko Sulistyono, M.T.

Pembimbing 2



Evvin Faristasari, M.Sc.

Penguji 1



Indra Dwisaputra, M.T.

Penguji 2



Riztamala Diana, M.Tr.T.

PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan di bawah ini:

Nama Mahasiswa 1 : Hardi Perwira NIM: 1052241

Nama Mahasiswa 2 : Merizta Aulia NIM: 1052246

Dengan Judul : Rancang Bangun Alat Kesehatan Pengukur dan
Sistem Monitoring Denyut Nadi, Suhu, dan Tekanan
Darah Berbasis *Internet of Things* (IoT)

Menyatakan bahwa laporan akhir ini adalah hasil kerja kami sendiri dan bukan merupakan plagiat. Pernyataan ini kami buat dengan sebenarnya dan bila ternyata dikemudian hari ternyata melanggar pernyataan ini, kami bersedia menerima sanksi yang berlaku.

Sungailiat, 16 Juli 2025

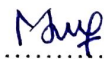
Nama Mahasiswa

Tanda Tangan

1. Hardi Perwira

.....


2. Merizta Aulia

.....


ABSTRAK

Kesehatan manusia sangat bergantung pada fungsi alat vital, meliputi denyut nadi, suhu, dan tekanan darah. Ketiga parameter ini memainkan peran penting dalam menentukan kondisi kesehatan secara keseluruhan. Penelitian ini mengembangkan alat pengukuran dan sistem monitoring denyut nadi, suhu, serta tekanan darah yang bersifat portable dan berbasis Internet of Things (IoT) yang bisa beroperasi secara bersamaan. Alat pengukuran denyut nadi, suhu, dan tekanan darah berbasis Internet of Things (IoT) dikontrol menggunakan Arduino Uno R3 ATmega328P dan ESP32, serta dilengkapi dengan tiga sensor, yaitu pulse sensor digunakan untuk mengukur denyut nadi, sensor MLX90614 digunakan untuk mengukur suhu tubuh, dan sensor MPX5700DP digunakan untuk mengukur tekanan darah. Hasil pengukuran ditampilkan secara real-time pada layar LCD TFT 3,5 inci, serta dimonitoring melalui server Blynk dan data disimpan secara offline menggunakan Firebase. Hasil pengujian menunjukkan tingkat akurasi untuk pengukuran denyut nadi sebesar 85,46%, pengukuran suhu sebesar 98,97%, dan pengukuran tekanan darah untuk sistolik sebesar 91,16% dan untuk diastolik sebesar 81,58%. Berdasarkan hasil tersebut, dapat disimpulkan bahwa alat ukur ini memiliki tingkat keakuratan yang cukup serta menawarkan efisiensi dalam pengukuran karena seluruh parameter dapat diukur secara terintegrasi dalam satu perangkat.

Kata kunci: Arduino Uno R3 ATmega328P, ESP32, Internet of Things, MLX90614, MPX5700DP, Pulse Sensor, Sistem Monitoring

ABSTRACT

Human health is highly dependent on the functioning of vital organs, including pulse rate, temperature and blood pressure. These three parameters play an important role in determining overall health conditions. This research develops a portable, Internet of Things (IoT)-based pulse, temperature, and blood pressure measurement and monitoring system that can operate simultaneously. The Internet of Things (IoT)-based pulse, temperature, and blood pressure measurement tool is controlled using Arduino Uno R3 ATmega328P and ESP32, and is equipped with three sensors, namely the pulse sensor used to measure pulse rate, MLX90614 sensor used to measure body temperature, and MPX5700DP sensor used to measure blood pressure. The measurement results are displayed in real-time on a 3.5-inch TFT LCD screen, and monitored through the Blynk server and the data is stored offline using Firebase. The test results show the accuracy rate for pulse measurement is 85.46%, temperature measurement is 98.97%, and blood pressure measurement for systolic is 91.16% and for diastolic is 81.58%. Based on these results, it can be concluded that this measuring instrument has a sufficient level of accuracy and offers efficiency in measurement because all parameters can be measured in an integrated manner in one device.

Keywords: Arduino Uno R3 ATmega328P, ESP32, Internet of Things, MLX90614, MPX5700DP, Pulse Sensor, Monitoring System

KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa, yang telah melimpahkan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan laporan proyek akhir ini yang berjudul **“Rancang Bangun Alat Kesehatan Pengukur dan Sistem Monitoring Denyut Nadi, Suhu, dan Tekanan Darah Berbasis *Internet of Things* (IoT)”** dengan baik dan tepat waktu.

Adapun tujuan penulisan laporan proyek akhir ini adalah untuk memenuhi salah satu syarat kelulusan dalam Program Sarjana Terapan di Politeknik Manufaktur Negeri Bangka Belitung. Laporan ini membahas mengenai alat yang dapat mengukur denyut nadi, suhu, dan tekanan darah. Alat ini dirancang untuk memudahkan proses pemantauan kesehatan menggunakan teknologi berbasis *Internet of Things* (IoT), sehingga informasi mengenai data hasil pengukuran dapat dilihat secara *real-time* dan bisa diakses dari jarak jauh. Dengan adanya alat ini, diharapkan dapat memberikan manfaat yang luas dan kemudahan dalam penggunaannya, khususnya dalam bidang kesehatan.

Dalam penyusunan dan pembuatan proyek akhir ini, penulis tidak terlepas dari dukungan berbagai pihak yang telah memberikan semangat, motivasi, serta masukan yang sangat berharga. Oleh karena itu, penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapak I Made Andik Setiawan, M.Eng., Ph.D. selaku Direktur Politeknik Manufaktur Negeri Bangka Belitung.
2. Bapak Zanu Saputra, M.Tr.T. selaku Ketua Jurusan Rekayasa Elektro dan Industri Pertanian Politeknik Manufaktur Negeri Bangka Belitung.
3. Bapak Aan Febriansyah, M.T. selaku Ketua Program Studi D-IV Teknik Elektronika Politeknik Manufaktur Negeri Bangka Belitung.
4. Bapak Eko Sulisty, M.T. selaku Dosen Pembimbing 1, atas bimbingan, ilmu, serta masukan berharga dalam penyelesaian proyek akhir ini.
5. Ibu Evvin Faristasari, M.Sc. selaku Dosen Pembimbing 2, atas bimbingan, arahan, serta kritik dan saran yang membangun.

6. Bapak Surojo, M.T. selaku Dosen Wali Kelas STE B Politeknik Manufaktur Negeri Bangka Belitung.
7. Seluruh tenaga pendidik dan kependidikan Politeknik Manufaktur Negeri Bangka Belitung.
8. Orang tua penulis, yang selalu memberikan dukungan, baik secara moral maupun materiil, serta senantiasa mendoakan yang terbaik. Motivasi, saran, dan masukan yang diberikan menjadi dorongan bagi penulis dalam menyelesaikan laporan proyek akhir ini dengan hasil yang maksimal.
9. Teman-teman seperjuangan, khususnya kelas STE B, yang telah memberikan dukungan, motivasi, serta masukan yang berharga dalam penyelesaian laporan proyek akhir ini. Kebersamaan dan semangat yang terjalin selama ini menjadi inspirasi bagi penulis untuk menyelesaikan tugas ini dengan baik.

Penulis menyadari bahwa laporan ini masih memiliki kekurangan dan jauh dari kata sempurna. Oleh karena itu, penulis sangat menerima masukan, kritik, dan saran yang membangun sangat diharapkan untuk penyempurnaan laporan ini di masa mendatang. Penulis berharap bahwa laporan ini dapat memberikan kontribusi yang bermanfaat bagi ilmu pengetahuan, khususnya dalam bidang yang terkait. Semoga laporan ini dapat memberikan manfaat bagi pembaca serta menjadi referensi yang berguna bagi penelitian selanjutnya

Sungailiat, 16 Juli 2025

Penulis

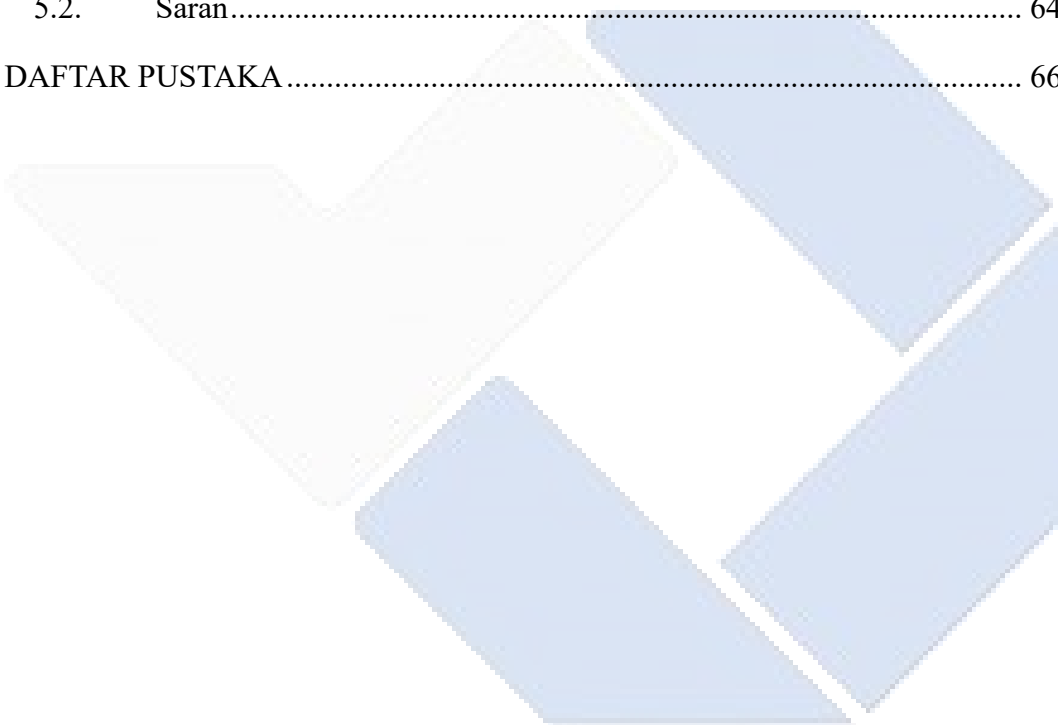
DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN	ii
PERNYATAAN BUKAN PLAGIAT	iii
ABSTRAK	iv
ABSTRACT.....	v
KATA PENGANTAR.....	vi
DAFTAR ISI	viii
DAFTAR TABEL.....	xii
DAFTAR GAMBAR	xiii
DAFTAR LAMPIRAN	xv
BAB I	1
PENDAHULUAN	1
1.1. Latar Belakang Masalah.....	1
1.2. Rumusan Masalah	3
1.3. Hipotesis.....	3
1.4. Tujuan Proyek Akhir	4
BAB II.....	5
DASAR TEORI	5
2.1. Denyut Nadi	5
2.2. Suhu Tubuh	6
2.3. Tekanan Darah.....	7
2.4. Arduino Uno R3 ATmega328P	8
2.5. NodeMCU ESP32	9

2.6.	Pulse Sensor	10
2.7.	Sensor MLX90614	12
2.8.	Sensor MPX5700DP	13
2.9.	Blynk	14
2.10.	Firestore	16
2.11.	Monitoring.....	17
2.12.	<i>Internet of Things</i> (IoT).....	18
BAB III		20
METODE PELAKSANAAN.....		20
3.1.	Jenis dan Desain Penelitian	21
3.2.	Populasi	21
3.3.	Waktu dan Tempat.....	21
3.4.	Variabel Penelitian.....	21
3.5.	Definisi Operasional.....	22
3.6.	Jenis dan Teknik Pengumpulan Data.....	23
3.7.	Perancangan Alat dan Konstruksi	24
3.8.	Pembuatan Alat	24
3.9.	Pengujian Alat	25
3.10.	Pengambilan Data	25
3.11.	Analisis Hasil	26
3.12.	Pembuatan Laporan.....	27
BAB IV		28
PEMBAHASAN		28
4.1.	Deskripsi Alat.....	28
4.1.1.	Sistem Kerja Alat	28

4.1.2.	Rancangan <i>Hardware</i>	30
4.2.	Perancangan dan Pembuatan Konstruksi	31
4.2.1.	Perancangan Konstruksi.....	31
4.2.2.	Pembuatan Konstruksi.....	32
4.3.	Perancangan dan Pembuatan Kontrol Alat Pengukuran.....	33
4.3.1.	Perancangan dan Pembuatan Sistem Catu Daya dan Sistem Kontrol	34
4.3.1.1.	Perancangan dan Pembuatan Sistem Penyimpanan Energi dan Penurun Tegangan	34
4.3.1.2.	Perancangan dan Pembuatan Sistem Kontrol	35
4.3.2.	Perancangan dan Pembuatan Kontrol Alat Pengukur Denyut Nadi...	35
4.3.3.	Perancangan dan Pembuatan Kontrol Alat Pengukur Suhu	36
4.3.4.	Perancangan dan Pembuatan Kontrol Alat Pengukur Tekanan Darah	37
4.3.4.1.	Perancangan dan Pembuatan Sistem Solenoid dan Pompa Pump 37	
4.3.5.	Perancangan dan Pembuatan LCD TFT 3,5 Inchi.....	38
4.4.	Pengujian Kontrol Alat Pengukuran.....	38
4.4.1.	Pengujian Kontrol Alat Pengukur Denyut Nadi.....	38
4.4.2.	Pengujian Kontrol Alat Pengukur Suhu	40
4.4.3.	Pengujian Kontrol Alat Pengukur Tekanan Darah	42
4.5.	Pengujian Alat Pengukur Secara Keseluruhan	44
4.6.	Tampilan dan Pengiriman Data Hasil Pengukuran Secara Keseluruhan 52	
4.6.1.	Tampilan LCD TFT 3,5 Inchi.....	53
4.6.2.	Tampilan Blynk	55
4.6.3.	Tampilan Firebase	56
4.6.4.	Tampilan Website dan WhatsApp	56

4.7.	Analisis Data	57
4.7.1.	Hasil Analisis Pengukuran Denyut Nadi.....	58
4.7.2.	Hasil Analisis Pengukuran Suhu	59
4.7.3.	Hasil Analisis Pengukuran Tekanan Darah	60
BAB V		63
KESIMPULAN DAN SARAN		63
5.1.	Kesimpulan.....	63
5.2.	Saran.....	64
DAFTAR PUSTAKA		66



DAFTAR TABEL

Tabel 2.1 Lokasi Nadi	5
Tabel 2.2 Rentang Denyut Nadi	6
Tabel 2.3 Rentang Suhu	7
Tabel 2.4 Rentang Tekanan Darah	7
Tabel 3.1 Definisi Operasional Variabel	23
Tabel 4.1 Hasil Kalibrasi Pengukuran Denyut Nadi	39
Tabel 4.2 Hasil Kalibrasi Pengukuran Tekanan Darah.....	42
Tabel 4.3 Hasil Pengujian Denyut Nadi	48
Tabel 4.4 Hasil Pengujian Suhu	49
Tabel 4.5 Hasil Pengujian Tekanan Darah	50

DAFTAR GAMBAR

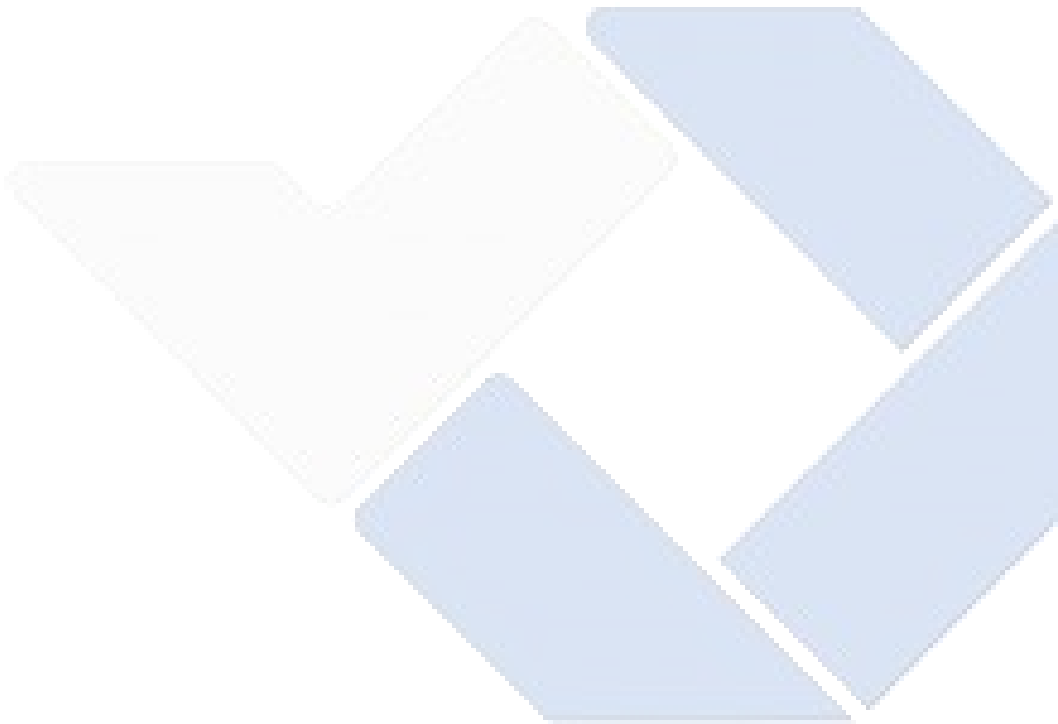
Gambar 2.1 Arduino Uno R3 ATmega328P.....	9
Gambar 2.2 Modul NodeMCU ESP32.....	10
Gambar 2.3 Diagram Block ESP32	10
Gambar 2.4 Tampak Depan Pulse Sensor	11
Gambar 2.5 Tampak Belakang Pulse Sensor	11
Gambar 2.6 Skematik Diagram Modul Pulse Heart Rate Sensor	11
Gambar 2.7 Tipe-tipe MLX90614.....	13
Gambar 2.8 MPX5700DP	14
Gambar 2.9 Pin-Pin MPX5700DP	14
Gambar 2.10 Logo Blynk.....	15
Gambar 2.11 Fitur Firebase.....	16
Gambar 2.12 Logo Firebase	17
Gambar 2.13 Tahapan Monitoring	18
Gambar 2.14 Konsep Internet of Things (IoT)	19
Gambar 3.1 Flowchart Alur Penelitian.....	20
Gambar 4.1 Blok Diagram	28
Gambar 4.2 Wiring Diagram.....	30
Gambar 4.3 Perancangan Konstruksi Tampak Depan.....	32
Gambar 4.4 Perancangan Konstruksi Tampak Samping.....	32
Gambar 4.5 Perancangan Konstruksi Tampak Belakang	32
Gambar 4.6 Konstruksi Tampak Depan	33
Gambar 4.7 Konstruksi Tampak Samping	33
Gambar 4.8 Konstruksi Tampak Belakang	33

Gambar 4.9 Rangkaian Alat Pengukur Denyut Nadi	36
Gambar 4.10 Rangkaian Alat Pengukur Suhu.....	36
Gambar 4.11 Rangkaian Alat Pengukur Tekanan Darah.....	37
Gambar 4.12 Rangkaian Seluruh Sistem	38
Gambar 4.13 Blok Pengujian Alat Pengukur Denyut Nadi	39
Gambar 4.14 Blok Pengujian Alat Pengukur Suhu.....	41
Gambar 4.15 Blok Pengujian Alat Pengukur Tekanan Darah.....	42
Gambar 4.16 Pengujian Alat Secara Keseluruhan	45
Gambar 4.17 Tampilan LCD TFT 3,5 Inchi.....	54
Gambar 4.18 Tampilan Blynk	55
Gambar 4.19 Tampilan Firebase	56
Gambar 4.20 Tampilan Website	57
Gambar 4.21 Tampilan WhatsApp.....	57
Gambar 4.22 Grafik Perbandingan Denyut Nadi.....	58
Gambar 4.23 Grafik Perbandingan Suhu	59
Gambar 4.24 Grafik Perbandingan Tekanan Darah Sistolik.....	61
Gambar 4.25 Grafik Perbandingan Tekanan Darah Diastolik.....	61

DAFTAR LAMPIRAN

Lampiran 1: Daftar Riwayat Hidup

Lampiran 2: Coding



BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Kesehatan manusia sangat bergantung pada fungsi alat vital, meliputi denyut nadi, suhu, dan tekanan darah. Ketiga parameter ini memainkan peran penting dalam menentukan kondisi kesehatan secara keseluruhan. Menurut *National Center for Biotechnology Information* (NCBI), denyut nadi normal pada orang dewasa memiliki rentang antara 60 hingga 100 *Beats Per Minute* (BPM). Denyut nadi yang berdetak lebih dari 100 *Beats Per Minute* (BPM) disebut *takikardia*, sedangkan denyut nadi yang berdetak kurang dari 60 *Beats Per Minute* (BPM) disebut *bradikardia* [1]. Suhu normal tubuh manusia berkisar antara 36,5°C hingga 37,5°C. Suhu tubuh melebihi 37,5°C disebut *hipertermia*, sedangkan suhu tubuh dibawah 35°C disebut *hipotermia* [2]. Klasifikasi tekanan darah terdiri dari tiga kategori, yaitu kurang atau sama dengan 120/80 mmHg disebut tekanan darah normal, angka di atas 120/80 mmHg hingga 139/89 mmHg disebut *prahipertensi*, dan angka di atas 140/90 mmHg disebut *hipertensi* [3].

Monitoring kesehatan membutuhkan peralatan kesehatan yang dapat mengukur denyut nadi, suhu, dan tekanan darah secara bersamaan. Namun, permasalahan yang sering ditemui pada produk kesehatan yang tersedia di pasaran memiliki keterbatasan alat pengukur yang mampu melakukan ketiga pengukuran tersebut dalam satu alat dan satu kali pengukuran. Saat ini, pengukurannya masih menggunakan beberapa alat secara terpisah, seperti tensimeter digital untuk mengukur denyut nadi dan tekanan darah, serta termometer inframerah digital untuk mengukur suhu tubuh. Jurnal “Alat Pengukur Tekanan Darah Disertai Pengukur Suhu Tubuh” (2024) menyebutkan bahwa pengukuran suhu tubuh masih dilakukan secara terpisah dari pengukuran tekanan darah [4]. Selain itu, terdapat alat kesehatan yang tidak dilengkapi dengan fitur penyimpanan data. Jurnal "Sistem Monitoring Saturasi Oksigen dan Denyut Nadi dalam Darah Menggunakan Sensor MAX30100 Via Telegram Berbasis IoT" (2021) serta jurnal "Alat Pengukur Denyut Nadi Dengan Tampilan OLED Berbasis Arduino" (2023)

sama-sama mengusulkan alat pemantauan kesehatan yang lebih efektif [5,6]. Namun, kedua penelitian tersebut tidak dilengkapi dengan fitur penyimpanan data, sehingga pengguna hanya dapat melihat hasil pengukuran secara *real-time*. Ketiadaan akses ke data historis ini menjadi kendala dalam melakukan analisis kesehatan jangka panjang secara lebih mendalam dan berkelanjutan. Hal tersebut menyebabkan hasil pengukuran pada setiap individu tidak tercatat dengan baik, kesulitan dalam melakukan analisis, dan tidak bisa menggunakan data hasil monitoring untuk keperluan lainnya.

Berdasarkan penjelasan di atas, telah dikembangkan peralatan untuk mengukur denyut nadi, suhu, dan tekanan darah yang bersifat *portable* serta berbasis *Internet of Things* (IoT) yang bisa beroperasi secara bersamaan. Alat yang penulis kembangkan ini merupakan pengembangan dari penelitian sebelumnya, sebagaimana dijelaskan oleh Milda Muryati dan Rini Angganita (2018) dalam penelitian mereka yang berjudul “Monitoring Suhu dan Detak Jantung Berbasis Arduino”, yang membahas tentang pengukuran suhu dan detak jantung namun belum menerapkan teknologi *Internet of Things* (IoT) [7]. Penelitian ini bertujuan untuk membuat alat yang dapat melakukan pengukuran dan memonitoring kesehatan individu dari jarak jauh sebagai bentuk pengembangan alat yang dapat mengukur denyut nadi, suhu, dan tekanan darah berbasis *Internet of Things* (IoT). LCD TFT 3,5 inci digunakan untuk memberikan data hasil pengukuran secara *real-time*, server Blynk digunakan untuk monitoring, sementara Firebase berfungsi untuk menyimpan data secara *offline*. Selain itu, mengembangkan sebuah *website* yang terintegrasi dengan Firebase untuk mengambil data hasil pengukuran secara otomatis berdasarkan ID pengukuran dan dikirimkan ke WhatsApp penerima yang dituju. Sistem ini dilengkapi dengan fitur notifikasi di *platform* IoT yang memberikan informasi tentang status hasil pengukuran, apakah berada dalam rentang normal atau tidak. Melalui alat ini, operator dapat memonitoring kesehatan secara *real-time* menggunakan sensor untuk mengumpulkan data fisiologis berbasis *Internet of Things* (IoT).

1.2. Rumusan Masalah

Adapun rumusan masalah yang diangkat pada proyek akhir ini adalah sebagai berikut:

1. Bagaimana merancang dan membuat alat kesehatan yang dapat mengukur denyut nadi?
2. Bagaimana merancang dan membuat alat kesehatan yang dapat mengukur suhu?
3. Bagaimana merancang dan membuat alat kesehatan yang dapat mengukur tekanan darah?
4. Bagaimana cara mengirimkan dan menampilkan data hasil pengukuran secara *real-time* ke LCD TFT 3,5 inci, server Blynk, dan Firebase dalam sistem berbasis *Internet of Things* (IoT), serta mengembangkan *website* terintegrasi dengan Firebase yang dapat mengambil data berdasarkan ID pengukuran dan mengirimkannya ke WhatsApp?

1.3. Hipotesis

Adapun hipotesis dalam penelitian ini dirumuskan sebagai berikut:

1. Hipotesis Nol (H_0)
Alat pengukuran dan sistem monitoring denyut nadi, suhu, dan tekanan darah berbasis *Internet of Things* (IoT) yang dikembangkan tidak mampu memberikan data hasil pengukuran secara *real-time* melalui LCD TFT 3,5 inci, tidak bisa memonitoring melalui server Blynk, data hasil pengukuran tidak tersimpan secara *offline* di Firebase, *website* yang terintegrasi dengan Firebase tidak dapat mengambil data hasil pengukuran secara otomatis melalui ID pengukuran, data hasil pengukuran tidak dapat dikirimkan ke WhatsApp penerima yang dituju, serta memiliki tingkat akurasi tinggi dengan deviasi pengukuran kurang jika dibandingkan dengan alat medis.
2. Hipotesis Alternatif (H_1)
Alat pengukuran dan sistem monitoring denyut nadi, suhu, dan tekanan darah berbasis *Internet of Things* (IoT) yang dikembangkan mampu memberikan data hasil pengukuran secara *real-time* melalui LCD TFT 3,5 inci, bisa

memonitoring melalui server Blynk, data hasil pengukuran tersimpan secara *offline* di Firebase, *website* yang terintegrasi dengan Firebase dapat mengambil data hasil pengukuran secara otomatis melalui ID pengukuran, data hasil pengukuran dapat dikirimkan ke WhatsApp penerima yang dituju, serta memiliki tingkat akurasi tinggi dengan deviasi pengukuran kurang jika dibandingkan dengan alat medis.

