

**RANCANG BANGUN *REAL-TIME MONITORING DATA SYSTEM*
UNTUK PERANGKAT BUOY U-TEWS DENGAN TEKNOLOGI PERN
*STACK***

(Skripsi)

Oleh
Muhammad Rafsanjani
2115031015



**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2025**

ABSTRAK

RANCANG BANGUN REAL-TIME MONITORING DATA SYSTEM UNTUK PERANGKAT BUOY U-TEWS DENGAN TEKNOLOGI PERN STACK

Oleh

MUHAMMAD RAFSANJANI

Peristiwa tsunami di Selat Sunda bersifat near-field dengan kesempatan evakuasi yang kerap hanya dalam hitungan menit; perubahan muka air berlangsung cepat dan pola gelombang dapat bergeser secara dinamis. Karena itu, pemantauan yang benar-benar real time, berlatensi rendah, aliran data kontinu, dan tahan terhadap fluktuasi jaringan pesisir, menjadi prasyarat untuk deteksi dini, verifikasi kejadian, dan pengambilan keputusan operasional. Kebutuhan ini dijawab melalui pengembangan buoy Unila–Tsunami Early Warning System (U-TEWS) oleh Universitas Lampung.. Buoy ini dilengkapi dengan multi sensor yang meliputi sensor tekanan air, kecepatan dan arah angin, koordinat GPS, serta sensor pemantauan kondisi perangkat. Semua data dikirim secara real-time menggunakan protokol komunikasi MQTT. Penelitian ini merancang dan membangun sistem monitoring data real-time untuk buoy U-TEWS menggunakan teknologi PERN Stack, yang terdiri dari PostgreSQL sebagai basis data, Express.js dan Node.js sebagai backend, dan React untuk frontend. Sistem monitoring ini menerima input data sensor dari enam topik MQTT, menyimpannya secara terstruktur dalam enam tabel PostgreSQL, menyediakan REST API untuk integrasi, serta menampilkan visualisasi data interaktif melalui dashboard web. Pengujian performa menunjukkan latency rata-rata pengiriman data dari broker MQTT ke PostgreSQL sebesar 62,8 milidetik. Selain itu, dari 25 endpoint API yang diuji menggunakan Postman, diperoleh respon rata rata 9,256 detik. Optimasi kueri basis data dengan pemanfaatan Index Scan juga menghasilkan waktu eksekusi rata-rata di bawah 1 milidetik. Evaluasi antarmuka pengguna dengan Google Lighthouse memberikan skor performa sebesar 95 (desktop) dan 77 (mobile). Temuan ini menunjukkan sistem telah memenuhi kebutuhan pemantauan data real time dan menjadi komponen penting dalam mendukung mitigasi bencana di wilayah pesisir.

Kata Kunci: U-TEWS, PERN *Stack*, MQTT, *monitoring real-time*, *dashboard web*, sistem peringatan dini tsunami, PostgreSQL, React.js, Express.js, Node.js.

ABSTRACT

DESIGN AND DEVELOPMENT OF A REAL-TIME DATA MONITORING SYSTEM FOR U-TEWS BUOY USING PERN STACK TECHNOLOGY

By

MUHAMMAD RAFSANJANI

Tsunami events in the Sunda Strait are near-field, often leaving only minutes for evacuation; sea level can change rapidly and wave patterns shift dynamically. Consequently, truly real-time monitoring, low latency, continuous data flow, and resilience to fluctuating coastal networks, is essential for early detection, event verification, and operational decision-making. This need is addressed through the University of Lampung's development of the Unila-Tsunami Early Warning System (U-TEWS) buoy. The buoy is equipped by multiple sensors, including water pressure, wind speed and direction, GPS coordinates, and device condition monitoring sensors. All data is transmitted in real time using the MQTT communication protocol. This study designs and implements a real-time data monitoring system for the U-TEWS buoy using the PERN Stack, consisting of PostgreSQL as the database, Express.js and Node.js for the backend, and React for the frontend. The system ingests sensor data from six MQTT topics, stored them in six structured PostgreSQL tables provides REST APIs for integration purposes, and displaying interactive data visualizations through a web dashboard. Performance tests showed an average latency of 62.8 millisecond for transferring data from the MQTT broker to PostgreSQL. In addition, the tests on 25 API endpoints using Postman yielded an average response time of 9.256 seconds. Database query optimization using Index Scan achieved average execution time below 1 millisecond. User interface evaluation using Google Lighthouse showed a performance score of 95 for the desktop view and 77 for the mobile view. These results indicate that The system meets real-time monitoring requirements and can serve as a key component in supporting disaster mitigation systems in coastal areas.

Keywords: U-TEWS, PERN Stack, MQTT, real-time monitoring, web dashboard, tsunami early warning system, PostgreSQL, React.js, Express.js, Node.js.

**RANCANG BANGUN *REAL-TIME MONITORING DATA SYSTEM*
UNTUK PERANGKAT BUOY U-TEWS DENGAN TEKNOLOGI PERN
STACK**

**Oleh
MUHAMMAD RAFSANJANI**

Skripsi

**Sebagai Salah Satu Syarat untuk Mencapai Gelar
SARJANA TEKNIK**

Pada

**Jurusan Teknik Elektro
Fakultas Teknik Universitas Lampung**



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2025**

Judul Skripsi : **RANCANG BANGUN MONITORING DATA
SYSTEM REAL-TIME UNTUK
PERANGKAT BUOY U-TEWS DENGAN
TEKNOLOGI PERN STACK**

Nama Mahasiswa : **Muhammad Rafsanjani**

Nomor Pokok Mahasiswa : 2115031015

Program Studi : Teknik Elektro

Fakultas : Teknik

MENYETUJUI

1. Komisi Pembimbing

Pembimbing Utama

Pembimbing Pendamping

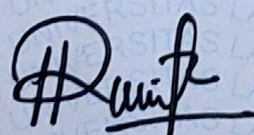

Ing. Ardian Ulvan, S.T., M.Sc., Ph.D.
NIP. 19731128 199903 1 005

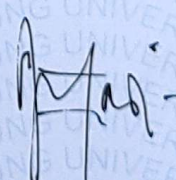

Mona Arif Muda, S.T., M.T.
NIP. 19711112 200003 1 002

2. Mengetahui

Ketua Jurusan
Teknik Elektro

Ketua Program Studi
Teknik Elektro


Herlinawati, S.T., M.T.
NIP. 19710314 199903 2 001


Sumadi, S.T., M.T.
NIP. 19731104 200003 1 001

MENGESAHKAN

1. Tim Penguji

Ketua : Ing. Ardian Ulvan, S.T., M.Sc., Ph.D.

Sekretaris : Mona Arif Muda, S.T., M.T.

Penguji : Ing. Melvi, S.T., M.T., Ph.D.



2. Dekan Fakultas Teknik

Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc. ✓

NIP. 19750928 200112 1 002

Tanggal Lulus Ujian Skripsi : 17 Juli 2025

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini menyatakan bahwa karya tulis ilmiah saya yang berjudul “Rancang Bangun *Monitoring Data System Real-Time* Untuk Perangkat Buoy U-TEWS Dengan Teknologi *PERN Stack*” merupakan hasil pekerjaan saya sendiri. Segala isi dan hasil yang disajikan dalam skripsi ini telah disusun sesuai dengan pedoman penulisan ilmiah di Universitas Lampung. Apabila di kemudian hari terbukti bahwa skripsi ini merupakan hasil penjiplakan atau dibuat oleh pihak lain, saya siap menerima sanksi sesuai dengan peraturan hukum maupun ketentuan akademik yang berlaku.

Bandar Lampung, 17 Juli 2025

Pembuat pernyataan,



Muhammad Rafsanjani

NPM. 2115031015

RIWAYAT HIDUP



Penulis dilahirkan di Tulang Bawang pada tanggal 21 Desember 2002. Penulis merupakan anak kedua dari dua bersaudara dari pasangan Bapak Muhammad Yusuf dan Ibu Siti Umi Zahrotul Mudmainah.

Penulis memulai pendidikan di TK Bratasena Adiwarna pada tahun 2008-2009, SDN 1 Bratasena Adiwarna pada tahun 2009-2015, SMP IT Miftahul Jannah pada tahun 2015-2018, dan SMAN 13 Bandar Lampung pada tahun 2018-2021.

Penulis menjadi mahasiswa Jurusan Teknik Elektro Universitas Lampung pada tahun 2021 melalui jalur SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri). Selama menjadi mahasiswa, penulis pernah menjadi anggota Himpunan Mahasiswa Teknik Elektro (HIMATRO) sebagai anggota divisi Pengabdian Masyarakat pada periode 2022 dan periode 2023. Penulis mengambil konsentrasi Telekomunikasi dan Teknologi Informasi dan secara aktif mengikuti kegiatan akademik. Pada tahun 2023 penulis mengikuti Lomba Inovasi Digital Mahasiswa yang berhasil menjadi finalis dan menghasilkan paten, mengikuti Studi Independen Kampus Merdeka, dan Riset Sistem Pendeteksi Dini Tsunami yang dilaksanakan di Wilayah Hutan Mangrove Petengoran, Pesisir Desa Canti Kalianda, dan Perairan Anak Gunung Krakatau

PERSEMBAHAN

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Dengan rasa syukur yang mendalam kepada Allah SWT atas segala rahmat dan karunia-Nya, karya ini penulis persembahkan kepada:

“Ayahanda Muhammad Yusuf dan Ibunda Siti Umi Zahrotul Mudmainah”

Yang dengan penuh cinta, doa, dan pengorbanan tiada henti telah membimbing dan mendampingi penulis dalam setiap langkah kehidupan. Terima kasih atas kasih sayang, keikhlasan, serta semangat yang tak pernah padam.

“Kakakku Annisa Rodhiyaturrizki”

Yang selalu menjadi motivasi dan penyemangat dalam menghadapi tantangan.

“Dosen pembimbing dan seluruh dosen Jurusan Teknik Elektro Universitas Lampung”

Atas ilmu, bimbingan, dan arahan yang telah diberikan selama proses studi hingga penyusunan skripsi ini.

“Teman-teman seperjuangan di Teknik Elektro 2021”

Khususnya rekan-rekan satu konsentrasi dan tim proyek, terima kasih atas kebersamaan, bantuan, dan semangat yang telah kita bangun bersama.

“Diri Sendiri”

yang telah bertahan, berjuang, dan tidak menyerah meskipun dalam berbagai keterbatasan, tekanan, dan tantangan. Terima kasih telah terus melangkah hingga titik ini.

MOTTO HIDUP

“Jika kamu bersyukur, niscaya Aku akan menambah (nikmat) kepadamu.”

(QS. Ibrahim: 7)

“Allah tidak akan membebani seseorang melainkan sesuai dengan kesanggupannya.”

(QS. Al-Baqarah: 286)

"Jika kamu ingin dunia, carilah ilmu. Jika kamu ingin akhirat, carilah ilmu. Jika kamu ingin keduanya, carilah ilmu."

(Imam Syafi’I)

"Bekerjalah untuk duniamu seolah-olah kamu akan hidup selamanya. Dan bekerjalah untuk akhiratmu seolah-olah kamu akan mati besok."

(Ali bin Abi Thalib)

SANWACANA

Puji syukur kehadirat Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul “Rancang Bangun *Monitoring Data System Real-Time* Untuk Perangkat Buoy U-TEWS Dengan Teknologi *PERN Stack*” sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik di Jurusan Teknik Elektro, Fakultas Teknik, Universitas Lampung.

Skripsi ini disusun dengan tujuan untuk mengembangkan sebuah sistem *monitoring* data secara *real-time* yang dapat digunakan pada perangkat buoy U-TEWS sebagai bagian dari upaya penguatan sistem pemantauan lingkungan berbasis teknologi. Selama proses penyusunan, penulis telah mendapatkan banyak bantuan, arahan, dan dukungan dari berbagai pihak.

Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Ibu Prof. Dr. Ir. Lusmeilia Afriani, D.E.A., I.P.M., selaku Rektor Universitas Lampung
2. Bapak Dr. Eng. Helmy Fitriawan, S.T., M.Sc. selaku Dekan Fakultas Teknik Universitas Lampung.
3. Ibu Herlinawati, S.T., M.T. selaku Ketua Jurusan Teknik Elektro Fakultas Teknik Universitas Lampung.
4. Bapak Sumadi, S.T., M.T. selaku Kepala Program Studi Teknik Elektro Universitas Lampung.
5. Bapak Dr. Eng. F.X Arinto Setyawan, S.T., M.T. selaku dosen pembimbing akademik.
6. Bapak Ing. Ardian Ulvan, S.T., M.Sc., Ph.D. selaku dosen pembimbing utama yang telah memberikan bimbingan, saran, dan arahan dengan penuh kesabaran selama proses penyusunan skripsi ini.

