

**PENGEMBANGAN *FRONTEND WEBSITE* PUMMA PEMANTAUAN DAN
PERINGATAN DINI BENCANA TSUNAMI MENGGUNAKAN *LIBRARY*
REACTJS DAN METODE SCRUM**

(Skripsi)

**Oleh
AKBAR ANDIKA KHADAFI**



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

**PENGEMBANGAN *FRONTEND WEBSITE* PUMMA PEMANTAUAN DAN
PERINGATAN DINI BENCANA TSUNAMI MENGGUNAKAN *LIBRARY*
REACTJS DAN METODE SCRUM**

**Oleh
AKBAR ANDIKA KHADAFI**

Skripsi

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

Pada

Program Studi S1 Teknik Informatika



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

ABSTRAK

PENGEMBANGAN *FRONTEND WEBSITE* PUMMA PEMANTAUAN DAN PERINGATAN DINI BENCANA TSUNAMI MENGGUNAKAN *LIBRARY REACTJS* DAN METODE SCRUM

Oleh

AKBAR ANDIKA KHADAFI

PUMMA (Perangkat Ukur Murah Muka Air laut) merupakan *website* sistem informasi pemantauan dan peringatan dini bencana tsunami berbasis IoT yang dikembangkan oleh tim U-TEWS Krakatau Universitas Lampung. Sistem ini menyajikan data *real-time* dari perangkat *monitoring* seperti PUMMA dan Buoy, termasuk informasi *battery voltage*, *device temperature*, *sea water level*, dan status perangkat. Namun, versi awal *website* masih memiliki kekurangan dalam desain antarmuka dan fitur *dashboard*. Penelitian ini bertujuan untuk mengembangkan *frontend website* PUMMA menggunakan *React JS* dan *Tailwind CSS* dengan pendekatan metode Scrum. Pengembangan dilakukan selama 8 minggu dalam 4 iterasi, dimulai dari implementasi *user interface* hingga integrasi *API*. Sistem yang dikembangkan mencakup fitur utama seperti tampilan *landing page*, daftar alat, *card status*, tabel, grafik, dan lokasi perangkat. Pengujian sistem dilakukan menggunakan metode *Blackbox Testing* pada 6 skenario tiap iterasi dan memperoleh tingkat keberhasilan 100%. Evaluasi kebergunaan menggunakan *System Usability Scale (SUS)* memperoleh skor 83,8 (kategori *acceptable*, *grade B*, *adjective rating excellent*). Pengujian *A/B Testing* menunjukkan bahwa *website* versi baru lebih disukai dengan skor rata-rata 96,6% dibandingkan versi lama sebesar 3,3%. Hasil penelitian menunjukkan bahwa pengembangan *website* PUMMA berhasil meningkatkan kualitas, kegunaan, dan pengalaman pengguna secara signifikan.

Kata Kunci: PUMMA, *React JS*, *Tailwind CSS*, *Scrum*, *System Usability Scale*, *Blackbox Testing*, *A/B Testing*

ABSTRACT

FRONTEND DEVELOPMENT OF THE PUMMA WEBSITE FOR TSUNAMI MONITORING AND EARLY WARNING USING THE REACTJS LIBRARY AND THE SCRUM METHOD

BY

AKBAR ANDIKA KHADAFI

PUMMA (Affordable Sea Surface Level Measuring Device) is an information system website for tsunami monitoring and early warning based on IoT, developed by the U-TEWS Krakatau team at the University of Lampung. The system presents real-time data from monitoring devices such as PUMMA and Buoy, including battery voltage, device temperature, sea water level, and device status. However, the initial version of the website had several shortcomings in terms of interface design and dashboard features. This study aims to develop the frontend of the PUMMA website using React JS and Tailwind CSS through the Scrum methodology. The development was carried out over 8 weeks in 4 iterations, covering the implementation of the user interface to API integration. The system includes key features such as a landing page, device list, status cards, tables, charts, and device location visualization. The system's functionality was tested using the blackbox testing method across 6 scenarios per iteration, achieving a 100% success rate. Usability evaluation using the System Usability Scale (SUS) resulted in an acceptance score of 83.8, categorized as acceptable, with a grade of B and an adjective rating of excellent. A/B testing results showed that the newly developed website version was preferred, scoring an average of 96.6%, compared to 3.3% for the previous version. The findings indicate that the PUMMA website redevelopment successfully enhanced the system's quality, usability, and user experience.

Keywords: PUMMA, React JS, Tailwind CSS, Scrum, System Usability Scale, Blackbox Testing, A/B Testing

Judul Skripsi

**: PENGEMBANGAN *FRONTEND WEBSITE*
PUMMA PEMANTAUAN DAN PERINGATAN
DINI BENCANA TSUNAMI MENGGUNAKAN
LIBRARY REACTJS DAN METODE SCRUM**

Nama Mahasiswa : **Akbar Andika Khadafi**

Nomor Pokok Mahasiswa : 2015061078

Program Studi : Teknik Informatika

Jurusan : Teknik Elektro

Fakultas : Teknik

MENYETUJUI

I. Komisi Pembimbing

Pembimbing Utama

Pembimbing Pendamping

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

NIP. 197111122000031002

Resty Annisa, S.ST., M.Kom

NIP. 199008302019032019

2. Mengetahui

Ketua Jurusan Teknik Elektro

Ketua Program Studi Teknik
Informatika

Herlinawati, S.T., M.T.

NIP. 197103141999032001

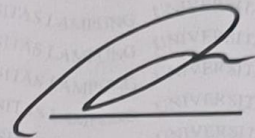
Yessi Mulyani, S.T., M.T.

NIP. 197312262000122001

MENGESAHKAN

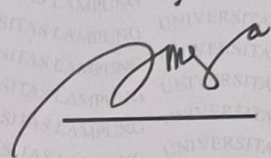
1 Tim Penguji
Ketua

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



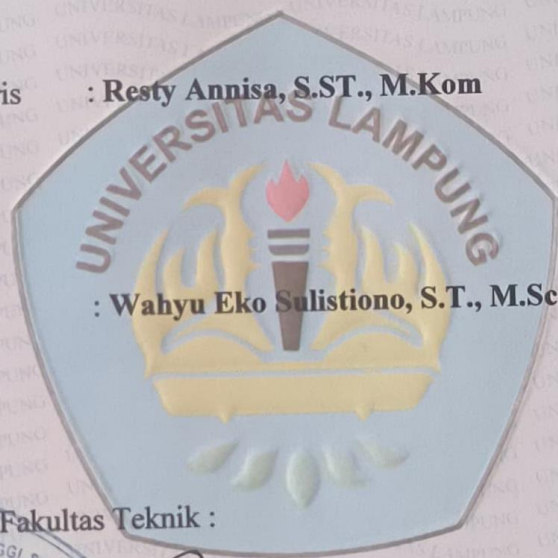
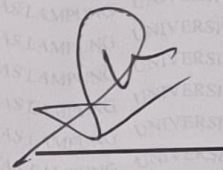
Sekretaris

: Resty Annisa, S.ST., M.Kom



Penguji

: Wahyu Eko Sulistiono, S.T., M.Sc.