1.4. Tujuan Proyek Akhir

Adapun tujuan penulisan dalam penyusunan proyek akhir ini diantaranya adalah:

1. Merancang dan membuat alat kesehatan yang dapat mengukur denyut nadi.
2. Merancang dan membuat alat kesehatan yang dapat mengukur suhu.
3. Merancang dan membuat alat kesehatan yang dapat mengukur tekanan darah.
4. Merancang dan membuat sistem pengiriman serta penampilan data hasil pengukuran secara *real-time* ke LCD TFT 3,5 inci, server Blynk, dan Firebase dalam sistem berbasis *Internet of Things* (IoT), serta mengembangkan *website* terintegrasi dengan Firebase yang dapat mengambil data berdasarkan ID pengukuran dan mengirimkannya ke WhatsApp.

BAB II

DASAR TEORI

2.1. Denyut Nadi

Denyut nadi adalah suatu gelombang yang dapat dideteksi pada arteri apabila darah di pompa keluar jantung. Denyut nadi akan mudah dideteksi di suatu tempat yang dimana tempat tersebut terdapat arteri yang melintas [8]. Denyut nadi dapat diukur di berbagai titik tubuh, terutama pada area yang dekat dengan permukaan kulit, yaitu radialis dan karotis [9]. Radialis adalah tempat yang paling sering digunakan untuk mengukur denyut nadi, dimana denyut ini berada pada lengan bawah, tepat di sebelah proksimal sendi pergelangan tangan, sedangkan karotis sering dilakukan oleh dokter, yang dimana lokasinya berada pada leher [10].

Tabel 2.1 Lokasi Nadi [10]

Nadi	Lokasi
Radialis	Menjalar sepanjang tulang radial, sejajar ibu jari dibagian dalam pergelangan tangan
Karotis	Pada sisi leher di antara trakea dan otot sternokleidomastoideus

Denyut nadi normal orang dewasa berkisar diantara 60 hingga 100 *Beats Per Minute* (BPM), peningkatan dan penurunan denyut nadi pasien menandakan orang tersebut memiliki gangguan yang berhubungan dengan nadi, yang disebut *takikardia* dan *bradikardia*. *Takikardia* adalah kondisi dimana denyut nadi pasien mencapai lebih dari 100 *Beats Per Minute* (BPM). Kelebihan denyut nadi normal dapat menyebabkan jantung berdebar, sulit bernapas, dan berkeringat. Sedangkan *bradikardia* adalah kondisi denyut nadi pasien dibawah 60 *Beats Per Minute* (BPM). Kekurangan denyut nadi normal dapat menyebabkan berbagai masalah kesehatan seperti pusing, lelah, dan jantung berdebar [1]. Beberapa faktor yang mempengaruhi frekuensi denyut nadi, yaitu jenis kelamin, umur, posisi tubuh, obat-obatan, suhu, emosi, dan aktivitas fisik. Peningkatan aktivitas fisik pasien akan

meningkatkan frekuensi denyut nadi, begitu juga sebaliknya akan terjadi penurunan apabila aktivitas fisik diturunkan [10].

Tabel 2.2 Rentang Denyut Nadi [1]

Usia	Rentang Denyut Nadi Normal <i>Beats Per Minute</i> (BPM)
Bayi	100-180
Balita	98-140
Prasekolah	80-120
Usia sekolah	75-118
Dewasa	60-100

2.2. Suhu Tubuh

Suhu tubuh merupakan keseimbangan antara jumlah panas yang dihasilkan oleh tubuh dan jumlah panas yang hilang dari tubuh [11]. Suhu tubuh pada manusia dapat dengan mudah berubah sesuai dengan beberapa faktor, baik dari faktor eksternal maupun faktor internal [12]. Salah satu faktor yang mempengaruhi suhu tubuh adalah hormon, pada wanita suhu tubuh biasanya bergantung pada saat dalam daur haid, kemudian jika pada anak-anak suhu tubuhnya lebih tinggi dari orang dewasa, sedangkan pada usia lanjut ataupun bayi yang baru lahir suhunya lebih rendah [13]. Untuk mengetahui berapa suhu tubuh digunakan alat termometer. Alat pengukur suhu tubuh ini banyak jenisnya, yaitu termometer air raksa, termometer digital, dan termometer berbentuk strip [14].

Suhu normal pada manusia dibagi menjadi tiga, di antaranya suhu normal pada bayi (36,3–37,7 °C), suhu normal pada anak (36,1–37,7 °C), dan suhu normal pada orang dewasa (36,5–37,5 °C) [17]. Menurut WHO, normalnya tubuh manusia memiliki suhu sebesar 36,5°C hingga 37,5°C. Ketika suhu tubuh sudah melebihi dari angka normal, maka hal tersebut menunjukkan adanya infeksi, peradangan, atau lainnya. *Hipotermia* merupakan kondisi suhu tubuh turun dibawah 36,5°C. *Hipertermia* merupakan kondisi suhu tubuh naik diatas 37,5°C [15].

Tabel 2.3 Rentang Suhu [10]

Usia	Daerah Tubuh	Suhu Normal (Derajat Celsius)
0 – 2 bulan	Rectal (Dubur)	36.5 – 38.0
2 – 5 tahun	Rectal (Dubur)	36.5 – 37.8
6 – 12 tahun	Oral (Mulut)	35.9 – 37.6
Dewasa	Oral (Mulut)	35.8 – 37.5
Dewasa	Axillary (Ketiak)	34.7 – 37.3
Dewasa	Rectal (Dubur)	36.8 – 38.0
Lansia	Oral (Mulut)	35.8 – 36.8

2.3. Tekanan Darah

Tekanan darah adalah kemampuan untuk dapat meregangkan pada setiap dinding pembuluh terhadap volume darah. Ada beberapa faktor yang mempengaruhi tekanan darah, yaitu pola hidup, genetik, usia, dan jenis kelamin [16]. Secara umum, klasifikasi tekanan darah menurut *World Health Organization* (WHO) ada tiga, yaitu kurang atau sama dengan 120/80 mmHg disebut tekanan darah normal, tekanan yang di bawah 90/60 mmHg disebut *hipotensi*, angka diatas 120/80 mmHg hingga 139/89 mmHg disebut *prahipertensi*, dan angka diatas 140/90 mmHg disebut *hipertensi*.

Tabel 2.4 Rentang Tekanan Darah [10]

Kategori	Tekanan Darah Sistolik	Tekanan Darah Diastolik
	(mmHg)	(mmHg)
Tekanan Darah Normal	<120	<80
Tekanan Darah Tinggi	120 – 139	80 – 89
Hipertensi Tahap 1	140 – 159	90 – 99
Hipertensi Tahap 2	>160	>100

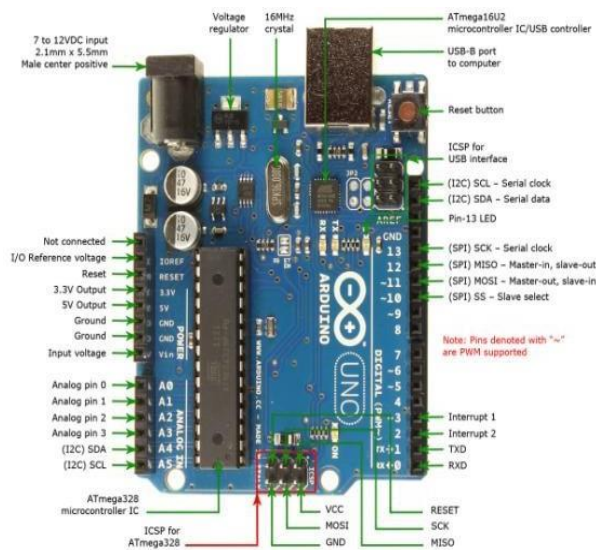
Pada umumnya penyakit yang berkaitan dengan tekanan darah adalah *hipertensi* dan *hipotensi*. *Hipertensi* dapat menyebabkan penyakit kardiovaskular, seperti serangan jantung dan stroke, serta meningkatkan risiko gagal ginjal dan

demensia akibat kerusakan pembuluh darah. Diperkirakan sekitar 80% kenaikan kasus *hipertensi* terutama terjadi di negara berkembang pada tahun 2025, dari jumlah total 639 juta kasus di tahun 2000. Jumlah ini diperkirakan meningkat menjadi 1,15 miliar kasus di tahun 2025. Di sisi lain, *hipotensi*, atau tekanan darah rendah, dapat menyebabkan pusing, lemas, dan pingsan, terutama saat berdiri tiba-tiba. Kondisi ini juga dapat mengakibatkan kelelahan, nyeri dada, dan dalam kasus parah, dapat mengakibatkan syok. Prevalensi kejadian *hipotensi* diperkirakan berkisar antara 5% hingga 34%, dengan kecenderungan meningkat pada usia remaja dan lanjut usia [17].

Terdapat 2 metode untuk melakukan pengukuran tekanan darah yaitu, metode langsung (*invasive*) dan metode tidak langsung (*non-invasive*), metode langsung (*invasive*) adalah metode yang memasang kateter ke dalam pembuluh darah untuk melakukan pengukuran tekanan darah, sedangkan metode tidak langsung (*non-invasive*) adalah metode yang menggunakan alat tensimeter (sphygmomanometer) dan stetoskop [18].

2.4. Arduino Uno R3 ATmega328P

Arduino adalah otak dari sebuah sistem untuk melakukan suatu pekerjaan. Setiap sistem operasi yang dikontrol Arduino akan menerima input, kemudian diproses, dan menghasilkan perintah output untuk melakukan pekerjaan tersebut. Pada Arduino Uno R3 ATmega328P memiliki pin sebanyak 28 pin, yang terdiri dari 14 pin digital *input/output* (dengan 6 pin *output* untuk PWM), 6 pin *input* analog, 2 pin resonator kristal keramik 16 MHz, 1 pin analog referensi, 1 pin reset, serta masing-masing 2 pin GND dan VCC [19].



Gambar 2.1 Arduino Uno R3 ATmega328P

Input analog pada Arduino Uno R3 ATmega328P ditandai dengan pin dari A0 hingga A5. Dari 6 pin analog digunakan untuk melakukan proses dengan *input* analog dan dapat digunakan dengan rentang 0 sampai 5V. Sementara itu, *input/output* digital pada Arduino Uno R3 ATmega328P terdiri dari D0 hingga D13. *Input* digital dapat diartikan sebagai sinyal waktu yang tidak kontinyu dengan pulsa *input* diskrit yang direpresentasikan sebagai 0 dan 1. Mikrokontroler Arduino Uno R3 ATmega328P ini dapat dihubungkan ke perangkat elektronik seperti komputer atau port USB. Kita juga dapat meng-*upload* program Arduino ke kabel *power jack* [20]. Pemrograman Arduino Uno R3 ATmega328P dapat dilakukan dengan Arduino *Integrated Development Environment* (IDE), pada *software* Arduino IDE terdapat beberapa *sketch* yang memiliki fungsi memberikan perintah kepada mikrokontroler atau Arduino Uno R3 ATmega328P [19].

2.5. NodeMCU ESP32

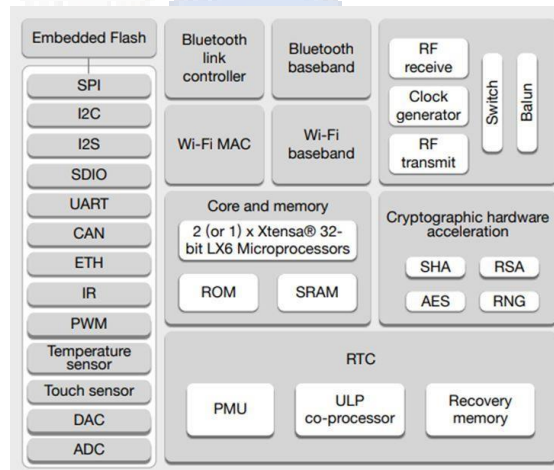
NodeMCU ESP32 adalah sebuah mikrokontroler yang memiliki Wi-Fi yang terintegrasi dan *Bluetooth Low Energy* (BLE) yang merupakan penerus dari ESP8266 yang hanya mendukung Wi-Fi untuk komunikasi [21]. Mikrokontroler ini memiliki arsitektur *dual-core* 32-bit dengan prosesor Tensilica Xtensa LX6, yang

memungkinkan pemrosesan data yang lebih cepat dan efisien yang memudahkan pengembangan aplikasi yang menggunakan *Internet of Things* (IoT).



Gambar 2.2 Modul NodeMCU ESP32

Memori ESP32 terdiri dari 448 kB ROM, 520 kB SRAM, dua memori RTC 8 kB, dan 4 MB memori flash. Chip ini memiliki empat unit SPI, dua unit I2C, dan 18 pin ADC 12-bit. Keunggulan utama mikrokontroler ini adalah harganya yang terjangkau dan kemampuan untuk digunakan dalam pemrograman dengan ESP32 [22]. Meskipun NodeMCU ESP32 menawarkan banyak keunggulan, tantangan dalam penggunaannya tetap ada, seperti keamanan data dalam komunikasi IoT dan keterbatasan jangkauan sinyal yang dapat dipengaruhi oleh faktor lingkungan [23].



Gambar 2.3 Diagram Block ESP32 [21]

2.6. Pulse Sensor

Pulse sensor adalah perangkat elektronik yang secara *non-invasive* mengukur detak jantung dengan mendeteksi perubahan volume darah dalam

pembuluh darah kapiler. Metodenya mirip dengan *photoplethysmography* (PPG), di mana cahaya inframerah atau LED merah digunakan untuk mengukur perubahan aliran darah seiring dengan detak jantung [24].

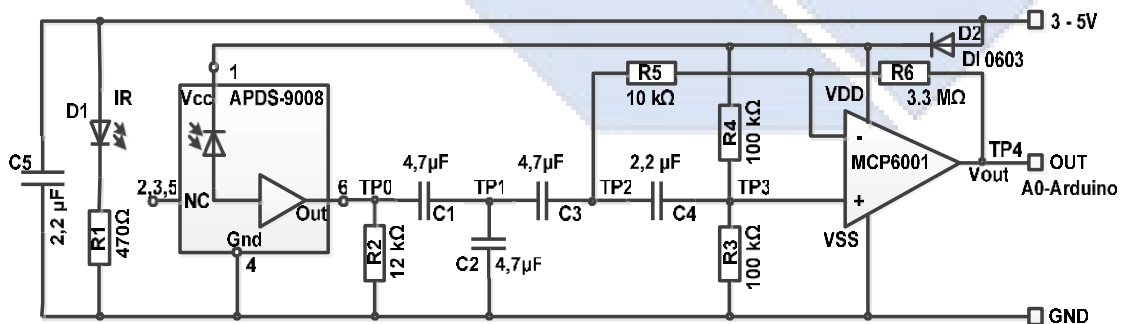


Gambar 2.4 Tampak Depan Pulse Sensor



Gambar 2.5 Tampak Belakang Pulse Sensor

Pulse sensor bekerja dengan mendeteksi perubahan intensitas cahaya yang ditransmisikan atau dipantulkan oleh darah dalam jaringan kapiler. Ada dua metode yang digunakan dalam sistem kerja dari pulse sensor, yaitu metode transmisi dan metode refleksi. Dalam metode transmisi, cahaya dari LED melewati jaringan tubuh dan diterima oleh photodiode di sisi lain. Dalam metode refleksi, cahaya dari LED dipantulkan oleh jaringan tubuh dan photodiode mengukur intensitas pantulannya. Selanjutnya, sensor mengubah intensitas cahaya menjadi sinyal listrik. Mikrokontroler, seperti Arduino atau Raspberry Pi, dapat memproses sinyal ini untuk menghitung jumlah detak jantung per menit *Beats Per Minute* (BPM).



Gambar 2.6 Skematik Diagram Modul Pulse Heart Rate Sensor

Pulse sensor banyak digunakan dalam berbagai aplikasi, terutama dalam bidang kesehatan dan kebugaran, karena dengan tingkat akurasi pulse sensor yang memiliki kesalahan (*error*) berkisar antara 0,69% hingga 1,84%, maka dari itu pulse

sensor mampu memberikan hasil yang baik dan dapat diandalkan untuk. Oleh karena itu, pulse sensor sering digunakan dalam perangkat medis seperti monitor detak jantung, alat pemantau kebugaran, serta aplikasi IoT yang berfokus pada kesehatan [26].

2.7. Sensor MLX90614

Sensor MLX90614 merupakan salah satu dari sensor yang dapat mengukur suhu ruangan dan objek tanpa memerlukan kontak langsung dengan objek yang diukur. Sensor ini bekerja dengan mendeteksi radiasi inframerah yang dipancarkan oleh objek dan mengubahnya menjadi nilai suhu [27]. Radiasi ini kemudian diubah menjadi sinyal listrik yang diproses oleh ASIC (*Application-Specific Integrated Circuit*) yang terintegrasi dalam sensor. ASIC ini dilengkapi dengan amplifier rendah noise, ADC 17-bit, dan unit DSP (*Digital Signal Processing*) yang memungkinkan sensor mencapai akurasi dan resolusi yang tinggi [28]. Sensor ini dikalibrasi secara digital dengan *output* SMBus, yang memungkinkan pengukuran suhu dengan resolusi $0,02^{\circ}\text{C}$ [27]. Sensor ini memiliki beberapa kelebihan dibandingkan dengan sensor suhu lainnya, diantaranya:

- Akurasi tinggi: sensor MLX90614 memiliki akurasi yang sangat baik dan kesalahan (*error*) yang cenderung lebih rendah. Beberapa penelitian telah dilakukan dengan menggunakan sensor ini dan menunjukkan bahwa *error* yang dihasilkan berkisar antara 0.02°C hingga 2.8% [27].
- Non kontak: sensor MLX90614 dapat melakukan pengukuran pada objek tanpa perlu bersentuhan/kontak langsung, sehingga sangat cocok digunakan untuk melakukan pengukuran cepat dan higienis, seperti pada masa pandemi covid-19 [29,30].
- Respon cepat: selain memiliki akurasi yang tinggi sensor MLX90614 ini memiliki respon yang cepat sampai dalam hitungan milidetik, sehingga sangat efektif untuk melakukan pengukuran secara *real-time* [27].

Walaupun memiliki beberapa kelebihan tidak menutup kemungkinan bahwa sensor MLX90614 memiliki keterbatasan dalam melakukan pengukuran, diantaranya:

- Rentang pengukuran: sensor MLX90614 memiliki keterbatasan dalam melakukan pengukuran. Jika jarak antara sensor dan objek yang diukur, maka akurasi pengukurannya dapat menurun [29,30]. Namun, ada beberapa tipe pada sensor MLX90614, yaitu type GY-906-BAA dengan rentang pengukuran maksimal 2 cm, type GY-906-BCC dengan rentang pengukuran maksimal 10 cm, dan type GY-906-DCI dengan rentang pengukuran maksimal 50 cm.
- Pengaruh lingkungan: sensor ini juga dapat dipengaruhi dengan lingkungan, terutama jika digunakan diluar lingkungan atau dalam kondisi lingkungan yang ekstrem [27].



Gambar 2.7 Tipe-tipe MLX90614

Dalam pengimplementasian dari sensor MLX90614 dapat digunakan dalam berbagai aplikasi, terutama yang membutuhkan pengukuran suhu non-kontak. Dalam penelitian yang dilakukan Yanuar dan Agoes (2020), sensor MLX90614 digunakan untuk mengukur suhu tubuh non-kontak dengan persentase *error* yang kecil, yaitu 0-2% tergantung pada jarak pengukuran, dengan batas optimal pada rentang 3-5 cm dan tingkat *error* tertinggi dengan persentase 2.8% [27].

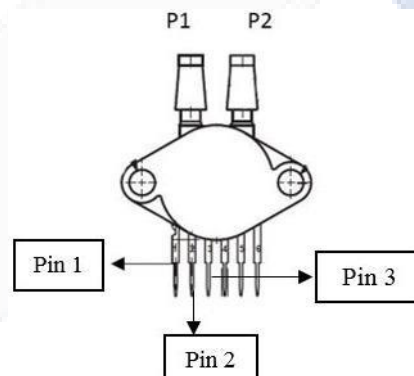
2.8. Sensor MPX5700DP

MPX5700DP adalah sebuah sensor tekanan yang menggunakan teknologi piezoresistif untuk mengukur tekanan udara atau gas dengan rentang berkisar 0 hingga 500 kPa. MPX5700DP mampu menghasilkan *output* analog yang proporsional terhadap tekanan yang diterima, dengan rentang tegangan berkisar 0,2 V hingga 4,7 V [31].



Gambar 2.8 MPX5700DP

Prinsip kerja dari MPX5700DP adalah dengan mendeteksi perbedaan tekanan antara dua port (P1 dan P2). Perbedaan tekanan tersebut akan dikonversi menjadi sinyal listrik dengan menggunakan sistem piezoresistif, kemudian *output* dari MPX5700DP akan masuk ke dalam mikrokontroler seperti Arduino, NodeMCU, dan Raspberry Pi yang nantinya akan digunakan dalam berbagai aplikasi [32].



Gambar 2.9 Pin-Pin MPX5700DP

MPX5700DP sudah banyak digunakan dalam dunia kesehatan, yang biasanya digunakan dalam alat tensimeter, sesuai dengan penelitian Arief Ramdhani (2019) yang menggunakan MPX5700DP sebagai sensor untuk mengukur tekanan darah pasien dengan tingkat akurasi sistolik 98,24% dan diastolik 96,4%, kemudian data hasil pengukuran tersebut dikirimkan melalui Wi-Fi menggunakan ESP8266 ke server, sehingga dapat memantau pasien dari jarak jauh [33].

2.9. Blynk

Blynk adalah sebuah *platform Internet of Things* (IoT) yang mendukung pengguna untuk dapat melakukan kontrol dan memantau perangkat IoT dari jarak

jauh melalui aplikasi *mobile* atau *website* yang mendukung berbagai jenis perangkat mikrokontroler, seperti NodeMCU ESP8266, ESP32, Arduino, dan Raspberry Pi [34,35]. Pada Blynk ini memiliki tiga komponen utama yang saling mendukung untuk memfasilitasi komunikasi dan kontrol dalam sistem IoT. Pertama, aplikasi *mobile* Blynk yang digunakan pengguna untuk mengontrol dan memantau perangkat IoT dari mana dan kapan saja melalui *platform* IOS dan android. Kedua, server Blynk yang berguna sebagai penghubung antara aplikasi/*web* dan perangkat IoT, yang memungkinkan untuk mengirim dan menerima data antara kedua pihak agar memastikan komunikasi dapat dilakukan secara *real-time*. Ketiga, *library* blynk yang digunakan untuk memprogram perangkat mikrokontroler yang memiliki fitur IoT, seperti NodeMCU, ESP8266, atau ESP32 agar dapat terhubung dengan server Blynk [36].



Gambar 2.10 Logo Blynk

Blynk memiliki banyak keunggulan yang membuatnya menjadi *platform* IoT yang disukai oleh pengembang dan pengguna. Pertama, antarmuka *drag-and-drop* yang mudah digunakan memungkinkan pengguna merancang dan mengelola proyek IoT tanpa pengetahuan pemrograman yang mendalam. Bahkan pemula dapat menggunakan hal ini untuk membuat aplikasi *Internet of Things* (IoT) yang berhasil dalam waktu singkat. Selanjutnya, kemampuan Blynk untuk berkoneksi dengan berbagai jenis jaringan, seperti Wi-Fi, Bluetooth, dan jaringan seluler, memungkinkannya digunakan dengan berbagai perangkat dan lingkungan, baik di rumah, kantor, maupun tempat kerja. Ketiga, keamanan adalah hal terpenting bagi Blynk. Untuk setiap proyek, *platform* ini menggunakan token akses unik sebagai alat otentikasi, yang memastikan bahwa hanya perangkat yang terdaftar yang dapat berinteraksi dengan server. [36,37]

Dalam implementasi Blynk sudah banyak digunakan dalam dunia kesehatan, salah satunya dalam penelitian (Pratama, Sulistiyanto, & Setyobudi,

2023) memanfaatkan Blynk untuk menampilkan data dari berbagai sensor, seperti suhu ruangan, kelembapan, kadar CO, CO₂, NH₃, dan NO₂, serta suhu tubuh pasien. Dengan fitur Blynk, data dari beberapa sensor tersebut dapat dikirim dan ditampilkan pada aplikasi Blynk, sehingga mempermudah akses informasi kesehatan pasien [38].

2.10. Firebase

Firebase adalah sebuah *platform* layanan berbasis cloud dari Google yang mendukung pengembangan dari aplikasi *web* dan *mobile*. Ada beberapa fitur yang dapat digunakan dalam Firebase, diantaranya *autentifikasi*, *develop* (penyimpanan data secara *real-time*, *stroge*, dan *cloud messaging*), dan *grow* dengan beberapa fitur yang disediakan membuat pengguna dapat membuat aplikasi *web* dan *mobile* dengan mudah [39,40].



Gambar 2.11 Fitur Firebase

Berbagai fitur *develop* terdapat layanan utama yang dapat digunakan untuk melakukan pengembangan aplikasi diantaranya [40].

- *Firebase Realtime Database*

Realtime database ini digunakan untuk menyimpan data dalam format JSON dan dapat memperbarui informasi secara otomatis ke semua pengguna yang terhubung

- *Firebase Authentication*

Authentication ini dapat digunakan untuk meningkatkan keamanan dan akses pengguna, dengan cara login melalui Email, Google, Facebook, dan metode lainnya.

- **Firestore**

Layanan ini memungkinkan untuk dapat melakukan penyimpanan berbasis cloud dengan berbagai jenis file seperti gambar, video, dan dokumen.

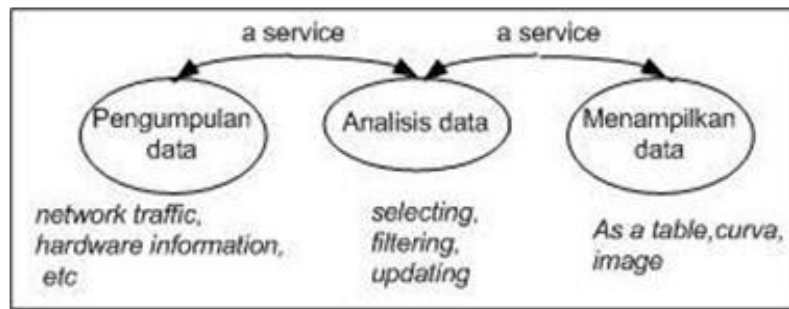


Gambar 2.12 Logo Firebase

Firestore telah banyak digunakan dalam berbagai bidang yang berguna untuk meningkatkan efisiensi dan kinerja aplikasi, salah satunya dalam aplikasi *mobile*, dalam penelitian yang dilakukan Setyawan (2024) dengan menggunakan Firestore sebagai aplikasi catatan harian bagi pasien yang menderita penyakit diabetes melitus yang memungkinkan melakukan penyimpanan dan pengolahan data pasien agar lebih efisien [41].

2.11. Monitoring

Monitoring adalah suatu proses pengumpulan, analisis, dan interpretasi data yang sistematis dan berkelanjutan. Tujuan dari monitoring adalah untuk memantau suatu kinerja sistem atau program dan sebagai mekanisme deteksi dini terhadap suatu masalah yang berpotensi melakukan penyimpangan yang dapat mempengaruhi efektivitas dari suatu sistem [42]. Secara garis besar memiliki beberapa tahapan utama, yaitu pengumpulan data, analisis data, pelaporan dan evaluasi, dan tindakan korektif [43].



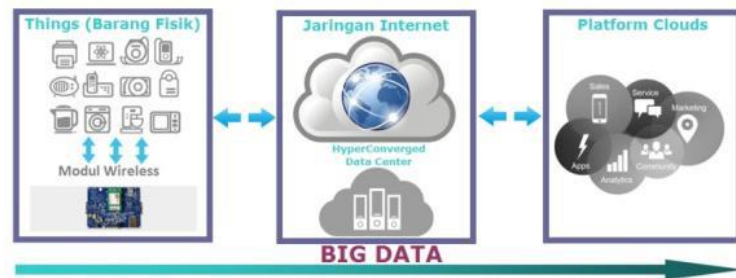
Gambar 2.13 Tahapan Monitoring

Dalam dunia kesehatan, sistem monitoring adalah suatu metode untuk mengamati, mencatat, dan menganalisis kesehatan pasien untuk memastikan kondisi kesehatan yang baik dan optimal. Monitoring dapat dilakukan secara manual oleh tenaga medis atau secara otomatis menggunakan teknologi seperti IoT, memonitoring secara otomatis menggunakan IoT dapat meningkatkan efisiensi dan akurasi dalam pemantauan pasien [44].