7. Bapak Mona Arif Muda, S.T., M.T. selaku dosen pembimbing pendamping yang telah memberikan bimbingan, saran, dan arahan dengan penuh kesabaran selama proses penyusunan skripsi ini.
8. Ibu Ing. Melvi, S.T., M.T., Ph.D. selaku dosen penguji yang telah memberikan bimbingan, saran, dan arahan dengan penuh kesabaran selama proses penyusunan skripsi ini.
9. Segenap dosen Jurusan Teknik Elektro Fakultas Teknik Universitas Lampung, yang telah memberikan ilmu dan wawasan selama masa perkuliahan.
10. Diri penulis sendiri, yang telah bertahan, tidak menyerah, dan terus melangkah meski dihadapkan pada berbagai rintangan. Terima kasih telah percaya bahwa semua proses ini akan membawa hasil.
11. Ayahanda Muhammad Yusuf dan Ibunda Siti Umi Zahrotul Mudmainah, atas doa yang tulus, kasih sayang, serta dukungan yang tiada henti.
12. Kakak tercinta Annisa Rodhiyaturrizki, atas semangat, perhatian, dan dukungan moral yang tak pernah putus.
13. Keluarga PUMMA 21 dan PUMMA 22 yang telah menemani penulis selama masa akhir perkuliahan, dan menjadi penyemangat dalam perjalanan masa skripsi.
14. Tegar, Frissa, dan Destalia yang telah menemani penulis dari saat menjadi mahasiswa baru hingga sekarang dan telah membantu selama masa perkuliahan.
15. Seluruh teman-teman konsentrasi Telekomunikasi dan Teknologi Informasi 2021 yang telah memberikan dukungan dan semangat.
16. Keluarga besar Excalto 21 yang telah memberikan pengalaman, kebersamaan selama masa perkuliahan.
17. Keluarga Besar Himatro Unila yang telah memberikan pengalaman tak terlupakan.
18. Seluruh pihak yang telah membantu penulis menyelesaikan skripsi ini.

Akhir kata, penulis mengucapkan terima kasih kepada semua pihak yang telah memberikan dukungan, baik secara langsung maupun tidak langsung, dalam penyusunan skripsi ini. Penulis berharap skripsi ini dapat memberikan manfaat, tidak hanya sebagai syarat kelulusan, tetapi juga sebagai kontribusi kecil dalam

pengembangan teknologi monitoring berbasis Internet of Things (IoT), khususnya pada perangkat buoy U-TEWS.

Penulis juga menyadari bahwa skripsi ini masih memiliki kekurangan dan keterbatasan. Oleh karena itu, saran dan kritik yang membangun dari pembaca sangat penulis harapkan demi perbaikan di masa yang akan datang.

Semoga Allah SWT senantiasa memberkahi setiap langkah kita dalam mencari dan menyebarkan ilmu.

Bandar Lampung, 17 Juli 2025

Muhammad Rafsanjani

DAFTAR ISI

	Halaman
PERSEMBAHAN.....	i
MOTTO HIDUP	ii
SANWACANA	iii
DAFTAR ISI.....	vi
DAFTAR GAMBAR.....	viii
DAFTAR TABEL	x
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	4
1.6 Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA.....	6
2.1 Penelitian Terkait	6
2.2 Buoy U-TEWS	9
2.3 Full-stack Developer	10
2.4 Lingkungan Server	10
2.5 Teknologi Pembuatan Website	11
2.5.1 PostgreSQL	11
2.5.2 Express.js	12
2.5.3 React.....	12
2.5.4 Node.js	13
2.6 Komunikasi Interface	13
2.7 Teks Editor	14
2.8 API Tester	14
2.9 Protokol Komunikasi	15
2.10 Web Hosting	16
2.11 Metode Penelitian yang Digunakan.....	16
2.12 Framework CSS.....	17
2.13 Package Builder.....	18
BAB III METODOLOGI PENELITIAN	20
3.1 Waktu dan Tempat Penelitian.....	20
3.2 Capstone Project	20
3.3 Alat dan Bahan.....	22
3.3.1 Alat.....	22
3.3.2 Bahan.....	25
3.4 Tahapan Penelitian.....	29

3.5 Tahapan Visualisasi Data	31
3.6 Studi Literatur	32
3.7 Analisa Kebutuhan	33
3.8 Pengembangan Sistem	35
3.8.1 Tahap Task To Do.....	35
3.8.2 Tahap Work in Progress.....	38
3.8.3 Tahap Testing.....	39
3.8.4 Tahap <i>Done</i>	39
3.9 Analisa Hasil	40
BAB IV HASIL DAN PEMBAHASAN	42
4.1 Proses Pengembangan Sistem.....	42
4.1.1 Membuat Kartu Tugas	42
4.1.2 Flow Pertama	43
4.1.3 Flow Ke Dua.....	53
4.1.4 Flow Ke Tiga	61
4.1.5 Flow ke Empat.....	66
4.1.6 Flow ke Lima	71
4.1.7 Flow ke Enam	74
4.1.8 Flow ke Tujuh.....	77
4.1.9 Flow ke Delapan	81
4.1.10 Flow ke Sembilan	84
4.1.11 Flow ke Sepuluh	88
4.2 Trial and Error.....	92
4.3 Release	93
4.4 Analisa Hasil	96
4.4.1 Kecepatan pengiriman data MQTT ke <i>backend</i>	96
4.4.2 Kecepatan respon backend.....	98
4.4.3 Analisis Performa Query dengan EXPLAIN ANALYZE.....	99
4.4.3 <i>Testing web</i>	101
BAB V KESIMPULAN DAN SARAN	103
5.1 Kesimpulan.....	103
5.2 Saran.....	103
DAFTAR PUSTAKA	104
LAMPIRAN.....	107

DAFTAR GAMBAR

	Halaman
Gambar 1. 1 Anak Gunung Krakatau[2]	1
Gambar 2. 1 Perangkat Buoy U-TEWS	9
Gambar 2. 2 <i>Virtual Private Server</i> [10]	10
Gambar 2. 4 React[15]	12
Gambar 2. 7 Message Queue Telemetry Transport (MQTT)[21].....	15
Gambar 3. 1 Diagram keseluruhan proyek perangkat Buoy U-TEWS.....	21
Gambar 3. 2 Diagram alur tahapan penelitian Web Buoy U-TEWS	30
Gambar 3. 3 Flowchart visualisasi data	31
Gambar 3. 4 Tahapan pengembangan sistem dengan metode kanban.....	35
Gambar 3. 5 Tampilan kanban method dalam aplikasi Notion.....	40
Gambar 4. 1 Kartu tugas	42
Gambar 4. 2 Flow pertama pengembangan sistem	43
Gambar 4. 3 Diagram aliran data sistem.....	44
Gambar 4. 4 Activity diagram handler program MQTT save data into backend .	46
Gambar 4. 5 <i>Activity diagram REST-API request and response</i>	47
Gambar 4. 6 <i>Activity diagram Front-end</i>	48
Gambar 4. 7 .ENV untuk <i>subscribe topic</i> MQTT.....	51
Gambar 4. 8 Kode untuk menghubungkan <i>backend</i> ke MQTT broker dengan mengambil alias dari .env	51
Gambar 4. 9 Kode <i>listener</i> untuk MQTT.....	52
Gambar 4. 10 <i>Flow</i> Ke Dua	53
Gambar 4. 11 File model untuk <i>database</i>	58
Gambar 4. 12 Kode program membuat kolom di database.....	59
Gambar 4. 13 kode program penulisan koneksi <i>database</i>	60
Gambar 4. 14 Hasil table database di pgadmin.....	60
Gambar 4. 15 <i>Flow</i> ke Tiga.....	61
Gambar 4. 16 kode program menampilkan data terakhir.....	62
Gambar 4. 17 Kode untuk <i>routing</i> menampilkan data terakhir	62
Gambar 4. 18 Hasil <i>testing</i> URL memanggil data terakhir.....	63
Gambar 4. 19 Kode program untuk menampilkan data perhari.....	64
Gambar 4. 20 Kode program untuk menampilkan data perminggu.....	65
Gambar 4. 21 Kode program untuk menampilkan data perbulan	66
Gambar 4. 22 <i>Flow</i> ke empat	66
Gambar 4. 23 Kode program menampilkan 10 data terakhir.....	67
Gambar 4. 24 Kode program user.routes.js.....	68
Gambar 4. 25 Kode program di buoy.js	70
Gambar 4. 26 <i>middleware</i>	71
Gambar 4. 27 <i>Flow</i> ke lima.....	71
Gambar 4. 28 Potongan kode program untuk navbar	72

Gambar 4. 29 Navbar web buoy U-TEWS	73
Gambar 4. 30 Potongan kode program <i>landing page</i>	73
Gambar 4. 31 Landing page web Buoy U-TEWS.....	74
Gambar 4. 32 Flow ke enam	74
Gambar 4. 33 Potongan kode program sidebar.....	75
Gambar 4. 34 Sidebar web Buoy U-TEWS	76
Gambar 4. 35 Potongan kode program untuk <i>card component</i>	76
Gambar 4. 36 Card component	77
Gambar 4. 37 flow ke tujuh.....	77
Gambar 4. 38 Potongan kode program <i>table component</i>	78
Gambar 4. 39 Data table web buoy U-TEWS.....	79
Gambar 4. 40 potongan kode program <i>chart component</i>	79
Gambar 4. 41 <i>chart</i> pada web buoy U-TEWS	80
Gambar 4. 42 <i>Flow</i> ke delapan	81
Gambar 4. 43 Potongan kode program pembuatan <i>maps component</i>	82
Gambar 4. 44 Map pada web buoy U-TEWS	83
Gambar 4. 45 Potongan kode program pembuatan <i>footer component</i>	83
Gambar 4. 46 <i>footer</i> pada web buoy U-TEWS	84
Gambar 4. 47 <i>flow</i> ke sembilan.....	84
Gambar 4. 48 Potongan kode program pembuatan <i>Time Selector component</i>	85
Gambar 4. 49 Time Selector	86
Gambar 4. 50 Potongan kode program pembuatan halaman <i>dashboard</i> buoy	86
Gambar 4. 51 Halaman buoy	87
Gambar 4. 52 Flow ke sepuluh	88
Gambar 4. 53 Potongan kode untuk membuat web responsive	88
Gambar 4. 54 file untuk program logger.....	90
Gambar 4. 55 Kode program di Apps Script.....	90
Gambar 4. 56 Kode program untuk mengirim data ke google apps script	91
Gambar 4. 57 Folder yang sudah dibuat berdasarkan topic	91
Gambar 4. 58 File logger setiap satu bulan.....	92
Gambar 4. 59 perintah masuk ke folder	93
Gambar 4. 60 perintah menjalankan program utama	93
Gambar 4. 61 Proses <i>backend</i> yang berjalan pada server	94
Gambar 4. 62 perintah untuk <i>build</i> proyek	94
Gambar 4. 63 domain untuk akses web buoy	95
Gambar 4. 64 isi folder buoy di hpanel.....	95
Gambar 4. 65 alamat web buoy U-TEWS	95
Gambar 4. 66 Command untuk menjalankan tcpdump.....	96
Gambar 4. 67 Filter paket MQTT	97
Gambar 4. 68 Filter paket PostgreSQL yang mengandung table modbusnew	97
Gambar 4. 69 Query explain analyze	100
Gambar 4. 70 Hasil pengujian web dalam bentuk <i>desktop</i>	101
Gambar 4. 71 Hasil pengujian web dalam bentuk <i>mobile</i>	102

DAFTAR TABEL

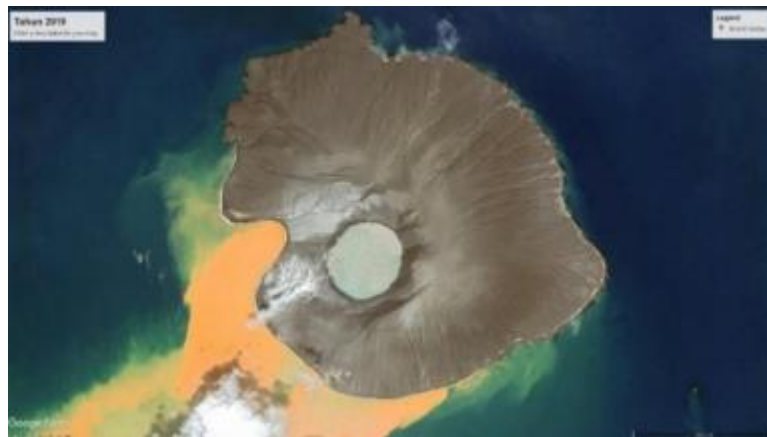
	Halaman
Tabel 3. 2 Alat berupa Hardware dan Software yang digunakan.	22
Tabel 3. 3 Bahan berupa data sensor serial dari raspberry pi.....	25
Tabel 3. 4 Bahan berupa dari Modbus	26
Tabel 3. 5 Bahan berupa dari scc	27
Tabel 3. 6 Bahan berupa dari GPS	28
Tabel 3. 7 Bahan berupa dari cpu.....	28
Tabel 3. 8 Tugas <i>High Priority</i>	36
Tabel 3. 9 Tugas <i>Medium Priority</i>	38
Tabel 3. 10 Tugas <i>Low Priority</i>	38
Tabel 4. 1 Topic yang digunakan pada perangkat buoy.....	50
Tabel 4. 2 Tabel modbus.....	53
Tabel 4. 3 tabel serial	54
Tabel 4. 4 tabel scc.....	55
Tabel 4. 5 tabel gpsnew.....	55
Tabel 4. 6 tabel CPU	56
Tabel 4. 7 tabel RAW.....	57
Tabel 4. 8 breakpoints by default	89
Tabel 4. 9 Pengukuran kecepatan antara MQTT ke database	98
Tabel 4. 10 Kecepatan respon backend.....	98
Tabel 4. 11 Hasil query explain analyze	100

BAB I

PENDAHULUAN

1.1.Latar Belakang

Provinsi Lampung, yang terletak di ujung selatan Pulau Sumatera, memiliki posisi geografis yang sangat strategis namun rentan terhadap ancaman bencana alam, terutama tsunami. Sejarah mencatat bahwa letusan Gunung Krakatau pada tahun 1883 telah menghasilkan tsunami dahsyat yang menghancurkan wilayah sekitar Selat Sunda, menewaskan lebih dari 36.000 jiwa dan mengubah lanskap geologi regional [1]



Gambar 1. 1 Anak Gunung Krakatau[2]

Perkembangan teknologi sistem peringatan dini tsunami (*Tsunami Early Warning System/TEWS*) menjadi salah satu strategi paling kritis dalam mitigasi risiko bencana. Universitas Lampung (Unila) telah mengembangkan U-TEWS sebagai solusi inovatif untuk pemantauan *real-time* perubahan ketinggian permukaan air laut di sekitar wilayah rawan tsunami. Sistem ini memanfaatkan teknologi *Internet of Things* (IoT) untuk mengumpulkan dan mentransmisikan data secara kontinyu dan akurat.