2 Dekan Fakultas Teknik :



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

NIP. 197509282001121002

Tanggal Lulus Ujian Skripsi : 21 April 2025

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini, menyatakan bahwa skripsi saya dengan judul “Pengembangan *Frontend Website* PUMMA Pemantauan Dan Peringatan Dini Bencana Tsunami Menggunakan *Library ReactJS* Dan Metode Scrum” dibuat oleh saya sendiri. Semua hasil yang tertuang dalam skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti bahwa skripsi ini merupakan salinan atau dibuat oleh orang lain, maka saya bersedia menerima sanksi sesuai dengan ketentuan hukum atau akademik yang berlaku.

Bandar Lampung, 28 April 2025

Pembuat Pernyataan,



Akbar Andika Khadafi

NPM. 2015061078

RIWAYAT HIDUP



Penulis yang bernama Akbar Andika Khadafi dilahirkan di Bandar Lampung pada tanggal 28 Februari 2002, sebagai anak terakhir dari tiga bersaudara, dari Bapak Hermanto dan Ibu Sumiyati. Pendidikan yang ditempuh penulis, yaitu Sekolah Dasar (SD) Negeri 2 Beringin Raya diselesaikan pada tahun 2014,

Sekolah Menengah Pertama (SMP) Negeri 14 Bandar Lampung diselesaikan pada tahun 2017, Sekolah Menengah Atas (SMA) Negeri 3 Bandar Lampung diselesaikan pada tahun 2020. Penulis terdaftar sebagai mahasiswa Program Studi Teknik Informatika Jurusan Teknik Elektro Fakultas Teknik Universitas Lampung pada tahun 2020 melalui jalur Seleksi Bersama Masuk Perguruan Tinggi Negeri (SBMPTN). Selama menjadi mahasiswa, penulis melaksanakan perkuliahan dengan mengikuti prosedur yang telah ditetapkan. Untuk melatih hardskill dan softskill, penulis mengikuti program Kampus Merdeka Studi Independen di Dicoding pada bulan Agustus – Desember 2022. Penulis juga melaksanakan kegiatan Kuliah Kerja Nyata Pada bulan Januari 2023 di desa Reno Basuki, Kecamatan Rumbia, Kabupaten Lampung Tengah, Provinsi Lampung. Selain itu, penulis juga magang di Bank Lampung KCP Menggala pada bulan Juli – Agustus 2023 dan mengikuti MSIB Batch 5 di Maritim Muda Future Leaders Program dengan posisi Web Developer & Administrator pada bulan Agustus-Desember 2023.

MOTTO

"Aku percaya masa depan milik mereka yang berani bermimpi dan bertindak. Karena seperti kata Eleanor Roosevelt, *'The future belongs to those who believe in the beauty of their dreams.'*"

(Sumber: Eleanor Roosevelt, *You Learn by Living*, 1960)

PERSEMBAHAN

Bismillahirrahmanirrahim

Puji syukur ke hadirat Allah Swt. atas nikmat iman, kesehatan, dan kesempatan yang diberikan, sehingga skripsi ini dapat terselesaikan dengan baik.

Karya ini kupersembahkan kepada:

Kedua Orang Tua,

yang telah menjadi teladan hidupku, mendidikku dengan kasih sayang tanpa batas, mengiringi setiap langkahku dengan doa yang tulus, dan memberikan segala yang terbaik dalam hidupku. Terima kasih atas cinta, pengorbanan, dan keikhlasan kalian, yang takkan pernah mampu kubalas sepenuhnya.

Keluarga Dan Teman-Teman Terbaikku,

yang selalu ada dalam suka dan duka, memberikan semangat, tawa, doa, dan dukungan tanpa henti. Kehadiran kalian adalah anugerah yang begitu berharga dalam perjalanan hidupku.

Dosen Pembimbing dan Dosen Penguji,

yang dengan penuh kesabaran dan kebijaksanaan telah memberikan arahan, ilmu, dan motivasi selama proses penulisan skripsi ini.

Almamater Tercinta

Teknik Informatika Universitas Lampung

SANWACANA

Puji syukur kehadiran Allah SWT yang telah memberikan rahmat dan hidayat-Nya sehingga penulis dapat menyelesaikan penelitian yang berjudul “Pengembangan *Frontend Website* PUMMA Pemantauan Dan Peringatan Dini Bencana Tsunami Menggunakan *Library ReactJS* Dan Metode Scrum”. Selama pelaksanaan penelitian ini penulis menerima banyak dukungan, bimbingan serta bantuan dari berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Kedua orang tua tercinta dan seluruh keluarga yang selalu memberikan doa, motivasi dan kasih sayang tiada terkira yang selalu mengingatkan penulis untuk menyelesaikan penelitian ini.
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 Universitas Lampung.
4. Ibu Yessi Mulyani, S.T., M.T. selaku Ketua Program Studi Teknik Informatika Universitas Lampung dan telah membantu proses kelancaran pengerjaan penelitian dan selaku Pembimbing Akademik yang selalu memberikan dukungan serta bimbingan agar menjadi lebih baik.
5. Bapak Mona Arif Muda, S.T., M.T. selaku Pembimbing Utama yang selalu memberikan dukungan berupa waktu maupun ilmu kepada penulis dalam menyelesaikan penelitian ini.
6. Ibu Resty Annisa, S.ST., M.Kom selaku Pembimbing Pendamping yang telah membimbing dan memberikan dukungan dalam menyelesaikan penelitian ini.

7. Bapak Wahyu Eko Sulistiono, S.T., M.Sc. selaku Penguji yang telah memberikan banyak saran dan masukan terhadap penelitian ini.
8. Mbak Rika selaku Admin Program Studi Teknik Informatika yang telah banyak membantu penulis dalam segala urusan administrasi selama perkuliahan dan penelitian.
9. Seluruh dosen dan staf Jurusan Teknik Informatika Unila yang memberi masukan dan mempermudah proses penelitian ini.
10. Tim Penelitian U-TEWS Krakatau yang telah membantu penulis selama pembuatan aplikasi.
11. Abrorizt Talentio, Fiky Nurahman, Muhammad Akbar, Muhammad Rendy Pratama, Okky Damayanti, Syafa Ramadhani Harja, dan seluruh teman-teman PSTI Angkatan 2020 telah mewarnai masa perkuliahan penulis dan menulis banyak cerita bersama.
12. Keluarga besar Teknik Elektro Angkatan 2020 yang telah menjadi teman seperjuangan sejak mahasiswa baru. Terima kasih telah mewarnai masa perkuliahan penulis dan menulis banyak cerita bersama.