Dalam pengimplementasian sistem monitoring dalam dunia kesehatan, ada beberapa contoh aplikasi yang telah dikembangkan, yaitu diantaranya monitoring pasien rawat inap berbasis IoT, monitoring ini dapat memantau suhu tubuh, detak jantung, dan jumlah cairan infus pasien secara otomatis, yang dapat diakses oleh tenaga medis melalui aplikasi *mobile*, selain itu pengaplikasian sistem monitoring ini dapat melakukan pemantauan pasien dengan penyakit kronis, seperti diabetes dan hipertensi yang memudahkan tenaga medis untuk memberikan tindakan yang lebih cepat [44, 45].

2.12. *Internet of Things (IoT)*

Internet of Things (IoT) adalah sebuah konsep yang memiliki beberapa perangkat fisik yang berbeda satu sama lain yang dapat terhubung dan bertukar data melalui internet tanpa perlu campur tangan manusia [46]. IoT bertujuan untuk meningkatkan penggunaan konektivitas internet yang saling terhubung satu sama lain, termasuk memungkinkan kontrol jarak jauh terhadap alat-alat elektronik seperti lampu ruangan, sistem keamanan, dan infrastruktur industri [47].



Gambar 2.14 Konsep *Internet of Things* (IoT)

Konsep utama dalam *Internet of Things* (IoT) terdiri dari tiga bagian yang bekerja secara bersamaan dan mengendalikan perangkat dari jarak jauh [46].

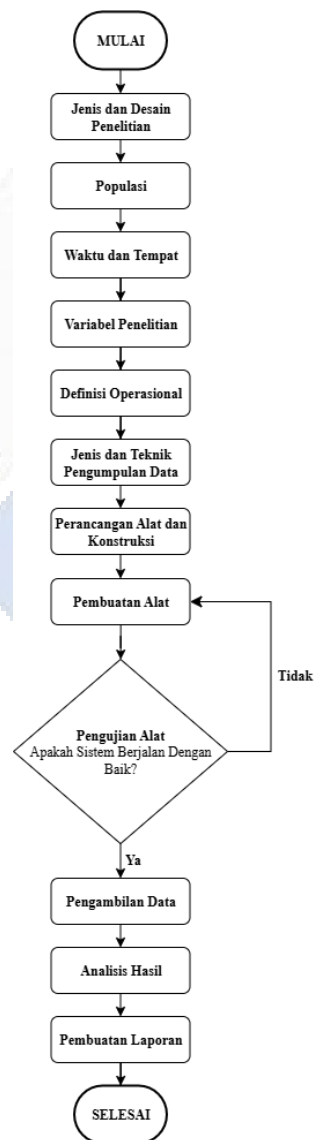
- *Things* (Barang Fisik)
Mencakup beberapa perangkat sensor, aktuator, dan perangkat lainnya yang dilengkapi dengan modul IoT yang digunakan untuk mengumpulkan dan mengeksekusi perintah.
- Konektivitas
Memungkinkan perangkat dapat terhubung dengan jaringan internet melalui Bluetooth, Wi-Fi, dan jaringan seluler yang digunakan untuk mentransmisikan data secara real time.
- *Platform Clouds*
Platform cloud berperan dalam menyimpan, menganalisis, dan mengolah data, sehingga pengguna dapat dengan mudah mengakses informasi secara relevan dan mengontrol perangkat dari mana dan kapan saja.

Implementasi IoT telah banyak diterapkan di berbagai bidang, salah satunya dalam smart home, sesuai dengan penelitian yang dilakukan oleh Efendi (2018), di mana IoT digunakan untuk mengontrol beberapa perangkat elektronik di rumah, seperti lampu dan kamera keamanan yang dikontrol melalui smartphone, sehingga pengguna dapat menghidupkan dan mematikan lampu dan memantau kondisi rumah dari jarak jauh [46]. Selain dalam smart home, IoT juga dapat diterapkan dalam bidang pertanian yang digunakan untuk memantau kelembapan, lingkungan dari jarak jauh secara otomatis. Dengan menggunakan bantuan dari beberapa sensor yang nantinya data akan dikirimkan secara *real-time* ke server, sehingga dapat membantu para petani untuk mendapatkan hasil panen yang efektif dan efisien [47].

BAB III

METODE PELAKSANAAN

Pelaksanaan proyek akhir yang berjudul "Rancang Bangun Alat Kesehatan Pengukur dan Sistem Monitoring Denyut Nadi, Suhu, dan Tekanan Darah Berbasis *Internet of Things* (IoT)", penulis memiliki beberapa tahapan penelitian untuk mempermudah pembuatannya. Berikut adalah tahapan penelitian dalam bentuk *flowchart* pada Gambar 3.1.



Gambar 3.1 *Flowchart* Alur Penelitian

3.1. Jenis dan Desain Penelitian

Jenis penelitian yang digunakan adalah penelitian analitik, yaitu metode penelitian yang bertujuan untuk menguji hubungan sebab-akibat antara dua variabel. Dalam bidang kesehatan, penelitian analitik memiliki peranan penting untuk memahami faktor-faktor yang mempengaruhi kondisi kesehatan serta menilai efektivitas dari intervensi kesehatan.

3.2. Populasi

Populasi dalam penelitian ini terdiri dari mahasiswa Politeknik Manufaktur Negeri Bangka Belitung.

a. Kriteria inklusi:

1. Mahasiswa Politeknik Manufaktur Negeri Bangka Belitung.
2. Mahasiswa yang bersedia menjadi partisipan dalam penelitian ini.

b. Kriteria eksklusi:

1. Selain mahasiswa Politeknik Manufaktur Negeri Bangka Belitung.
2. Mahasiswa yang tidak bersedia menjadi partisipan dalam penelitian ini.

3.3. Waktu dan Tempat

Pengambilan data dilakukan pada akhir Juni dan awal Juli 2025 di lokasi yang berbeda-beda.

3.4. Variabel Penelitian

Variabel dalam penelitian ini adalah gambaran mengenai pengumpulan data untuk hasil dari Tugas Akhir yang dibuat dengan judul "Rancang Bangun Alat Kesehatan Pengukur dan Sistem Monitoring Denyut Nadi, Suhu, dan Tekanan Darah Berbasis *Internet of Things* (IoT)".

1. Variabel Independen

- Denyut Nadi

Denyut nadi merupakan jumlah denyutan jantung per menit dengan memiliki berbagai faktor yang dapat mempengaruhi atau mengubah kecepatan dan kekuatannya.

- Suhu

Suhu merupakan indikator untuk menilai kondisi kesehatan saat terjadinya perubahan suhu.

- Tekanan Darah

Tekanan darah merupakan pengukuran yang menghasilkan nilai sistolik dan diastolik dalam sistem peredaran darah.

2. Variabel Terikat

- Hasil Pengukuran Denyut Nadi

Hasil pengukuran denyut nadi merupakan data output yang diperoleh dari pulse sensor berbasis *Internet of Things* (IoT). Data yang dihasilkan menunjukkan nilai jumlah denyutan jantung per menit secara akurat dan *real-time* untuk memantau kondisi kesehatan pengguna.

- Hasil Pengukuran Suhu

Hasil pengukuran suhu merupakan data output yang diperoleh dari sensor MLX90614 tipe GY-906-BCC berbasis *Internet of Things* (IoT). Data yang dihasilkan berupa informasi mengenai kondisi suhu tubuh pengguna dikarenakan perubahan suhu secara akurat dan *real-time* untuk memantau kondisi kesehatan pengguna.

- Hasil Pengukuran Tekanan Darah

Hasil pengukuran tekanan darah merupakan data output yang diperoleh dari sensor MPX5700DP dengan tipe GY-906-BCC berbasis *Internet of Things* (IoT). Data yang dihasilkan mencakup nilai tekanan darah sistolik dan diastolik yang menilai sistem peredaran darah secara akurat dan *real-time* untuk memantau kondisi kesehatan pengguna.

3.5. Definisi Operasional

Definisi operasional merupakan penjelasan secara operasional mengenai makna dan pelaksanaan variabel dalam penelitian yang akan dilaksanakan. Berikut disajikan tabel definisi operasional yang memuat variabel, definisi operasional, alat ukur, skala ukur, dan hasil ukur yang digunakan dalam penelitian ini.

Tabel 3.1 Definisi Operasional Variabel

Variabel	Definisi Operasional	Alat Ukur	Skala Ukur	Hasil Ukur
Denyut Nadi	Denyut nadi adalah suatu gelombang yang dapat dideteksi pada arteri apabila darah di pompa keluar jantung.	Tensimeter Digital	Ordinal	Normal: 60-100 BPM Takikardia: >100 BPM Bradikardia: <100 BPM
Suhu	Suhu tubuh merupakan keseimbangan antara jumlah panas yang dihasilkan oleh tubuh dan jumlah panas yang hilang dari tubuh.	Termometer Inframerah Digital	Ordinal	Normal: 36,5-37,5°C Hipertermia: >37,5°C Hipotermia: <35°C
Tekanan Darah	Tekanan darah adalah kemampuan untuk dapat meregangkan pada setiap dinding pembuluh terhadap volume darah.	Tensimeter Digital	Ordinal	Normal: ≤120/80 mmHg Prahipertensi: >120/80-139/89 mmHg Hipertensi: >140/90 mmHg

3.6. Jenis dan Teknik Pengumpulan Data

Pengumpulan data menjadi tahap paling krusial dalam memperoleh informasi yang akurat. Oleh karena itu, dijelaskanlah jenis data yang digunakan dan teknik pengumpulan data agar penelitian berjalan sesuai dengan yang diharapkan.

1. Jenis Data

Data yang dikumpulkan dalam penelitian ini merupakan data primer yang diperoleh langsung dari responden.

2. Teknik Pengumpulan Data

Pengumpulan data dilakukan melalui pengukuran langsung dari *prototype* yang dirancang langsung untuk mengukur denyut nadi, suhu, dan tekanan darah kepada mahasiswa Politeknik Manufaktur Negeri Bangka Belitung.

3.7. Perancangan Alat dan Konstruksi

Tahapan perancangan ini merupakan langkah awal untuk menetapkan gambaran alat fisik yang akan dibuat, meliputi pemilihan komponen, pemilihan bahan konstruksi, rancangan konstruksi, rancangan keseluruhan alat, serta penentuan posisi komponen yang digunakan. Pembuatan rancangan alat menggunakan aplikasi Fritzing dan rancangan konstruksi menggunakan aplikasi Autodesk. Proses ini bertujuan agar alat yang dirancang sesuai dengan kebutuhan, berfungsi secara optimal, dan mendukung keberhasilan pembuatan alat.

3.8. Pembuatan Alat

Tahap awal dalam pembuatan alat pengukur denyut nadi, suhu, dan tekanan darah berbasis IoT dimulai dengan perancangan alat dan desain fisik alat tersebut. Pertama, pembuatan box atau *casing* sebagai tempat peletakan komponen yang diperlukan dalam pembuatan alat, kotak ini dirancang sedemikian rupa agar lebih ekonomis dan mudah digunakan oleh operator. Ukuran dari kotak tersebut adalah 19 cm × 19 cm untuk panjang dan lebar, dengan tinggi 178 mm. Setelah kotak selesai dibuat, tahapan yang kedua adalah merakit komponen-komponen yang diperlukan, seperti pulse sensor, sensor MLX90614, sensor MPX5700DP, Arduino Uno R3 ATmega328P, NodeMCU ESP32, LCD TFT 3,5 inci, dan beberapa komponen penting lainnya. Selanjutnya, pada tahap ketiga dilakukan pemrograman mikrokontroler menggunakan *software* Arduino IDE, yang berfungsi untuk membaca data dari masing-masing sensor, mengolah data, dan mengirimkannya secara *real-time* ke LCD TFT 3,5 inci, *platform* IoT Blynk, Firebase, dan WhatsApp penerima yang dituju.

3.9. Pengujian Alat

Pada tahapan ini pengujian alat dilakukan untuk memastikan bahwa semua sensor berfungsi dengan baik dan data dapat dikirimkan secara *real-time* ke LCD TFT 3,5 inci, *platform* IoT, dan WhatsApp penerima.

1. Menguji pulse sensor untuk memastikan apakah sensor dapat membaca denyut nadi pengguna secara akurat ketika jari telunjuk diletakkan pada sensor. Pengujian dilakukan dengan membandingkan hasil dengan alat tensimeter digital.
2. Menguji sensor MLX90614 untuk memverifikasi apakah sensor mampu membaca suhu tubuh dengan stabil dan tepat ketika sensor mendeteksi pada bagian dahi pengguna. Pengujian dilakukan dengan membandingkan hasil dengan termometer digital berbasis inframerah.
3. Menguji sensor MPX5700DP untuk memastikan apakah sensor dapat membaca tekanan darah sistolik dan diastolik dengan benar ketika manset mulai bekerja. Pengujian dilakukan dengan membandingkan hasil dengan alat tensimeter digital atau manual.
4. Menguji tampilan LCD TFT 3,5 inci apakah semua data hasil pengukuran dapat ditampilkan *real-time* dan benar.
5. Menguji pengiriman data ke aplikasi IoT Blynk dan Firebase melalui ESP32 untuk memastikan apakah hasil pembacaan sensor dapat ditransmisikan melalui jaringan Wi-Fi dan ditampilkan secara *real-time* di aplikasi.
6. Menguji *website* yang telah dibuat dan terintegrasi dengan Firebase dengan cara melakukan penginputan data pengguna terlebih dahulu. Setelah data dikirimkan, *website* melakukan pengambilan data hasil pengukuran berdasarkan ID pengukuran yang terdapat di Firebase, kemudian mengirimkan data tersebut ke WhatsApp penerima yang dituju.

3.10. Pengambilan Data

Pengambilan data dilakukan setelah proses perakitan dan pengujian alat selesai secara keseluruhan dan berfungsi dengan baik. Data yang diambil melibatkan tiga jenis sensor, yaitu pulse sensor untuk mengukur denyut nadi, sensor

MLX90614 tipe GY-906-BCC <10 cm untuk mengukur suhu tubuh, dan sensor MPX5700DP untuk mengukur tekanan darah. Proses pengambilan data melibatkan pengguna dari rentang usia yang beragam.

Pertama, pengambilan data denyut nadi menggunakan pulse sensor dengan menempatkan jari telunjuk pengguna pada permukaan sensor, kemudian nilai dari pengukuran tersebut dicatat dan dibandingkan dengan alat pengukur denyut nadi tensimeter digital. Kedua, untuk pengambilan data suhu tubuh menggunakan sensor MLX90614 tipe GY-906-BCC dengan mengarahkan sensor ke dahi pengguna dalam jangkauan maksimal 10 cm, kemudian sensor tersebut akan otomatis mendeteksi suhu tubuh dan data hasil pengukuran tersebut juga akan dibandingkan dengan alat pengukur suhu tubuh, seperti termometer inframerah digital.

Terakhir, untuk pengambilan data tekanan darah menggunakan sensor MPX5700DP dengan cara menghubungkan sensor ke manset yang telah dibalutkan ke lengan pengguna. Sensor ini akan mendeteksi perubahan tekanan udara pada saat proses inflasi dan deflasi pada manset berlangsung. Data tekanan yang terbaca merupakan nilai analog kemudian akan dikonversi oleh Arduino Uno R3 ATmega328P menjadi nilai tekanan sistolik dan diastolik dalam satuan mmHg menggunakan rumus kalibrasi, kemudian data hasil pengukuran tersebut akan dibandingkan dengan alat tensimeter digital sebagai alat pembanding tingkat akurasi sensor.

3.11. Analisis Hasil

Analisis hasil dilakukan setelah proses pengambilan data selesai dilakukan. Tujuan dari analisis ini adalah untuk mengevaluasi data yang diperoleh dari masing-masing data pengujian sensor. Data denyut nadi, suhu, dan tekanan darah dianalisis dengan membandingkan data hasil pengukuran dari alat yang dikembangkan dengan data hasil pengukuran dari alat medis standar yang umum digunakan oleh tenaga kesehatan, seperti tensimeter digital dan termometer inframerah digital. Proses analisis data mencakup pengolahan hasil pembacaan sensor untuk menghitung nilai rata-rata, menghitung selisih antara data sensor dengan alat pembanding, serta menganalisis kestabilan data, waktu respons, dan kepekaan

sensor terhadap faktor eksternal. Hasil analisis akan menjadi penilaian alat secara keseluruhan dan kemampuannya kesehatan secara *real-time*.

3.12. Pembuatan Laporan

Pembuatan laporan dilakukan setelah seluruh tahapan selesai dilakukan, meliputi pembuatan alat, pengujian alat, pengambilan data, serta analisis hasil. Laporan ini disusun secara sistematis agar mudah dipahami dan dapat menjadi dokumentasi resmi dari proses pengembangan alat pengukur dan sistem monitoring denyut nadi, suhu, dan tekanan darah berbasis *Internet of Things* (IoT).

Laporan ini disusun sedemikian rupa dengan mengikuti format penulisan yang telah disediakan oleh perguruan tinggi, yang mencakup pendahuluan, landasan teori, metodologi penelitian, perancangan dan pembuatan alat, pengujian, pengambilan data, analisis hasil data, kesimpulan, dan saran. Setiap bagian ditulis sesuai dengan data dan proses yang dilakukan secara langsung selama kegiatan pembuatan alat berlangsung.

Laporan ini diharapkan bisa digunakan sebagai referensi untuk pengembangan alat yang serupa di masa yang akan datang. Dengan adanya laporan ini, setiap langkah dan hasil yang telah didapatkan dapat dijadikan bahan dasar perbaikan dan pengembangan lebih lanjut.

BAB IV

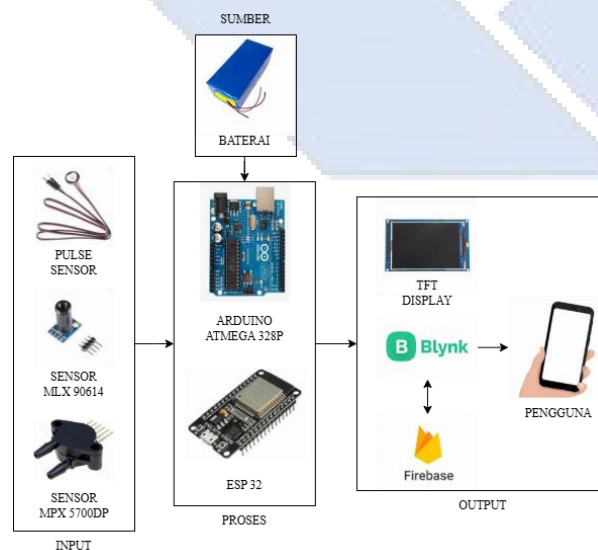
PEMBAHASAN

4.1. Deskripsi Alat

Alat pengukuran dan sistem monitoring alat kesehatan bertujuan untuk memonitoring tiga parameter vital, yaitu denyut nadi, suhu, dan tekanan darah. Alat pengukuran kesehatan ini diintegrasikan oleh tiga sensor secara bersamaan dan dapat digunakan oleh anak-anak hingga dewasa. Data hasil pengukuran dapat ditampilkan secara *real-time* melalui layar LCD TFT 3,5 inci. Sistem ini menggunakan *Internet of Things* (IoT) sehingga hasil pengukuran dapat ditampilkan melalui server Blynk, disimpan di Firebase untuk keperluan analisis, serta mengembangkan *website* yang terintegrasi dengan Firebase, dan mengirimkan data hasil pengukuran ke WhatsApp penerima yang dituju.

4.1.1. Sistem Kerja Alat

Sistem kerja dari alat kesehatan pengukur dan sistem monitoring denyut nadi, suhu, dan tekanan darah berbasis *Internet of Things* (IoT) dapat dijelaskan melalui blok diagram berikut pada Gambar 4.1.



Gambar 4.1 Blok Diagram

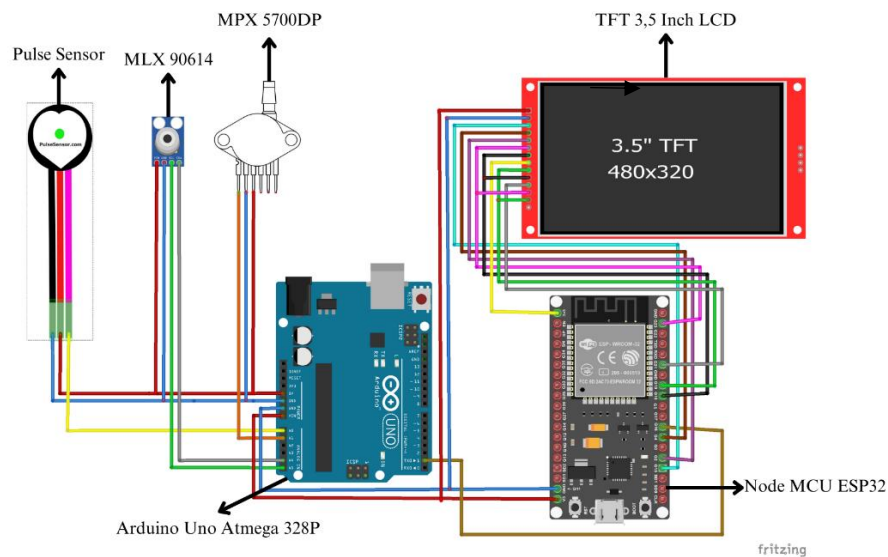
Sistem ini dirancang untuk memantau parameter kesehatan dengan tiga bagian utama, yaitu *input*, proses, dan *output*. Bagian *input* terdiri dari beberapa sensor yang digunakan untuk melakukan pengukuran pada kondisi tubuh. Pulse sensor digunakan untuk mengukur denyut nadi, MLX90614 digunakan untuk mengukur suhu tubuh secara non-kontak dengan mendeteksi radiasi inframerah di dahi pengguna, dan MPX5700DP digunakan untuk mengukur tekanan darah.

Data dari sensor dikirimkan ke bagian proses, yang terdapat Arduino Uno R3 ATmega328P yang berperan dalam pengolahan data dari sensor, lalu mengonversinya dari sinyal analog ke digital. Setelah itu, data akan dikirimkan ke ESP32, yang berperan sebagai modul komunikasi dalam menghubungkan sistem ke *platform* IoT agar dapat dipantau dari jarak jauh.

Pada bagian *output*, hasil pengukuran kesehatan ditampilkan melalui tiga metode utama. Pertama, LCD TFT 3,5 inci digunakan untuk menampilkan data secara *real-time* kepada pengguna, sehingga pengguna dapat melihat langsung hasil pengukuran. Kedua, data dikirimkan ke server Blynk yang berfungsi sebagai *platform* monitoring berbasis *online*, sehingga memungkinkan operator memantau kondisi kesehatan pengguna secara *real-time* melalui *web* atau aplikasi. Ketiga, Firebase digunakan sebagai penyimpanan data secara offline, sehingga data hasil tersebut bisa digunakan untuk analisis dan riwayat jangka panjang. Selain ketiga metode tersebut, sistem ini juga mengembangkan sebuah situs *web* terintegrasi dengan Firebase yang secara otomatis mengambil data hasil pengukuran melalui ID pengukuran dan mengirimkannya ke WhatsApp penerima yang dituju sebagai fitur notifikasi. Informasi tersebut dapat diakses kapan saja sesuai kebutuhan.

4.1.2. Rancangan *Hardware*

Pada gambar 4.2 menunjukan *wiring* diagram yang dimana terdapat beberapa komponen seperti pulse sensor, sensor MLX90614, sensor MPX5700DP, Arduino Uno R3 ATmega328P, ESP32, dan LCD TFT 3,5 inci.



Gambar 4.2 *Wiring* Diagram

Pada sistem *wiring* diatas, pulse sensor dihubungkan dengan pin A0 pada Arduino Uno R3 ATmega 328P, sementara pin GND dan VCC dihubungkan dengan GND dan VCC pada Arduino Uno R3 ATmega328P. Selain itu, untuk pengukuran tekanan darah digunakan sensor MPX5700DP yang terintegrasi dengan manset cuff, solenoid, dan motor DC, sensor ini memiliki *output* yang dihubungkan dengan pin A1 pada Arduino, serta pin GND dan VCC di jumper dengan GND dan VCC pulse sensor. Sementara itu, sensor MLX90614 digunakan untuk mendeteksi radiasi inframerah dengan memiliki empat pin, yaitu VIN yang dihubungkan dengan VCC, GND dihubungkan ke GND, serta SCL dan SDA masing-masing dihubungkan dengan pin A5 dan A4 pada Arduino. ESP32 digunakan sebagai sistem monitoring yang menampilkan hasil pengukuran dari ketiga sensor yang dikirimkan oleh Arduino Uno R3 ATmega328P melalui komunikasi serial pada pin TX ke RX ESP32. Data tersebut kemudian ditampilkan secara *real-time* pada LCD

TFT 3,5 inci dan melalui *platform* Blynk sebagai bagian dari sistem IoT yang menampilkan hasil pengukuran dari ketiga sensor. Selain itu, data juga disimpan secara *offline* di Firebase dan dapat diakses melalui *website* yang terintegrasi dengan Firebase. Selanjutnya, hasil pengukuran diambil dari Firebase menggunakan ID pengukuran, kemudian dikirimkan ke WhatsApp penerima yang dituju.

4.2. Perancangan dan Pembuatan Konstruksi

Pembuatan alat melibatkan beberapa tahapan, yaitu perancangan dan pembuatan konstruksi. Berikut ini dijelaskan tahapan yang dilakukan dalam perancangan dan pembuatan konstruksi pada proyek akhir ini.

4.2.1. Perancangan Konstruksi

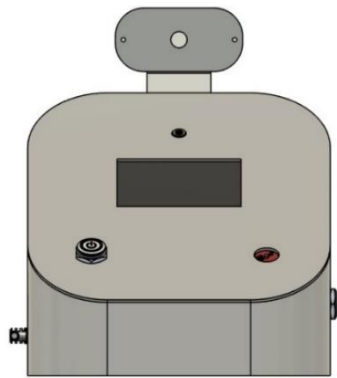
Pembuatan konstruksi memerlukan perhatian khusus agar hasilnya optimal dan sesuai dengan rancangan desain. Pada alat ini, rancangan konstruksi yang digunakan memiliki tiga utama, yaitu box sebagai wadah yang berisi komponen elektronik, penutup sebagai pelindung alat yang berada di dalam box, dan tempat khusus untuk pemasangan sensor suhu.

Box memiliki ukuran 19 cm × 19 cm untuk panjang dan lebar, dengan tinggi 178 cm. Ukuran ini disesuaikan dengan penempatan komponen elektronik yang dipasang di dalamnya agar memudahkan pengoperasian serta memastikan komponen terintegrasi dengan baik tanpa ada bagian yang bertabrakan atau longgar.

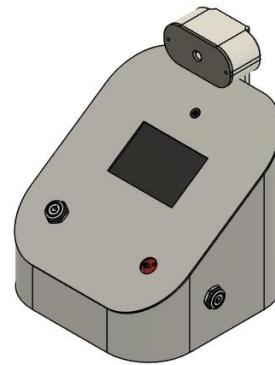
Bahan yang digunakan berupa plastik jenis Polyethylene Terephthalate Glycol-modified (PETG), yang dicetak menggunakan teknologi 3D printing. Penggunaan plastik PETG membuat desain konstruksi memiliki bobot yang ringan dan kuat, sehingga memudahkan dalam memindahkan atau membawanya. Selain itu, PETG memiliki ketahanan terhadap bahan kimia, ketahanan benturan, dan memiliki transparansi yang sangat baik, sehingga cocok digunakan dalam berbagai aplikasi, termasuk dalam pembuatan konstruksi alat pengukuran kesehatan ini.