Kemajuan teknologi informasi dan komunikasi telah membuka peluang baru dalam pengembangan sistem pemantauan bencana. Integrasi sensor IoT dengan platform digital memungkinkan analisis data yang lebih cepat, presisi, dan komprehensif. Namun, tantangan utama dalam implementasi sistem peringatan dini tidak hanya terletak pada akuisisi data, tetapi juga pada kemampuan visualisasi dan interpretasi informasi secara *real-time*.

Pendekatan berbasis teknologi *PERN stack*, yang mengintegrasikan pengembangan *frontend* dan *backend*, menjadi solusi strategis untuk mengatasi kompleksitas pengolahan dan visualisasi data sensor. *Backend* berbasis Node.js dan Express.js memberikan keunggulan dalam pengelolaan data yang efisien, skalabel, dan mendukung analisis *real-time*, sementara *frontend* yang responsif dan intuitif memastikan data tersebut dapat disajikan secara informatif dan mudah diakses oleh pengguna. Kombinasi ini dikenal karena keterpaduannya serta kemudahan dalam penggunaan JavaScript di seluruh lapisan pengembangan, menjadikannya pilihan ideal untuk berbagai skenario pengembangan web.[3]

Berdasarkan konteks tersebut, penelitian ini bertujuan untuk rancang bangun *dashboard* IoT *real-time* untuk U-TEWS menggunakan teknologi *PERN stack* (PostgreSQL, Express.js, React, Node.js). *Backend* akan dibangun dengan menggunakan Node.js dan Express.js untuk mengelola data sensor secara efisien, serta PostgreSQL untuk penyimpanan data yang terstruktur dan skalabel. Sementara itu, *frontend* akan dikembangkan menggunakan React untuk menyajikan data secara interaktif dan responsif. Tujuan utamanya adalah menciptakan platform visualisasi data yang komprehensif, mudah digunakan, dan memberikan informasi

akurat tentang perubahan ketinggian air laut secara instan, mendukung pemantauan *real-time* dan pengambilan keputusan yang tepat.

1.2. Rumusan Masalah

Rumusan masalah dari penelitian ini adalah:

1. Bagaimana data yang dikirimkan oleh sensor menggunakan *protocol Message Queuing Telemetry Transport* (MQTT) dapat disimpan baik kedalam *database server*.
2. Bagaimana menjalankan sistem *backend* pada *Virtual Private Server* (VPS).
3. Bagaimana cara mengambil dan mengolah data yang telah disimpan menjadi sebuah API untuk mendukung penelitian lainnya.
4. Bagaimana cara agar data dari sensor dapat di visualisasikan secara *real-time* pada sebuah *dashboard*

1.3 Batasan Masalah

Batasan-batasan masalah penelitian adalah:

1. Penelitian ini berfokus pada implementasi teknologi PERN Stack yang terhubung dengan Perangkat Buoy U-TEWS di wilayah Perairan Pulau Sebesi.
2. Penelitian ini tidak mencakup proses pembuatan perangkat keras dan sistem Buoy U-TEWS.

1.4 Tujuan Penelitian

Tujuan dilakukan penelitian adalah sebagai berikut:

1. Menerapkan teknologi PERN Stack untuk proses pengiriman dan penerimaan data sensor secara *real-time* antara perangkat buoy U-TEWS dan *dashboard monitoring*.
2. Mengelola data dari sensor pada perangkat buoy U-TEWS dengan menyimpannya ke dalam *database* PostgreSQL menggunakan Node.js.
3. Merancang A P I untuk mengakses data yang telah tersimpan di *database*.
4. Mengembangkan *frontend* menggunakan react untuk menampilkan visualisasi data secara dinamis, memastikan pengalaman pengguna yang interaktif dan

responsif dalam memantau perubahan ketinggian air laut secara *real-time*.

1.5 Manfaat Penelitian

Beberapa manfaat dari penelitian ini adalah :

1. Memudahkan para peneliti dan masyarakat untuk mengolah data yang diperlukan.
2. Memberikan data ketinggian permukaan laut, arah angin, dan parameter lainnya secara langsung, yang berguna untuk analisis dan pengambilan keputusan cepat.
3. Mengintegrasikan *frontend* dan *backend* dalam satu platform visualisasi yang interaktif dan responsif, memungkinkan pengguna untuk dengan mudah memantau data sensor dan memahami informasi yang disajikan.

1.6 Sistematika Penulisan

Dalam pembahasan laporan penelitian ini, sistematika penulisan yang digunakan adalah sebagai berikut :

BAB I : PENDAHULUAN

Pada bab ini menjelaskan tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian serta sistematika penulisan laporan mengenai penelitian Rancang Bangun *Monitoring Data System Real-Time* untuk Perangkat Buoy U-Tews dengan Teknologi PERN Stack.

BAB II : TINJAUAN PUSTAKA DAN LANDASAN TEORI

Membahas tentang penelitian-penelitian sebelumnya pada tinjauan pustaka, dan dasar-dasar teori dari penelitian Rancang Bangun *Monitoring Data System Real-Time* untuk Perangkat Buoy U-Tews dengan Teknologi PERN Stack.

BAB III : METODE PENELITIAN

Menjelaskan waktu, tempat, alat dan bahan, dan metode penelitian beserta tahapannya mengenai Rancang Bangun *Monitoring Data System Real-Time* untuk Perangkat Buoy U-TEWS dengan Teknologi PERN Stack.

BAB IV : HASIL DAN PEMBAHASAN

Menjelaskan bagaimana proses pelaksanaan penelitian dan juga hasil yang didapatkan pada penelitian mengenai Rancang Bangun *Monitoring Data System Real-Time* untuk Perangkat Buoy U-TEWS dengan Teknologi PERN Stack.

BAB V : KESIMPULAN DAN SARAN

Memaparkan kesimpulan apa saja yang didapat dari proses pelaksanaan penelitian mengenai Rancang Bangun *Monitoring Data System Real-Time* untuk Perangkat Buoy U-TEWS dengan Teknologi PERN Stack dan memberikan saran untuk dilakukan pada penelitian selanjutnya.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian yang dilakukan oleh Anh Vu (2021) menggunakan pendekatan arsitektur *microservices* dan *serverless* untuk mendukung pengelolaan data secara *real-time* dengan latensi rendah. Sebagai studi kasus, penelitian ini berhasil merancang dan mengimplementasikan *Minimum Viable Product* (MVP) untuk aplikasi pemesanan taksi, yang mencakup fitur seperti pengecekan ketersediaan taksi, pemesanan, dan manajemen perusahaan[4]

Perbedaan penelitian ini dengan penelitian Anh Vu yaitu fokusnya pada pengembangan *backend real-time* untuk sistem IoT menggunakan Node.js, Express.js, dan PostgreSQL, dengan memanfaatkan MQTT untuk menerima data sensor secara langsung dari perangkat di lapangan. Sementara itu, penelitian Anh Vu membahas pengembangan *backend* untuk aplikasi pemesanan taksi berbasis Node.js dan Express.js, tetapi menggunakan layanan *Google Cloud Platform* seperti *Cloud Firestore* dan *Cloud Functions* dalam arsitektur *microservices* dan *serverless*. Penelitian ini menonjolkan pengelolaan data sensor secara *real-time* dalam lingkungan IoT tanpa ketergantungan pada layanan *cloud* tambahan, yang berbeda dari pendekatan berbasis layanan *cloud* dalam penelitian Anh Vu, yang lebih berfokus pada pengelolaan transaksi pemesanan taksi dengan skalabilitas dan ketersediaan tinggi.

Selanjutnya penelitian yang dilakukan oleh Hao Sun dan Li Chen (2023) membahas pengembangan sistem *monitoring* jarak jauh berbasis IoT dengan memanfaatkan MySQL sebagai *database*, Vue untuk pengembangan *frontend*, dan C# untuk *backend*. Studi ini berhasil merancang sebuah platform perangkat lunak yang memungkinkan pengguna untuk melihat data *real-time*, mengontrol perangkat IoT, serta mengelola data melalui antarmuka aplikasi[5].

Perbedaan penelitian ini dengan penelitian yang dilakukan Hao Sun dan Li Chen yaitu pada penelitian Hao Sun dan Li Chen memanfaatkan MySQL sebagai *database*, sedangkan dalam penelitian ini menggunakan Postgresql, kemudian penelitian yang dilakukan Hao Sun dan Li Chen menggunakan Vue untuk pengembangan *frontend*, dan C# untuk *backend*, berbeda dengan penelitian ini yang menggunakan react sebagai *frontend* dan Node.js sebagai *backend*.

Dalam penelitian yang dilakukan oleh Ramadhan dan Haq (2021) berfokus pada pengembangan sistem pemantauan untuk pembangkit listrik tenaga surya (PLTS) yang memanfaatkan teknologi *Internet of Things* (IoT) dengan tujuan untuk meningkatkan efisiensi dan keandalan dalam pemantauan kinerja sistem energi terbarukan. Dalam penelitian ini, penulis mengimplementasikan protokol komunikasi MQTT yang memungkinkan interaksi antara perangkat yang berfungsi sebagai penerbit (publisher) dan server *cloud* sebagai pelanggan (*subscriber*). Sistem yang gunakan menggunakan ESP32 sebagai mikrokontroler yang bertugas mengumpulkan data dari sensor INA219 dan PZEM004t, yang digunakan untuk memantau produksi energi dan konsumsi daya. Data yang diperoleh kemudian dikirim secara *real-time* ke *Google Cloud Platform*, yang berfungsi sebagai *broker* MQTT dan tempat penyimpanan basis data. Hasil penelitian menunjukkan bahwa sistem ini mampu beroperasi dengan baik, dengan waktu respons yang minimal dan akurasi pengukuran yang tinggi[6]

Penelitian ini berfokus pada pengembangan *backend real-time* berbasis Node.js, Express.js, dan PostgreSQL, dengan integrasi protokol MQTT untuk menerima data sensor secara langsung dari perangkat IoT, serta menyimpan dan mengelola data tersebut dalam basis data untuk kebutuhan *monitoring*. Sementara itu, penelitian yang dilakukan Ramadhan mengembangkan sistem *monitoring* berbasis

IoT untuk pembangkit listrik tenaga surya menggunakan protokol komunikasi MQTT *Publisher/Subscriber* yang diimplementasikan pada *Google Cloud Platform* (GCP), dengan ESP32 sebagai pengendali mikro dan *database* MySQL yang terletak pada server virtual GCP.

Selanjutnya terdapat penelitian sebelumnya dilakukan oleh Preston Porter dari 2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems Workshops (MASSW) dengan judul "*The Design and Implementation of a RESTful IoT Service Using the MERN Stack*[7]". Penelitian ini fokus pada tantangan yang dihadapi oleh *cloud* dalam memberikan respons *real-time* kepada perangkat *Internet of Things (IoT)* yang semakin banyak terhubung ke internet, mendesain sebuah layanan khusus bernama "*RESTful IoT service*" dengan menggunakan teknologi *MERN stack* (Mongodb, Express.js, React.js, Node.js). Layanan ini berperan sebagai penghubung antara perangkat IoT, seperti sensor nirkabel, dan *Cloud*, memfasilitasi komunikasi yang lebih cepat dan efisien. Dengan mengadopsi model *Representational State Transfer (REST)*, sistem ini menunjukkan kinerja yang efisien dalam kecepatan respons dan adaptabilitasnya.

Perbedaan antara penelitian Preston Porter dan penelitian ini terletak pada fokus dan pendekatan teknologi yang digunakan. Penelitian Preston Porter berpusat pada respons *real-time* terhadap perangkat *Internet of Things (IoT)* dengan mengembangkan layanan bernama "*RESTful IoT Service*" menggunakan teknologi *MERN Stack*. Tujuan utamanya adalah untuk mempercepat dan meningkatkan efisiensi komunikasi serta menawarkan adaptabilitas yang tinggi. Sementara itu, penelitian ini mengimplementasikan teknologi *PERN Stack* (PostgreSQL, Express.js, React.js, Node.js) untuk memvisualisasikan data yang diperoleh dari perangkat IoT Buoy U-TEWS. Protokol MQTT digunakan dalam penelitian ini dengan fokus pada efisiensi dalam penerimaan data ke server *database*.

Penelitian yang dilakukan oleh Agus Rianto (2020) menggunakan pendekatan berbasis IoT dengan sensor ultrasonik HC-SR04 dan mikrokontroler Wemos D1 untuk memantau ketinggian air secara *real-time*[8]. Penelitian ini mengandalkan platform Thingier.io untuk memvisualisasikan data dan menyimpan hasil pengukuran secara *real-time*, yang diaplikasikan untuk mitigasi banjir di daerah

rawan.