Bandar Lampung, 28 April 2025
Penulis,

Akbar Andika Khadafi

DAFTAR ISI

DAFTAR TABEL	xiii
DAFTAR GAMBAR.....	xv
I. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian.....	2
1.4 Manfaat Penelitian.....	3
1.5 Batasan Masalah.....	3
1.6 Sistematika Penulisan.....	3
II. TINJAUAN PUSTAKA.....	5
2.1 Redesain Website.....	5
2.2 <i>Early Warning System (EWS)</i>	5
2.3 <i>Internet Of Things (IoT)</i>	6
2.4 PUMMA	6
2.5 Ubuoy	9
2.6 JavaScript	12
2.7 ReactJS	13
2.8 REST-API	14
2.9 Tailwind CSS	14
2.10 Visual Studio Code.....	14
2.11 Figma.....	15
2.12 <i>Scrum</i>	15
2.13 <i>Use Case Diagram</i>	18
2.14 Activity Diagram	19

2.15	<i>System Usability Scale</i>	19
2.16	<i>Black Box Testing</i>	20
2.17	<i>A/B Testing</i>	21
2.18	Penelitian Terkait	21
III.	METODOLOGI PENELITIAN	26
3.1	Waktu dan Tempat Penelitian	26
3.2	Alat Penelitian	27
3.3	Tahapan Penelitian	28
3.3.1	Analisis Kebutuhan	30
3.3.2	Perancangan Sistem	30
3.3.3	Pengembangan Sistem	56
3.3.4	Pengujian Sistem	59
3.3.5	Penulisan Laporan	62
IV.	HASIL DAN PEMBAHASAN	63
4.1	<i>Product Backlog</i>	63
4.2	Iterasi I	67
4.3	Iterasi II.....	80
4.4	Iterasi III	117
4.5	Iterasi IV	133
4.6	<i>Black Box Testing</i>	148
4.7	<i>Usability Testing</i>	155
4.8	<i>A/B Testing Website</i>	158
4.9	Pembahasan	159
V.	KESIMPULAN DAN SARAN	161
5.1	Kesimpulan.....	161
5.2	Saran	162
	DAFTAR PUSTAKA	163

DAFTAR TABEL

Tabel 1. Simbol Use Case Diagram	18
Tabel 2. Simbol Activity Diagram	19
Tabel 3. Jadwal Penelitian.....	26
Tabel 4. Alat Penelitian.....	27
Tabel 5. Bahan berupa data sensor dari raspberry pi yang dikirimkan oleh server menggunakan <i>REST API</i>	28
Tabel 6. Kebutuhan Sistem	30
Tabel 7. Kebutuhan Fungsional	31
Tabel 8. <i>Environment Project</i>	33
Tabel 9. <i>Team Scrum</i>	56
Tabel 10. Pertanyaan <i>SUS</i>	60
Tabel 11. Pilihan Jawaban <i>SUS</i>	60
Tabel 12. Pertanyaan <i>SUS</i>	61
Tabel 13. Pilihan Jawaban A/B Testing.....	62
Tabel 14. <i>Produck Backlog</i>	63
Tabel 15. <i>Sprint Backlog</i> Iterasi I	67
Tabel 16. <i>Sprint Backlog</i> Iterasi II	81
Tabel 17. <i>Sprint Backlog</i> Iterasi III.....	118
Tabel 18. API Pumma	121
Tabel 19. <i>Feedback</i> Iterasi III	132
Tabel 20. <i>Sprint Backlog</i> Iterasi IV	133
Tabel 21. Pengujian Black Box.....	148
Tabel 22. Pengujian Melihat Landing Page	149
Tabel 23. Pengujian Melihat List Alat	150
Tabel 24. Melihat Data Alat Dalam Bentuk Card Status	151
Tabel 25. Melihat Data Alat Dalam Bentuk Table	152
Tabel 26. Melihat Data Dalam Bentuk Chart	153
Tabel 27. Melihat Lokasi Alat	154
Tabel 28. Profil Responden <i>SUS</i>	155
Tabel 29. Hasil Pengujian <i>SUS</i>	156
Tabel 30. Profil Responden A/B Testing	158
Tabel 31. Hasil Pengujian A/B Testing	159

DAFTAR GAMBAR

Gambar 1. Desain PUMMA.....	7
Gambar 2. Rangkaian Skematik dari PUMMA	7
Gambar 3. Penempatan Pumma di Pulau Panjang dan Marina Jambu	8
Gambar 4. Penempatan Pumma di Pantai Canti	8
Gambar 5. Desain Buoy	9
Gambar 6. Rangkaian Skematik Buoy	10
Gambar 7. Penempatan Buoy di laut lingkungan sekitar anak Gunung Krakatau.....	11
Gambar 8. Posisi Ubuoy 001 tanggal 29 Agustus 2024.....	12
Gambar 9. Alur Kerja Scrum	16
Gambar 10. Skor Penilaian SUS.	20
Gambar 11. Tahapan Penelitian	29
Gambar 12. <i>Environment Project</i>	32
Gambar 13. <i>Use Case Diagram</i> Pengguna	34
Gambar 14. <i>AD</i> Mengakses <i>Landing Page</i>	36
Gambar 15. <i>AD</i> Memilih <i>list device</i> yang ingin dibuka.....	37
Gambar 16. <i>AD</i> Melihat Detail Pada Alat.....	39
Gambar 17. Halaman <i>Dashboard</i> Sebelum Redesain.....	41
Gambar 18. Halaman <i>Dashboard</i> Sesudah Redesain	42
Gambar 19. Rancangan Halaman Detail <i>Device</i> Sebelum Redesain	43
Gambar 20. Halaman Detail <i>Device</i> Pada <i>Card Status</i>	44
Gambar 21. Halaman Detail <i>Device</i> Pada <i>Table</i>	45
Gambar 22. Halaman Detail <i>Device</i> Pada Grafik	46
Gambar 23. Halaman Detail <i>Device</i> Pada Lokasi	48
Gambar 24. Halaman <i>List Device</i> Buoy.....	49
Gambar 25. Halaman Detail Buoy Pada <i>Card Status</i>	50
Gambar 26. Halaman Detail Buoy Pada <i>Table</i>	52
Gambar 27. Halaman Detail Buoy Pada Grafik.....	54
Gambar 28. Halaman Detail Lokasi Buoy	55
Gambar 29. <i>Trello Product Backlog</i>	66
Gambar 30. <i>Trello Iterasi I</i>	68
Gambar 31. Implementasi <i>Sidebar</i>	69
Gambar 32. <i>Source Code Sidebar</i>	70
Gambar 33. Implementasi halaman <i>list</i> alat Pumma.....	71
Gambar 34. <i>Source code</i> halaman <i>list</i> alat Pumma.....	72
Gambar 35. Implementasi halaman <i>list</i> alat Climatologi.....	73