Desain konstruksi terbagi menjadi tiga bagian utama, yaitu desain box, desain penutup, dan desain sensor suhu MLX90614. Pada desain box, di dalamnya

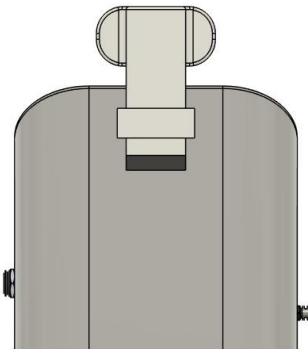
terdapat komponen-komponen elektronik yang sudah terintegrasi. Pada desain penutup, bagian depan alat, terdapat layar LCD TFT berukuran 3,5 inci yang berfungsi sebagai tampilan data hasil pengukuran secara *real-time*. Pada desain sensor suhu MLX90614 terdapat bagian diatas box sebagai penempatan sensor suhu MLX90614 untuk melakukan pengukuran suhu. Desain rancangan konstruksi dapat dilihat pada gambar berikut.



Gambar 4.3 Perancangan Konstruksi Tampak Depan



Gambar 4.4 Perancangan Konstruksi Tampak Samping



Gambar 4.5 Perancangan Konstruksi Tampak Belakang

4.2.2. Pembuatan Konstruksi

Pada pembuatan konstruksi, proses dilakukan berdasarkan desain yang telah dirancang di bagian perancangan konstruksi. Pemilihan bahan material sangat penting dilakukan supaya alat pengukur kesehatan denyut nadi, suhu, dan tekanan darah berbasis *Internet of Things* (IoT) dapat bekerja dengan baik dan tahan lama.

Alat ini dibuat menggunakan bahan plastik jenis *Polyethylene Terephthalate Glycol-modified* (PETG) dan dicetak menggunakan teknologi 3D printing, yang bisa mencetak benda tiga dimensi sehingga memungkinkan pembuatannya presisi dengan bentuk yang dibuat sesuai desain. Konstruksi alat terdiri dari tiga bagian utama, yaitu box sebagai wadah yang berisi komponen elektronik, penutup sebagai pelindung alat yang berada di dalam box, dan tempat khusus untuk pemasangan sensor suhu. Hasil pembuatan konstruksi alat dapat dilihat pada gambar berikut.



Gambar 4.6 Konstruksi Tampak Depan



Gambar 4.7 Konstruksi Tampak Samping



Gambar 4.8 Konstruksi Tampak Belakang

4.3. Perancangan dan Pembuatan Kontrol Alat Pengukuran

Perancangan dan pembuatan kontrol alat pengukuran terdiri dari komponen-komponen yang saling terintegrasi untuk menghasilkan pengukuran yang akurat. Komponen-komponen tersebut meliputi sistem catu daya dan sistem kontrol, alat

pengukur denyut nadi, suhu, dan tekanan darah, serta tampilan data menggunakan LCD TFT dengan ukuran 3,5 inci. Berikut merupakan penjelasan secara rinci proses perancangan dan pembuatan kontrol alat pengukuran.

4.3.1. Perancangan dan Pembuatan Sistem Catu Daya dan Sistem Kontrol

Sistem catu daya dan kontrol merupakan bagian dalam pengukuran yang penting untuk menyediakan energi dan mengontrol keseluruhan alat. Proses perancangan dan pembuatan terbagi menjadi dua, yaitu sistem penyimpanan energi dan penurun tegangan, serta sistem kontrol menggunakan push button.

4.3.1.1. Perancangan dan Pembuatan Sistem Penyimpanan Energi dan Penurun Tegangan

Pada bagian alat pengukuran ini membahas proses perancangan dan pembuatan sistem yang berfungsi untuk menyimpan energi dan menurunkan tegangan sesuai kebutuhan alat, sehingga komponen elektronik dapat bekerja secara optimal.

Pembuatan alat pengukuran menggunakan baterai yang memiliki daya tahan lama dan dapat diisi ulang ketika baterai sudah habis. Baterai yang digunakan adalah battery pack lithium dengan kapasitas arus sebesar 20A dan tegangan sebesar 12V, yang dapat memberikan energi secara optimal untuk mendukung seluruh rangkaian alat selama waktu pengoperasian yang telah ditentukan. Tegangan kerja komponen elektronik, terutama rangkaian sensor dan mikrokontroler seperti Arduino Uno R3 ATmega328P dan ESP32, adalah 3,3V atau 5V. Oleh karena itu, untuk menyesuaikan tegangan baterai terhadap kebutuhan komponen, maka diperlukan rangkaian penurun tegangan (step-down).

Rangkaian penurun tegangan (step-down) yang digunakan adalah modul LM2596, yang berfungsi untuk menjaga kestabilan tegangan dari baterai dengan menurunkan tegangan dari nilai yang tinggi ke nilai yang rendah guna mencegah perubahan tegangan yang berpotensi merusak komponen atau kinerja alat.

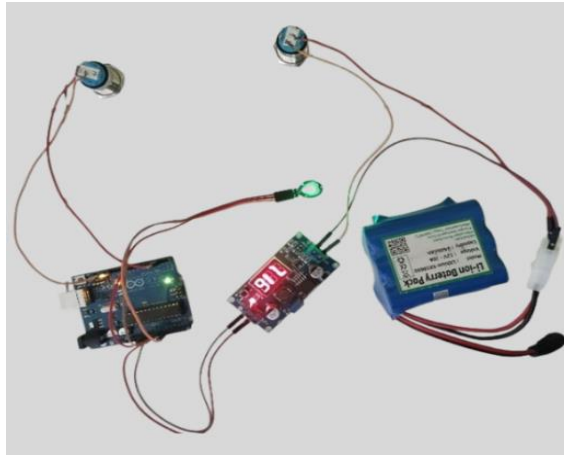
4.3.1.2. Perancangan dan Pembuatan Sistem Kontrol

Pembuatan alat pengukuran dan sistem monitoring denyut nadi, suhu, dan tekanan darah berbasis *Internet of Things* (IoT) menggunakan sistem kontrol yang terdiri dari dua buah push button sebagai pengendali utama pada alat pengukuran. Tombol pertama berfungsi untuk menghidupkan dan mematikan alat pengukuran. Saat tombol ditekan, sistem melakukan inisialisasi dan mengaktifkan semua sensor, sehingga alat siap melakukan pengukuran. Ketika tombol ditekan kembali, alat akan mati secara keseluruhan untuk menghentikan semua proses pengukuran.

Tombol kedua berfungsi untuk memulai proses pengukuran secara bersamaan pada parameter vital, yaitu denyut nadi, suhu, dan tekanan darah. Saat tombol ditekan, alat akan memulai proses pengukuran dimulai dari pembacaan sensor, kemudian data diproses oleh mikrokontroler seperti Arduino Uno R3 ATmega328P dan ESP32, serta mengirimkan data hasil pengukuran secara *real-time* ke ketiga *platform* dalam sistem monitoring berbasis *Internet of Things* (IoT). Dengan menggunakan dua tombol, maka dapat memudahkan dalam mengendalikan alat secara berurutan, dimulai dari menghidupkan alat pengukuran, kemudian melakukan proses pengukuran, hingga mematikan alat pengukuran.

4.3.2. Perancangan dan Pembuatan Kontrol Alat Pengukur Denyut Nadi

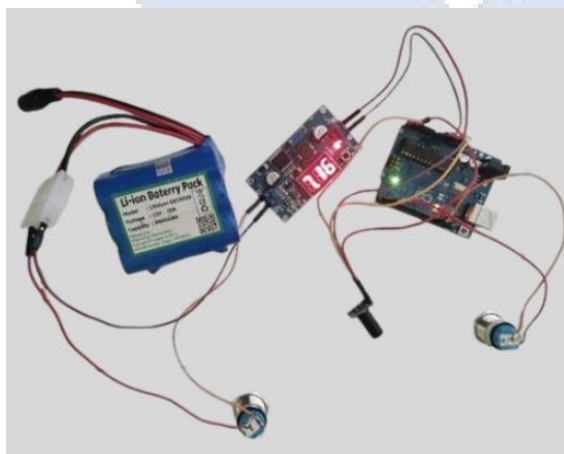
Perancangan alat pengukur denyut nadi dilakukan menggunakan aplikasi Fritzing, yang bertujuan untuk merancang alat sesuai kebutuhan. Pada sistem kontrol dalam pengukuran ini, digunakan mikrokontroler Arduino Uno R3 ATmega328P yang berfungsi membaca data dari sensor pengukuran. Data hasil pembacaan kemudian dikirimkan ke mikrokontroler ESP32, yang berperan sebagai modul komunikasi untuk mengirimkan data tersebut secara *real-time* ke tiga *platform* IoT. Berikut merupakan gambar rangkaian dari alat pengukur denyut nadi.



Gambar 4.9 Rangkaian Alat Pengukur Denyut Nadi

4.3.3. Perancangan dan Pembuatan Kontrol Alat Pengukur Suhu

Perancangan alat pengukur suhu dilakukan menggunakan aplikasi Fritzing, yang bertujuan untuk merancang alat sesuai kebutuhan. Pada sistem kontrol dalam pengukuran ini, digunakan mikrokontroler Arduino Uno R3 ATmega328P yang berfungsi membaca data dari sensor pengukuran. Data hasil pembacaan kemudian dikirimkan ke mikrokontroler ESP32, yang berperan sebagai modul komunikasi untuk mengirimkan data tersebut secara *real-time* ke tiga *platform* IoT. Berikut merupakan gambar rangkaian dari alat pengukur suhu tubuh.



Gambar 4.10 Rangkaian Alat Pengukur Suhu

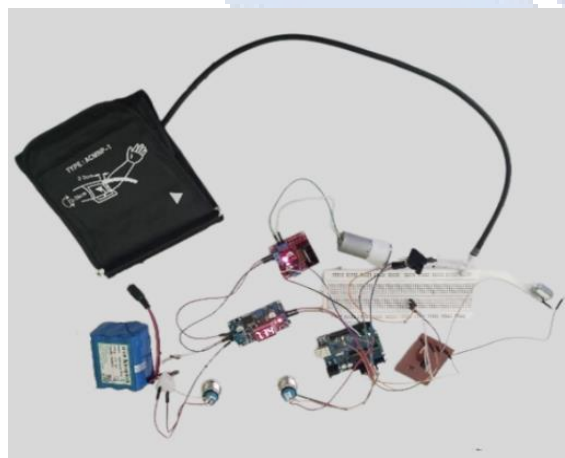
4.3.4. Perancangan dan Pembuatan Kontrol Alat Pengukur Tekanan Darah

Perancangan alat pengukur tekanan darah dilakukan menggunakan aplikasi Fritzing, yang bertujuan untuk merancang alat sesuai kebutuhan. Pada sistem kontrol pengukuran ini, digunakan mikrokontroler Arduino Uno R3 ATmega328P yang berfungsi membaca data dari sensor pengukuran. Data hasil pembacaan kemudian dikirimkan ke mikrokontroler ESP32, yang berperan sebagai modul komunikasi untuk mengirimkan data secara *real-time* ke tiga *platform* IoT.

Pada alat pengukur tekanan darah ini, terdapat beberapa komponen utama yang saling terintegrasi untuk menghasilkan pengukuran yang akurat dan stabil. Sistem alat pengukuran tekanan darah terdiri dari sensor, sistem kontrol, serta sistem solenoid dan pompa pump. Penjelasan mengenai perancangan dan pembuatan sistem solenoid dan pompa pump dapat dilihat pada subbab 4.3.4.1.

4.3.4.1. Perancangan dan Pembuatan Sistem Solenoid dan Pompa Pump

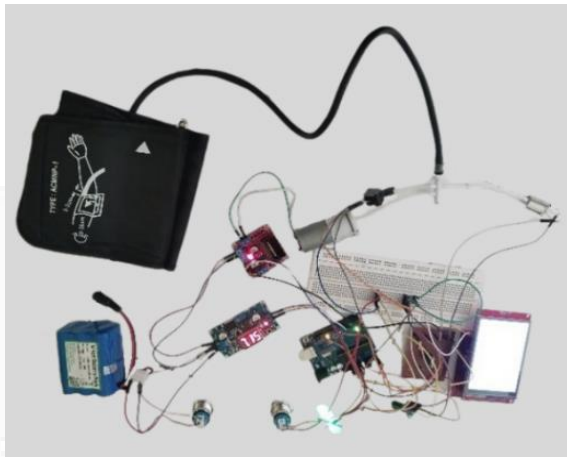
Sistem solenoid dan pompa pump merupakan bagian penting dari alat pengukuran tekanan darah. Pompa pump berfungsi untuk memompa udara ke manset hingga mencapai nilai tertentu, sedangkan solenoid valve berfungsi sebagai katup udara yang masuk dan keluar manset, sehingga pengukuran bisa berjalan dengan baik. Berikut merupakan gambar rangkaian sistem solenoid dan pompa pump yang terintegrasi dengan kontrol alat pengukur tekanan darah.



Gambar 4.11 Rangkaian Alat Pengukur Tekanan Darah

4.3.5. Perancangan dan Pembuatan LCD TFT 3,5 Inchi

Pengukuran parameter vital seperti denyut nadi, suhu, dan tekanan darah memerlukan data hasil pengukuran secara *real-time* yang akurat dan mudah dibaca. Pada alat pengukuran ini, digunakan layar LCD TFT berukuran 3,5 inci sebagai tampilan utama untuk menampilkan hasil pengukuran pengguna, yang meliputi nilai pengukuran denyut nadi, suhu, dan tekanan darah, serta klasifikasi hasil pengukuran berdasarkan parameter tersebut. Berikut merupakan gambar rangkaian seluruh sistem yang telah terintegrasi dengan LCD TFT 3,5 inci.



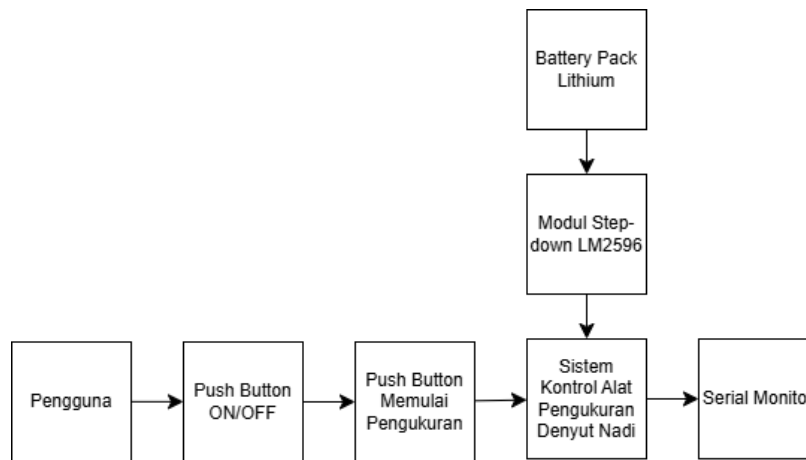
Gambar 4.12 Rangkaian Seluruh Sistem

4.4. Pengujian Kontrol Alat Pengukuran

Pengujian kontrol terbagi menjadi beberapa bagian, yaitu pengujian sistem catu daya, pengujian alat pengukur denyut nadi, pengujian alat pengukur suhu, dan pengujian alat pengukur tekanan darah.

4.4.1. Pengujian Kontrol Alat Pengukur Denyut Nadi

Pengujian dilakukan untuk mengetahui apakah pulse sensor mampu membaca denyut nadi secara akurat dan stabil. Berikut adalah gambar blok pengujian pada alat pengukur denyut nadi.



Gambar 4.13 Blok Pengujian Alat Pengukur Denyut Nadi

Pengujian dilakukan dengan mengkalibrasi sensor dengan membandingkan hasil pengukuran alat yang dibuat dengan hasil pengukuran menggunakan alat ukur tensimeter digital. Proses pengujian dilakukan dengan meletakkan ujung jari telunjuk pengguna pada sensor denyut nadi (pulse sensor), yang kemudian membaca sinyal analog. Sinyal ini diolah oleh mikrokontroler, yaitu Arduino Uno R3 ATmega328P dan ESP32, sehingga hasil pengukuran denyut nadi dapat ditampilkan pada serial monitor melalui aplikasi Arduino IDE.

Berikut adalah hasil kalibrasi pengukuran denyut nadi yang telah dibandingkan dengan pengukuran menggunakan alat ukur digital.

Tabel 4.1 Hasil Kalibrasi Pengukuran Denyut Nadi

Waktu (Detik)	Nilai Sensor (Analog)	Pulse Sensor Terdeteksi (Ya/Tidak)	Keterangan
1	505	Ya	Pulse Terdeteksi
2	511	Ya	Pulse Terdeteksi
3	508	Ya	Pulse Terdeteksi
4	512	Ya	Pulse Terdeteksi
5	523	Ya	Pulse Terdeteksi
6	512	Ya	Pulse Terdeteksi
7	527	Ya	Pulse Terdeteksi
8	514	Ya	Pulse Terdeteksi
9	524	Ya	Pulse Terdeteksi
10	514	Ya	Pulse Terdeteksi
11	515	Ya	Pulse Terdeteksi

12	516	Ya	Pulse Terdeteksi
13	518	Ya	Pulse Terdeteksi
14	514	Ya	Pulse Terdeteksi
15	515	Ya	Pulse Terdeteksi
16	513	Ya	Pulse Terdeteksi
17	514	Ya	Pulse Terdeteksi
18	519	Ya	Pulse Terdeteksi
19	512	Ya	Pulse Terdeteksi

Tabel 4.1 menunjukkan bahwa selama durasi pengujian kalibrasi 19 detik, pulse sensor memberikan output nilai analog yang berada dalam rentang antara 505 hingga 527. Seluruh data menunjukkan keberhasilan mendeteksi denyut nadi pada setiap detik pengujian. Hal ini menunjukkan bahwa konsistensi pembacaan sensor bekerja dengan baik dan mampu mendeteksi denyut jantung secara stabil setiap detiknya. Persentase *error* antara alat pengukur yang dibuat dengan alat pengukur denyut nadi secara digital dapat dihitung sebagai berikut:

$$\text{Persentase Error (\%)} = \left| \frac{\bar{X}_b - X_s}{X_s} \right| \times 100\%$$

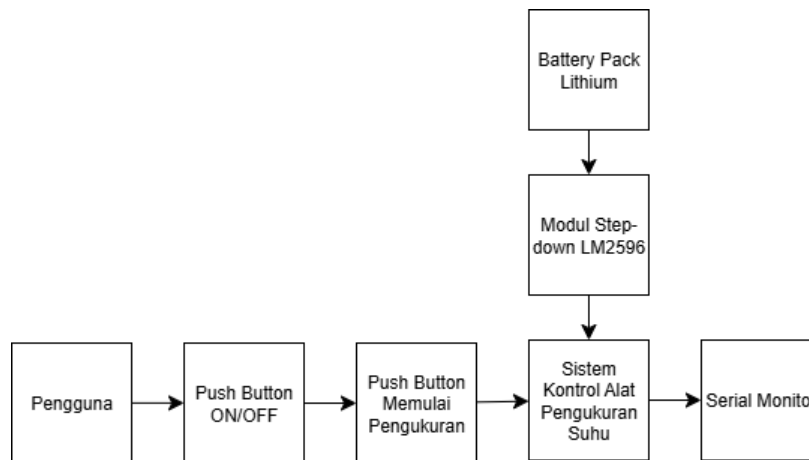
Keterangan:

- $\bar{X}_b$ = rata-rata pengukuran alat buatan
- X_s = nilai pengukuran alat standar

Persentase *error* yang diperoleh pada pengukuran denyut nadi menunjukkan bahwa alat pengukur denyut nadi ini berfungsi dengan baik dan memenuhi standar. Sehingga alat ini layak digunakan untuk mengukur denyut nadi secara akurat dan dapat diandalkan.

4.4.2. Pengujian Kontrol Alat Pengukur Suhu

Pengujian dilakukan untuk mengetahui apakah sensor MLX90614 GY-BCC <10 cm mampu membaca suhu secara akurat dan stabil. Berikut adalah gambar blok pengujian pada alat ukur suhu.



Gambar 4.14 Blok Pengujian Alat Pengukur Suhu

Pengujian dilakukan dengan mengkalibrasi sensor dengan membandingkan hasil pengukuran alat yang dibuat dengan hasil pengukuran menggunakan alat ukur termometer digital inframerah. Proses pengujian dilakukan dengan mengarahkan sensor suhu (MLX90614 GY-BCC) ke dahi pengguna dengan jarak <10 cm, yang kemudian membaca sinyal analog. Sinyal ini diolah oleh mikrokontroler, yaitu Arduino Uno R3 ATmega328P dan ESP32, sehingga hasil pengukuran suhu dapat ditampilkan pada serial monitor melalui aplikasi Arduino IDE.

Persentase *error* antara alat pengukur yang dibuat dengan alat pengukur suhu secara digital dapat dihitung sebagai berikut:

$$\text{Persentase Error (\%)} = \left| \frac{\bar{X}_b - X_s}{X_s} \right| \times 100\%$$

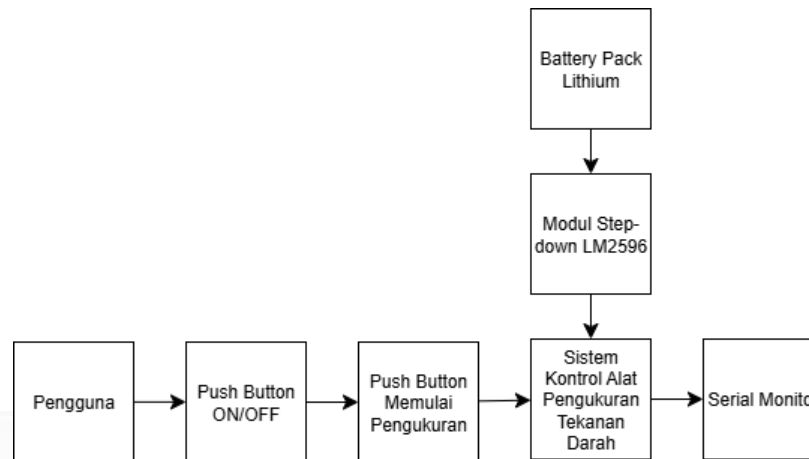
Keterangan:

- $\bar{X}_b$ = rata-rata pengukuran alat buatan
- X_s = nilai pengukuran alat standar

Persentase *error* yang diperoleh pada pengukuran suhu menunjukkan bahwa alat pengukur suhu ini berfungsi dengan baik dan memenuhi standar. Sehingga alat ini layak digunakan untuk mengukur suhu secara akurat dan dapat diandalkan.

4.4.3. Pengujian Kontrol Alat Pengukur Tekanan Darah

Pengujian dilakukan untuk mengetahui apakah sensor MPX5700DP mampu membaca tekanan darah secara akurat dan stabil. Berikut adalah gambar blok pengujian pada alat ukur tekanan darah.



Gambar 4.15 Blok Pengujian Alat Pengukur Tekanan Darah

Pengujian dilakukan dengan mengkalibrasi sensor dengan membandingkan hasil pengukuran alat yang dibuat dengan hasil pengukuran menggunakan alat ukur tensimeter digital. Proses pengujian dilakukan dengan mengaktifkan pompa motor sehingga manset mengembang untuk mencapai nilai sistolik, lalu dibiarkan mengempis secara perlahan melalui katup solenoid untuk pengambilan nilai diastolik. Proses ini dikombinasikan dengan sensor tekanan darah (MPX5700DP), yang kemudian membaca sinyal analog. Sinyal ini diolah oleh mikrokontroler, yaitu Arduino Uno R3 ATmega328P dan ESP32, sehingga hasil pengukuran tekanan darah dapat ditampilkan pada serial monitor melalui aplikasi Arduino IDE.

Berikut adalah hasil kalibrasi pengukuran tekanan darah yang telah dibandingkan dengan pengukuran menggunakan alat ukur digital.

Tabel 4.2 Hasil Kalibrasi Pengukuran Tekanan Darah

mmHg (Referensi)	Analog
0	39
10	40
20	42

30	44
40	45
50	47
60	48
70	50
80	52
90	53
100	55
110	57
120	59
130	61
140	62
150	64
160	66
170	68
180	70
190	72
200	73

Tabel 4.2 menunjukkan bahwa kalibrasi sensor tekanan darah terdapat adanya hubungan linier antara nilai analog yang dibaca dengan tekanan darah dalam satuan mmHg. Untuk proses kalibrasi, digunakan dua titik acuan, yaitu nilai analog 39 saat tekanan darah 0 mmHg dan nilai analog 73 saat tekanan darah 200 mmHg. Berdasarkan dua titik tersebut, dibuat persamaan garis lurus untuk mengonversi nilai analog menjadi tekanan darah.

Persamaan yang digunakan mengikuti format umum $y = mx + c$, di mana y adalah tekanan darah (mmHg), x adalah nilai analog, m adalah gradien, dan c adalah konstanta. Gradien diperoleh dari selisih nilai tekanan dibagi nilai selisih analog:

$$m = \frac{200 - 0}{73 - 39} = \frac{200}{34} \approx 5.88$$

Untuk mencari nilai c , digunakan titik (39, 0):

$$0 = 5.88 \times 39 + c \rightarrow c = -229.32$$

Sehingga persamaan konversi lengkap menjadi:

$$\text{Tekanan Darah (mmHg)} = 5.88 \times \text{Nilai Analog} - 229.32$$

Persamaan ini dapat diterapkan dalam sistem pembacaan sensor tekanan darah menggunakan mikrokontroler seperti Arduino Uno R3 ATmega328P dan ESP32. Sebagai contoh, jika sensor membaca nilai analog sebesar 60, maka tekanan darah dihitung sebagai berikut:

$$5.88 \times 60 - 229.32 = 123.48 \text{ mmHg}$$

Berdasarkan hasil kalibrasi pengukuran di atas, persentase *error* antara alat pengukur yang dibuat dengan alat pengukur tekanan darah secara digital dapat dihitung sebagai berikut:

$$\text{Persentase Error Sistolik (\%)} = \left| \frac{\bar{X}_b - X_s}{X_s} \right| \times 100\%$$

$$\text{Persentase Error Diastolik (\%)} = \left| \frac{\bar{X}_b - X_s}{X_s} \right| \times 100\%$$

Keterangan:

- $\bar{X}_b$ = rata-rata pengukuran alat buatan
- X_s = nilai pengukuran alat standar

Persentase *error* yang diperoleh pada pengukuran tekanan darah menunjukkan bahwa alat pengukur tekanan darah ini berfungsi dengan baik dan memenuhi standar. Sehingga alat ini layak digunakan untuk mengukur tekanan darah secara akurat dan dapat diandalkan.

4.5. Pengujian Alat Pengukur Secara Keseluruhan

Pembuatan alat pengukuran dan sistem monitoring denyut nadi, suhu, serta tekanan darah berbasis *Internet of Things* (IoT) menggunakan dua buah push button sebagai kontrol utama dalam alat pengukuran. Tombol pertama berfungsi untuk menghidupkan dan mematikan alat pengukuran. Saat tombol ditekan, sistem melakukan inisialisasi dan mengaktifkan semua sensor, sehingga alat siap untuk

melakukan pengukuran. Ketika tombol ditekan kembali, alat akan dimatikan secara keseluruhan untuk menghentikan semua proses pengukuran.

Parameter vital seperti denyut nadi, suhu, dan tekanan darah sangat penting dilakukan untuk menilai kondisi fisiologis seseorang secara cepat dan akurat. Ketiga parameter ini diukur secara bersamaan dengan sistem yang telah diintegrasikan dan hasil pengukuran dapat dilihat secara *real-time* melalui sistem monitoring.