Perbedaan penelitian ini dengan penelitian Agus Rianto adalah fokusnya pada pengembangan *backend real-time* untuk sistem IoT menggunakan Node.js, Express.js, dan PostgreSQL, dengan integrasi MQTT untuk menerima data langsung dari perangkat Buoy U-TEWS. Sementara itu, penelitian Agus Rianto berfokus pada pengukuran ketinggian air menggunakan perangkat sederhana dan visualisasi data melalui platform bawaan seperti Thinger.io. Penelitian ini menonjolkan kemampuan membangun sistem *monitoring real-time* yang lebih kompleks dan fleksibel, termasuk visualisasi data yang sepenuhnya disesuaikan melalui *dashboard*, berbeda dengan pendekatan perangkat IoT sederhana dan platform *cloud* bawaan dalam penelitian Agus Rianto.

2.2 Buoy U-TEWS

U-TEWS adalah singkatan dari *Unila-Tsunami Early Warning System* merupakan alat yang diimplementasikan di perairan sekitar Pulau Sebesi dengan fokus pengukuran tinggi muka air laut, namun memiliki pengukuran lain seperti arah mata angin dan kecepatan angin, serta dilengkapi GPS. Perangkat ini bekerja dengan cara mengukur tekanan air laut. Buoy U-TEWS dilengkapi dengan sensor khusus yang dapat mengukur tekanan air laut, kecepatan angin, serta arah angin. Data yang dihasilkan perangkat ini sangat penting sebagai monitor bagaimana keadaan perairan di sekitar Pulau Sebesi.



Gambar 2. 1 Perangkat Buoy U-TEWS

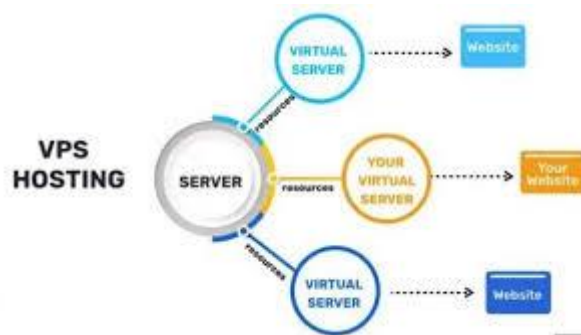
[sumber: dokumentasi pribadi]

2.3 Full-stack Developer

Fullstack developer adalah seorang pengembang perangkat lunak yang memiliki keahlian dalam menangani baik sisi *frontend* (antarmuka pengguna) maupun *backend* (logika server dan *database*) dari sebuah aplikasi atau *website*. Dengan kata lain, seorang *fullstack developer* dapat mengembangkan keseluruhan aplikasi dari ujung ke ujung. Hal ini mencakup desain dan implementasi tampilan pengguna di sisi *frontend* menggunakan teknologi seperti HTML, CSS, dan JavaScript. Di sisi *backend*, pengembangan fokus pada pengelolaan *database* dan pembuatan API (*Application Programming Interface*) yang berfungsi sebagai penghubung antara *frontend* dan *backend*, memastikan bahwa data dapat diakses, diproses, dan disampaikan dengan efisien dan aman[9]

2.4 Lingkungan Server

Untuk pembuatan dan menjalankan aplikasi layanan dalam *backend*. Penelitian ini menggunakan *Virtual Private Server* (VPS).



Gambar 2. 2 *Virtual Private Server*[10]

Virtual Private Server adalah server pribadi yang keseluruhan *resource* digunakan oleh satu *user* saja dan tidak dipengaruhi oleh *user* lainnya. Teknologi yang digunakan VPS adalah virtualisasi *hardware* server fisik yang kemudian dibagi menjadi beberapa *resource* berbeda. Disebut virtual karena pada pembagian ini menggunakan *software* sehingga dalam satu server fisik terdapat beberapa VPS yang dijalankan[11].

Teknologi virtualisasi server yang dikenal dengan VPS (*Virtual Private Server*) saat

ini banyak digunakan oleh pengguna untuk berbagai keperluan, antara lain penggunaan pribadi, profesional, perkantoran, dan bisnis. Karena menyewa VPS lebih murah daripada membeli atau menyewa server khusus, banyak orang menggunakannya untuk berbagai keperluan[12].

2.5 Teknologi Pembuatan *Website*

Untuk pembuatan *website* dalam penelitian ini. Peneliti menggunakan teknologi yaitu teknologi *PERN Stack*. *PERN* adalah singkatan dari PostgreSQL, Express.js, React.js, dan Node.js, yang merupakan kumpulan teknologi untuk membangun aplikasi web fullstack secara efisien. Dengan kombinasi ini, *PERN* menjadi pilihan populer untuk membangun aplikasi web yang modern, skalabel, dan berperforma tinggi.

2.5.1 PostgreSQL

PostgreSQL adalah sistem manajemen basis data relasional sumber terbuka yang terkenal karena keandalan, fleksibilitas, dan kinerja tinggi. Dikembangkan pertama kali pada tahun 1986 sebagai bagian dari proyek POSTGRE di Universitas California, Berkeley, PostgreSQL mendukung berbagai jenis data dan fitur tingkat lanjut seperti transaksi ACID (*Atomicity, Consistency, Isolation, Durability*), sistem indexing yang kuat, dan ekstensibilitas.

PostgreSQL dapat digunakan sebagai aplikasi manajemen *database* dengan aplikasi bawaannya *pgAdmin*. PostgreSQL merupakan database relasional yang mendukung perintah SQL maupun PL/pgSQL (*Procedural Language*)[13]

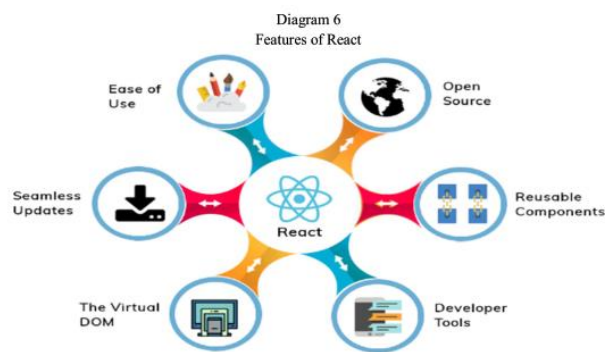
PostgreSQL dipilih dalam penelitian ini sebagai *database* penyimpanan data dikarenakan beberapa kelebihan dari PostgreSQL yaitu *Open-source*, skalabilitas nya yang baik, mudah dikonfigurasi, dapat menangani volume data yang besar, dan lebih cepat dibandingkan beberapa *Database Management System* (DBMS) yang lain seperti MySQL atau MongoDB [14].

2.5.2 Express.js

Express.js adalah *framework web* minimalis untuk Node.js yang dirancang untuk membangun aplikasi web dan API dengan cepat dan mudah. Sebagai salah satu framework paling populer di ekosistem Node.js, Express menyediakan berbagai fitur untuk menangani rute, *middleware*, permintaan HTTP, dan manajemen respons. Express memungkinkan pengembang untuk membuat aplikasi server yang efisien dan dapat disesuaikan dengan sedikit kode, serta mendukung berbagai format data, seperti JSON, HTML, dan XML. Dengan menggunakan Express, pengembang dapat mengelola rute aplikasi, mengatur parameter dan kueri, serta menghubungkan aplikasi ke basis data dengan lebih mudah. Kelebihan utama dari Express adalah fleksibilitasnya yang memungkinkan pengguna untuk menambahkan *middleware* untuk memproses permintaan dan *respons*, serta kemudahan dalam membangun aplikasi yang skalabel, seperti aplikasi API RESTful dan aplikasi *real-time*.

2.5.3 React

React adalah *open-source* library JavaScript deklaratif, efisien dan fleksibel untuk membangun antarmuka pengguna.



Gambar 2. 3 React[15]

React memungkinkan untuk membuat *user interface* yang kompleks dengan set kode kecil yang terisolasi yang disebut "komponen". React JS ini digunakan untuk menangani lapisan tampilan dalam aplikasi satu halaman dan pengembangan *mobile application*. React JS dikelola oleh facebook, instagram, komunitas

pengembang dan korporasi. React berusaha untuk memberikan kecepatan, kesederhanaan, dan skalabilitas. Beberapa fitur yang paling mencolok adalah JSX, Komponen Stateful, Model Objek Dokumen Virtual[16].

2.5.4 Node.js

Node.js adalah platform yang dibangun di atas mesin JavaScript V8 milik Google, yang memungkinkan pengembang untuk menjalankan JavaScript di sisi server. Dengan arsitektur *non-blocking* dan *event-driven*, Node.js sangat efisien dalam menangani banyak koneksi secara bersamaan, menjadikannya pilihan yang ideal untuk aplikasi *real-time*.

Dalam penggunaan Node.js, pengembang menulis kode JavaScript yang dapat dijalankan di sisi *server*, mengakses sistem file, dan melakukan tugas lainnya yang memerlukan akses ke sumber daya sistem.

Selain itu, kemampuan Node.js untuk berintegrasi dengan berbagai API, seperti yang ditunjukkan dalam penelitian Fauzi mengenai integrasi API *Google Classroom*, menunjukkan bahwa Node.js dapat berfungsi sebagai jembatan antara berbagai layanan dan aplikasi[17].

2.6 Komunikasi *Interface*

Dalam membangun *komunikasi interface*, Peneliti menggunakan *REST API* (*Representational State Transfer API*) yang merupakan arsitektur yang digunakan untuk membangun antarmuka komunikasi antara aplikasi, biasanya berbasis web, dengan menggunakan protokol HTTP. REST API memungkinkan klien (seperti aplikasi *frontend*) dan server (seperti *backend*) untuk bertukar data melalui metode HTTP standar seperti *GET*, *POST*, *PUT*, dan *DELETE* [18]. Arsitektur ini mengikuti prinsip REST, termasuk *statelessness* (tidak menyimpan status sesi pada server), representasi sumber daya melalui URL, dan penggunaan format data seperti JSON atau XML.

REST API dikenal karena kesederhanaannya, skalabilitasnya, dan kemampuannya

untuk diintegrasikan dengan berbagai sistem. Contohnya, REST API digunakan untuk mengambil data dari *database*, mengirimkan formulir, atau menghubungkan aplikasi ke layanan pihak ketiga. Fleksibilitas dan kompatibilitasnya menjadikannya pilihan populer dalam pengembangan aplikasi web, *mobile*, dan IoT.

2.7 Teks Editor

Dalam pembuatan dan penulisan kode program dalam penelitian ini. Penulis menggunakan *Visual Studio Code* yang merupakan editor teks gratis dan *open-source* yang dikembangkan oleh Microsoft. Visual Studio Code memungkinkan pengguna untuk menulis kode pada berbagai bahasa pemrograman, termasuk HTML, CSS, JavaScript, PHP, Python, dan lain-lain. VS Code sering digunakan dalam berbagai proyek pengembangan perangkat lunak, baik untuk aplikasi web maupun desktop. Misalnya, dalam pengembangan sistem informasi berbasis web atau aplikasi berbasis PHP dan JavaScript, VS Code menjadi pilihan utama karena kemampuannya untuk mendukung berbagai teknologi dan bahasa pemrograman[19].

Visual Studio Code memiliki beberapa kelebihan yang membuatnya populer di kalangan pengembang perangkat lunak. Kelebihan-kelebihan tersebut antara lain adalah :

1. Gratis dan *open-source*
2. Ringan dan cepat
3. Mudah digunakan
4. Mendukung banyak bahasa pemrograman
5. Memiliki banyak ekstensi dan *plugin*

2.8 API Tester

Dalam pengujian API di penelitian ini, peneliti menggunakan *software* Postman yaitu sebuah alat yang digunakan oleh para pengembang perangkat lunak untuk berinteraksi dengan API (*Application Programming Interface*). API seperti pelayan

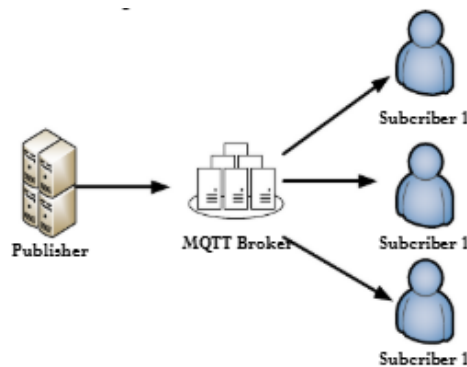
di restoran, yang mengambil pesanan (permintaan data atau fungsi) dan memberikan balasan dari dapur (sistem atau aplikasi).

Dengan Postman, pengembang bisa menguji API dengan mengirimkan permintaannya dan melihat apa yang dikembalikan. Misalnya, jika ingin melihat data pengguna dari sebuah layanan, bisa "meminta" data tersebut melalui Postman, dan layanan akan "memberikan" data yang diminta.

Postman sendiri kemudian digunakan untuk mengirim permintaan ke API. “Permintaan dapat mengambil (*GET*), menambah (*POST*), menghapus (*DELETE*), atau memperbarui data (*PUT*)”. Oleh karena itu, Postman merupakan perangkat lunak pengujian API yang canggih, kuat, dan serbaguna. Memiliki infrastruktur pengujian berkelanjutan yang sepenuhnya fleksibel untuk API sangat memungkinkan dengan adanya Newman sebagai tambahan.[20]

2.9 Protokol Komunikasi

Untuk protokol komunikasi yang menghubungkan antara hasil pembacaan nilai sensor dengan *backend*. Peneliti menggunakan protokol MQTT (*Message Queuing Telemetry Transport*) yang merupakan sebuah protokol standar pada lapisan aplikasi untuk ekosistem Internet of Things (IoT).



Gambar 2. 4 Message Queue Telemetry Transport (MQTT)[21]

Protokol ini menggunakan mekanisme *publish/subscribe*, yang mengorganisir sekelompok klien di sekitar server yang disebut broker, yang bertugas untuk mendistribusikan pesan antara klien[22].