Gambar 36. <i>Source code</i> halaman <i>list</i> alat <i>Climatologi</i>	74
Gambar 37. Implementasi halaman <i>list</i> alat <i>Buoy</i>	76
Gambar 38. <i>Source code</i> halaman <i>list</i> alat <i>Buoy</i>	77
Gambar 39. Implementasi <i>Tabs</i>	78
Gambar 40. <i>Source Code Tabs</i>	79
Gambar 41. Trello Iterasi II	81
Gambar 42. Implementasi halaman <i>Card Status</i> detail alat <i>Pumma</i>	82
Gambar 43. <i>Source Code</i> Implementasi <i>UI</i> halaman <i>Card Status</i> detail alat <i>Pumma</i>	84
Gambar 44. Implementasi halaman <i>Table</i> detail alat <i>Pumma</i>	85
Gambar 45. <i>Source Code</i> halaman <i>Table</i> detail alat <i>Pumma</i>	86
Gambar 46. Implementasi halaman <i>Chart</i> detail alat <i>Pumma</i>	87
Gambar 47. <i>Source Code</i> halaman <i>Chart</i> detail alat <i>Pumma</i>	89
Gambar 48. Implementasi halaman <i>Location</i> detail alat <i>Pumma</i>	91
Gambar 49. <i>Source Code</i> halaman <i>Location</i> detail alat <i>Pumma</i>	92
Gambar 50. Implementasi halaman <i>Card Status</i> detail alat <i>Climatologi</i>	94
Gambar 51. <i>Source Code</i> Implementasi halaman <i>Card Status</i> detail alat <i>Climatologi</i>	95
Gambar 52. Implementasi halaman <i>Table</i> detail alat <i>Climatologi</i>	96
Gambar 53. <i>Source Code</i> halaman <i>Table</i> detail alat <i>Climatologi</i>	97
Gambar 54. Implementasi halaman <i>Chart</i> detail alat <i>Climatologi</i>	98
Gambar 55. <i>Souce Code</i> Implementasi halaman <i>Chart</i> detail alat <i>Climatologi</i>	99
Gambar 56. Implementasi halaman <i>Location</i> detail alat <i>Climatologi</i>	100
Gambar 57. <i>Source Code</i> halaman <i>Location</i> detail alat <i>Climatologi</i>	101
Gambar 58. Implementasi halaman <i>Card Status</i> detail alat <i>Buoy</i>	103
Gambar 59. <i>Source Code</i> halaman <i>Card Status</i> detail alat <i>Buoy</i>	104
Gambar 60. Implementasi halaman <i>Table</i> detail alat <i>Buoy</i>	106
Gambar 61. <i>Source Code</i> halaman <i>Table</i> detail alat <i>Ubuoy</i>	107
Gambar 62. Implementasi halaman <i>Table</i> detail alat <i>FC Buoy</i>	109
Gambar 63. <i>Source Code</i> halaman <i>Table</i> detail alat <i>FC Buoy</i>	110
Gambar 64. Implementasi halaman <i>Chart</i> detail alat <i>Buoy</i>	111
Gambar 65. <i>Source Code</i> halaman <i>Chart</i> detail alat <i>Ubuoy</i>	113
Gambar 66. Implementasi halaman <i>Location</i> detail alat <i>Ubuoy</i>	115
Gambar 67. <i>Source Code</i> halaman <i>Location</i> detail alat <i>Ubuoy</i>	116
Gambar 68. Trello Iterasi III	118
Gambar 69. Dokumentasi <i>API</i> <i>Pumma</i>	119
Gambar 70. <i>Response API</i> <i>Pumma</i>	120
Gambar 71. <i>Source Code API GET Latest</i>	122
Gambar 72. <i>Source Code API GET AGG</i>	123
Gambar 73. <i>Source Code API GET IMAGE</i>	125
Gambar 74. Dokumentasi <i>API</i> <i>Buoy</i>	126
Gambar 75. <i>Response API</i> <i>Buoy</i>	127

Gambar 76. Response API Buoy server offline	128
Gambar 77. Repository Github	129
Gambar 78. <i>Deployment Vercel</i>	130
Gambar 79. Responsive Website	131
Gambar 80. Trello Iterasi IV	134
Gambar 81. UI/UX Halaman Landing Page	135
Gambar 82. Implementasi Halaman Landing Page.....	137
Gambar 83. Source Code Halaman Landing Page.....	139
Gambar 84. Membuat UI/UX design halaman Pumma 006 Pulau Sebesi.....	141
Gambar 85. Implementasi Penambahan list alat Pumma U-TEWS 006 Pulau Sebesi	143
Gambar 86. Penambahan list alat Pumma U-TEWS 006	144
Gambar 87. Implementasi halaman Card Status detail alat Pumma U-TEWS 006 Pulau Sebesi	145
Gambar 88. <i>Source Code</i> halaman <i>Card Status</i> detail alat Pumma U-TEWS 006 Pulau Sebesi	146
Gambar 89. Hasil Pengujian Melihat <i>Landing Page</i>	149
Gambar 90. Hasil Pengujian Melihat <i>List Alat</i>	150
Gambar 91. Hasil Pengujian Melihat Data Alat Dalam Bentuk <i>Card Status</i>	151
Gambar 92. Hasil Pengujian Melihat Data Alat Dalam Bentuk <i>Table</i>	152
Gambar 93. Hasil Pengujian Melihat Data Alat Dalam Bentuk <i>Chart</i>	153
Gambar 94. Hasil Pengujian Melihat Lokasi Alat	154
Gambar 95. Penentuan Hasil Penilaian Skor <i>SUS</i>	157

I. PENDAHULUAN

1.1 Latar Belakang

Tsunami adalah bencana alam yang sangat membahayakan manusia, terutama mereka yang tinggal di daerah pantai. Bencana ini datang tiba-tiba dan bisa menyebabkan kerusakan parah. Perubahan iklim membuat cuaca menjadi semakin ekstrem dan meningkatkan risiko terjadinya tsunami. Negara Indonesia terletak di wilayah pertemuan beberapa lempeng tektonik yang aktif bergerak, sehingga sangat rentan terhadap bencana tsunami.[9].

Data dari Badan Nasional Penanggulangan Bencana (BNPB) memperkirakan akan terjadi sebanyak 76 kali gelombang pasang atau abrasi di wilayah Indonesia selama periode 2021 hingga 2023. Sistem pemantauan dan upaya pencegahan yang tidak memadai dapat menyebabkan kejadian ini berpotensi menjadi bencana alam yang serius. Gelombang laut yang melebihi batas normal akan sangat mengancam daerah pesisir, baik ekosistem laut maupun kawasan daratan di sekitarnya. [4].

Peneliti terdahulu yang dilakukan oleh Ranto membangun *front-end website* sistem informasi pemantauan dan peringatan dini bencana tsunami berbasis IoT (*Internet of Things*) dengan menggunakan library ReactJS. Penelitian tersebut berhasil menciptakan *website* yang mendukung pengolahan data secara *real-time* dari sensor-sensor IoT yang dipasang di area rawan bencana. *Website* Pumma memiliki sistem *monitoring Early Warning System* (EWS) tsunami yang dapat memberikan informasi *real-time* terkait potensi bahaya tsunami. Sistem ini menyajikan data berupa *battery voltage*, *device temperature*, *sea water level*, *forecast30*, *forecast300*, dan *status* dalam bentuk *card*

status, table, dan chart untuk memudahkan pemantauan kondisi terkini dan memberikan peringatan dini kepada masyarakat di wilayah yang berisiko, khususnya di area sekitar Anak Krakatau. [16].