Proses perancangan sistem kontrol alat pengukuran, sistem monitoring, serta pemrograman telah selesai dilakukan. Tahap selanjutnya adalah melakukan pengujian secara menyeluruh untuk memastikan alat yang dibuat berfungsi sesuai dengan tujuan.

Pengujian alat pengukuran dan sistem monitoring dilakukan secara menyeluruh dengan tujuan mengevaluasi tingkat keakuratan alat serta mengidentifikasi faktor-faktor yang mempengaruhinya. Pengujian ini dilakukan pada pengguna dengan variasi jenis kelamin, usia, dan kondisi fisik yang berbeda. Berikut adalah gambar pengujian alat pengukuran secara keseluruhan.



Gambar 4.16 Pengujian Alat Secara Keseluruhan

Sebelum melakukan pengukuran, terdapat langkah-langkah yang harus dilakukan oleh operator dan pengguna agar hasil pengukuran akurat, yaitu:

1. Operator memasukkan data pengguna melalui *website* yang telah dibuat sebagai langkah awal sebelum memulai pengukuran.
2. Pengguna meletakkan ujung jari tangan telunjuk ke pulse sensor untuk mengukur denyut nadi, mengarahkan sensor MLX90614 ke arah dahi yang menggunakan kabel panjang yang terletak di tengah box dengan jarak <10 cm untuk mengukur suhu, dan memasang manset tensi yang sudah terintegrasi oleh sensor MPX5700DP untuk mengukur tekanan darah.
3. Operator menekan tombol pertama untuk mengaktifkan alat.
4. Operator menekan tombol kedua untuk memulai pengukuran secara bersamaan.
5. Data hasil pengukuran ditampilkan secara *real-time* di LCD TFT 3,5 inci, server Blynk, dan Firebase.
6. Notifikasi akan muncul jika hasil pengukuran berada dalam rentang klasifikasi tertentu sesuai standar medis.
7. Operator menekan tombol pertama untuk menonaktifkan alat pengukuran setelah proses pengukuran selesai.
8. Data hasil pengukuran beserta data pengguna yang telah diinput sebelumnya akan dikirimkan ke WhatsApp pengguna sebagai laporan hasil pengukuran.

Proses perancangan, pembuatan alat, serta pemrograman telah selesai dilakukan. Tahap berikutnya adalah melakukan pengujian untuk memastikan fungsionalitas alat sesuai dengan yang diharapkan serta menghasilkan data pengukuran yang akurat. Pengujian ini bertujuan untuk mengevaluasi kinerja seluruh komponen yang terintegrasi secara menyeluruh di dalam sistem alat.

Pengujian dilakukan sesuai dengan prosedur yang telah ditetapkan untuk mengukur parameter vital seperti denyut nadi, suhu, dan tekanan darah. Data hasil pengujian kemudian dibandingkan dengan alat ukur digital sebagai tolak ukur tingkat akurasi alat yang telah dibuat. Berikut ini penjelasan mengenai pengujian alat pengukuran dan sistem monitoring denyut nadi, suhu, serta tekanan darah berbasis *Internet of Things* (IoT). Pengujian dilakukan secara bertahap, dimulai dengan pengujian masing-masing parameter secara terpisah. Setelah itu, pengujian

dilanjutkan pada sistem secara keseluruhan setelah seluruh komponen terintegrasi dalam satu perangkat.

Pengujian alat pengukur pada parameter kesehatan vital, seperti denyut nadi, suhu, dan tekanan darah, dilakukan oleh 15 orang pengguna dengan variasi usia, jenis kelamin, dan kondisi fisik yang berbeda. Alat pengukuran menggunakan komponen yang sudah terintegrasi, sehingga bisa melakukan pengukuran secara bersamaan. Data hasil pengukuran dapat dikirim secara *real-time* ke ketiga *platform*, yaitu LCD TFT 3,5 inci, server Blynk, dan Firebase. Selain itu, dikembangkan juga sebuah *website* yang terintegrasi dengan Firebase untuk mengambil data hasil pengukuran secara otomatis berdasarkan ID pengukuran, kemudian mengirimkan data tersebut ke WhatsApp penerima yang dituju.

Hasil data perbandingan pengujian secara keseluruhan antara alat pengukur yang dibuat dan alat pengukur standar manua dapat dilihat pada tabel berikut.

- Denyut Nadi

Tabel 4.3 Hasil Pengujian Denyut Nadi

No.	Nama Pengguna	Pengukuran Alat Standar Digital (BPM)	Pengukuran Alat Buatan (BPM)			Rata-rata Pengukuran Alat Buatan (BPM)	Persentase Akurasi (%)
			P1	P2	P3		
1.	Pengguna 1	66	64	84	84	77,33	82,83%
2.	Pengguna 2	75	84	84	84	84	88%
3.	Pengguna 3	80	84	84	88	85,33	93,34%
4.	Pengguna 4	72	80	76	80	78,67	90,74%
5.	Pengguna 5	104	84	88	80	84	80,77%
6.	Pengguna 6	72	84	88	88	86,67	79,62%
7.	Pengguna 7	80	85	84	80	83	96,25%
8.	Pengguna 8	74	84	84	84	84	86,49%
9.	Pengguna 9	75	96	80	84	86,67	84,44%
10.	Pengguna 10	80	84	92	88	88	90%
11.	Pengguna 11	94	88	84	76	82,67	87,95%
12.	Pengguna 12	81	80	80	85	81,67	99,18%
13.	Pengguna 13	70	84	88	80	84	80%
14.	Pengguna 14	80	88	96	88	90,67	86,67%
15.	Pengguna 15	60	88	88	84	86,67	55,55%
Rata-rata Persentase							85,46%

- Suhu

Tabel 4.4 Hasil Pengujian Suhu

No.	Nama Pengguna	Pengukuran Alat Standar Digital (°C)	Pengukuran Alat Buatan (°C)			Rata-rata Pengukuran Alat Buatan (°C)	Persentase Akurasi (%)
			P1	P2	P3		
1.	Pengguna 1	36,5	36,2	36,7	36,8	36,57	99,81%
2.	Pengguna 2	36,4	36,1	36	36,4	36,17	99,37%
3.	Pengguna 3	36,9	36,9	36,8	36,6	36,77	99,65%
4.	Pengguna 4	37,2	37	37	36	36,67	98,57%
5.	Pengguna 5	36,9	37,1	37,2	37	37,10	99,46%
6.	Pengguna 6	37,3	36,8	37,3	37,1	37,07	99,38%
7.	Pengguna 7	36,8	36,7	36,5	36,6	36,60	99,46%
8.	Pengguna 8	37,5	37,8	35,9	37,9	37,20	99,20%
9.	Pengguna 9	37,3	36,9	37	37,3	37,07	99,38%
10.	Pengguna 10	36,6	36,3	36,2	36,1	36,20	98,91%
11.	Pengguna 11	36,5	35,7	35,3	36,1	35,70	97,81%
12.	Pengguna 12	36,5	35	35	36	35,33	96,80%
13.	Pengguna 13	36,5	36	36	36	36	98,63%
14.	Pengguna 14	37	37,2	37,2	36,4	36,93	99,81%
15.	Pengguna 15	37,1	35,7	37,1	36,6	36,47	98,31%
Rata-rata Persentase							98,97%

- Tekanan Darah

Tabel 4.5 Hasil Pengujian Tekanan Darah

No.	Nama Pengguna	Pengukuran Alat Standar Digital (mmHg)		Pengukuran Alat Buatan (mmHg)						Rata-rata Pengukuran Alat Buatan (mmHg)		Persentase Akurasi (%)	
		Alat Standar		P1		P2		P3		Sis	Dia	Sis	Dia
		Sis	Dia	Sis	Dia	Sis	Dia	Sis	Dia				
1.	Pengguna 1	123	71	89	91	123	91	114	84	108,67	86,67	88,35%	77,93%
2.	Pengguna 2	106	71	112	84	111	90	110	84	111	86	95,29%	78,87%
3.	Pengguna 3	134	89	121	90	128	90	127	84	125,33	88	93,53%	98,88%
4.	Pengguna 4	100	70	116	90	99	62	112	76	109	76	91%	91,43%
5.	Pengguna 5	99	67	105	91	105	90	112	92	107,33	91	91,59%	64,18%
6.	Pengguna 6	103	74	87	90	127	84	105	84	106,33	86	96,77%	83,79%
7.	Pengguna 7	110	71	120	80	122	90	100	84	114	84,67	96,37%	80,75%
8.	Pengguna 8	101	63	94	84	128	84	97	84	106,33	84	94,73%	66,67%
9.	Pengguna 9	133	80	95	84	129	84	104	84	109,33	84	82,21%	95,00%
10.	Pengguna 10	120	83	87	90	101	90	128	84	105,33	88	87,78%	93,98%
11.	Pengguna 11	126	73	105	84	118	90	130	91	117,67	88,33	93,33%	79,04%
12.	Pengguna 12	124	70	116	90	121	90	132	84	123	88	99,2%	74,29%
13.	Pengguna 13	135	80	120	90	131	90	115	90	122	90	90,38%	87,5%
14.	Pengguna 14	88	60	112	90	111	85	102	84	108,33	86,33	76,9%	56,12%
15.	Pengguna 15	127	84	128	84	117	90	98	90	114,33	88	90,03%	95,24%
Rata-rata Persentase											91,16%	81,58%	

Berdasarkan data hasil pengukuran secara keseluruhan, dilakukan perhitungan untuk menentukan persentase *error* pada alat pengukur yang dibuat dengan acuan alat pengukur standar digital. Rumus yang digunakan untuk menghitung persentase *error* dan persentase akurasi pada tabel di atas adalah sebagai berikut.

Persentase error dihitung dengan rumus sebagai berikut:

$$\text{Persentase Error (\%)} = \left| \frac{\bar{X}_b - X_s}{X_s} \right| \times 100\%$$

Keterangan:

- $\bar{X}_b$ = rata-rata pengukuran alat buatan
- X_s = nilai pengukuran alat standar

Persentase akurasi dihitung dengan rumus sebagai berikut:

$$\text{Persentase Akurasi} = 100\% - \text{Persentase Error (\%)}$$

➤ Perhitungan pada pengguna pertama

- Pengukuran Denyut Nadi

$$\text{Persentase Error (\%)} = \left| \frac{77,33 - 66}{66} \right| \times 100\% = 17,17\%$$

$$\text{Persentase Akurasi} = 100\% - 17,17\% = 82,83\%$$

- Pengukuran Suhu

$$\text{Persentase Error (\%)} = \left| \frac{36,57 - 36,5}{36,5} \right| \times 100\% = 0,19\%$$

$$\text{Persentase Akurasi} = 100\% - 0,19\% = 99,81\%$$

- Pengukuran Tekanan Darah

$$\text{Persentase Error Sistolik (\%)} = \left| \frac{108,67 - 123}{123} \right| \times 100\% = 11,65\%$$

$$\text{Persentase Akurasi Sistolik} = 100\% - 11,65\% = 88,35\%$$

$$\text{Persentase Error Diastolik (\%)} = \left| \frac{86,67 - 71}{71} \right| \times 100\% = 22,07\%$$

$$\text{Persentase Akurasi Diastolik} = 100\% - 22,07\% = 77,93\%$$

➤ Perhitungan pada pengguna kedua

- Pengukuran Denyut Nadi

$$\text{Persentase Error (\%)} = \left| \frac{84 - 75}{75} \right| \times 100\% = 12\%$$

$$\text{Persentase Akurasi} = 100\% - 12\% = 88\%$$

- Pengukuran Suhu

$$\text{Persentase Error (\%)} = \left| \frac{36,17 - 36,4}{36,4} \right| \times 100\% = 0,63\%$$

$$\text{Persentase Akurasi} = 100\% - 0,63\% = 99,37\%$$

- Pengukuran Tekanan Darah

$$\text{Persentase Error Sistolik (\%)} = \left| \frac{111 - 106}{106} \right| \times 100\% = 4,71\%$$

$$\text{Persentase Akurasi Sistolik} = 100\% - 4,71\% = 95,29\%$$

$$\text{Persentase Error Diastolik (\%)} = \left| \frac{86 - 71}{71} \right| \times 100\% = 21,13\%$$

$$\text{Persentase Akurasi Diastolik} = 100\% - 21,13\% = 78,87\%$$

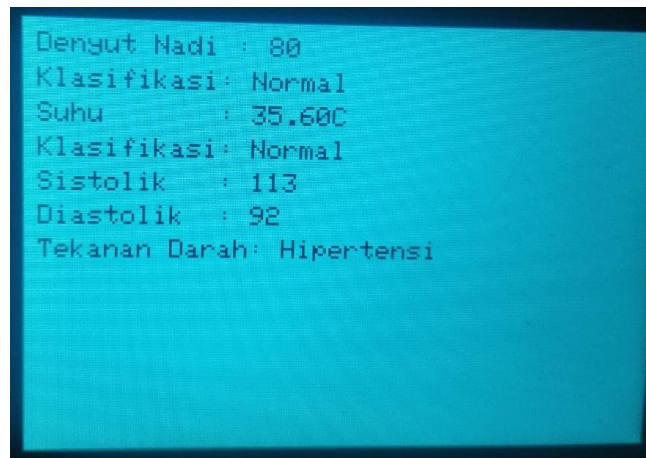
4.6. Tampilan dan Pengiriman Data Hasil Pengukuran Secara Keseluruhan

Sistem monitoring tiga parameter vital, yaitu denyut nadi, suhu, dan tekanan darah, dirancang untuk menampilkan dan mengirimkan data hasil pengukuran secara *real-time* serta terintegrasi. Data yang diperoleh dari sensor akan diproses

oleh mikrokontroler, yaitu Arduino Uno R3 ATmega328P, kemudian diteruskan ke ESP32 untuk dikirimkan pada ke tiga *platform* utama, yaitu LCD TFT 3,5 inci, server Blynk, dan Firebase. Selain itu, sistem ini juga mengembangkan sebuah *website* yang terintegrasi dengan Firebase untuk pengiriman hasil pengukuran melalui WhatsApp, sehingga pengguna dapat menerima dan memantau kondisi kesehatan diri sendiri dengan melihat data hasil yang diperoleh secara berkelanjutan. Berikut penjelasan mengenai tampilan data hasil pengukuran pada masing-masing platform.

4.6.1. Tampilan LCD TFT 3,5 Inci

Proses pengukuran dilakukan terlebih dahulu hingga diperoleh data hasil pengukuran yang valid dan akurat. Data tersebut ditampilkan secara *real-time* pada layar LCD TFT 3,5 inci, sehingga pengguna dapat melihat hasil pengukuran secara langsung. Pada layar ini, informasi yang ditampilkan meliputi waktu pengukuran, yang mencakup tanggal dan jam saat melakukan pengukuran, nilai denyut nadi dalam satuan *Beats Per Minute* (BPM), suhu dalam derajat Celsius (°C), serta tekanan darah yang terdiri dari tekanan sistolik dan diastolik dalam satuan mmHg. Selain itu, layar juga menampilkan klasifikasi untuk masing-masing parameter, sehingga pengguna dapat mengetahui dan memantau kondisi kesehatannya secara lebih mudah. Berikut merupakan gambar tampilan data hasil pengukuran pada LCD TFT 3,5 inci.



Gambar 4.17 Tampilan LCD TFT 3,5 Inchi

LCD TFT 3,5 inci yang digunakan merupakan versi dengan interface SPI (Serial Peripheral Interface) dan driver ILI9488. Pemilihan tipe SPI didasarkan pada kebutuhan efisiensi pin mikrokontroler dan kemudahan integrasi dalam sistem embedded. Interface SPI memiliki jalur komunikasi serial yang relatif sederhana dengan jumlah pin yang lebih sedikit jika dibandingkan interface paralel, sehingga menghemat penggunaan pin pada mikrokontroler. SPI memungkinkan kecepatan pengiriman data yang cukup tinggi, sehingga mampu menampilkan data dan grafik secara *real-time* dengan responsif. Driver ILI9488 mendukung resolusi layar 480 × 320 piksel dengan warna 16-bit, memberikan tampilan yang tajam dan warna yang cukup kaya untuk menampilkan data hasil pengukuran secara jelas. Berikut adalah kelebihan dan kekurangan LCD TFT 3,5 inci versi SPI.

- Kelebihan

1. Penggunaan pin yang efisien.
2. Kecepatan transfer data yang cukup tinggi.
3. Kemudahan integrasi.
4. Resolusi dan warna baik.
5. Konsumsi daya relatif rendah.

- Kekurangan

1. Kecepatan transfer data lebih rendah dibandingkan interface paralel.
2. Pengaturan clock dan sinkronisasi lebih kompleks.
3. Rentan terhadap interferensi sinyal pada jarak kabel panjang.

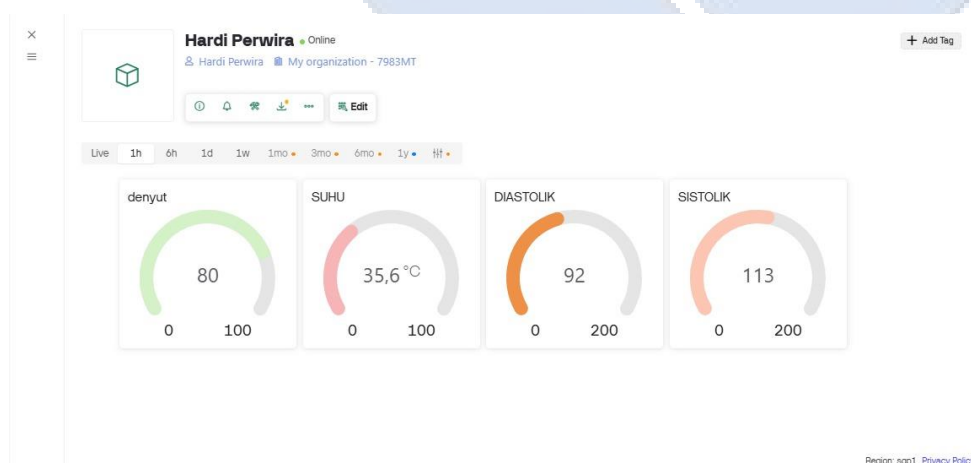
4. Sudut pandang dan kualitas warna terbatas.

4.6.2. Tampilan Blynk

Blynk merupakan platform *Internet of Things* (IoT) yang digunakan untuk memantau data hasil pengukuran secara *real-time* melalui jaringan internet. Dalam sistem monitoring ini, Blynk berfungsi untuk menampilkan data hasil pengukuran secara *real-time* melalui aplikasi atau *website* sesuai kebutuhan pengguna, sehingga memungkinkan pemantauan dari jarak jauh.

Penghubung perangkat dengan *platform* Blynk memerlukan konfigurasi khusus berupa Template ID dan Auth Token yang diperoleh dari akun Blynk Cloud yang telah dibuat. Template ID berfungsi sebagai identifikasi jenis perangkat yang dikontrol melalui *platform* Blynk, sedangkan Auth Token memungkinkan perangkat keras berkomunikasi dengan proyek tertentu di Blynk.

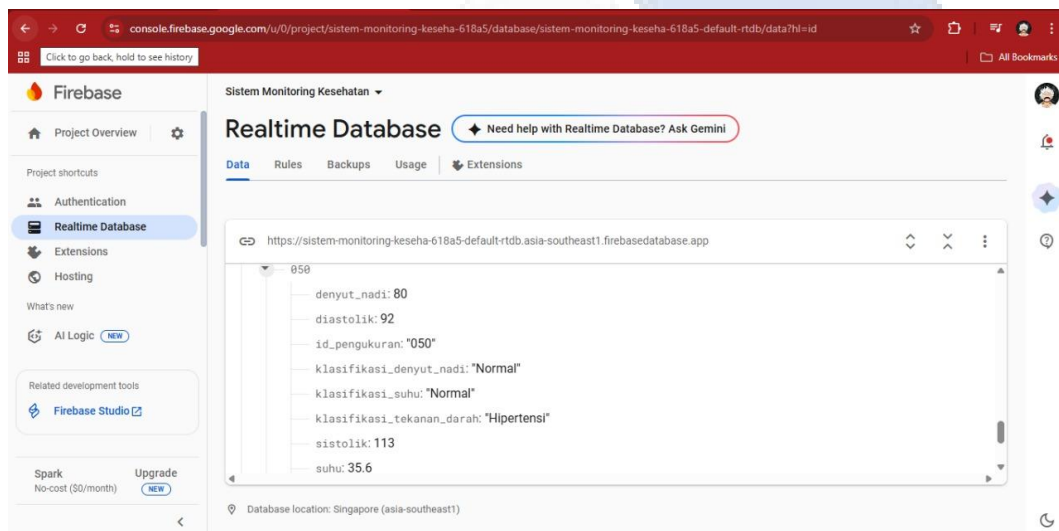
Data hasil pengukuran ditampilkan dalam bentuk gauge semi-melingkar yang menunjukkan nilai pengukuran secara visual. Informasi yang ditampilkan meliputi nilai denyut nadi dalam satuan *Beats Per Minute* (BPM), suhu dalam satuan derajat Celsius ($^{\circ}\text{C}$), serta tekanan darah yang terdiri dari tekanan sistolik dan diastolik dalam satuan mmHg. Berikut merupakan gambar tampilan data hasil pengukuran pada Blynk.



Gambar 4.18 Tampilan Blynk

4.6.3. Tampilan Firebase

Firebase digunakan sebagai *platform* penyimpanan cloud untuk semua data hasil pengukuran yang telah diperoleh secara *real-time*. Data yang dikirimkan oleh mikrokontroler ESP32 disimpan ke dalam database Firebase Realtime. Database ini menyimpan nilai pengukuran denyut nadi dalam satuan *Beats Per Minute* (BPM), suhu dalam satuan derajat Celsius ($^{\circ}\text{C}$), serta tekanan darah yang terdiri dari tekanan sistolik dan diastolik dalam satuan mmHg. Keberadaan Firebase Realtime Database memungkinkan data hasil pengukuran tersimpan dalam jangka panjang dan dapat diakses kapan saja untuk keperluan pemantauan lebih lanjut. Berikut merupakan gambar tampilan data hasil pengukuran pada *platform* Firebase.



Gambar 4.19 Tampilan Firebase

4.6.4. Tampilan Website dan WhatsApp

WhatsApp digunakan sebagai *platform* media komunikasi dalam sistem monitoring ini, khususnya untuk menerima dan mengirimkan pesan berisi data hasil pengukuran kepada pengguna secara *real-time*. Proses dimulai dengan input data pengguna, termasuk ID pengukuran, pada *website* moitoring yang terhubung dengan Firebase. Website kemudian mengambil data hasil pengukuran berdasarkan ID yang dimasukkan, dan menampilkan informasi secara lengkap, meliputi data pengguna, hasil pengukuran, serta klasifikasinya, lalu mengirimkannya secara

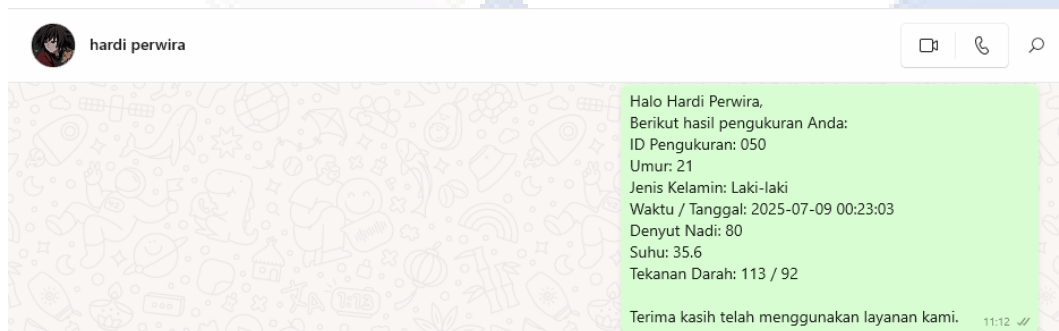
otomatis ke WhatsApp pengguna. Dengan adanya data ini, pengguna dapat mengetahui dan memantau data hasil pengukuran sebelumnya sehingga dapat memantau kondisi kesehatan secara berkelanjutan. Berikut merupakan gambar tampilan data hasil pengukuran pada *website* dan WhatsApp.

The screenshot shows a web browser window with the address bar displaying 'File: C:/Users/X280/Documents/Form%20Data%20Pasien.html'. The page title is 'Monitoring Pengguna - Sinkron F:'. The main content area is titled 'Monitoring Pengguna - Input Data' and contains a form with the following fields:

- ID Pengukuran ***: A text input field with the placeholder 'Masukkan ID pengukuran'.
- Nama Pengguna ***: A text input field with the placeholder 'Masukkan nama pengguna'.
- Umur ***: A text input field with the placeholder 'Masukkan umur (tahun)'.
- Jenis Kelamin ***: A dropdown menu with the placeholder '-- Pilih Jenis Kelamin --'.
- Nomor WhatsApp (format +62) ***: A text input field with the placeholder '+6281234567890'.

At the bottom of the form is a blue button labeled 'Kirim Data'. In the top right corner of the page, there are two blue buttons: 'Beranda' and 'Riwayat'.

Gambar 4.20 Tampilan Website



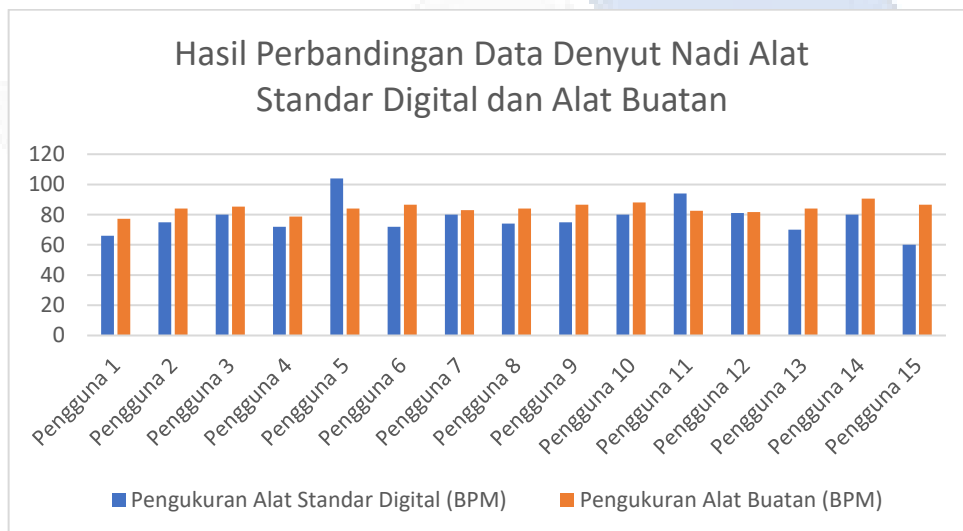
Gambar 4.21 Tampilan WhatsApp

4.7. Analisis Data

Tahap ini merupakan tahap analisa data yaitu dengan membandingkan data hasil pengukuran alat standar digital terhadap alat buatan, beserta persentase akurasi dari pengukuran tersebut. Selain itu, pada tahap ini juga membahas berbagai faktor yang dapat menyebabkan terjadinya kesalahan dalam proses pengukuran maupun pada alat ukur yang digunakan. Berikut merupakan penjelasan mengenai analisa keseluruhan dari hasil pengukuran.