MQTT berfungsi sebagai penghubung antara penerbit (*publisher*) dan pelanggan (*subscriber*) melalui topik. Topik ini bisa dianggap sebagai saluran di mana klien dapat mendengarkan atau mengirim data. Ketika data diterbitkan oleh klien pada saluran tertentu, semua klien yang berlangganan pada saluran tersebut akan menerima data tersebut. Broker MQTT mengelola semua penerbit dan pelanggan, serta menangani semua rincian pengiriman data antara klien secara end-to-end.

2.10 Web Hosting

Agar *website* dapat diakses oleh semua orang, peneliti menggunakan Layanan *web hosting* yaitu Niagahoster. Niagahoster adalah salah satu penyedia layanan *hosting web* terkemuka di Indonesia. Mereka menyediakan berbagai solusi untuk kebutuhan hosting, domain, dan layanan terkait lainnya untuk website atau aplikasi *online*. Niagahoster dikenal karena kemudahan penggunaan, dukungan pelanggan yang baik, dan layanan yang terjangkau.

Keunggulan utama Niagahoster terletak pada harga yang terjangkau, dukungan pelanggan 24/7, serta fitur keamanan seperti SSL gratis dan perlindungan DDoS. Layanan mereka juga menyediakan kecepatan dan stabilitas server yang tinggi, serta backup harian untuk melindungi data situs. Dengan panel kontrol yang mudah digunakan seperti cPanel dan Softaculous, Niagahoster mempermudah pengguna dalam mengelola situs mereka. Dengan berbagai fitur lengkap dan layanan yang dapat diandalkan, Niagahoster menjadi pilihan populer bagi banyak pengguna di Indonesia yang membutuhkan solusi hosting yang handal dan efisien.

2.11 Metode Penelitian yang Digunakan

Dalam pembuatan *website monitoring* pada penelitian ini. Peneliti menggunakan metode Kanban sebagai metode penelitian. Kanban adalah metode untuk mengelola dan mengendalikan proses kerja yang awalnya dikembangkan untuk digunakan dalam industri manufaktur. Namun, dengan berjalannya waktu, metode ini telah diadaptasi untuk digunakan di berbagai industri lain, termasuk pengembangan perangkat lunak, kesehatan, dan manajemen rantai pasokan. Sistem Kanban adalah

salah satu alat dalam sistem manufaktur ramping (lean) yang dapat mencapai jumlah inventaris minimum pada satu waktu [23].

Beberapa fitur kunci dari Kanban adalah:

1. Manajemen Visual: Kanban menggunakan sistem visual untuk mengelola pekerjaan. Biasanya, metode ini menggunakan papan dengan kartu atau catatan lengket untuk mewakili item pekerjaan dan statusnya.
2. Batasan Pekerjaan Dalam Proses (WIP): Salah satu prinsip utama Kanban adalah membatasi jumlah pekerjaan yang dapat sedang berlangsung pada saat tertentu. Hal ini dilakukan untuk mencegah kelebihan beban pada sistem dan meningkatkan aliran pekerjaan.
3. Sistem Berbasis Tarik: Berbeda dengan pendekatan tradisional yang mendorong pekerjaan ke dalam sistem berdasarkan jadwal atau rencana, Kanban adalah sistem berbasis tarik. Artinya, pekerjaan hanya ditarik ke dalam sistem ketika ada kapasitas untuk menanganinya.
4. Peningkatan Berkelanjutan: Kanban menekankan pada peningkatan berkelanjutan. Proses secara rutin ditinjau dan disesuaikan sesuai kebutuhan untuk meningkatkan efisiensi dan mengurangi pemborosan.

Secara keseluruhan, Kanban merupakan alat manajemen yang kuat yang tidak hanya meningkatkan efisiensi operasional tetapi juga mendorong kolaborasi tim dan peningkatan kualitas produk. Implementasi yang tepat dari metode ini dapat memberikan keuntungan kompetitif yang signifikan bagi tim.

2.12 Framework CSS

Untuk pengembangan antarmuka pengguna. Peneliti menggunakan *framework* CSS yaitu Tailwind CSS yang merupakan *framework* CSS berbasis *utility-first* yang dirancang untuk mempermudah proses pengembangan antarmuka pengguna (UI). Tidak seperti *framework* CSS konvensional yang menyediakan komponen siap pakai, Tailwind CSS menawarkan kelas-kelas utilitas kecil (*utility classes*) yang dapat langsung digunakan untuk mengatur desain dan tata letak, seperti warna, *margin*, *padding*, ukuran *font*, dan lainnya.

Setiap utilitas Tailwind memiliki tipe responsif yang memudahkan untuk membangun antarmuka responsif tanpa perlu menggunakan perintah CSS khusus. Tailwind menggunakan *prefix {screen}*: yang intuitif untuk mengidentifikasi kelas-kelas responsif dalam markup sambil tetap mempertahankan nama kelas aslinya. Ramah terhadap komponen, Tailwind menyediakan alat untuk mengekstraksi kelas komponen dari pola utilitas berulang dan memperbarui beberapa *instance* dalam sebuah komponen dari satu tempat. Tailwind ditulis dalam format PostCSS dan dikonfigurasi dalam JavaScript. Tailwind CSS menawarkan kustomisasi dan fleksibilitas yang lebih besar, sehingga lebih cocok untuk proyek-proyek yang kompleks.[24]

Selain itu, Tailwind juga ramah terhadap pembuatan komponen. *Framework* ini menyediakan alat yang memungkinkan developer untuk mengekstrak kelas-kelas utilitas yang berulang menjadi komponen yang lebih terorganisir dan dapat digunakan kembali. Misalnya, dengan menggunakan direktif *@apply*, yang bisa menggabungkan beberapa kelas utilitas menjadi sebuah komponen yang terstruktur dengan lebih baik. Ini sangat berguna ketika ingin mengubah beberapa *instance* dalam komponen yang sama dari satu tempat saja, tanpa perlu mengedit setiap elemen secara manual. Dengan demikian, Tailwind memfasilitasi pengelolaan kode yang lebih bersih dan modular, serta membantu meminimalkan duplikasi kode di seluruh proyek. Hal ini membuat pengembangan aplikasi dan situs web menjadi lebih terstruktur, mudah dipelihara, dan skalabel.

2.13 Package Builder

Untuk meningkatkan performa dalam pengembangan *web monitoring* ini. Peneliti menggunakan *build tool* yaitu Vite. Vite adalah *build tool* yang sangat populer untuk pengembangan aplikasi React karena menawarkan kecepatan, efisiensi, dan pengalaman pengembangan yang mulus. Vite mendukung React secara bawaan dan menyediakan template siap pakai untuk memulai proyek React dengan mudah.

Dengan menggunakan Vite, pengembang React dapat memanfaatkan keunggulan seperti *instant server start*, *hot module replacement* (HMR) yang sangat cepat, serta

waktu *build* produksi yang singkat.

Keunggulan ini sangat penting ketika menangani aplikasi besar dan kompleks, karena dapat mempercepat proses *build* dan mengurangi waktu tunggu. Selain itu, Vite mendukung berbagai fitur modern, termasuk TypeScript, JSX/TSX, dan berbagai ekstensi atau plugin yang kompatibel dengan ekosistem React, seperti *React Refresh* untuk memastikan pengalaman pengembangan yang lebih produktif. Dengan *React Refresh*, pengembang dapat dengan mudah melihat pembaruan komponen React secara langsung, tanpa kehilangan status aplikasi, yang sangat membantu dalam pengembangan aplikasi dinamis dan interaktif. Vite juga memungkinkan penggunaan berbagai *plugin* ekosistem React lainnya, seperti alat untuk optimasi, pengelolaan dependensi, atau integrasi dengan alat pengujian, membuatnya menjadi pilihan yang sangat fleksibel dan efisien untuk pengembangan aplikasi *React modern*.

BAB III

METODOLOGI PENELITIAN

3.1 Waktu dan Tempat Penelitian

Penelitian ini dilaksanakan dalam waktu 6 bulan, dimulai dari bulan November 2024 sampai April 2025. Tahapan penelitian meliputi studi literatur, setting VPS, pembuatan *database*, pengembangan *backend*, seminar proposal, pembuatan *frontend*, integrasi *backend* dan *frontend*, pengujian dan *debugging*, penulisan laporan, seminar hasil, dan seminar komprehensif. Proses penelitian ini dilakukan di beberapa tempat yaitu:

1. Gedung Baja, Jurusan Teknik Sipil, Fakultas Teknik, Universitas Lampung.
2. Laboratorium Terpadu Teknik Elektro, Jurusan Teknik Elektro, Fakultas Teknik Universitas Lampung.
3. Pulau Sebesi, Kecamatan Rajabasa, Kabupaten Lampung Selatan.

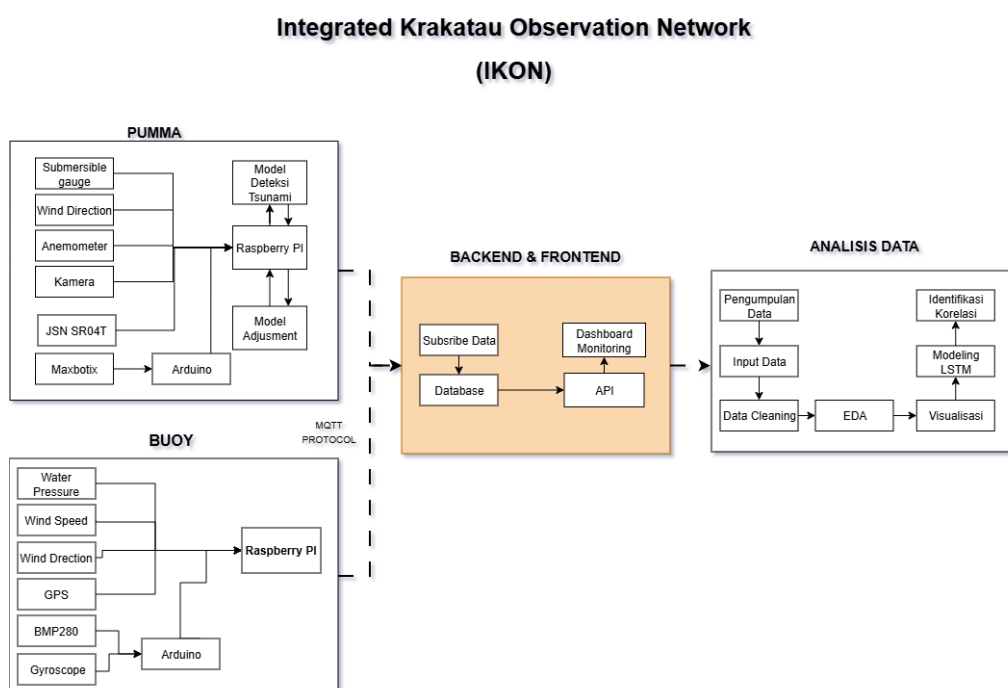
3.2 Capstone Project

Penelitian ini berkaitan dengan berbagai penelitian sebelumnya yang berkolaborasi untuk mengembangkan sistem perangkat Buoy U-TEWS yang diletakkan di daerah perairan Pulau Sebesi. Dalam proyek yang akan dilaksanakan, terdapat beberapa bagian yang akan dikerjakan secara terpisah.

Penulis merupakan anggota tim *capstone project* IKON (*Integrated Krakatau Observation Network*) dan terlibat langsung dalam subproyek BUOY. Proyek IKON merupakan inisiatif penelitian kebencanaan tsunami yang bertujuan mengembangkan sistem pemantauan dini tsunami berbasis IoT untuk kawasan Selat Sunda. Proyek ini merupakan kelanjutan dari penelitian sebelumnya yang dikenal

sebagai *Krakatoa Research Center* (KRC) atau dikenal *K-Project* dan melibatkan kolaborasi dengan PT Drone Nirwana Bentala serta Unila Robotika dan Automasi. Tim yang tergabung dalam capstone project ini dikenal dengan U-TEWS (*Unila-Tsunami Early Warning Sistem*).

Bagian *hardware* perangkat Buoy U-TEWS dikerjakan oleh dua anggota tim, selanjutnya untuk *software* integrasi dikerjakan oleh dua anggota tim, untuk analisis data dikerjakan satu anggota tim, dan untuk *backend* dan *frontend* dikerjakan oleh penulis.



Gambar 3. 1 Diagram keseluruhan proyek perangkat Buoy U-TEWS

[sumber: dokumentasi pribadi]

Pada gambar 3.1 merupakan diagram keseluruhan proyek perangkat buoy U-TEWS, terdapat berbagai tugas yang dibagi kepada anggota tim, masing-masing sesuai dengan peran dan tanggung jawab mereka. Kotak yang diberi warna merah pada diagram menunjukkan tugas-tugas yang akan dikerjakan secara spesifik dalam penelitian ini, yaitu mencakup pengembangan pada bagian *backend* dan *frontend*.

Backend dalam proyek ini memiliki peran penting sebagai penghubung antara data hasil pembacaan sensor dengan berbagai komponen lainnya dalam sistem. Fungsi

utama *backend* meliputi pengambilan data secara langsung dari protokol komunikasi MQTT, pengolahan dan penyimpanan data tersebut ke dalam *database*, serta pembuatan API yang memungkinkan data tersebut diteruskan dan diakses oleh bagian *frontend*. Selain itu, *backend* dirancang untuk menangani berbagai kebutuhan sistem, seperti memastikan keamanan, performa, dan integritas data dalam proses transmisi maupun penyimpanan.