Hasil penelitian Ranto menyarankan beberapa peningkatan pada *website* Pumma. Desain *User Interface (UI)* perlu diperbarui agar tampil lebih modern dan menarik. Sistem juga perlu dilengkapi dengan penambahan alat dan fitur baru pada *dashboard* untuk meningkatkan fungsionalitas sistem. [16]. Untuk mencapai tujuan tersebut, beberapa masalah desain dan fitur *website* perlu diperbaiki, termasuk tampilan menu yang kurang menarik, *font* monoton, warna yang tidak sesuai tema, serta tata letak yang perlu dioptimalkan. Perbaikan ini diharapkan dapat meningkatkan kualitas dan daya tarik *website* Pumma. Penambahan *Landing Page* dan alat Buoy yang sedang dikembangkan oleh tim U-TEWS Krakatau diintegrasikan ke dalam *website*.

Permasalahan tersebut mendorong perlunya penelitian berjudul "PENGEMBANGAN FRONTEND WEBSITE PUMMA PEMANTAUAN DAN PERINGATAN DINI BENCANA TSUNAMI MENGGUNAKAN LIBRARY REACTJS DAN METODE SCRUM" untuk tim peneliti U-TEWS Krakatau Unila.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah diuraikan, rumusan masalah dari penelitian ini adalah:

1. Bagaimana membuat desain pada *frontend website* menjadi lebih *user-friendly* dan mudah diakses oleh pengguna dibandingkan sebelumnya?
2. Bagaimana membuat *frontend Ubuoy* dan *Landing Page* kedalam *website*?

1.3 Tujuan Penelitian

Tujuan penelitian ini dilaksanakan adalah sebagai berikut:

1. Mengembangkan desain *frontend website* menggunakan Figma dan *Framework* React JS.
2. Mengembangkan *frontend* Ubuoy dan *Landing Page* ke dalam *website* untuk menampilkan data dari alat secara *real-time*.

1.4 Manfaat Penelitian

Manfaat Penelitian ini adalah sebagai berikut :

1. Penelitian ini membantu pengguna dalam melakukan memonitoring dan menganalisis data bencana gelombang pasang dan tsunami yang berpotensi terjadi.
2. Penelitian ini dapat menjadi referensi untuk pengembangan selanjutnya yang berkaitan dengan pembuatan aplikasi *monitoring Early Warning System (EWS)* berbasis *website*.

1.5 Batasan Masalah

Adapun batasan masalah dalam penelitian ini meliputi hal-hal sebagai berikut:

1. Penelitian ini akan berfokus dalam redesain *Front End* dan Penambahan fitur dashboard Ubuoy pada *website* Pumma.
2. *Website* ini dibuat berbasis *web* dengan menggunakan React JS dan Tailwind CSS.

1.6 Sistematika Penulisan

Sistematika penulisan yang digunakan oleh penulis dalam menyusun laporan kerja praktik ini adalah sebagai berikut:

BAB I : PENDAHULUAN

Pada bab ini secara umum memuat latar belakang

penelitian, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah dan sistematika penulisan penelitian.

BAB II : TINJAUAN PUSTAKA

Pada bab ini berisi teori dasar dari Redesain Website, Early Warning System, Internet Of Things, PUMMA, Ubuoy, JavaScript, ReactJS, REST-API, Tailwind CSS, Visual Studio Code, Figma, Scrum, Use Case Diagram, Activity Diagram, System Usability Scale, Black Box Testing, dan penelitian terkait.

BAB III : METODOLOGI PENELITIAN

Pada bab ini memuat metodologi penelitian yang digunakan dalam menyusun penelitian

BAB IV HASIL DAN PEMBAHASAN

Pada bab ini berisi hasil dan pembahasan pembuatan website Pumma

BAB V : KESIMPULAN DAN SARAN

Pada bab ini berisi kesimpulan dan saran hasil dari penelitian.

DAFTAR
PUSTAKA
LAMPIRAN

II. TINJAUAN PUSTAKA

2.1 Redesain Website

Redesain *website* adalah langkah penting bagi seorang pengembang *web* untuk memperbaiki tampilan *website* dan menambahkan fitur yang sesuai guna meningkatkan pengalaman pengguna. Pembaruan ini, diharapkan dapat menciptakan dampak positif, seperti peningkatan jumlah pengguna dikarenakan pengalaman pengguna saat mengunjungi *website* nyaman dan mudah untuk digunakan. Dengan begitu, tidak hanya citra positif yang terangkat, namun juga kontribusi positif terhadap pemantauan dan peringatan dini bencana tsunami [1].

2.2 Early Warning System (EWS)

Early Warning System (EWS) atau Sistem peringatan dini adalah sebuah sistem yang dirancang untuk mendeteksi dan memberikan peringatan dini terhadap ancaman atau kejadian yang mungkin terjadi di masa depan. Tujuan utama dari EWS adalah untuk mengidentifikasi risiko potensial dan memberikan informasi yang memadai kepada pihak yang berwenang agar mereka dapat mengambil langkah-langkah pencegahan atau mitigasi yang diperlukan [20].

EWS dapat diterapkan dalam berbagai konteks, termasuk bidang keamanan, keuangan, kesehatan, dan lingkungan. Contoh penerapan EWS antara lain adalah sistem peringatan dini tsunami yang mengamati aktivitas gempa bumi di laut untuk memberikan peringatan kepada masyarakat di wilayah pesisir, sistem peringatan dini cuaca ekstrem yang memantau kondisi atmosfer untuk memberikan peringatan terhadap badai atau

banjir yang akan datang, dan sistem peringatan dini pandemi yang mengumpulkan data epidemiologi untuk mengidentifikasi penyebaran penyakit yang cepat.

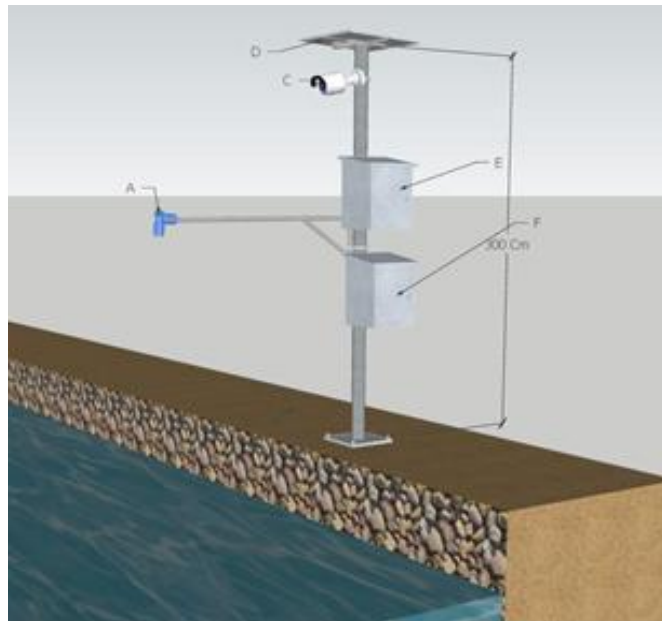
2.3 *Internet Of Things (IoT)*

Internet of Things (IoT) adalah jaringan perangkat fisik kendaraan, bangunan, dan barang lainnya tertanam dengan elektronik, perangkat lunak, sensor, dan konektivitas jaringan yang memungkinkan benda-benda ini mengumpulkan dan bertukar data. Pengertian lain yang menjelaskan bahwa sistem perangkat komputasi yang saling terkait, mesin mekanik dan digital, objek, hewan, atau manusia yang dilengkapi dengan pengidentifikasi unik dan kemampuan untuk mentransfer data melalui jaringan tanpa memerlukan interaksi manusia ke manusia atau manusia ke komputer [18].