4.7.1. Hasil Analisis Pengukuran Denyut Nadi

Berdasarkan data uji coba alat pengukur denyut nadi, alat buatan menunjukkan nilai kesalahan (error) jika dibandingkan dengan alat pengukur dari produk Omron. Tingkat akurasi pengukuran denyut nadi mencapai 84,68% dengan rata-rata persentase error sebesar 15,32%. Setelah dilakukan penelitian, alat pengukur denyut nadi ini terbukti memiliki kalibrasi yang baik. Oleh karena itu, dapat disimpulkan bahwa alat ini layak digunakan karena tingkat error yang sangat kecil. Berdasarkan hasil analisa data tersebut, dibuatlah grafik yang menggambarkan hasil pengukuran denyut nadi, yaitu sebagai berikut.



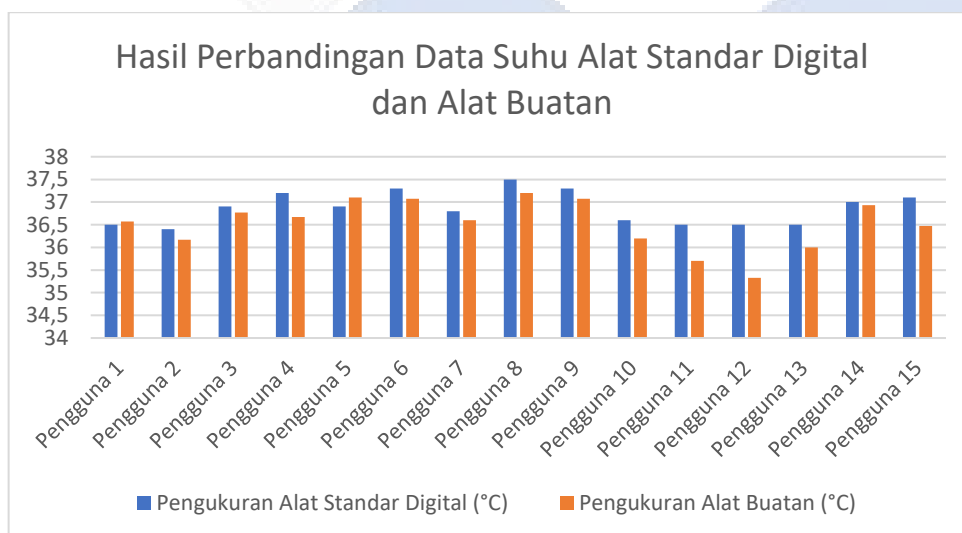
Gambar 4.22 Grafik Perbandingan Denyut Nadi

Pengujian alat pengukur denyut nadi dilakukan dengan membandingkan alat pengukur digital standar dengan alat pengukur buatan, sehingga menghasilkan data kalibrasi yang baik sehingga data yang diperoleh sesuai dengan pengukuran digital. Namun, pembuatan alat ini masih dilakukan secara manual dimulai dari perakitan hingga kalibrasi, tentu terdapat beberapa kendala yang menyebabkan hasil pembacaan sensor. Faktor-faktor yang dapat menyebabkan kesalahan dalam pengukuran antara lain kualitas sensor kurang optimal, sambungan kabel kurang baik, serta kondisi saat pengujian sensor ditempatkan pada posisi yang tidak tepat sehingga menghasilkan nilai yang tidak stabil. Selain itu, gerakan tubuh dan

penekanan terhadap sensor selama pengukuran dapat memperoleh sinyal pengukuran yang tidak sesuai dan fluktuasi pembacaan denyut nadi. Kondisi lingkungan saat melakukan pengukuran juga berpengaruh terhadap hasil yang diperoleh seperti cahaya yang berlebihan atau interferensi elektromagnetik dari perangkat lain sehingga memengaruhi sensitivitas sensor. Oleh karena itu, dalam melakukan pengukuran dilakukan sesuai dengan prosedur sehingga memperoleh hasil yang akurat.

4.7.2. Hasil Analisis Pengukuran Suhu

Berdasarkan data uji coba alat pengukur suhu, alat buatan menunjukkan nilai kesalahan (error) jika dibandingkan dengan alat pengukur digital. Tingkat akurasi pengukuran suhu mencapai 98,794% dengan rata-rata persentase error sebesar 1,21%. Setelah dilakukan penelitian, alat pengukur suhu ini terbukti memiliki kalibrasi yang baik. Oleh karena itu, dapat disimpulkan bahwa alat ini layak digunakan karena tingkat error yang sangat kecil. Berdasarkan hasil analisa data tersebut, dibuatlah grafik yang menggambarkan hasil pengukuran suhu, yaitu sebagai berikut.



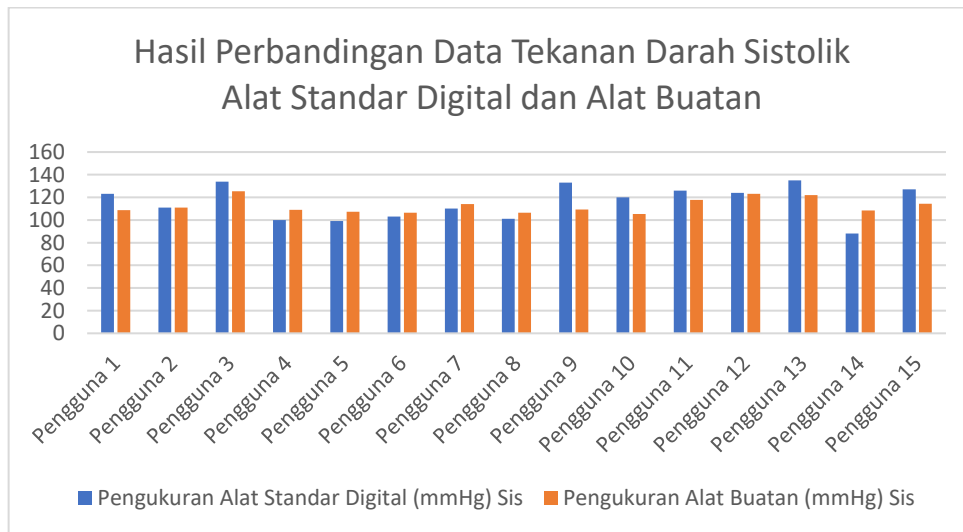
Gambar 4.23 Grafik Perbandingan Suhu

Penguian alat pengukur suhu dilakukan dengan membandingkan alat pengukur digital dengan alat pengukur buatan. Namun, pembuatan alat ini masih

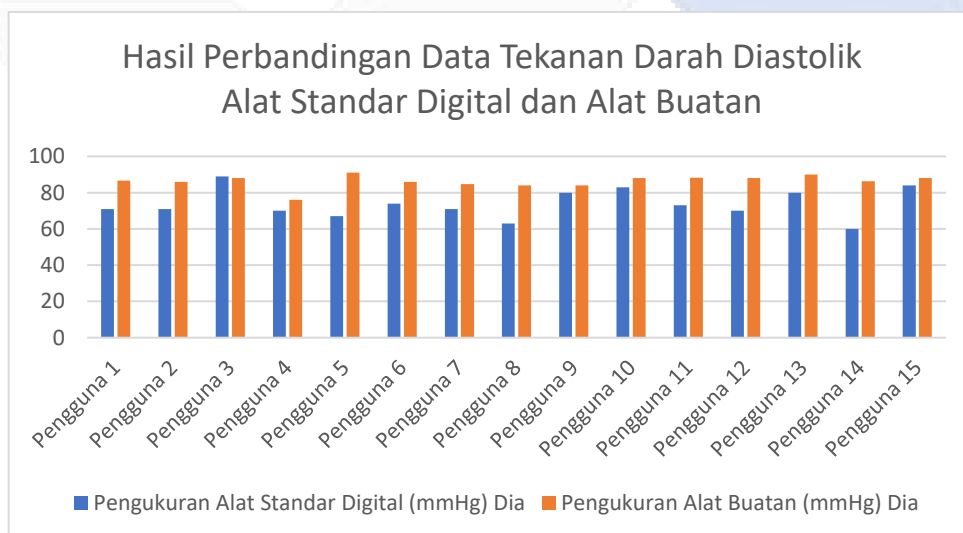
dilakukan secara manual dimulai dari perakitan, tentu terdapat beberapa kendala yang memengaruhi hasil pembacaan sensor. Faktor-faktor yang dapat menyebabkan kesalahan dalam pengukuran antara lain kualitas sensor kurang optimal, sambungan kabel kurang baik, serta kondisi saat pengujian sensor ditempatkan pada posisi yang tidak tepat sehingga menghasilkan nilai yang tidak stabil. Contohnya, pengukuran pada permukaan yang tidak rata atau terkena pengaruh lingkungan sekitar sehingga menyebabkan fluktuasi nilai suhu yang terukur. Selain itu, dalam melakukan pengukuran suhu ke tubuh pengguna, penempatan sensor pada dahi pengguna dengan jarak yang bervariasi namun kurang dari 10 cm dapat mempengaruhi akurasi pembacaan, karena jarak yang tidak konsisten dapat menyebabkan perbedaan hasil pengukuran. Oleh karena itu, pengaturan jarak pengukuran yang tepat dan bersifat konsisten sangat penting untuk memperoleh hasil pengukuran yang akurat dan stabil.

4.7.3. Hasil Analisis Pengukuran Tekanan Darah

Berdasarkan data uji coba alat pengukur tekanan darah yang terdiri dari nilai sistolik dan diastolik, alat buatan menunjukkan nilai kesalahan (error) jika dibandingkan dengan alat pengukur dari produk Omron. Tingkat akurasi pengukuran tekanan darah untuk nilai sistolik mencapai 91,4% dengan rata-rata persentase error sebesar 8,6% dan nilai diastolik mencapai 83,19% dengan rata-rata persentase error sebesar 16,81%. Setelah dilakukan penelitian, alat pengukur tekanan darah ini terbukti memiliki kalibrasi yang baik. Oleh karena itu, dapat disimpulkan bahwa alat ini layak digunakan karena tingkat error yang sangat kecil. Berdasarkan hasil analisa data tersebut, dibuatlah grafik yang menggambarkan hasil pengukuran tekanan darah untuk nilai sistolik dan diastolik, yaitu sebagai berikut.



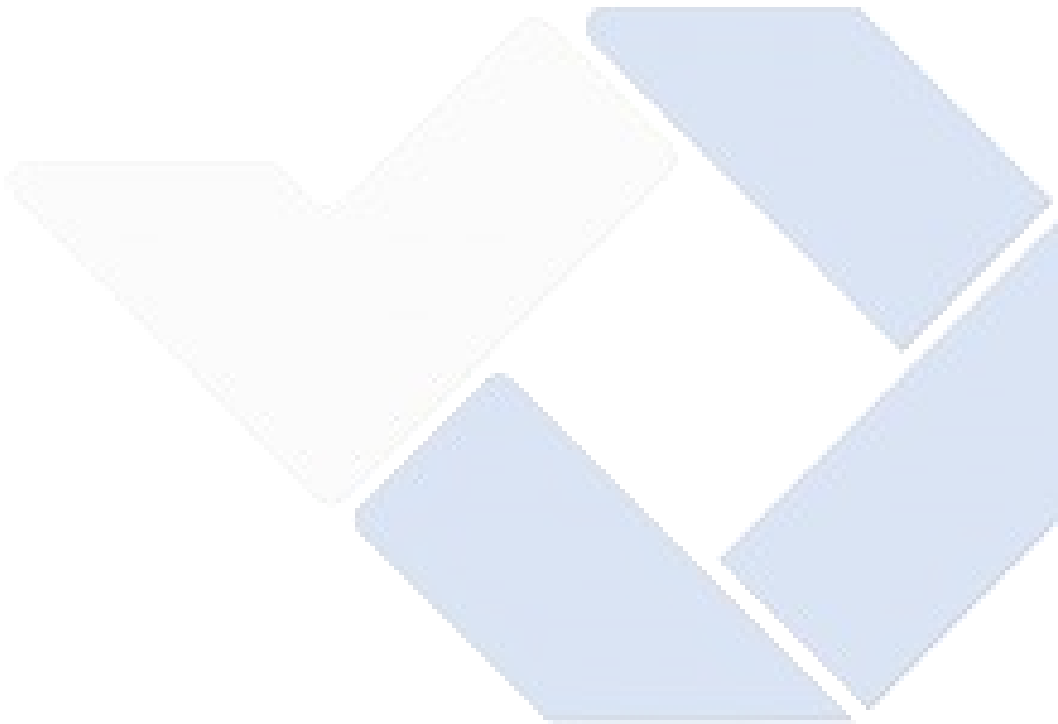
Gambar 4.24 Grafik Perbandingan Tekanan Darah Sistolik



Gambar 4.25 Grafik Perbandingan Tekanan Darah Diastolik

Penguian alat pengukur tekanan darah dilakukan dengan membandingkan alat pengukur digital dengan alat pengukur buatan. Namun, pembuatan alat ini masih dilakukan secara manual dimulai dari perakitan, tentu terdapat beberapa kendala yang memengaruhi hasil pembacaan sensor. Faktor-faktor yang dapat menyebabkan kesalahan dalam pengukuran antara lain kualitas sensor kurang optimal, sambungan kabel kurang baik, serta kondisi saat pengujian sensor ditempatkan pada posisi yang tidak tepat sehingga menghasilkan nilai yang tidak

stabil. Penempatan manset yang tidak pas di lengan, lengan tidak rileks, posisi lengan tidak sejajar dengan jantung, dapat menyebabkan fluktuasi nilai tekanan darah. Dalam melakukan pengukuran terdapat variasi penempatan manset di lengan, misalnya longgar atau ketat, perubahan posisi pengguna saat sedang melakukan pengukuran, dapat memengaruhi hasil pembacaan. Oleh karena itu, posisi lengan dan penempatan manset yang sesuai dengan tidak adanya kesalahan seperti longgar atau ketat dapat memperoleh hasil pengukuran tekanan darah yang akurat dan stabil.



BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa alat kesehatan berbasis *Internet of Things* (IoT) ini berhasil dirancang dan dibuat untuk mengukur denyut nadi, suhu, dan tekanan darah secara bersamaan dalam satu sistem terintegrasi. Alat ini mampu mengukur parameter kesehatan secara akurat sehingga memperoleh hasil pengukuran yang bisa diandalkan untuk melakukan pemantauan kondisi tubuh secara menyeluruh.

Pengujian dan pengambilan data sudah dilakukan pada alat pengukur yang dibuat dengan dibandingkan dengan alat pengukur digital, dapat disimpulkan bahwa:

1. Rata-rata tingkat akurasi pengukuran denyut nadi mencapai 85,46% jika dibandingkan dengan alat pengukur digital.
2. Rata-rata tingkat akurasi pengukuran suhu mencapai 98,97% jika dibandingkan dengan alat pengukur digital.
3. Rata-rata tingkat akurasi pengukuran tekanan darah sistolik mencapai 91,16% dan diastolik mencapai 81,58% jika dibandingkan dengan alat pengukur digital.
4. Sistem pengiriman dan penampilan data hasil pengukuran secara *real-time* ke LCD TFT 3,5 inci, server Blynk, dan Firebase dalam sistem berbasis *Internet of Things* (IoT), serta pengembangan *website* terintegrasi dengan Firebase yang dapat mengambil data berdasarkan ID pengukuran dan mengirimkannya ke WhatsApp, berhasil dirancang dan dibuat. Sistem ini memungkinkan pemantauan kesehatan dari jarak jauh secara efektif dengan respons data cepat dan akurasi pengiriman yang baik.

Alat pengukuran ini mampu menghasilkan data kalibrasi yang baik dan nilai pengukuran yang dihasilkan dari alat buatan mendekati alat standar digital, meskipun terdapat kendala yang memengaruhi akurasi dan kestabilan hasil pengukuran, seperti kualitas sensor belum optimal, sambungan kabel rusak atau kurang baik, serta pengujian kurang ideal. Faktor-faktor tersebut dapat

menyebabkan fluktuasi dan ketidakstabilan nilai pengukuran, sehingga penting dilakukannya penempatan sensor yang konsisten, pengaturan jarak atau posisi yang tepat, serta kondisi lingkungan saat melakukan pengujian. Dengan dilakukannya perbaikan pada faktor-faktor tersebut, maka alat pengukur yang dikembangkan memiliki potensi besar untuk digunakan secara efektif dengan tingkat akurasi tinggi dan nilai error yang minimal.

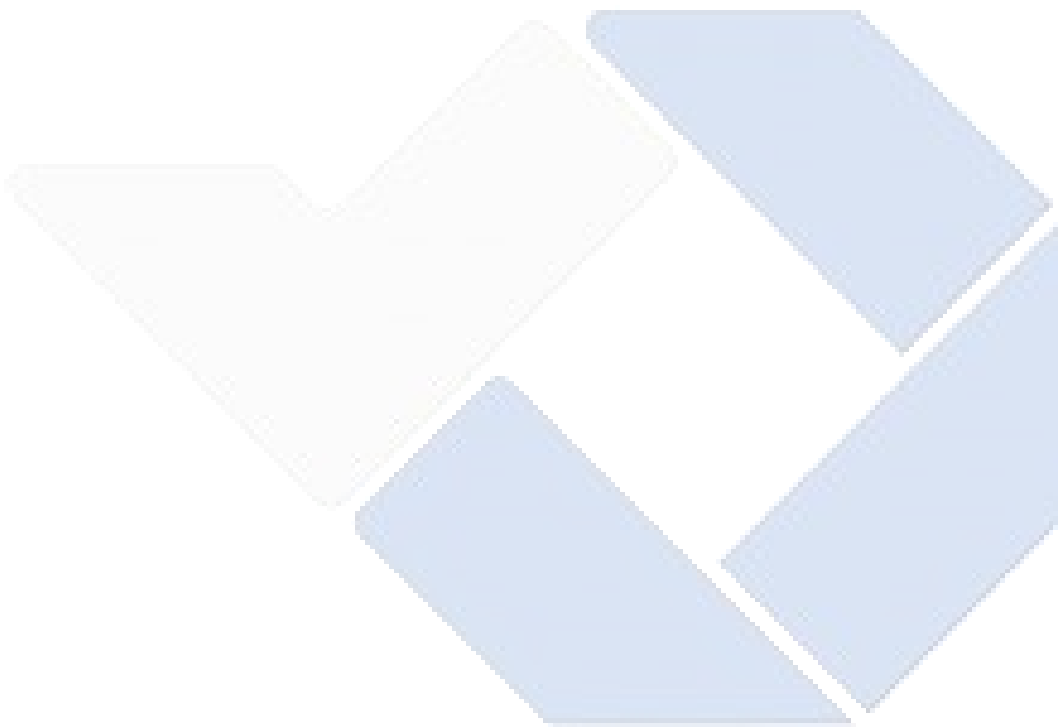
Secara keseluruhan, alat kesehatan berbasis *Internet of Things* (IoT) yang sudah dirancang dan dibuat cukup berhasil mengukur secara bersamaan, serta potensi pemanfaatannya dapat meningkatkan efektivitas pelayanan kesehatan jarak jauh secara *real-time*. Pengembangan lebih lanjut diperlukan untuk mengoptimalkan performa alat agar dapat digunakan secara lebih luas dan andal.

5.2. Saran

Proses penyusunan makalah ini masih memiliki beberapa kekurangan dan hal yang perlu diperbaiki, terutama pada tata bahasa dan cara penulisan agar menjadi lebih jelas dan sistematis. Terkait alat pengukur kesehatan yang bisa mengukur denyut nadi, suhu, dan tekanan darah yang telah dibuat, penulis memberikan beberapa saran untuk pengembangan dan perbaikan lebih lanjut, antara lain:

1. Kalibrasi sensor perlu ditingkatkan supaya memperoleh hasil pengukuran yang akurat pada setiap parameter.
2. Penambahan pengukuran parameter kesehatan lain yang relevan jika diintegrasikan secara bersamaan dengan alat sekarang ini agar memberikan informasi yang lebih lengkap.
3. Pengujian alat harus dilakukan pada berbagai kondisi untuk memastikan alat yang dibuat efektif jika digunakan di berbagai situasi.
4. Penentuan posisi sensor serta proses pengujian pada kondisi lingkungan saat pengujian harus diperhatikan agar memperoleh hasil yang stabil dan akurat.
5. Perawatan dan pemeliharaan alat secara berkala untuk menjaga performa sensor agar berfungsi secara optimal dalam jangka waktu yang panjang.

Penerapan saran-saran diatas diharapkan dapat memberikan pengarahannya terhadap kualitas alat sehingga menjadi alat pengukur kesehatan yang lebih akurat dan bermanfaat bagi para pengguna dalam memantau kesehatan tubuh secara rutin.



DAFTAR PUSTAKA

- [1] National Center for Biotechnology Information. (2023, 1 Mei). Vital Sign Assessment. Diakses pada 12 Maret 2025, dari <https://www.ncbi.nlm.nih.gov/books/NBK553213/>
- [2] Halim, A. R., Saiful, M., & Kertawijaya, L. (2022). Rancang Bangun Alat Pengukur Suhu Tubuh Pintarberbasis Internet Of Things. Infotek J. Inform. dan Teknol, 5(1), 117-127.
- [3] Alona Situmeang, S. T., & Anggietta, M. Y. K. RANCANG BANGUN ALAT PENGUKUR TEKANAN DARAH DAN DENYUT JANTUNG BERBASIS ARDUINO NANO TEKS, SUARA & CAHAYA. DENGAN BENTUK KELUARAN
- [4] Supriyanto, S., Nugraha, R. I., & Sugiharto, S. (2024). Alat Pengukur Tekanan Darah Disertai Pengukur Suhu Tubuh. ProBisnis: Jurnal Manajemen, 15(4), 635-642.
- [5] Rahmawarni, D., & Harmadi, H. (2021). Sistem Monitoring Saturasi Oksigen dan Denyut Nadi dalam Darah Menggunakan Sensor MAX30100 Via Telegram Berbasis IoT. Jurnal Fisika Unand, 10(3), 377-383..
- [6] Timor, A. R., & Erliwati, E. (2023). Alat Pengukur Denyut Nadi Dengan Tampilan OLED Berbasis Arduino. Jurnal Teknik, Komputer, Agroteknologi Dan Sains, 2(1), 92-97.
- [7] Milda, M., & Rini, A. (2018). MONITORING SUHU DAN DETAK JANTUNG BERBASIS ARDUINO (Doctoral dissertation, Politeknik Manufaktur Negeri Bangka Belitung).
- [8] Arifushalat, A. (2019). *Pengaruh Recovery Aktif dan Pasif Terhadap Denyut Nadi Pemulihan pada Atlet Sepak Bola SMA Negeri Keberbakatan Olahraga* (Doctoral dissertation, UNIVERSITAS NEGERI MAKASSAR).

- [9] Manaf, M. A., Nabilla, F., Wahyudianto, N. F., Setiawan, D. B., Nurhayati, A., & Mufidah, I. (2023). *Pemeriksaan denyut nadi*. Praktikum Biomekanika & Biotransportasi, Program Studi S1 Teknobiomedik, Fakultas Sains dan Teknologi, Universitas Airlangga. kasaan denyut nadi
- [10] Sulistyowati, A. (2018). Pemeriksaan Tanda-Tanda Vital.
- [11] Budiono, M. L. A. (2021). Rancang Bangun Pendeteksi Detak Jantung dan Pengukur Suhu Tubuh Menggunakan Pulsesensor dan Sensor DS18B20. *SinarFe7*, 4(1), 481-485.
- [12] Fikri, M. F. R., & Suyanto, S. (2013). Rancang bangun prototipe monitoring suhu tubuh manusia berbasis OS android menggunakan koneksi bluetooth. *Jurnal Teknik ITS*, 2(1), A213-A216.
- [13] Kukus, Y., Supit, W., & Lintong, F. (2009). Suhu tubuh: homeostasis dan efek terhadap kinerja tubuh manusia. *Jurnal Biomedik: JBM*, 1(2).
- [14] SAYUTI, I. GAMBARAN SUHU INTI TUBUH PREANESTESI DAN PASCAANESTESI PADA PASIEN SECTIO CASAREA DENGAN SPINAL ANESTESI DI KAMAR OPERASI RUMAH SAKIT BHAYANGKARA BANDA ACEH TAHUN 2021.
- [15] Halim, A. R., Saiful, M., & Kertawijaya, L. (2022). Rancang Bangun Alat Pengukur Suhu Tubuh Pintar berbasis Internet Of Things. *Infotek J. Inform. dan Teknol*, 5(1), 117-127.
- [16] Dumalang, E. R., Lintong, F., & Danes, V. R. (2022). Analisa Perbandingan Pengukuran Tekanan Darah antara Posisi Tidur dan Posisi Duduk pada Lansia. *Jurnal Biomedik: JBM*, 14(1), 96-101.
- [17] Fadlilah, S., Rahil, N. H., & Lanni, F. (2020). Analisis Faktor Yang Mempengaruhi Tekanan Darah Dan Saturasi Oksigen Perifer (Spo2). *Jurnal Kesehatan Kusuma Husada*, 21-30.

- [18] Sulista, A., Nehru, N., & Fuady, S. (2021). Rancang Bangun Alat Monitoring Tekanan Darah Berbasis Internet of Things (IoT). *Jurnal Engineering*, 3(1), 13-26.
- [19] Ramadhan, K., Mukti, L. H., Mahfudz, M. R. A., Rezaputra, M. D. D., & Hafizh, M. T. (2021). Modification of the automatic control system for arduino ATmega 328P-based water gallon cleaner. *Bulletin of Computer Science and Electrical Engineering*, 2(2), 67-73.
- [20] Kumar, R. H., Roopa, A. U., & Sathiya, D. P. (2015). Arduino ATMEGA-328 microcontroller. *Int. J. Innov. Res. Electr. Electron. Instrum. Control Eng*, 3(4), 27-29.
- [21] ESP32, P. K. PENDETEKSI KEHADIRAN MENGGUNAKAN ESP32 UNTUK SISTEM PENGUNCI PINTU OTOMATIS.
- [22] Pratama, E. W., & Kiswantono, A. (2022). Electrical analysis using esp-32 module in realtime. *JEECS (Journal of Electrical Engineering and Computer Sciences)*, 7(2), 1273-1284
- [23] Al Sarfini, A. A., & Irawan, D. (2024). Sistem Kontrol Jarak Jauh Plc Menggunakan Esp32 Berbasis Iot. *Jurnal Amplifier: jurnal ilmiah bidang teknik elektro dan komputer*, 14(1), 51-55.
- [24] Irawan, Y., Fernando, Y., & Wahyuni, R. (2019). Detecting Heart Rate Using Pulse Sensor As Alternative Knowing Heart Condition. *Journal of Applied Engineering and Technological Science (JAETS)*, 1(1), 30-42.
- [25] Rachmat, H. H., & Ambaransari, D. R. (2018). Sistem Perekam Detak Jantung Berbasis Pulse Heart Rate Sensor pada Jari Tangan. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 6(3), 344.
- [26] Mukhammad, Y., & Hyperastuty, A. S. (2021). Sensitivitas Sensor MLX90614 Sebagai Alat Pengukur Suhu Tubuh Tubuh Non-Contact Pada Manusia. *Indonesian Journal of Professional Nursing*, 1(2), 51-53.