Sementara itu, bagian *frontend* akan berfokus pada visualisasi data yang diperoleh dari perangkat Buoy U-TEWS. Visualisasi ini bertujuan untuk menyajikan data secara interaktif dan mudah dipahami oleh pengguna, sehingga memungkinkan mereka untuk memantau kondisi secara *real-time*. Selain itu, *frontend* juga dirancang agar memiliki antarmuka pengguna yang responsif, ringan, dan dapat diakses pada berbagai perangkat, sehingga memberikan pengalaman pengguna yang optimal.

Dengan pembagian tugas yang jelas ini, penelitian diharapkan dapat menghasilkan sistem yang terintegrasi dengan baik, mulai dari pengambilan data sensor hingga penyajian informasi kepada pengguna dalam bentuk visualisasi yang informatif dan akurat.

3.3 Alat dan Bahan

3.3.1 Alat

Alat yang digunakan dalam pengerjaan penelitian dan pengembangan sistem yang dibuat adalah sebagai berikut:

Tabel 3.1 Alat berupa Hardware dan Software yang digunakan.

No	Perangkat	Spesifikasi	Kegunaan	Keterangan
1	Laptop	Processor Intel Core i5 1135G7, RAM 16GB, SSD 512GB, Sistem Operasi	Perangkat pembuatan dan pengujian sistem.	Perangkat keras yang digunakan sebagai <i>compiler</i> dalam

		Windows		pemrograman
2	<i>Virtual Private Server (VPS)</i>	Penyedia: Niagahoster.co.id, Lokasi: Singapore, 3 CPU Core, RAM 3GB, Storage 60GB.	<i>Virtual private server</i> yang digunakan sebagai pusat layanan sistem yang akan dibuat.	Diakses melalui <i>Secure Shell (SSH)</i>
3	MQTT Explorer	MQTT Explorer Versi 0.4.0.0	<i>Software</i> yang digunakan untuk pengujian broker mqtt dalam mengirim dan menerima data	Terinstall di Laptop
4	<i>Visual Studio Code</i>	VS Code versi 1.95.1.0	<i>Software</i> yang digunakan dalam penulisan kode program untuk pengambilan dan penyimpanan data dari sensor	Terinstall di Laptop
3	PostgreSQL	PostgreSQL <i>database</i> versi 9.2.24	Program <i>database</i> yang digunakan	Terinstall di VPS

			dalam penyimpanan data dari sensor	
4	Postman	Postman V.11.18.10	Program yang digunakan untuk mencoba Endpoint API	Terinstall di Laptop
5	Node.js & NPM	Node.js versi v16.19.0. NPM versi 8.19.3	Aplikasi yang digunakan untuk menjalankan kode program untuk penyimpanan data	Terinstall di VPS
6	Notion	Notion Desktop version	Aplikasi yang digunakan untuk visualisasi metode Kanban dalam pengembangan sistem	Terinstall di Laptop
7	pgAdmin 4	pgAdmin 4 versi 0.77.0.0	memudahkan pengelolaan database PostgreSQL	Terinstall di Laptop

			tanpa perlu menggunakan baris perintah	
7	Niaga Hoster	Website Hosting	Aplikasi yang digunakan untuk melakukan publikasi website	Terinstall di provider
10	Vite	<i>Build tool</i> ringan untuk Reactjs.	Mempercepat pengembangan <i>frontend</i> .	Menawarkan pengembangan cepat dengan HMR.

3.3.2 Bahan

Bahan yang digunakan dalam pengerjaan penelitian dan pengembangan sistem Buoy U-TEWS ini berupa data yang didapatkan dari sensor. Data yang ada dalam pemantauan tersebut adalah sebagai berikut :

Tabel 3.2 Bahan berupa data sensor serial dari raspberry pi.

NO	Data yang diterima	Keterangan
1	timestamp	Data yang menunjukkan tanggal dan waktu
2	xacc	Data yang menunjukkan percepatan ke depan atau belakang
3	yacc	Data yang menunjukkan percepatan ke kanan atau kiri

4	zacc	Data yang menunjukkan percepatan ke atas atau bawah
5	xgyro	Mengukur rotasi atau kecepatan sudut pada sumbu X
6	ygyro	Mengukur rotasi atau kecepatan sudut pada sumbu Y
7	zgyro	Mengukur rotasi atau kecepatan sudut pada sumbu Z
8	xangle	Miring ke depan atau belakang.
9	yangle	Miring ke kanan atau kiri.
10	zangle	Memutar ke kiri atau kanan di tempatnya.
11	temperature	Mengukur suhu
12	pressure	Mengukur tekanan
13	altitude	pengukuran ketinggian relatif terhadap permukaan laut

Tabel 3. 3 Bahan berupa dari Modbus

NO	Data yang diterima	Keterangan
1	timestamp	Data yang menunjukkan waktu
2	water_column_height	Data yang menunjukkan tinggi kolom air laut

3	AnemometerSpeed	Data yang menunjukkan kecepatan angin
4	Beaufort_scale	mengklasifikasikan angin berdasarkan efeknya terhadap lingkungan, tanpa mengukur kecepatan angin secara langsung.
5	Angle	Data yang menunjukkan sudut
6	Direction	Data yang menunjukkan arah angin

Tabel 3. 4 Bahan berupa dari scc

NO	Data yang diterima	Keterangan
1	timestamp	Data yang menunjukkan waktu
2	pv_voltage	Data yang menunjukkan tegangan panel surya
3	pv_current	Data yang menunjukkan arus panel surya
4	pv_power	Data yang menunjukkan daya panel surya.
5	battery_voltage	Data yang menunjukkan tegangan baterai
6	battery_charge_current	Data yang menunjukkan arus <i>charge</i> baterai
7	device_current	Data yang menunjukkan arus <i>device</i>
8	device_power	Data yang menunjukkan daya <i>device</i>

Tabel 3. 5 Bahan berupa dari GPS

NO	Data yang diterima	Keterangan
1	timestamp	Data yang menunjukkan waktu
2	latitude	Data yang menunjukkan garis bujur
3	longitude	Data yang menunjukkan garis lintang
4	speed	Data yang menunjukkan kecepatan <i>device</i>
5	true_course	Data yang menunjukkan arah atau jalur yang ditempuh <i>device</i>

Tabel 3. 6 Bahan berupa dari cpu

NO	Data yang diterima	Keterangan
1	timestamp	Data yang menunjukkan waktu
2	cpu_usage	Data yang menunjukkan penggunaan cpu
3	mem_gpu	Data yang menunjukkan <i>memory</i> gpu
4	mem_arm	Data yang menunjukkan <i>memory</i> arm
5	temp	Data yang menunjukkan suhu cpu
6	total_space	Data yang menunjukkan total penyimpanan
7	used_space	Data yang menunjukkan penyimpanan yang digunakan

8	free_space	Data yang menunjukkan sisa penyimpanan
---	------------	--

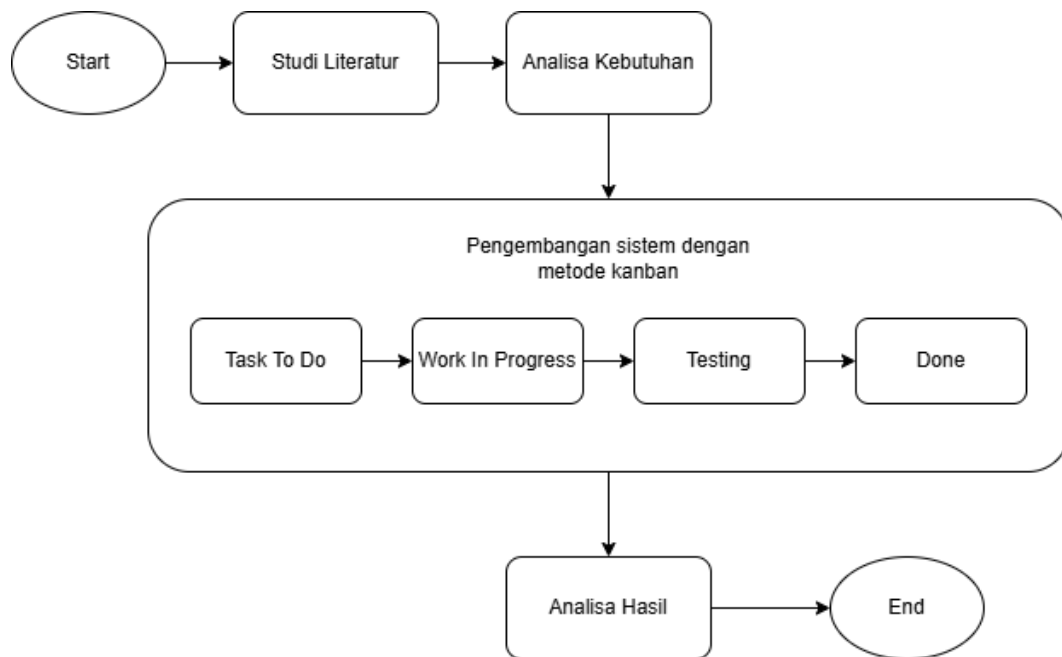
3.4 Tahapan Penelitian

Penelitian ini dilakukan secara bertahap untuk memastikan hasil yang terstruktur dan optimal. Tahapan dimulai dengan studi literatur di Bab 2 untuk memahami konsep, metode, dan teknologi yang mendukung pengembangan sistem Buoy U-TEWS. Selanjutnya, dilakukan analisis kebutuhan untuk mengidentifikasi apa saja yang diperlukan.

Proses pengembangan dilakukan menggunakan metode Kanban, yang membantu memvisualisasikan alur kerja, mengelola prioritas, dan mengontrol jumlah pekerjaan dalam proses. Backend dikembangkan dengan Node.js, Express.js, dan PostgreSQL, sementara frontend menggunakan React.js untuk menghasilkan antarmuka yang responsif dan ringan.

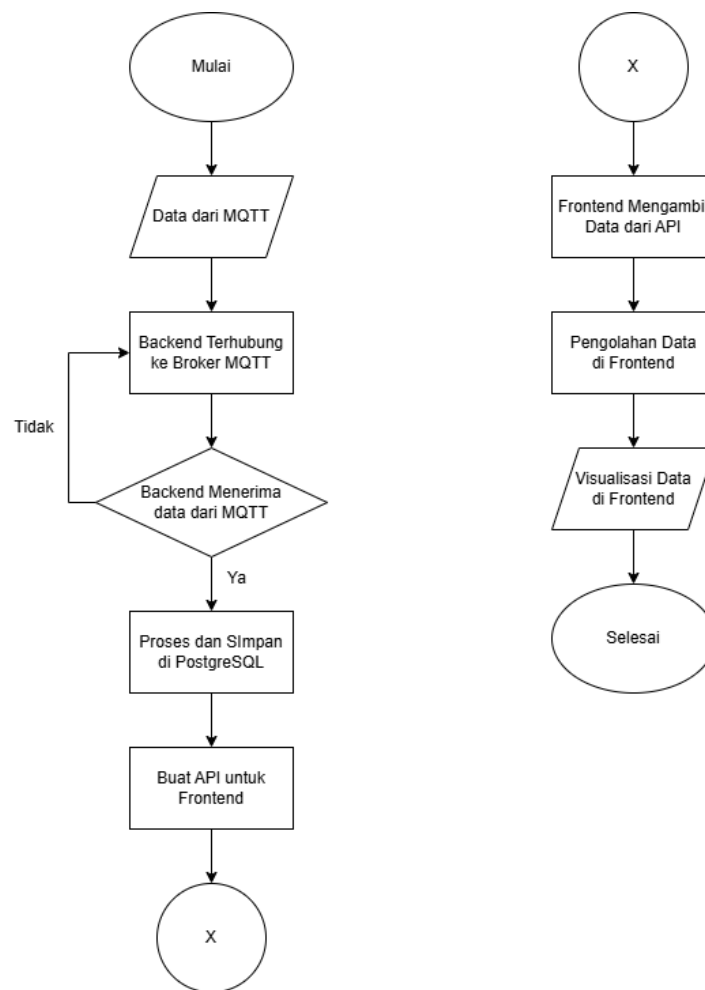
Tahap akhir adalah evaluasi hasil, meliputi pengujian sistem, validasi data buoy, dan pengumpulan umpan balik dari pengguna untuk menyempurnakan sistem.

Berikut ini adalah diagram alur yang menggambarkan tahapan penelitian yang telah dilakukan:



Gambar 3. 2 Diagram alur tahapan penelitian Web Buoy U-TEWS

3.5 Tahapan Visualisasi Data



Gambar 3. 3 Flowchart visualisasi data

Tahap visualisasi data merupakan suatu tahapan yang dimulai dari data yang dikirimkan melalui MQTT hingga data tersebut dapat ditampilkan pada *dashboard*. Dimana tahapan pertama yaitu hasil pembacaan sensor akan dikirimkan melalui MQTT, kemudian tahapan ke dua yaitu *backend* memasukkan topik untuk *subscribe* ke MQTT, kemudian pada tahapan ke tiga *backend* melakukan penyesuaian *message* yang dikirimkan oleh sensor sesuai yang ada di MQTT, jika berhasil maka data tersebut akan diolah dan disimpan dalam database PostgreSQL, setelah data disimpan dalam *database* maka *backend* membuat API agar dapat terhubung dengan *frontend*.

Setelah API dibuat, maka *frontend* dapat mengambil data berdasarkan dengan *end*

point yang dikirimkan dari *backend*. Setelah itu, data dari *backend* diolah di *frontend* dan divisualisasikan sesuai dengan *end point* dari *backend*.