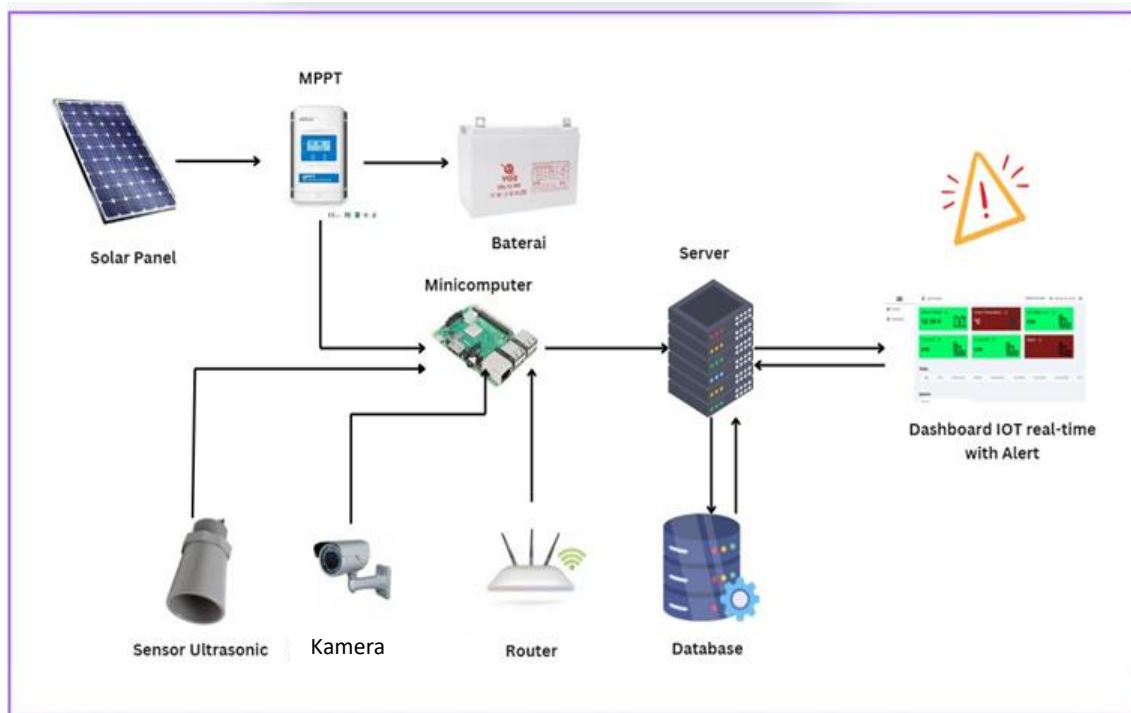
Tujuan utama dari IoT adalah untuk menciptakan lingkungan yang terhubung secara digital, di mana objek-objek sehari-hari dapat saling berinteraksi dan berbagi informasi secara otomatis tanpa adanya campur tangan manusia. Melalui penggunaan sensor dan perangkat yang terhubung, data yang dikumpulkan dapat digunakan untuk memantau, mengontrol, dan mengelola sistem secara efisien.

2.4 PUMMA

PUMMA (Perangkat Ukur Murah Muka Air laut) adalah alat pendeteksi tsunami yang dikembangkan oleh tim dari Universitas Lampung sebagai respons terhadap tsunami Selat Sunda pada 22 Desember 2018. Alat ini dilengkapi dengan berbagai komponen utama, antara lain dua panel surya berkapasitas 100 WP yang didukung MPPT untuk efisiensi pengisian daya, baterai berkapasitas 100Ah yang dapat menyuplai daya hingga tiga hari tanpa pengisian ulang, serta minicomputer (Raspberry Pi) sebagai pusat pemrosesan data. Untuk mengukur ketinggian air laut, PUMMA menggunakan sensor ultrasonik, dan kamera yang memantau kondisi sekitar setiap dua menit untuk validasi visual [3].



Gambar 1. Desain PUMMA



Gambar 2. Rangkaian Skematik dari PUMMA

Data yang dikumpulkan dikirim melalui router dan disimpan di server serta database, di mana semuanya diakses melalui dashboard IoT yang dilengkapi dengan sistem notifikasi. PUMMA bekerja dalam tiga lapisan utama: lapisan *input* yang mengukur ketinggian air, kondisi lingkungan, dan performa panel surya; lapisan controller yang memproses dan mengirim data melalui jaringan global; serta lapisan *network* yang mengelola dan menyimpan data untuk keperluan *monitoring*. Alat ini dipasang di beberapa lokasi strategis, seperti Gebang, Canti, Panjang, Jambu, dan Pangandaran, dengan tujuan utama mendeteksi kenaikan muka air laut secara *real-time* [3].