- [27] Irawan, Y., Fernando, Y., & Wahyuni, R. (2019). Detecting Heart Rate Using Pulse Sensor As Alternative Knowing Heart Condition. *Journal of Applied Engineering and Technological Science (JAETS)*, 1(1), 30-42.
- [28] Abdullah, Z. S. (2022). The Design and Build a Children's Temperature Monitoring System Using The MLX90614 Temperature Sensor and NODEMCU ESP-12E Based On Android. *Journal of Engineering and Scientific Research*, 4(1), 18-22.
- [29] Karsono, K., Irawan, B., & Sulistio, A. (2022). Design Of Non-Contact Human Body Temperature Detection Based On Internet Of Things (Iot) To Open The Door Automatically. *International Journal of Science, Technology & Management*, 3(4), 1012-1017.
- [30] Afifah, N., & Saufik, I. (2021). Pengukura Suhu dengan Ir MLX90614 dan NoDeMCU dan Membandingkan dengan Ds18B20 untuk pencegahan Covid 19. *Elkom: Jurnal Elektronika dan Komputer*, 14(2), 256-260.
- [31] Yogiarditha, I. N. A., Rachman, A. S., & Kanata, B. (2023). Perancangan alat ukur tekanan darah dan jumlah denyut jantung berbasis Arduino Uno [Design of blood pressure measure and heart rate based on Arduino Uno]. Universitas Mataram.
- [32] Maharani, A., Muid, A., & Nurhasanah, N. (2020). Rancang Bangun Alat Pengukur Volume Paru-paru Berbasis Sensor Tekanan Gas MPX5700DP dan Arduino Uno. *PRISMA FISIKA*, 7(3), 231-237.
- [33] Ramdhani, A. (2019). Telemedicine tensimeter digital untuk mengetahui tekanan darah (blood pressure) berbasis Internet of Thing (Tugas Akhir, Program Studi Teknik Elektro, Universitas Semarang). ESKRIPSI.USM.AC.ID
- [34] Rostini, A. N., & Junfithrana, A. P. (2020). Aplikasi smart home node mcu iot untuk blynk. *Jurnal Rekayasa Teknologi Nusa Putra*, 7(1), 1-7.

- [35] Herlina, A., Syahbana, M. I., Gunawan, M. A., & Rizqi, M. M. (2022). Sistem kendali lampu berbasis iot menggunakan aplikasi blynk 2.0 dengan modul nodemcu esp8266. *INSANtek*, 3(2), 61-66.
- [36] Rakhmawati, P. U., Rizdania, & Sumantri. (2024). Analisis komunikasi platform Internet of Things aplikasi Blynk. *Seminar Nasional TEKNOKA*, 9, 40-46. FT - UHAMKA.<https://journal.uhamka.ac.id/index.php/teknoka/article/download/17558/4897/55801>
- [37] Saputra, D., & Arinal, V. (2021). Perancangan Home Automation dalam Mengontrol Lampu dan Kipas Menggunakan Blynk Berbasis NodeMCU. *Jurnal Sosial Teknologi*, 1(7), 597-606.
- [38] Pratama, A. F., Sulistiyanto, S., & Setyobudi, R. (2023). Sistem Monitoring Smart Klinik Berbasis Internet Of Things (IOT). *JEECOM Journal of Electrical Engineering and Computer*, 5(1), 24-30.
- [39] Ilhami, M. (2017). Pengenalan Google Firebase Untuk Hybrid Mobile Apps Berbasis Cordova. *Jurnal Ilmiah IT CIDA*, 3(1).
- [40] Sanad, E. A. W., Achmad, A., & Dewiani, D. (2018). Pemanfaatan Realtime Database di Platform Firebase Pada Aplikasi E-Tourism Kabupaten Nabire. *Jurnal Penelitian Enjiniring*, 22(1), 20-26.
- [41] Setyawan, R. A. (2024). Penerapan Firebase Realtime Database Pada Aplikasi Catatan Harian Diabetes Melitus. *Jurnal Informatika Komputer, Bisnis dan Manajemen*, 22(1), 1-9.
- [42] Bimantara, D. T. (2023). Perancangan sistem monitoring dan evaluasi pelaksanaan puslatkab kabupaten lumajang. *Indonesia Strength Conditioning and Coaching Journal*, 1(1), 1-5.
- [43] Samsinar, R., Mulyadi, R. R. F., & Prambudi, D. A. (2018). Sistem Monitoring Besaran Listrik dan Energi Penerangan Jalan Umum Secara Realtime Berbasis

Web. RESISTOR (elektRONika kEndali telekomunikaSI tenaga liSTrik kOmputeR), 1(1), 7-12.

- [44] Mugni, S., Hirawan, D., Kom, S., & Kom, M. (2018). Rancang Bangun Sistem Monitoring Kesehatan Pasien Rawat Inap Berbasis Internet Of Things.
- [45] Abdulmalek, S., Nasir, A., Jabbar, W. A., Almuahaya, M. A., Bairagi, A. K., Khan, M. A. M., & Kee, S. H. (2022, October). IoT-based healthcare-monitoring system towards improving quality of life: A review. In *Healthcare* (Vol. 10, No. 10, p. 1993). MDPI.
- [46] IRIYANTA, K. (2023). MONITORING KELEMBAPAN TANAH BERBASIS IOT DAN PREDIKSI KELEMBAPAN TANAH MENGGUNAKAN REGRESI LINEAR (STUDI KASUS TANAMAN VINCA) (Doctoral dissertation, Universitas Teknologi Digital Indonesia).
- [47] Efendi, Y. (2018). Internet of Things (IOT) sistem pengendalian lampu menggunakan Raspberry PI berbasis mobile. *Jurnal Ilmiah Ilmu Komputer Fakultas Ilmu Komputer Universitas Al Asyariah Mandar*, 4(1), 19-26.

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama Lengkap	: Hardi Perwira
Tempat & tanggal lahir	: Kelapa Kampit, 9 Juni 2004
Alamat rumah	: Dusun Lalang, Kecamatan Kelapa Kampit, Kabupaten Belitung Timur
Telp	: 085269338397
HP	: 085269338397
Email	: hardipaw17@gmail.com
Jenis kelamin	: Laki-laki
Agama	: Islam



2. Riwayat Pendidikan

SD Negeri 14 Kelapa Kampit	2010 - 2016
SMP Negeri 1 Kelapa Kampit	2016 – 2019
SMA Negeri 1 Kelapa Kampit	2019 – 2022

3. Pendidikan Non-Formal

Magang Mandiri (MBKM) di PT Parit Sembada Palm Oil Mill Kelapa Kampit 2024.

Sungailiat, 16 Juli 2025

Hardi Perwira

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama Lengkap	: Merizta Aulia
Tempat & tanggal lahir	: Sempan, 6 Mei 2004
Alamat rumah	: Jalan Pusaka Sempan, Kecamatan Pemali, Kabupaten Bangka belitung
Telp	: 083175250110
HP	: 083175250110
Email	: meriztaaulia@gmail.com
Jenis kelamin	: Perempuan
Agama	: Islam



2. Riwayat Pendidikan

SD Negeri 4 Pemali	2010 - 2016
MTs Negeri 1 Bangka	2016 – 2019
SMA Negeri 1 Pemali	2019 – 2022

3. Pendidikan Non-Formal

Magang Mandiri (MBKM) di PT PLN (Persero) ULP Sungailiat 2024.

Sungailiat, 16 Juli 2025

Merizta Aulia

CODING

- **Arduino Uno**

```
#include <Wire.h>
#include <Adafruit_MLX90614.h>

// ===== KONFIGURASI PIN =====
const int PulseSensorPin = A0;
const int buttonPin = 2;
const int pressurePin = A1;
const int solenoidPin = 8;
const int motorIn1 = 6;
const int motorIn2 = 5;
const int enablePin = 9;

// ===== OBJEK & VARIABEL =====
Adafruit_MLX90614 mlx = Adafruit_MLX90614();

int thresholdMin = 300;
int thresholdMax = 530;

unsigned long previousMillis = 0;
unsigned long previousBPMMillis = 0;
unsigned long pulseCount = 0;
float BPM = 0;
unsigned long lastPulseTime = 0;

float bodyTemperature = 0.0;
float pressureBuffer[200];
```

```

int index = 0;

bool startMeasurement = false;
bool isFirstDetection = true;
bool hasMeasuredBPM = false;
bool isTemperatureValid = false;
bool measurementDone = false;

// ===== KONVERSI TEKANAN =====
const int ADC1 = 39;
const int Pressure1 = 0;
const int ADC2 = 73;
const int Pressure2 = 200;
float m = float(Pressure2 - Pressure1) / float(ADC2 - ADC1);
float b = Pressure1 - m * ADC1;
#define DATA_SIZE 60
struct DataPoint {
    float pressure;
    float oscillation;
};
DataPoint data[DATA_SIZE];
// ===== SETUP =====
void setup() {
    Serial.begin(9600);
    pinMode(buttonPin, INPUT_PULLUP);
    mlx.begin();

    pinMode(solenoidPin, OUTPUT);
    pinMode(motorIn1, OUTPUT);
    pinMode(motorIn2, OUTPUT);
    pinMode(enablePin, OUTPUT);

```

```

digitalWrite(solenoidPin, LOW); // default mati

Serial.println("=====");
Serial.println("SISTEM MONITOR KESEHATAN");
Serial.println("Tekan tombol untuk memulai pengukuran...");
Serial.println("=====");
}

// ===== LOOP UTAMA =====
void loop() {
    unsigned long currentMillis = millis();
    int sensorValue = analogRead(PulseSensorPin);

    // Tombol ditekan → mulai proses
    if (digitalRead(buttonPin) == LOW && !startMeasurement &&
!measurementDone) {
        startMeasurement = true;
        Serial.println("Tombol ditekan. Pengukuran dimulai...");
        Serial.println("Memulai pengukuran"); // =====> Untuk ESP32 mendeteksi &
tampilkan "Mengukur Denyut Nadi..."
        delay(500);
    }

    // 1. PENGUKURAN DENYUT NADI
    if (startMeasurement && !hasMeasuredBPM) {
        if (sensorValue < 50 && isFirstDetection) {
            isFirstDetection = false;
            pulseCount = 0;
            previousBPMMillis = currentMillis;
        }
    }
}

```

```

if (sensorValue >= thresholdMin && sensorValue <= thresholdMax) {
    if (currentMillis - lastPulseTime > 500) {
        pulseCount++;
        lastPulseTime = currentMillis;
    }
}

if (currentMillis - previousMillis >= 1000) {
    previousMillis = currentMillis;

    Serial.print(currentMillis / 1000);
    Serial.print(" s | Sensor: ");
    Serial.print(sensorValue);
    Serial.print(" | Pulse: ");
    Serial.println((sensorValue >= thresholdMin && sensorValue <=
thresholdMax) ? "Ya" : "Tidak");

    if (!isFirstDetection && currentMillis - previousBPMMillis >= 15000) {
        BPM = pulseCount * 4;
        hasMeasuredBPM = true;

        if (BPM >= 70 && BPM <= 90) {
            Serial.print("BPM terukur: "); Serial.println(BPM);
        } else {
            Serial.println("BPM tidak valid.");
        }
    }
}
}
}

```

```

// 2. PENGUKURAN SUHU
if (hasMeasuredBPM && !isTemperatureValid) {
    Serial.println("Mengukur suhu tubuh...");

    unsigned long startTempMillis = millis();
    float tempSum = 0.0;
    int validReadCount = 0;

    // Tunggu hingga suhu valid ditemukan
    while (true) {
        float temp = mlx.readObjectTempC() + 1.5; // offset kalibrasi
        if (temp >= 35.0 && temp <= 40.0) {
            bodyTemperature = temp; // Simpan suhu valid pertama
            Serial.println("Suhu valid ditemukan, menunggu 3 detik untuk konfirmasi...");

            // Ambil data suhu selama 3 detik (rata-rata dari beberapa pembacaan)
            startTempMillis = millis(); // Mulai pengambilan 3 detik
            tempSum = 0.0;
            validReadCount = 0;

            while (millis() - startTempMillis < 3000) {
                float tempRead = mlx.readObjectTempC() + 1.3;
                if (tempRead >= 35.0 && tempRead <= 40.0) {
                    tempSum += tempRead;
                    validReadCount++;
                }
                delay(1000); // Sampling setiap 100ms (bisa disesuaikan)
            }

            if (validReadCount > 0) {
                bodyTemperature = tempSum / validReadCount; // Ambil rata-rata suhu
            }
        }
    }
}

```

```

    }

    isTemperatureValid = true;
    Serial.println("=====");
    Serial.print("BPM: "); Serial.println(BPM);
    Serial.print("Suhu Akhir (rata-rata): "); Serial.println(bodyTemperature);
    Serial.println("=====");

    Serial.println("Menunggu 7 detik sebelum tekanan darah...");
    delay(7000);
    measureBloodPressure();
    break;
} else {
    Serial.println("Suhu tidak valid. Ulangi deteksi...");
    delay(500); // Delay kecil agar tidak terlalu cepat polling
}
}
}

// 3. RESET UNTUK PENGUKURAN ULANG
if (measurementDone) {
    Serial.println("\nTekan tombol lagi untuk mengulang pengukuran...\n");
    while (digitalRead(buttonPin) == HIGH);
    delay(500);
    resetMeasurement();
}
}

// ===== FUNGSI RESET =====
void resetMeasurement() {
    startMeasurement = false;

```

```

isFirstDetection = true;
hasMeasuredBPM = false;
isTemperatureValid = false;
measurementDone = false;

previousMillis = 0;
previousBPMMillis = 0;
pulseCount = 0;
BPM = 0;
bodyTemperature = 0.0;
index = 0;

digitalWrite(solenoidPin, LOW);
Serial.println("Sistem di-reset. Siap pengukuran ulang.");
}

// ===== KONVERSI =====
float kalibrasiSistolik(float x) { return x - 1.0; }
float kalibrasiDiastolik(float x) { return x - 1.0; }
float adcToPressure(int adc) { return m * adc + b; }

float movingAverage(float newValue) {
    static float buffer[10];
    static int i = 0;
    static float sum = 0.0;
    sum -= buffer[i];
    buffer[i] = newValue;
    sum += buffer[i];
    i = (i + 1) % 10;
    return sum / 10;
}

```

```

// ===== INFLASI =====
void inflateCuff() {
    int pwmSpeed = 70;
    float maxPressure = 170;
    while (true) {
        int adc = analogRead(pressurePin);
        float pressure = movingAverage(adcToPressure(adc));
        Serial.print("Inflasi Tekanan: "); Serial.println(pressure);
        if (pressure >= maxPressure) break;
        analogWrite(enablePin, pwmSpeed);
        digitalWrite(motorIn1, HIGH);
        digitalWrite(motorIn2, LOW);
        delay(100);
    }
    analogWrite(enablePin, 0);
    digitalWrite(motorIn1, LOW);
    digitalWrite(motorIn2, LOW);
}

// ===== PENGUKURAN =====
void measureBloodPressure() {
    digitalWrite(solenoidPin, HIGH); // TERTUTUP untuk menahan udara sampai
    tekanan < 70
    inflateCuff();
    delay(500); // Jeda sebelum mulai pembacaan osilasi
    measureOscillations();
    measurementDone = true;
}

// ===== ANALISIS OSILASI =====

```

```

void measureOscillations() {
    Serial.println(F("Mengukur osilasi tekanan..."));

    const int maxLoop = 300;
    const int filterSize = 5;
    float oscBuffer[filterSize] = {0};
    int oscIndex = 0;
    float oscSum = 0;

    int dataIndex = 0;
    float maxOsc = 0;
    float systolicPressure = -1;
    float diastolicPressure = -1;
    float maxOsc80_95 = -1;

    bool solenoidReleased = false;
    bool foundSystolic = false;

    for (int i = 0; i < maxLoop; i++) {
        int adc = analogRead(pressurePin);
        float pressure = adcToPressure(adc);
        float smoothed = movingAverage(pressure);
        float rawOsc = fabs(pressure - smoothed);

        // Filter osilasi (moving average)
        oscSum -= oscBuffer[oscIndex];
        oscBuffer[oscIndex] = rawOsc;
        oscSum += oscBuffer[oscIndex];
        oscIndex = (oscIndex + 1) % filterSize;
        float filteredOsc = oscSum / filterSize;
    }
}

```

```

if (dataIndex < DATA_SIZE) {
    data[dataIndex].pressure = smoothed;
    data[dataIndex].oscillation = filteredOsc;
}
dataIndex++;

if (i % 5 == 0) {
    Serial.print("Tekanan: "); Serial.print(smoothed);
    Serial.print(" | Osilasi: "); Serial.println(filteredOsc);
}

if (filteredOsc <= 4.7) {

    // Cari sistolik: osilasi tertinggi di antara tekanan 90–135
    if (smoothed >= 90.0 && smoothed <= 136.0) {
        if (filteredOsc > maxOsc) {
            maxOsc = filteredOsc;
            systolicPressure = smoothed;
        }
    }

    // Diastolik: osilasi terendah di rentang 75–90 mmHg
    if (smoothed >= 78.0 && smoothed <= 90.0) {
        if (diastolicPressure < 0 || filteredOsc < maxOsc80_95) {
            maxOsc80_95 = filteredOsc;
            diastolicPressure = smoothed;
        }
    }
}

// Buka solenoid saat tekanan < 70
if (!solenoidReleased && smoothed < 70.0) {

```

```

    digitalWrite(solenoidPin, LOW);
    solenoidReleased = true;
    Serial.println(F(">> Solenoid dibuka karena tekanan < 70 mmHg"));
}

if (solenoidReleased && smoothed < 40.0) {
    Serial.println(F(">> Tekanan sangat rendah, pengukuran selesai."));
    break;
}

delay(120);
}

Serial.println("===== HASIL AKHIR =====");
Serial.print("BPM      : "); Serial.println(BPM);
Serial.print("Suhu Tubuh : "); Serial.print(bodyTemperature, 1); Serial.println("
°C");

float Sistolik = kalibrasiSistolik(systolicPressure) - 3.0;
float Diastolik = kalibrasiDiastolik(diastolicPressure) + 5.0;
float MAP = (Diastolik + 2 * Sistolik) / 3.0;

Serial.print("Sistolik  : "); Serial.println(Sistolik, 2);
Serial.print("Diastolik  : "); Serial.println(Diastolik, 2);
Serial.print("MAP      : "); Serial.println(MAP, 2);
Serial.println("=====");

// Kirim ke ESP32
Serial.print(Sistolik, 0); Serial.print("|");
Serial.print(Diastolik, 0); Serial.print("|");
Serial.print(bodyTemperature, 1); Serial.print("|");

```

```

    Serial.println(BPM, 0);
}

```

- **ESP32**

```

// ===== ESP32 FULL CODE (Terintegrasi & Sinkron) =====

#define BLYNK_TEMPLATE_ID "TMPL6TVt-tERP"
#define BLYNK_TEMPLATE_NAME "PROYEK AKHIR1"
#define BLYNK_AUTH_TOKEN "L9YqndHT9tokK3ulFIqSolrcDSVowlif"

#include <TFT_eSPI.h>
#include <SPI.h>
#include <WiFi.h>
#include <BlynkSimpleEsp32.h>
#include <Firebase_ESP_Client.h>
#include <time.h>

#define API_KEY "AIzaSyAAxz1nZ8nqQKWUysmrP5byoZd0ydvwXhI"
#define DATABASE_URL "https://sistem-monitoring-keseha-618a5-default-rtdb.asia-southeast1.firebaseio.com/"

char ssid[] = "Redmi Note 11";
char pass[] = "123456789";

#define RXD2 16
#define TXD2 -1

TFT_eSPI tft = TFT_eSPI();
String incomingData = "";

FirebaseData fbdo;

```

```

FirebaseAuth auth;
FirebaseConfig config;

const char* ntpServer = "pool.ntp.org";
const long gmtOffset_sec = 7 * 3600;
const int daylightOffset_sec = 0;

int measurementCounter = 0;
bool counterInitialized = false;

bool wifiConnected = false;
bool firebaseConnected = false;
bool blynkConnected = false;
bool pengukuranDimulai = false;
unsigned long waktuTerakhirTerima = 0;

bool statusMenampilkanInstruksi = false;
bool statusTampilNadi = false;
bool statusTampilSuhu = false;
bool statusTampilTekanan = false;
bool statusSelesai = false;

// ===== Fungsi Tambahan =====

String formatID(int counter) {
    char buffer[4];
    snprintf(buffer, sizeof(buffer), "%03d", counter);
    return String(buffer);
}

String klasifikasiDenyutNadi(int heartRate) {

```

```

    if (heartRate < 60) return "Bradikardia";
    else if (heartRate > 100) return "Takikardia";
    else return "Normal";
}

```

```

String klasifikasiSuhu(float suhu) {
    if (suhu < 35.0) return "Hipotermia";
    else if (suhu > 37.5) return "Hipertermia";
    else return "Normal";
}

```

```

String klasifikasiTekananDarah(int systolic, int diastolic) {
    if (systolic <= 120 && diastolic <= 80) return "Normal";
    else if (systolic <= 139 && diastolic <= 89) return "Prahipertensi";
    else return "Hipertensi";
}

```

```

void tampilkanSplashscreen() {
    tft.fillScreen(TFT_CYAN);
    tft.setTextColor(TFT_BLACK, TFT_CYAN);
    tft.setTextSize(4);
    String judul = "PROYEK AKHIR";
    int x_judul = (tft.width() - judul.length() * 24) / 2;
    tft.setCursor(x_judul, 30);
    tft.println(judul);
}

```

```

tft.setTextSize(2);
String nama1 = "Hardi Perwira (1052241)";
String nama2 = "Merizta Aulia (1052246)";
String univ1 = "Politeknik Negeri";
String univ2 = "Manufaktur Babel";

```

```

int x_nama1 = (tft.width() - nama1.length() * 11) / 2;
int x_nama2 = (tft.width() - nama2.length() * 11) / 2;
int x_univ1 = (tft.width() - univ1.length() * 11) / 2;
int x_univ2 = (tft.width() - univ2.length() * 11) / 2;

tft.setCursor(x_nama1, 90); tft.println(nama1);
tft.setCursor(x_nama2, 115); tft.println(nama2);
tft.setCursor(x_univ1, 145); tft.println(univ1);
tft.setCursor(x_univ2, 170); tft.println(univ2);

delay(4000);
}

void tampilkanTFT(String pesan, int delayDetik) {
    tft.fillScreen(TFT_CYAN);
    tft.setTextColor(TFT_BLACK, TFT_CYAN);
    tft.setTextSize(2);
    int x = (240 - (pesan.length() * 11)) / 2;
    tft.setCursor(x, 60);
    tft.println(pesan);
    delay(delayDetik * 1000);
}

void getCounterFromFirebase() {
    if (Firebase.RTDB.getInt(&fbdo, "/last_counter")) {
        if (fbdo.dataType() == "int") measurementCounter = fbdo.intData();
        else measurementCounter = 1;
    } else {
        measurementCounter = 1;
    }
}

```

```

    counterInitialized = true;
}

void tokenStatusCallback(TokenInfo info) {
    if (info.status == token_status_ready) firebaseConnected = true;
}

void tampilkanInstruksiPengukuran() {
    tft.fillScreen(TFT_CYAN);
    tft.setTextColor(TFT_BLACK, TFT_CYAN);
    tft.setTextSize(2);
    tft.setCursor(50, 40); tft.println("WiFi: " + String(wifiConnected ? "Terhubung" :
"Gagal"));
    tft.setCursor(50, 70); tft.println("Blynk: " + String(blynkConnected ?
"Terhubung" : "Gagal"));
    tft.setCursor(50, 100); tft.println("Firebase: " + String(firebaseConnected ?
"Terhubung" : "Gagal"));
    delay(3000);

    tft.fillScreen(TFT_CYAN);
    tft.setCursor(30, 60); tft.println("Tekan tombol untuk");
    tft.setCursor(65, 90); tft.println("memulai");
}

// ===== SETUP =====

void setup() {
    Serial.begin(9600);
    Serial2.begin(9600, SERIAL_8N1, RXD2, TXD2);
    tft.init(); tft.setRotation(1);
    tampilkanSplashscreen();
}

```

```

WiFi.begin(ssid, pass);
while (WiFi.status() != WL_CONNECTED) delay(500);
wifiConnected = true;

configTime(gmtOffset_sec, daylightOffset_sec, ntpServer);
Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass); blynkConnected = true;

config.api_key = API_KEY;
config.database_url = DATABASE_URL;
config.token_status_callback = tokenStatusCallback;
if (Firebase.signUp(&config, &auth, "", "")) firebaseConnected = true;
Firebase.begin(&config, &auth);

getCounterFromFirebase();
tampilkanInstruksiPengukuran();
}

// ===== LOOP =====

void loop() {
  Blynk.run();
  if (!counterInitialized) return;

  while (Serial2.available()) {
    char c = Serial2.read();
    if (c == '\n') {
      incomingData.trim();

      if (incomingData.length() > 0) {
        if (incomingData.indexOf("Memulai pengukuran") != -1) {

```

```

    pengukuranDimulai = true;
    statusTampilNadi = false;
    statusTampilSuhu = false;
    statusTampilTekanan = false;
    statusSelesai = false;
    tampilkanTFT("Mengukur Denyut Nadi...", 3);
    statusTampilNadi = true;
} else if (incomingData.indexOf("Mengukur suhu") != -1 &&
!statusTampilSuhu) {
    tampilkanTFT("Mengukur Suhu...", 3);
    statusTampilSuhu = true;
} else if (incomingData.indexOf("tekanan darah") != -1 &&
!statusTampilTekanan) {
    tampilkanTFT("Mengukur Tekanan Darah...", 7);
    statusTampilTekanan = true;
} else if (incomingData.indexOf("|") != -1) {
    prosesData(incomingData);
}

waktuTerakhirTerima = millis();
}
incomingData = "";
} else {
    incomingData += c;
}
}

```

```

if (pengukuranDimulai && millis() - waktuTerakhirTerima > 10000) {
    pengukuranDimulai = false;
    statusTampilNadi = false;
    statusTampilSuhu = false;
}

```

```

statusTampilTekanan = false;

if (!statusSelesai) {
    tampilkanTFT("Pengukuran selesai", 2);
    tampilkanTFT("Tekan tombol untuk ulangi", 3);
    statusSelesai = true;

    // Kosongkan tampilan Blynk (opsional)
    Blynk.virtualWrite(V3, 0);
    Blynk.virtualWrite(V2, 0);
    Blynk.virtualWrite(V0, 0);
    Blynk.virtualWrite(V5, 0);
}
}
}

// ===== PROSES DATA =====

void prosesData(String data) {
    int p1 = data.indexOf("|");
    int p2 = data.indexOf("|", p1 + 1);
    int p3 = data.indexOf("|", p2 + 1);

    if (p1 == -1 || p2 == -1 || p3 == -1) return;

    int systolic = data.substring(0, p1).toInt();
    int diastolic = data.substring(p1 + 1, p2).toInt();
    float suhu = data.substring(p2 + 1, p3).toFloat();
    int heartRate = data.substring(p3 + 1).toInt();

    String klasifikasiDenyut = klasifikasiDenyutNadi(heartRate);

```

```

String klasifikasiTemp = klasifikasiSuhu(suhu);
String klasifikasiTD = klasifikasiTekananDarah(systolic, diastolic);

// Waktu pengukuran
time_t now;
struct tm timeinfo;
time(&now);
localtime_r(&now, &timeinfo);
char timeString[30];
strftime(timeString, sizeof(timeString), "%Y-%m-%d %H:%M:%S", &timeinfo);

// Kirim ke Blynk
Blynk.virtualWrite(V3, systolic);
Blynk.virtualWrite(V2, diastolic);
Blynk.virtualWrite(V0, suhu);
Blynk.virtualWrite(V5, heartRate);

// Tampilkan di TFT
tft.fillScreen(TFT_CYAN);
tft.setTextColor(TFT_BLACK, TFT_CYAN);
tft.setTextSize(2);
int y = 10;
tft.setCursor(10, y); tft.printf("Denyut Nadi : %d", heartRate);
tft.setCursor(10, y += 25); tft.printf("Klasifikasi: %s", klasifikasiDenyut.c_str());
tft.setCursor(10, y += 25); tft.printf("Suhu      : %.2fC", suhu);
tft.setCursor(10, y += 25); tft.printf("Klasifikasi: %s", klasifikasiTemp.c_str());
tft.setCursor(10, y += 25); tft.printf("Sistolik   : %d", systolic);
tft.setCursor(10, y += 25); tft.printf("Diastolik  : %d", diastolic);
tft.setCursor(10, y += 25); tft.printf("Tekanan Darah: %s", klasifikasiTD.c_str());

// Simpan ke Firebase