3.6 Studi Literatur

Pada tahap studi literatur, peneliti berfokus untuk mencari, mengumpulkan, dan menganalisis referensi-referensi yang relevan dengan penelitian terkait pengembangan *dashboard* sistem *monitoring real-time* untuk perangkat Buoy U-TEWS menggunakan teknologi PERN stack. Proses ini dimulai dengan memahami latar belakang permasalahan yang melandasi penelitian, termasuk pentingnya mitigasi risiko bencana tsunami di wilayah Selat Sunda. Pemahaman ini memberikan gambaran yang lebih mendalam tentang isu yang diangkat sekaligus menjadi dasar untuk mengidentifikasi kebutuhan penelitian.

Dari pemahaman tersebut, peneliti menyusun berbagai kebutuhan penelitian yang mencakup teknologi, metodologi, serta perangkat pendukung. Beberapa kebutuhan utama meliputi penyimpanan data menggunakan PostgreSQL, pengelolaan data sensor dengan protokol MQTT, pembangunan *backend* dengan Node.js dan Express.js, serta pengembangan frontend interaktif menggunakan React.

Selain itu, literatur yang dikumpulkan mencakup penelitian-penelitian terdahulu yang relevan, seperti pengembangan sistem IoT berbasis *cloud*, implementasi protokol MQTT untuk komunikasi data, serta integrasi teknologi *full-stack* dalam visualisasi data *real-time*. Referensi-referensi ini digunakan untuk mendukung setiap elemen penelitian, baik dari segi teknis maupun konsep.

Dengan pendekatan sistematis ini, tahap studi literatur menjadi fondasi yang kokoh untuk mendukung pengembangan sistem *monitoring real-time* yang inovatif dan efektif. Referensi yang dikumpulkan memberikan wawasan mendalam untuk mengatasi tantangan teknis sekaligus memastikan penelitian ini memberikan kontribusi nyata terhadap mitigasi risiko bencana tsunami melalui visualisasi data yang komprehensif dan akurat.

3.7 Analisa Kebutuhan

Berdasarkan gambar 3.2, langkah selanjutnya dalam penelitian ini adalah analisa kebutuhan. Analisa kebutuhan diperlukan setelah mempelajari latar belakang yang menjadi dasar penelitian serta literasi yang didapat. Berdasarkan studi pustaka yang telah dilakukan dan melihat kondisi sistem Buoy U-TEWS yang saat ini telah berjalan, maka diperlukan adanya penyimpanan data yang mendukung untuk penelitian pada alat Buoy U-TEWS ini.

Adapun setelah dilakukan analisa terhadap kebutuhan dari sistem Buoy U-TEWS serta terhadap tujuan dari penelitian ini, didapat beberapa kebutuhan yang diperlukan untuk mendukung penyimpanan data dari Buoy U-TEWS ini, berdasarkan hal tersebut analisa kebutuhan yang diperlukan bisa disebut sebagai “tugas”, dapat dilihat dari data seperti berikut :

1. Pembuatan *backend* untuk mengolah serta mengatur data yang akan keluar atau masuk pada *database*.
2. MQTT Broker yang dapat digunakan sebagai protokol pengiriman data dari sensor.
3. *Virtual Private Server* (VPS) yang berfungsi untuk *broker* MQTT, menyimpan data pada *database*, serta membuat *endpoint* pada API.
4. Pembuatan *database* pada server dalam hal ini, menggunakan PostgreSQL PGAdmin 4 sebagai tempat untuk mengkonfigurasi *database*.
5. Pembuatan *collection* pada database berupa data *table* dengan nama “serial” untuk sensor serial dari alat Buoy.
6. Pembuatan *collection* pada database berupa data *table* dengan nama “scc” untuk solar charge controller pada Buoy.
7. Pembuatan *collection* pada database berupa data *table* dengan nama “gps” untuk sensor dari alat *gps* pada Buoy
8. Pembuatan *collection* pada database berupa data *table* dengan nama “modbus” untuk sensor dari modbus pada Buoy
9. Pembuatan *collection* pada database berupa data *table* dengan nama “cpu” untuk cpu pada Buoy
10. Membuat *controller* agar data dari alat yang di *publish* pada *topic*:

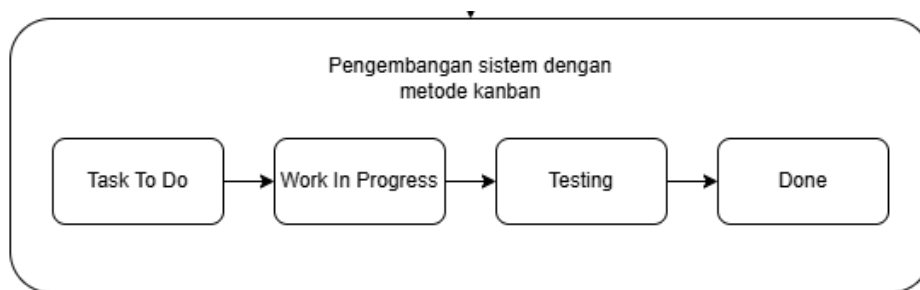
“buoyV2/serial” melalui MQTT dapat dibaca kemudian di masukan kedalam *collection database* yang bernama “serial”.

11. Membuat *controller* agar data dari alat yang di *publish* pada *topic*: “buoyV2/gps” melalui MQTT dapat dibaca kemudian dimasukkan kedalam *collection database* yang bernama “gps”.
12. Membuat *controller* agar data dari alat yang di *publish* pada *topic*: “buoyV2/scc” melalui MQTT dapat dibaca kemudian dimasukkan kedalam *collection database* yang bernama “scc”.
13. Membuat *controller* agar data dari alat yang di *publish* pada *topic*: “buoyV2/cpu” melalui MQTT dapat dibaca kemudian dimasukkan kedalam *collection database* yang bernama “cpu”.
14. Membuat *controller* agar data dari alat yang di *publish* pada *topic*: “buoyV2/modbus” melalui MQTT dapat dibaca kemudian dimasukkan kedalam *collection database* yang bernama “modbus”.
15. Membuat *routes* post untuk “buoyV2/serial” dengan nama “/serial” sehingga apabila data ingin dikirim, dapat melalui *endpoint* tersebut.
16. Membuat *routes* post untuk “buoyV2/gps” dengan nama “/gps” sehingga apabila data ingin dikirim, dapat melalui *endpoint* tersebut.
17. Membuat *routes* post untuk “buoyV2/scc” dengan nama “/scc” sehingga apabila data ingin dikirim, dapat melalui *endpoint* tersebut
18. Membuat *routes* post untuk “buoyV2/cpu” dengan nama “/cpu” sehingga apabila data ingin dikirim, dapat melalui *endpoint* tersebut
19. Membuat *routes* post untuk “buoyV2/modbus” dengan nama “/modbus” sehingga apabila data ingin dikirim, dapat melalui *endpoint* tersebut
20. Membuat dokumentasi sistem yang lengkap untuk memudahkan penggunaan dan *troubleshooting*.
21. Data yang didapat melalui *publisher* harus dapat disimpan dalam *database* dalam waktu yang lama untuk keperluan penelitian lebih lanjut.
22. Membuat data logger yang disimpan ke google drive.
23. Data yang tersimpan dalam *database* dapat ditampilkan dalam format *Javascript Object Notation* (JSON) untuk keperluan visualisasi data pada *frontend*.

24. Pembuatan *dashboard* untuk visualisasi data yang telah disimpan di dalam database agar data tersebut dapat ditampilkan.

3.8 Pengembangan Sistem

Dalam penelitian ini, metode yang digunakan adalah mengikuti metode kanban (*kanban method*). Metode Kanban dipilih karena fleksibilitasnya dalam memantau progres pekerjaan secara visual dan efisien. Metode ini membantu memastikan setiap tugas dapat diselesaikan sesuai prioritas dan waktu yang telah ditentukan. Berikut ini merupakan tahapan yang akan dilaksanakan dalam metode kanban.



Gambar 3. 4 Tahapan pengembangan sistem dengan metode kanban

Kanban memungkinkan setiap tugas yang dikerjakan untuk dipantau dengan jelas melalui empat tahapan utama, yaitu *Task To Do*, *Work in Progress*, *Testing*, dan *Done*. Setiap tahapan ini merepresentasikan status dari pekerjaan yang sedang berlangsung sehingga memudahkan tim untuk mengidentifikasi progres dan menyelesaikan pekerjaan secara sistematis.

3.8.1 Tahap Task To Do

Pada tahap "Task to Do", tugas-tugas yang telah diidentifikasi melalui analisis kebutuhan diletakkan di tahap ini sebagai persiapan. Ini menjamin bahwa tugas-tugas tersebut sudah siap dan dipahami sebelum kita mulai mengerjakannya, di samping itu, tahap ini memberi kita pemahaman yang jelas tentang apa yang perlu dilakukan selanjutnya.

Pada tahap ini, kartu diberi label berdasarkan tingkat urgensi (prioritas) untuk

menentukan tugas mana yang harus dikerjakan terlebih dahulu. Terdapat 3 macam label dalam menentukan skala prioritas:

1. *Priority-High*

Label *High* menunjukkan pentingnya tugas yang prioritas utama sehingga harus dilaksanakan. Tugas yang sangat penting perlu diselesaikan segera agar sistem dapat beroperasi dengan baik. Jika tugas ini diabaikan, sistem yang dibuat bisa jadi tidak berjalan dengan baik atau bahkan tidak dapat digunakan sama sekali.

Tabel 3. 7 Tugas *High Priority*

No	Tugas yang akan dikerjakan	Alasan
1.	Inisialisasi dan persiapan Virtual Private Server (VPS)	Persiapan server diletakkan di <i>high</i> prioritas dikarenakan VPS akan digunakan untuk membuat <i>backend</i> dalam proyek ini, jika tidak disiapkan maka sistem tidak akan berjalan dengan baik.
2.	Persiapan dan konfigurasi MQTT di VPS	Persiapan MQTT termasuk dalam <i>high</i> prioritas dikarenakan data yang akan diambil dan diproses dalam penelitian ini berdasarkan data yang dikirim sensor melalui MQTT

3.	Membuat <i>database</i>	Membuat <i>database</i> sanga diperlukan sebagai tempat penyimpanan data hasil sensor yang dikirimkan.
4.	Penulisan kode untuk menerima data dari MQTT ke <i>database</i>	Penulisan kode untuk menerima data dari MQTT ke <i>database</i> penting karena jika tidak dilakukan pengkodean maka data tidak akan diterima dan dimasukkan ke <i>database</i>
5.	Membuat REST-API HTTPS	Ini dilakukan agar data yang diterima dari sensor dapat divisualisasikan di <i>frontend</i>
6.	Koding untuk <i>Dashboard</i>	Dilakukan agar <i>Dashboard</i> dapat dijalankan
7.	Koding untuk menghubungkan <i>frontend</i> ke <i>backend</i>	Dilakukan agar data dari <i>backend</i> dapat ditampilkan di <i>dashboard</i>

2. *Priority-Medium*

Label *Medium* menunjukkan prioritas menengah atau tidak terlalu prioritas

dalam pelaksanaan tugas. Meskipun tugas dengan prioritas menengah penting, jika tugas tersebut belum dikerjakan, kinerja sistem tidak akan terganggu secara signifikan.

Tabel 3. 8 Tugas *Medium Priority*

No.	Tugas yang akan dikerjakan	Alasan
1.	Koding <i>dashboard responsive</i>	Dilakukan agar <i>dashboard</i> dapat dibuka di perangkat apa saja

3. *Priority-Low*

Label *Low* menunjukkan bahwa tugas tersebut memiliki prioritas yang rendah. Tugas dengan tingkat prioritas ini tidak wajib diselesaikan. Apabila tidak dikerjakan, sistem akan tetap berfungsi dengan baik. Namun, menyelesaikan tugas ini dapat memberikan peningkatan atau fitur tambahan yang dapat memperbaiki sistem secara keseluruhan.

Tabel 3. 9 Tugas *Low Priority*

No.	Tugas yang akan dikerjakan	Alasan
1.	Membuat <i>logger</i> data yang dikirimkan ke <i>google drive</i>	Tugas ini berfungsi untuk mengambil data dari MQTT dan menyimpan ke <i>Google Drive</i> sebagai <i>logger</i> agar ada <i>backup</i> data.

3.8.2 Tahap *Work in Progress*

Kartu tugas yang awalnya ada di bagian "*Task to Do*" akan dipindah ke tahap *Work*

In Progress saat sedang dikerjakan. Sesuai dengan prinsip Kanban yang menekankan pembatasan jumlah tugas yang berlangsung, jumlah tugas di tahap ini akan dibatasi dalam setiap alur kerjanya. Pembatasan ini bertujuan untuk mencegah *overproduction* dan pemborosan sumber daya dengan mengerjakan banyak tugas sekaligus. Dengan cara ini, peneliti dapat berkonsentrasi menyelesaikan tugas yang sedang dikerjakan sebelum memulai tugas baru.