Gambar 3. Penempatan Pumma di Pulau Panjang dan Marina Jambu

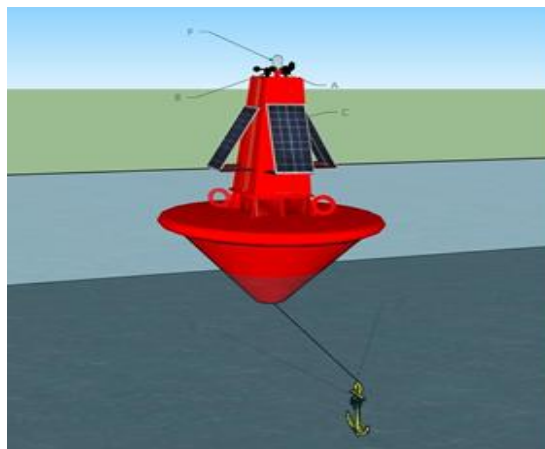


Gambar 4. Penempatan Pumma di Pantai Canti

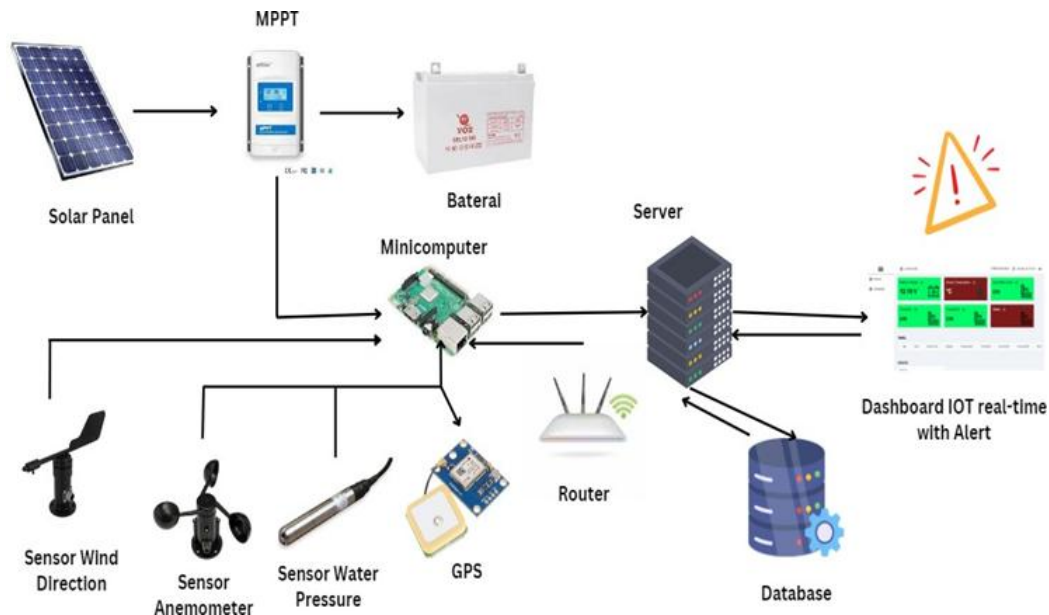
Data tersebut dikirim ke BMKG untuk mendukung sistem peringatan dini tsunami, sekaligus menyediakan informasi seperti ketinggian air laut, tegangan dan arus panel surya, *alert level*, dan *latency*, yang bermanfaat dalam penelitian kebencanaan di masa depan [3].

2.5 Ubuoy

Ubuoy adalah sebuah sistem peringatan dini tsunami berbasis buoy yang dikembangkan untuk mendeteksi dan mengirimkan data tsunami secara *real-time*. Sistem ini terdiri dari beberapa komponen utama, termasuk OBU (*Ocean Bottom Unit*) yang ditempatkan di dasar laut dan buoy permukaan yang dihubungkan dengan sistem *mooring*. Sensor tekanan (BPR) yang dipasang pada *Ocean Bottom Unit* (OBU) berfungsi untuk mendeteksi perubahan tinggi permukaan air laut yang disebabkan oleh tsunami. Setelah mendeteksi perubahan tersebut, sensor ini mengirimkan informasi ke buoy yang mengapung di permukaan laut melalui sinyal suara. *Mooring* adalah bagian yang menghubungkan buoy permukaan dengan pemberat, dan berfungsi untuk menahan buoy terhadap aksi gaya angin, gelombang, dan arus. Sistem ini memastikan buoy tetap berada di posisi yang tepat meskipun ada gangguan dari faktor lingkungan laut, sehingga dapat beroperasi dengan stabil dalam mendeteksi perubahan yang terjadi di sekitar laut [14].



Gambar 5. Desain Buoy



Gambar 6. Rangkaian Skematik Buoy

Sistem peringatan dini tsunami di Indonesia menggunakan kombinasi dua perangkat utama yaitu PUMMA (Perangkat Ukur Murah Muka Air laut) dan Buoy yang dikembangkan oleh tim Universitas Lampung setelah tragedi tsunami Selat Sunda 2018. PUMMA, yang ditempatkan di sepanjang garis pantai seperti Gebang, Canti, Panjang, Jambu, dan Pengandaran, dilengkapi dengan berbagai komponen canggih termasuk panel surya 100 WP, baterai 100Ah, minicomputer, sensor ultrasonik, dan kamera. Sistem ini bekerja dengan mengukur ketinggian permukaan air laut secara *real-time*, mengambil gambar kondisi lingkungan setiap 2 menit, dan mengirimkan data melalui jaringan ke server untuk dianalisis.

Buoy yang ditempatkan di perairan antara Pulau Sertung dan Pulau Panjang, dekat kawasan konservasi Anak Gunung Krakatau, berfungsi sebagai sistem deteksi awal di tengah lautan. Perangkat ini dilengkapi dengan berbagai sensor termasuk *Wind Direction* untuk mengukur arah angin, *Anemometer* untuk kecepatan angin, sensor tekanan air, dan GPS untuk pelacakan lokasi. Buoy menggunakan sistem tenaga surya dengan MPPT untuk operasionalnya dan mengirimkan data secara *real-time* ke pusat kontrol melalui sistem komunikasi nirkabel.



Gambar 7. Penempatan Buoy di laut lingkungan sekitar anak Gunung Krakatau

Sistem ini bekerja secara terintegrasi untuk memberikan peringatan dini yang lebih efektif. Data yang dikumpulkan dari PUMMA dan Buoy diolah dalam server pusat dan ditampilkan melalui *dashboard* IoT yang dilengkapi dengan sistem notifikasi. Informasi ini kemudian diteruskan ke BMKG sebagai otoritas yang berwenang dalam mitigasi bencana di Indonesia. Selain berfungsi sebagai sistem peringatan dini tsunami, perangkat-perangkat ini juga mengumpulkan data penting untuk penelitian kebencanaan di masa depan dan membantu dalam pemantauan kondisi ekosistem laut di sekitar kawasan konservasi Anak Gunung Krakatau.



Gambar 8. Posisi Buoy 001 tanggal 29 Agustus 2024

Buoy 001 sukses diluncurkan di perairan dekat Gunung Anak Krakatau pada tanggal 27 Agustus 2024. Alat ini berhasil mengirimkan data secara *real-time* selama beberapa hari. Namun, pada tanggal 29 Agustus, Buoy terdeteksi bergeser sejauh 528 meter ke arah Barat Laut menuju area tanpa sinyal, sehingga posisinya kini tidak diketahui.

Untuk mengatasi permasalahan ini, tim U-TEWS segera mempersiapkan peluncuran U-Buoy 002. Buoy kedua ini telah dilengkapi dengan teknologi yang lebih canggih, termasuk sistem penjangkaran yang telah dimodifikasi untuk meningkatkan stabilitas posisinya. Dengan memanfaatkan data lokasi terakhir Buoy 001, tim U-TEWS akan memulai misi pencarian .

2.6 JavaScript

JavaScript adalah bahasa pemrograman yang digunakan dalam pengembangan *web*, khususnya sebagai *Client Side Programming Language*. Bahasa ini berfungsi untuk meningkatkan interaktivitas dan dinamika halaman *web* dengan memungkinkan

pemrosesan dilakukan langsung di sisi klien, yaitu pada *browser* pengguna seperti Google Chrome atau Mozilla Firefox. JavaScript dapat disisipkan ke dalam dokumen HTML atau ditulis dalam *file* terpisah yang kemudian dihubungkan dengan dokumen HTML tersebut. JavaScript pertama kali dikembangkan pada pertengahan dekade 1990-an dan meskipun namanya mirip dengan bahasa pemrograman Java, keduanya adalah bahasa yang berbeda dengan tujuan dan sintaksis yang berbeda pula. JavaScript membantu dalam mengendalikan bagaimana halaman *web* berinteraksi dengan penggunanya, menjadikannya alat yang sangat berguna dalam pengembangan aplikasi *web* modern [7].