```

```

String id = formatID(measurementCounter);
String path = "pengukuran/" + id + "/";
if (Firebase.RTDB.setString(&fbdo, path + "id_pengukuran", id) &&
    Firebase.RTDB.setString(&fbdo, path + "waktu", String(timeString)) &&
    Firebase.RTDB.setInt(&fbdo, path + "denyut_nadi", heartRate) &&
    Firebase.RTDB.setString(&fbdo, path + "klasifikasi_denyut_nadi",
klasifikasiDenyut) &&
    Firebase.RTDB.setFloat(&fbdo, path + "suhu", suhu) &&
    Firebase.RTDB.setString(&fbdo, path + "klasifikasi_suhu", klasifikasiTemp)
&&
    Firebase.RTDB.setInt(&fbdo, path + "sistolik", systolic) &&
    Firebase.RTDB.setInt(&fbdo, path + "diastolik", diastolic) &&
    Firebase.RTDB.setString(&fbdo, path + "klasifikasi_tekanan_darah",
klasifikasiTD) &&
    Firebase.RTDB.setInt(&fbdo, "/last_counter", measurementCounter + 1)) {

    measurementCounter++;
    Serial.println("Data tersimpan ke Firebase ID: " + id);
} else {
    Serial.println("Gagal simpan data: " + fbdo.errorReason());
}
}

```

- **Codingan Website**

```

<!DOCTYPE html>
<html lang="id">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <title>Monitoring Pengguna - Sinkron Firebase & WhatsApp</title>
  <style>

```

```
body {
  font-family: Arial, sans-serif;
  max-width: 900px;
  margin: 20px auto;
  background: #f9f9f9;
  color: #333;
}

.nav {
  position: absolute;
  top: 20px;
  right: 30px;
  z-index: 10;
}

.nav button {
  background: #2c7be5;
  color: #fff;
  border: none;
  border-radius: 5px 5px 0 0;
  padding: 10px 18px;
  margin-left: 5px;
  font-size: 1em;
  cursor: pointer;
  transition: background 0.2s;
}

.nav button.active, .nav button:hover {
  background: #1a5dc9;
}

h2 {
  text-align: center;
  margin-bottom: 1em;
  margin-top: 40px;
```

```
}  
form {  
  background: white;  
  padding: 20px;  
  border-radius: 8px;  
  box-shadow: 0 0 10px rgba(0,0,0,0.1);  
  margin-bottom: 30px;  
}  
label {  
  display: block;  
  margin-bottom: 8px;  
  font-weight: 600;  
}  
input, select {  
  padding: 8px;  
  width: 100%;  
  box-sizing: border-box;  
  margin-bottom: 15px;  
  border-radius: 4px;  
  border: 1px solid #ccc;  
  font-size: 1em;  
}  
.err-msg {  
  color: #c00;  
  font-size: 0.85em;  
  margin-top: -12px;  
  margin-bottom: 10px;  
  display: none;  
}  
button[type="submit"], #batalEdit {  
  padding: 10px 18px;
```

```
font-size: 1em;
border: none;
border-radius: 5px;
cursor: pointer;
margin-right: 10px;
background-color: #2c7be5;
color: white;
transition: background-color 0.3s ease;
}
button[type="submit"]:hover {
  background-color: #1a5dc9;
}
#batalEdit {
  background-color: #888;
}
#batalEdit:hover {
  background-color: #555;
}
.message {
  padding: 12px;
  margin-bottom: 20px;
  border-radius: 6px;
  font-weight: 600;
  text-align: center;
  display: none;
}
.success {
  background-color: #d4edda;
  color: #155724;
  border: 1px solid #c3e6cb;
}
```

```
.error {
  background-color: #f8d7da;
  color: #721c24;
  border: 1px solid #f5c6cb;
}

#search-container {
  margin-bottom: 15px;
  text-align: right;
}

#search-input {
  padding: 6px 10px;
  font-size: 1em;
  width: 250px;
  border-radius: 4px;
  border: 1px solid #ccc;
}

table {
  width: 100%;
  border-collapse: collapse;
  background: white;
  border-radius: 8px;
  overflow: hidden;
  box-shadow: 0 0 8px rgba(0,0,0,0.1);
}

th, td {
  padding: 12px 15px;
  border-bottom: 1px solid #eee;
  text-align: center;
  font-size: 0.95em;
}

th {
```

```
background-color: #2c7be5;
color: white;
font-weight: 600;
}
tr:hover {
background-color: #f1f7ff;
}
.btn-edit, .btn-delete {
padding: 6px 12px;
border: none;
border-radius: 4px;
cursor: pointer;
font-size: 0.9em;
margin: 0 3px;
}
.btn-edit {
background-color: #28a745;
color: white;
}
.btn-edit:hover {
background-color: #218838;
}
.btn-delete {
background-color: #dc3545;
color: white;
}
.btn-delete:hover {
background-color: #c82333;
}
#loading {
display: none;
```

```

margin: 10px 0;
font-style: italic;
color: #555;
text-align: center;
}
/* Tab content */
.tab-content { display: none; }
.tab-content.active { display: block; }
@media (max-width: 600px) {
  .nav { right: 10px; }
  th, td { font-size: 0.85em; }
  #search-input { width: 100%; max-width: 300px; }
}
</style>

<script type="module">
  import { initializeApp } from
  "https://www.gstatic.com/firebasejs/9.23.0/firebase-app.js";
  import {
    getDatabase,
    ref,
    get,
    child,
    set,
    update,
    remove,
    onValue
  } from "https://www.gstatic.com/firebasejs/9.23.0/firebase-database.js";

  // Firebase config
  const firebaseConfig = {

```

```

    apiKey: "AIzaSyAAxz1nZ8nqQKWUysmrP5byoZd0ydvwXhI",
    authDomain: "pro-sistem-monitoring-keseha.firebaseio.com",
    databaseURL: "https://sistem-monitoring-keseha-618a5-default-rtdb.asia-southeast1.firebaseio.com",
    projectId: "pro-sistem-monitoring-keseha",
    storageBucket: "pro-sistem-monitoring-keseha.appspot.com",
    messagingSenderId: "1081489203399",
    appId: "1:1081489203399:web:2cd3fb15eaf44468971ee2"
  };

```

```

const app = initializeApp(firebaseConfig);
const db = getDatabase(app);

// Tab navigation
const tabButtons = document.querySelectorAll('.nav button');
const tabContents = document.querySelectorAll('.tab-content');
function showTab(tabId) {
  tabContents.forEach(tab => tab.classList.remove('active'));
  tabButtons.forEach(btn => btn.classList.remove('active'));
  document.getElementById(tabId).classList.add('active');
  document.querySelector('.nav button[data-tab="'+tabId+'"]').classList.add('active');
}
window.showTab = showTab;

```

```

// DOM Elements

```

```

let form, formMessage, batalEditBtn, historyBody, loadingEl, searchInput;
let editKey = null;
let riwayatData = {}; // simpan data riwayat asli untuk pencarian

```

```

window.addEventListener('DOMContentLoaded', () => {

```

```
form = document.getElementById("data-pasien");
formMessage = document.getElementById("form-message");
batalEditBtn = document.getElementById("batalEdit");
historyBody = document.getElementById("historyBody");
loadingEl = document.getElementById("loading");
searchInput = document.getElementById("search-input");
```

```
showTab('beranda');
```

```
// Validasi Form
```

```
function validateForm() {
  let valid = true;
  if (!form.id_pengukuran.value.trim()) {
    showError("err-id_pengukuran"); valid = false;
  } else { hideError("err-id_pengukuran"); }
  if (!form.nama.value.trim()) {
    showError("err-nama"); valid = false;
  } else { hideError("err-nama"); }
  const umurVal = Number(form.umur.value);
  if (isNaN(umurVal) || umurVal < 0 || umurVal > 120) {
    showError("err-umur"); valid = false;
  } else { hideError("err-umur"); }
  if (!form.jenis_kelamin.value) {
    showError("err-jenis"); valid = false;
  } else { hideError("err-jenis"); }
  const nomorWA = form.nomor_wa.value.trim();
  const patternWA = /^\\+62\\d{8,15}$\\;/;
  if (!patternWA.test(nomorWA)) {
    showError("err-wa"); valid = false;
  } else { hideError("err-wa"); }
  return valid;
}
```

```

    }
    function showError(id) { document.getElementById(id).style.display =
"block"; }
    function hideError(id) { document.getElementById(id).style.display = "none";
}

```

// Ambil data pengukuran dari Firebase berdasarkan ID Pengukuran

```

async function getPengukuran(idPengukuran) {
    try {
        const snapshot = await get(child(ref(db), `pengukuran/${idPengukuran}`));
        if (snapshot.exists()) {
            return snapshot.val();
        } else {
            return null;
        }
    } catch (error) {
        console.error("Error getting pengukuran:", error);
        return null;
    }
}

```

// Reset form dan UI setelah submit / batal edit

```

function resetFormUI() {
    form.reset();
    batalEditBtn.style.display = "none";
    editKey = null;
    formMessage.style.display = "none";
    loadingEl.style.display = "none";
}

```

// Render tabel riwayat dengan data yang sudah diurutkan dan filter pencarian

```

function renderTable(dataObj, filter = "") {
  historyBody.innerHTML = "";
  if (!dataObj) {
    historyBody.innerHTML = `<tr><td colspan="10" style="text-align:center;">Belum ada data riwayat</td></tr>`;
    return;
  }
  // Ubah objek ke array dan urutkan berdasarkan id_pengukuran ascending
  (numerik jika memungkinkan)
  const dataArr = Object.entries(dataObj)
    .map(([key, val]) => ({ key, ...val }));
  // PATCH: filter seluruh kolom, bukan hanya nama
  .filter(item => {
    const searchString = [
      item.id_pengukuran,
      item.nama,
      item.umur,
      item.jenis_kelamin,
      item.nomor_wa,
      item.waktu_tanggal,
      item.denyt_nadi,
      item.suhu,
      item.sistolik,
      item.diastolik
    ].map(v => (v ?? "").toString().toLowerCase()).join(' | ');
    return searchString.includes(filter.toLowerCase());
  })
  .sort((a, b) => {
    // Coba ubah id_pengukuran ke angka untuk sorting numerik, fallback ke
    string

```

```

        const idA = isNaN(Number(a.id_pengukuran)) ? a.id_pengukuran :
Number(a.id_pengukuran);
        const idB = isNaN(Number(b.id_pengukuran)) ? b.id_pengukuran :
Number(b.id_pengukuran);
        if (idA < idB) return -1;
        if (idA > idB) return 1;
        return 0;
    });

    if (dataArr.length === 0) {
        historyBody.innerHTML = `<tr><td colspan="10" style="text-
align:center;">Data tidak ditemukan</td></tr>`;
        return;
    }

    dataArr.forEach(item => {
        const row = document.createElement("tr");
        row.innerHTML = `
            <td>${item.id_pengukuran || '-'}</td>
            <td>${item.nama || '-'}</td>
            <td>${item.umur || '-'}</td>
            <td>${item.jenis_kelamin || '-'}</td>
            <td>${item.nomor_wa || '-'}</td>
            <td>${item.waktu_tanggal || '-'}</td>
            <td>${item.denyt_nadi ?? '-'}</td>
            <td>${item.suhu ?? '-'}</td>
            <td>${item.sistolik ?? '-'} / ${item.diastolik ?? '-'}</td>
            <td>
                <button class="btn-edit" data-key="${item.key}" aria-label="Edit data
pengguna ${item.nama}">Edit</button>
            </td>
        `;
    });

```

```

        <button class="btn-delete" data-key="${item.key}" aria-label="Hapus
data pengguna ${item.nama}">Hapus</button>
    </td>
    `;
    historyBody.appendChild(row);
  });
}

// Submit form handler
form.addEventListener("submit", async (e) => {
  e.preventDefault();
  formMessage.style.display = "none";
  if (!validateForm()) {
    formMessage.textContent = "Mohon isi form dengan benar.";
    formMessage.className = "message error";
    formMessage.style.display = "block";
    return;
  }
  loadingEl.style.display = "block";
  try {
    const idPengukuran = form.id_pengukuran.value.trim();
    const dataPengukuran = await getPengukuran(idPengukuran);
    if (!dataPengukuran) {
      formMessage.textContent = `ID Pengukuran "${idPengukuran}" tidak
ditemukan di database pengukuran.`;
      formMessage.className = "message error";
      formMessage.style.display = "block";
      loadingEl.style.display = "none";
      return;
    }
    const tanggalWaktu = dataPengukuran.waktu ?? '-';

```

```

const penggunaData = {
  id_pengukuran: idPengukuran,
  nama: form.nama.value.trim(),
  umur: Number(form.umur.value),
  jenis_kelamin: form.jenis_kelamin.value,
  nomor_wa: form.nomor_wa.value.trim(),
  waktu_tanggal: tanggalWaktu,
  sistolik: dataPengukuran.sistolik ?? '-',
  diastolik: dataPengukuran.diastolik ?? '-',
  suhu: dataPengukuran.suhu ?? '-',
  denyut_nadi: dataPengukuran.denyut_nadi ?? '-'
};

if (editKey) {
  await update(ref(db, `riwayat/${editKey}`), penggunaData);
  formMessage.textContent = "Data berhasil diupdate.";
} else {
  const newKey = Date.now().toString();
  await set(ref(db, `riwayat/${newKey}`), penggunaData);
  formMessage.textContent = "Data berhasil disimpan.";
}

formMessage.className = "message success";
formMessage.style.display = "block";

// Format nomor WA supaya bisa dipakai link wa.me
let nomorWAFormat = penggunaData.nomor_wa.replace(/[^0-9]/g, "");
if (nomorWAFormat.startsWith("0")) {
  nomorWAFormat = "62" + nomorWAFormat.slice(1);
}

const pesan = `Halo ${penggunaData.nama},

```

Berikut hasil pengukuran Anda:

ID Pengukuran: \${penggunaData.id_pengukuran}

Umur: \${penggunaData.umur}

Jenis Kelamin: $\{\text{penggunaData.jenis_kelamin}\}$
Waktu / Tanggal: $\{\text{penggunaData.waktu_tanggal}\}$
Denyut Nadi: $\{\text{penggunaData.denyut_nadi}\}$
Suhu: $\{\text{penggunaData.suhu}\}$
Tekanan Darah: $\{\text{penggunaData.sistolik}\} / \{\text{penggunaData.diastolik}\}$

Terima kasih telah menggunakan layanan kami.`;

```
const waUrl =  
`https://wa.me/${nomorWAMFormat}?text=${encodeURIComponent(pesan)}`;  
window.open(waUrl, "_blank");  
alert("Pesan WhatsApp telah dibuka. Silakan lanjutkan pengiriman di  
aplikasi WhatsApp.");  
resetFormUI();  
} catch (err) {  
  formMessage.textContent = "Terjadi kesalahan: " + err.message;  
  formMessage.className = "message error";  
  formMessage.style.display = "block";  
  loadingEl.style.display = "none";  
}  
});
```

```
// Ambil dan render riwayat data pengguna  
onValue(ref(db, "riwayat"), (snapshot) => {  
  riwayatData = snapshot.val();  
  renderTable(riwayatData, searchInput.value.trim());  
});
```

```
// Event handler edit dan delete  
historyBody.addEventListener("click", async (e) => {  
  const key = e.target.dataset.key;  
  if (!key) return;
```

```

if (e.target.classList.contains("btn-edit")) {
  try {
    const snapshot = await get(ref(db, `riwayat/${key}`));
    if (snapshot.exists()) {
      const pengguna = snapshot.val();
      showTab('beranda');
      form.id_pengukuran.value = pengguna.id_pengukuran || "";
      form.nama.value = pengguna.nama || "";
      form.umur.value = pengguna.umur || "";
      form.jenis_kelamin.value = pengguna.jenis_kelamin || "";
      form.nomor_wa.value = pengguna.nomor_wa || "";
      editKey = key;
      batalEditBtn.style.display = "inline-block";
      formMessage.style.display = "none";
      window.scrollTo({top:0,behavior:'smooth'});
    }
  } catch (err) {
    alert("Gagal mengambil data untuk edit: " + err.message);
  }
} else if (e.target.classList.contains("btn-delete")) {
  if (confirm("Yakin ingin menghapus data ini?")) {
    try {
      await remove(ref(db, `riwayat/${key}`));
      if (editKey === key) {
        resetFormUI();
      }
    } catch (err) {
      alert("Gagal menghapus data: " + err.message);
    }
  }
}
}

```

```

});

// Tombol batal edit
batalEditBtn.addEventListener("click", (e) => {
  e.preventDefault();
  resetFormUI();
});

// Event search input
searchInput.addEventListener("input", () => {
  const filter = searchInput.value.trim();
  renderTable(riwayatData, filter);
});
});
</script>
</head>
<body>

<div class="nav">
  <button data-tab="beranda" class="active"
onclick="showTab('beranda')">Beranda</button>
  <button data-tab="riwayat" onclick="showTab('riwayat')">Riwayat</button>
</div>

<div id="beranda" class="tab-content active">
  <h2>Monitoring Pengguna - Input Data</h2>
  <form id="data-pasien" novalidate>
    <label for="id_pengukuran">ID Pengukuran *</label>
    <input type="text" id="id_pengukuran" name="id_pengukuran"
placeholder="Masukkan ID pengukuran" required />

```

```
<div class="err-msg" id="err-id_pengukuran">ID Pengukuran wajib  
diisi.</div>
```

```
<label for="nama">Nama Pengguna *</label>  
<input type="text" id="nama" name="nama" placeholder="Masukkan nama  
pengguna" required />
```

```
<div class="err-msg" id="err-nama">Nama wajib diisi.</div>
```

```
<label for="umur">Umur *</label>  
<input type="number" id="umur" name="umur" placeholder="Masukkan  
umur (tahun)" min="0" max="120" required />
```

```
<div class="err-msg" id="err-umur">Umur harus antara 0 - 120.</div>
```

```
<label for="jenis_kelamin">Jenis Kelamin *</label>  
<select id="jenis_kelamin" name="jenis_kelamin" required>  
  <option value="">-- Pilih Jenis Kelamin --</option>  
  <option value="Laki-laki">Laki-laki</option>  
  <option value="Perempuan">Perempuan</option>  
</select>  
<div class="err-msg" id="err-jenis">Jenis kelamin wajib dipilih.</div>
```

```
<label for="nomor_wa">Nomor WhatsApp (format +62) *</label>  
<input type="tel" id="nomor_wa" name="nomor_wa"  
placeholder="+6281234567890" pattern="^\+62\d{8,15}$" required />  
<div class="err-msg" id="err-wa">Nomor WA harus diawali +62 dan diikuti  
8-15 digit angka.</div>
```

```
<button type="submit">Kirim Data</button>  
<button id="batalEdit" style="display:none;">Batal Edit</button>  
</form>  
<div id="loading">Sedang memproses data...</div>
```

```

<div id="form-message" class="message"></div>
</div>

<div id="riwayat" class="tab-content">
  <h2>Riwayat Data Pengguna</h2>
  <div id="search-container">
    <input type="text" id="search-input" placeholder="Cari data pengguna..."
aria-label="Cari nama pengguna" />
  </div>
  <table aria-label="Riwayat data pengguna">
    <thead>
      <tr>
        <th>ID Pengukuran</th>
        <th>Nama</th>
        <th>Umur</th>
        <th>Jenis Kelamin</th>
        <th>Nomor WA</th>
        <th>Waktu / Tanggal</th>
        <th>Denyut Nadi</th>
        <th>Suhu</th>
        <th>Tekanan Darah (SYS/DIA)</th>
        <th>Aksi</th>
      </tr>
    </thead>
    <tbody id="historyBody"></tbody>
  </table>
</div>

</body>
</html>

```

15% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Top Sources

- 13%  Internet sources
 - 5%  Publications
 - 5%  Submitted works (Student Papers)
-

Top Sources

- 13%  Internet sources
- 5%  Publications
- 5%  Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	repository.polman-babel.ac.id	2%
2	Student papers	Telkom University	<1%
3	Internet	www.researchgate.net	<1%
4	Internet	www.scribd.com	<1%
5	Internet	repository.bku.ac.id	<1%
6	Internet	hai.kemenkeu.go.id	<1%
7	Internet	ejournal.undiksha.ac.id	<1%
8	Internet	jtiik.ub.ac.id	<1%
9	Internet	repository.ub.ac.id	<1%
10	Internet	lebutokan.blogspot.com	<1%
11	Internet	docplayer.info	<1%

12	Publication	Septiar Rizkiansyah Sunarto. "PROTOTYPE PENGUKUR DETAK JANTUNG, SATURASI...	<1%
13	Internet	simdos.unud.ac.id	<1%
14	Publication	M Hizbul Wathan, Indra Irawan, Better Swengky, Irsan Cahyadi. "InfusCare: Smar...	<1%
15	Internet	etheses.uin-malang.ac.id	<1%
16	Internet	repository.stikstellamarismks.ac.id	<1%
17	Internet	eprints.uny.ac.id	<1%
18	Internet	123dok.com	<1%
19	Student papers	Forum Perpustakaan Perguruan Tinggi Indonesia Jawa Timur III	<1%
20	Internet	core.ac.uk	<1%
21	Internet	tambahpinter.com	<1%
22	Internet	lib.atim.ac.id	<1%
23	Student papers	Universitas Brawijaya	<1%
24	Internet	id.scribd.com	<1%
25	Internet	digilib.unila.ac.id	<1%

26	Student papers	Universitas Sebelas Maret	<1%
27	Publication	Endang Purnama Dewi, Joko Sumarsono, Amuddin Amuddin, I Gusti Made Kompy...	<1%
28	Internet	eprints.untirta.ac.id	<1%
29	Internet	ojs.unud.ac.id	<1%
30	Student papers	Politeknik Negeri Bandung	<1%
31	Internet	marostek.marospub.com	<1%
32	Internet	eprints.poltekkesjogja.ac.id	<1%
33	Internet	jurnal.iaii.or.id	<1%
34	Internet	repository.unja.ac.id	<1%
35	Student papers	Politeknik Negeri Jember	<1%
36	Student papers	SDM Universitas Gadjah Mada	<1%
37	Internet	newcomerscuerna.org	<1%
38	Internet	kc.umn.ac.id	<1%
39	Internet	repository.uinsu.ac.id	<1%

40	Publication	Masthura Masthura, Mulkan Iskandar Nasution, Rajai Sitorus. "ALAT MONITORIN...	<1%
41	Publication	Linda Handayani, Jesi Pebralia, Iful Amri, Ardiyaningsih Puji Lestari. "PENGEMBA...	<1%
42	Publication	Noor Ahda Fadillah, Muhammad Ikrar Fadhillah, Novita Agustina, Fitria Hidayati, ...	<1%
43	Publication	Rieke Revina Rahadian. "Hubungan tekanan Panas Dengan Denyut Nadi Pekerja ...	<1%
44	Student papers	Udayana University	<1%
45	Internet	jawapeh.com	<1%
46	Student papers	Universitas Muria Kudus	<1%
47	Internet	dewey.petra.ac.id	<1%
48	Internet	digilib.uinsgd.ac.id	<1%
49	Internet	ejournal.uki.ac.id	<1%
50	Internet	seputarkuliahkesehatan.blogspot.com	<1%
51	Internet	id.123dok.com	<1%
52	Student papers	itera	<1%
53	Internet	m.my.panasyslcd.com	<1%

54	Internet	repositori.usu.ac.id	<1%
55	Internet	static.buku.kemdikbud.go.id	<1%
56	Internet	www.akrabjuara.com	<1%
57	Internet	www.ghumi.id	<1%
58	Publication	Muhammad Fuad Nurillah, Bambang Guruh Irianto, I Dewa Gede Hari Wisana. "P...	<1%
59	Internet	closets-organizers.net	<1%
60	Internet	digilib.uin-suka.ac.id	<1%
61	Internet	djpk.kemenkeu.go.id	<1%
62	Internet	eprints.walisongo.ac.id	<1%
63	Internet	kumparan.com	<1%
64	Internet	pt.scribd.com	<1%
65	Internet	repository.its.ac.id	<1%
66	Internet	sariwiryanetty.blogspot.com	<1%
67	Internet	www.slideshare.net	<1%

68	Internet	www.trijurnal.lemliit.trisakti.ac.id	<1%
69	Publication	Ade Mandala Putra, Tedy Rismawan, Syamsul Bahri. "IMPLEMENTASI METODE FU...	<1%
70	Publication	Muhamad Nur Arifin, Kunto Eko Susilo, Aryo Nugroho. "RANCANG BANGUN ALAT ...	<1%
71	Publication	Muhammad Al Fiqri, Rieza Zulrian Aldio, Yoel Yonatan. "Rancang Bangun Alat Pen...	<1%
72	Publication	Patriot Teguh Santoso, Denny Irawan. "Rancang Bangun Alat Otomatisasi Penger...	<1%
73	Publication	Risma Putri Vandini, Demes Nurmayanti, Ernita Sari, Winarko. "Denyut Nadi Dan...	<1%
74	Publication	Sabang Firdaus, Tedy Rismawan, Uray Ristian. "SISTEM MANAJEMEN PENGAIRAN ...	<1%
75	Publication	Winandya Ayu Kemalla, M Rizqon Kamil, Muhammad Jamaluddin. "Efektivitas Tek...	<1%
76	Internet	adoc.pub	<1%
77	Internet	automationindo.com	<1%
78	Internet	edoc.ub.uni-muenchen.de	<1%
79	Internet	forumahlipengajar.blogspot.com	<1%
80	Internet	jom.unpak.ac.id	<1%
81	Internet	jurnalstmiksubang.ac.id	<1%

82	Internet	jurnalsyntaxadmiration.com	<1%
83	Internet	libraryeproceeding.telkomuniversity.ac.id	<1%
84	Internet	pdfcoffee.com	<1%
85	Internet	qastack.id	<1%
86	Internet	repository.maranatha.edu	<1%
87	Internet	repository.widyatama.ac.id	<1%
88	Internet	www.alodokter.com	<1%
89	Internet	www.coursehero.com	<1%
90	Publication	Muhammad Khoirul Hasin, Moh. Andy Wiranata, Aulia Nadia Rachmat. "Rancang ...	<1%
91	Publication	Tri Sakti Sutrisno, Koko Joni, Achmad Fiqhi Ibadillah, , Haryanto, Deni Tri Lakso...	<1%
92	Publication	Zulkifli Zulkifli, Muhlis Muhallim, Hasnahwati Hasnahwati. "PENGEMBANGAN SIS...	<1%
93	Internet	doku.pub	<1%
94	Internet	jurnal.ukh.ac.id	<1%
95	Internet	repository.telkomuniversity.ac.id	<1%

96

Internet

thousands-passed.xyz

<1%

SURAT PERNYATAAN

Saya/Kami yang bertandatangan dibawah ini telah menyelesaikan Proyek Akhir yang berjudul:
Rancang Bangun Alat Kesehatan Pengukur dan Sistem Monitoring Denyut Nadi, Suhu, dan Tekanan
Darah Berbasis *Internet of Things* (IoT)

Oleh :

1. Hardi Perwira /NPM 1052241
2. Merizta Aulia /NPM 1052246

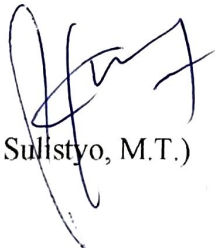
Dengan ini menyatakan bahwa isi laporan akhir proyek akhir sama dengan *hardcopy*.
Demikian surat pernyataan ini dibuat dengan sebenar-benarnya.

Sungailiat, 15 September 2025

1. Hardi Perwira (..... )
2. Merizta Aulia (..... )

Mengetahui,

Pembimbing 1,


(Eko Sulistyono, M.T.)

Pembimbing 2,


(Evvin Faristasari, M.Sc.)