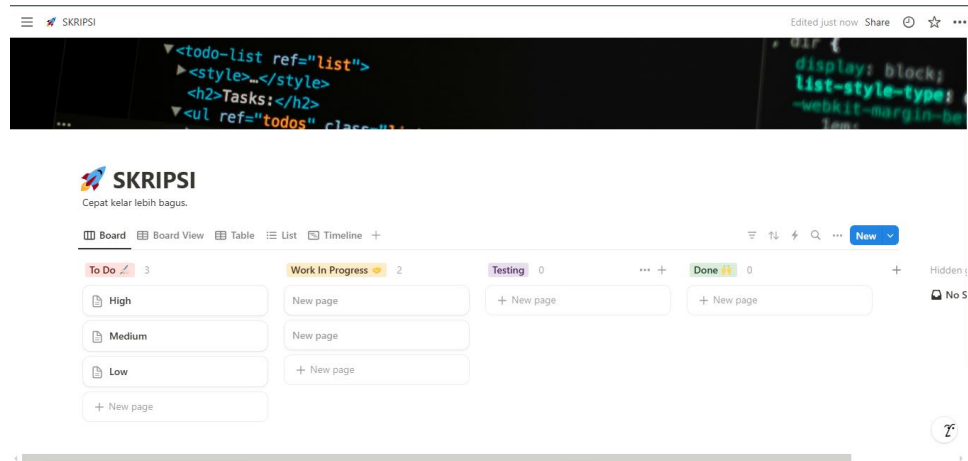
3.8.3 Tahap Testing

Pada tahap *testing* ini peneliti akan melakukan pengujian kualitas dan fungsionalitas pada *backend* dan *frontend*. Pada pengujian *backend*, penelitian ini menggunakan postman. *Postman* adalah platform pengembangan API yang populer yang memungkinkan pengembang untuk merancang, menguji, mendokumentasikan, memantau, dan mempublikasikan API mereka dengan mudah. Aplikasi ini menyediakan antarmuka pengguna yang ramah untuk mempermudah interaksi dengan API tanpa perlu menulis kode. Postman terdapat fungsi *GET*, *PUT*, *POST* dan *DELETE* untuk menguji sebuah *endpoint* pada *backend* tersebut dapat berjalan sebagaimana yang diharapkan.

Pada pengujian *frontend* menggunakan pengujian manual, dengan cara cek pada tampilan *dashboard* apakah data yang dikirimkan dari *backend* sudah terbaca atau belum.

3.8.4 Tahap Done

Pada tahap *done*, tugas yang sudah melalui tahapan-tahapan sebelumnya serta berjalan dengan baik sesuai dengan fungsionalitasnya, maka akan diletakkan pada bagian ini. Ketika tugas sudah berada pada tahap ini maka tugas bisa dikatakan selesai dan siap untuk digunakan. Pada penelitian ini *platform* yang digunakan adalah *Notion* dalam memvisualisasikan tugas yang berlangsung selama proses pengembangan sistem.



Gambar 3. 5 Tampilan kanban method dalam aplikasi Notion

[sumber: dokumentasi pribadi]

Notion dipilih oleh peneliti karena aplikasi ini menyediakan berbagai fitur yang terintegrasi, seperti membuat daftar, menjadwalkan, mengelola proyek, mencatat, dan mengelola database. Selain itu, terdapat beberapa elemen dalam Notion yang dapat diakses dengan mudah, termasuk kolom catatan, tabel, papan Kanban (papan sticky note secara daring), kalender, dan pengingat, yang semuanya dapat membantu meningkatkan kemampuan manajemen waktu.

3.9 Analisa Hasil

Tahap analisis hasil merupakan langkah penting dalam mengevaluasi keberhasilan pengembangan sistem monitoring nilai sensor pada perangkat Buoy U-TEWS. Pada tahap ini, pengujian terhadap sistem *backend* dan *frontend* yang telah dikembangkan dilakukan secara menyeluruh untuk memastikan bahwa sistem mampu berjalan sesuai dengan tujuan awal pengembangan. Proses pengembangan sistem, yang mencakup implementasi berbagai fitur dan integrasi antar komponen, telah selesai dilaksanakan, dan data hasil pengujian telah terkumpul untuk dianalisis lebih lanjut.

Dalam fase ini, hasil pengujian dilaporkan secara rinci. Laporan mencakup deskripsi *endpoint* pada sistem *backend* yang telah dirancang untuk mengelola data sensor secara *real-time*, serta *frontend* yang menyajikan data tersebut dalam

antarmuka yang ramah pengguna. Selain itu, sistem ini telah diimplementasikan pada *Virtual Private Server* (VPS) untuk memastikan ketersediaannya secara *online* dan dapat diakses melalui alamat *website* yang telah ditentukan. Informasi teknis seperti struktur endpoint, serta pengalaman pengguna dalam mengakses *website* juga menjadi bagian dari laporan.

Hasil analisis ini kemudian menjadi dasar dalam menyusun kesimpulan dan rekomendasi penelitian. Kesimpulan diambil berdasarkan keberhasilan sistem dalam mencapai tujuan awal, seperti kemampuan menyajikan data sensor secara *real-time* dan keandalan sistem dalam berbagai kondisi pengujian. Sementara itu, saran yang dihasilkan dapat memberikan panduan bagi pengembang selanjutnya, baik untuk meningkatkan performa sistem, menambahkan fitur baru, maupun melakukan integrasi dengan perangkat lain yang relevan.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan, terdapat beberapa kesimpulan yang didapat sebagai berikut:

1. Teknologi PERN Stack untuk proses pengiriman dan penerimaan data sensor secara *real-time* antara perangkat buoy U-TEWS dan *dashboard monitoring* telah berhasil diintegrasikan.
2. Berdasarkan hasil dari pengujian, dapat disimpulkan bahwa data dari sensor pada perangkat buoy U-TEWS telah berhasil diolah dan disimpan ke dalam database PostgreSQL menggunakan Node.js dengan rata rata waktu 62,812 ms.
3. Berdasarkan hasil pengujian, A P I untuk mengakses data yang telah tersimpan di *database* berhasil dibuat.
4. Berhasil mengembangkan *frontend* menggunakan react untuk menampilkan visualisasi data secara dinamis, memastikan pengalaman pengguna yang interaktif dan responsif dalam memantau perubahan ketinggian air laut secara *real-time* dengan performa 95 untuk mode desktop dan 77 untuk mobile.

5.2 Saran

Berdasarkan hasil penelitian dan evaluasi yang telah dilakukan, terdapat saran untuk penelitian selanjutnya, antara lain:

1. Untuk API disarankan untuk melakukan optimasi query *database*.
2. Skor SEO yang relatif rendah yaitu sebesar 67 dapat ditingkatkan dengan menambahkan meta tag yang lebih lengkap, struktur heading yang lebih terorganisir, dan optimalisasi kata kunci untuk meningkatkan visibilitas situs di mesin pencari.

DAFTAR PUSTAKA

- [1] D. G. Harbowo, "An Assessment of The Scientific Value of Krakatoa, Indonesia From a Geoheritage Perspective," *Juni*, vol. 2, no. 1, pp. 11–25, 2023, doi: 10.34312/jage.v2i1.19360.
- [2] O. Ariyanti and R. M. Fattah, "The process of establishing and activities of the Anak Krakatau Mountain," *Geographica Science and Education Journal*, vol. 2, no. 1, Dec. 2020, doi: 10.31327/gsej.v2i1.1339.
- [3] B. Bachina, "Comprehensive Development and Build Strategies for the PERN Stack," *Journal of Scientific and Engineering Research*, vol. 10, no. 2, pp. 189–201, 2023. [Online]. Available: www.jsaer.com
- [4] A. Vu, *Real-time backend architecture using Node.js, Express and Google Cloud Platform*, Bachelor's thesis, Metropolia University of Applied Sciences, Helsinki, Finland, Jan. 2021.
- [5] H. Sun and L. Chen, "Design of application layer software platform of remote monitoring system", *FCIS*, vol. 3, no. 2, pp. 124–126, Apr. 2023, doi: 10.54097/fcis.v3i2.7576.
- [6] J. M. Ramadhan, R. Mardiaty, and I. N. Haq, "IoT Monitoring System for Solar Power Plant Based on MQTT Publisher / Subscriber Protocol," in *Proc. 7th Int. Conf. Wireless and Telematics (ICWT)*, Bandung, Indonesia, 2021, pp. 1–6, doi: 10.1109/ICWT52862.2021.9678503.
- [7] P. Porter, S. Yang, and X. Xi, "The Design and Implementation of a RESTful IoT Service Using the MERN Stack," in *2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems Workshops (MASSW)*, IEEE, Nov. 2019, pp. 140–145. doi: 10.1109/MASSW.2019.00035.
- [8] R. Kristiyono and A. Riyanto, "APLIKASI SENSOR HC-SR04 UNTUK MENGUKUR JARAK KETINGGIAN AIR DENGAN MIKROKONTROL WEMOS D1 R2 BERBASIS IoT ", *JTe*, vol. 6, no. 4, pp. 141-148, Nov. 2020.
- [9] H. G. Suero Martínez, *Design & Development of a Full-stack ERP Marketing Web App*, Bachelor's thesis, Haaga-Helia University of Applied Sciences, Helsinki, Finland, 2022.
- [10] I. Ihsan, Z. Zulkarnain, and A. Y. Amsal, "DESAIN DAN PERANCANGAN VPS BERBASIS UBUNTU 22.04 UNTUK IMPLEMENTASI WEBSITE SEKOLAH DI SMK ADZKIYA BALIKPAPAN," *PROSISKO Jurnal Pengembangan Riset Dan Observasi*

- Sistem Komputer*, vol. 10, no. 2, pp. 173–178, Sep. 2023, doi: 10.30656/prosisko.v10i2.7103.
- [11] T. G. S. Putra and I. R. Widiyarsari, “Rancangan Virtual Private Server Pada Kantor Kelurahan Menggunakan ZeroTier,” *Building of Informatics, Technology and Science (BITS)*, vol. 4, no. 2, pp. 352–360, Sep. 2022, doi: 10.47065/bits.v4i2.1810.
 - [12] M. A. Pratama, H. Setiawan, and Z. R. Mair, “Implementasi Honeypot sebagai Pendeteksi Serangan Pada virtual private server (VPS),” *Jurnal Software Engineering and Computational Intelligence*, vol. 1, no. 1, pp. 26–39, Jun. 2023. doi:10.36982/jseci.v1i1.3045.
 - [13] D. A. B. Prasetyo, “Implementasi Information Schema Database Pada PostgreSQL untuk Pembuatan Tabel Informasi dengan Menggunakan Python Di PT XYZ,” *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, vol. 9, no. 3, pp. 1961–1972, Sep. 2022, doi: <https://doi.org/10.35957/jatisi.v9i3.2221>.
 - [14] Y. Y. Putra, O. Purwaningrum, and R. H. Winata, “Perbandingan Performa Respon Waktu Kueri MySQL, PostgreSQL, Dan mongodb,” *Jurnal Sistem Informasi dan Bisnis Cerdas*, vol. 15, no. 1, Mar. 2022. doi:10.33005/sibc.v15i1.2749.
 - [15] S. A. Bafna, “Review on Study and usage of MERN Stack for Web Development,” *International Journal for Research in Applied Science and Engineering Technology*, vol. 10, no. 2, pp. 178–186, Feb. 2022, doi: 10.22214/ijraset.2022.40209.
 - [16] Kohar and H. . Zakaria, “Implementasi Single Page Aplikasi (SPA) Pada Aplikasi Pengajuan Cuti Karyawan Berbasis Web Menggunakan React Js: Studi Kasus : PT Mitra Bisnis Sarana”, *logic*, vol. 1, no. 6, pp. 1653–1661, Oct. 2023.
 - [17] R. Fauzi, H. Ashaury, and R. Ilyas, “ANALISIS DAN INTEGRASI API GOOGLE CLASSROOM PADA SISTEM INFORMASI AKADEMIK MENGGUNAKAN CREDENTIAL UNTUK PENGAMBILAN DATA NILAI AKADEMIK DI SEKOLAH MENENGAH PERTAMA 3 CIMAHI,” *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 7, no. 4, pp. 2916–2922, Jan. 2024, doi: 10.36040/jati.v7i4.7294.
 - [18] M. Kim, S. Sinha, and A. Orso, “Adaptive REST API Testing with Reinforcement Learning,” *arXiv (Cornell University)*, Jan. 2023, doi: 10.48550/arxiv.2309.04583.
 - [19] S. P. Ichsan, I. A. Noer, M. Azra, G. B. Satrya, and M. Rosmiati, “Peningkatan Efektifitas Reservasi Online Photo Studio Menggunakan Aplikasi Picnette Berbasis Website,” *ILKOMNIKA: Journal of Computer Science and Applied Informatics*, vol. 5, no. 2, pp. 29–35, Aug. 2023, doi: 10.28926/ilkomnika.v5i2.540.

- [20] P. P. Kore, M. J. Lohar, M. T. Surve, and S. Jadhav, "API testing using Postman tool," *International Journal for Research in Applied Science and Engineering Technology*, vol. 10, no. 12, pp. 841–843, Dec. 2022, doi: 10.22214/ijraset.2022.48030.
- [21] G. Y. Saputra, A. D. Afrizal, F. K. R. Mahfud, F. A. Pribadi, and F. J. Pamungkas, "Penerapan protokol MQTT pada teknologi WAN (Studi Kasus Sistem Parkir Universitas Brawijaya)," *Informatika Mulawarman Jurnal Ilmiah Ilmu Komputer*, vol. 12, no. 2, p. 69, Aug. 2017, doi: 10.30872/jim.v12i2.653.
- [22] F. Azzedin and T. Alhazmi, "Secure Data Distribution Architecture in IoT Using MQTT," *Applied Sciences*, vol. 13, no. 4, p. 2515, Jan. 2023, doi: <https://doi.org/10.3390/app13042515>.
- [23] A. M. Gobachew, D. Kitaw, E. Berhan, and H. D. Haasis, "ABC/XYZ analysis for kanban system implementation in pharmaceutical supply chain: A case of ethiopian pharmaceutical supply agency," *International Journal of Information Systems and Supply Chain Management*, vol. 14, no. 3, pp. 63–78, Jul. 2021, doi: 10.4018/IJISSCM.2021070104.
- [24] N. S, U. S. R, and P. Mohan, "Comparison of Utility-First CSS Framework," *Journal of Innovation and Technology*, vol. 2024, no. 1, Nov. 2024, doi: 10.61453/joit.v2024no32.