2.7 ReactJS

React JS adalah *library* JavaScript yang dikembangkan oleh Facebook (sekarang Meta) untuk membangun *User Interface (UI)* yang interaktif dan dinamis. Sejak diperkenalkan pada tahun 2013, React telah menjadi salah satu alat paling populer dalam pengembangan aplikasi *web* dan *mobile*. React JS memungkinkan pengembang untuk membuat aplikasi dengan memecah UI menjadi komponen-komponen kecil yang dapat digunakan kembali (*reusable components*). Ini memberikan fleksibilitas dalam pengembangan, karena setiap komponen dapat dikelola secara independen, memudahkan pemeliharaan dan pengembangan lebih lanjut. React menggunakan konsep Virtual DOM, yang merupakan representasi virtual dari DOM (*Document Object Model*) yang ada di *browser*. Ketika ada perubahan pada state atau data, React akan memperbarui Virtual DOM terlebih dahulu dan kemudian membandingkannya dengan DOM asli untuk menentukan perubahan yang perlu diterapkan. Proses ini membuat pembaruan UI menjadi lebih efisien dan cepat, karena hanya bagian yang berubah yang akan diperbarui, bukan seluruh halaman [13].

2.8 REST-API

Aplikasi dengan arsitektur client-server dapat dibangun dengan sistem operasi yang berbeda pada sisi client dan sisi server. REST API digunakan sebagai perantara dua sistem operasi untuk saling berhubungan. REST API adalah kumpulan fungsi, aturan, perintah, dan protokol yang berisi standar umum tentang cara bertukar informasi antara klien dan server [5].

REST API adalah salah satu jenis dari API yang didasarkan pada arsitektur REST (Representational State Transfer). REST sendiri adalah sebuah arsitektur perangkat lunak yang terdiri dari aturan dan konvensi yang digunakan untuk membuat web service. Dalam REST API, setiap sumber daya (resource) diidentifikasi dengan URI (Uniform Resource Identifier) dan diakses melalui metode HTTP seperti GET, POST, PUT, DELETE, dan sebagainya.

2.9 Tailwind CSS

Tailwind CSS adalah sebuah *framework* CSS yang memungkinkan perancangan dan desain *User Interface (UI)* dengan cepat. *Framework* ini cocok digunakan oleh pengembang frontend pemula karena menyediakan komponen-komponen bawaan yang lengkap. Tailwind CSS menggunakan pendekatan *utility-first* yang memungkinkan pengembang untuk menyesuaikan konfigurasi dan tata letak UI, termasuk ukuran bordir, warna, jenis huruf, bayangan, titik putus, dan lain-lain, sesuai dengan kebutuhan mereka [11].

2.10 Visual Studio Code

Visual Studio Code adalah *editor* kode yang dikembangkan oleh Microsoft dan tersedia untuk sistem operasi Windows, Linux, dan macOS. Editor ini dilengkapi dengan berbagai fitur, termasuk dukungan untuk *debugging*, integrasi kontrol versi Git dengan GitHub, penyorotan sintaksis, penyelesaian kode yang cerdas, snippet, dan kemampuan *refactoring* kode. Selain itu, *Visual Studio Code* juga dapat disesuaikan secara luas,

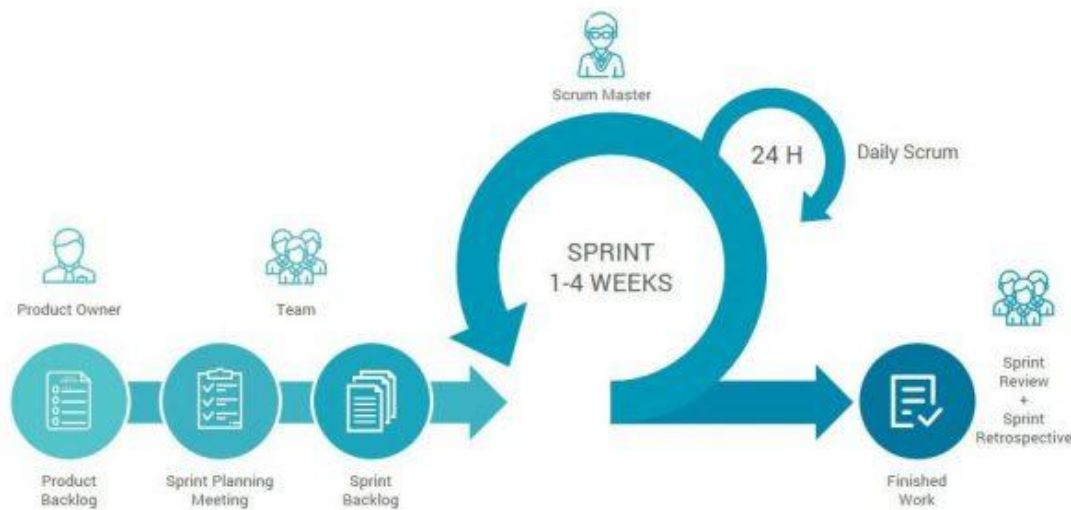
memungkinkan pengguna untuk mengubah tema, pintasan *keyboard*, dan preferensi lainnya, serta menginstal ekstensi untuk menambah fungsionalitas [3].

2.11 Figma

Figma adalah *tool* yang biasanya digunakan untuk membuat desain *User Interface (UI)* aplikasi *mobile*, situs *web*, desktop, serta berbagai *platform* lainnya. Umumnya, Figma dipergunakan oleh orang-orang untuk membuat desain UI/UX, desain *web*, dan bidang serupa lainnya. Salah satu keunggulan utama Figma, selain fitur yang lengkap seperti Adobe XD, adalah kemampuan kolaboratifnya. Figma memungkinkan *tim* untuk bekerja sama dalam satu *file* desain, meski anggota tim berada di lokasi yang berbeda. Setiap anggota *tim* dapat memberikan masukan, meninggalkan komentar, melihat perubahan yang dilakukan oleh anggota lain, dan melakukan penyuntingan desain secara simultan. Fitur kolaborasi ini menjadikan Figma sangat diminati oleh banyak desainer UI/UX untuk membuat prototipe desain *website* dan aplikasi dengan cara yang efektif. Selain itu, Figma juga menyediakan fitur kerja secara *real-time*, sehingga perubahan yang terjadi disimpan secara otomatis [10].

2.12 Scrum

Scrum adalah sebuah kerangka kerja yang dirancang untuk mengatasi masalah kompleks dan bersifat adaptif, dengan tujuan untuk menghasilkan produk yang bernilai tinggi melalui kreativitas dan produktivitas kerja secara bersamaan [8]. Prinsip dasar dari *Scrum* adalah melakukan pekerjaan secara iteratif dan bertahap hingga mencapai waktu yang telah ditentukan, sehingga perangkat lunak yang dikembangkan dapat memenuhi kebutuhan konsumen. Dalam pengembangan perangkat lunak, metode *Scrum* dibagi menjadi beberapa bagian iterasi, yang masing-masing disebut sebagai *Sprint*. Alur pengembangan metode *scrum* dapat dilihat pada Gambar 1.



Gambar 9. Alur Kerja Scrum

1. *Product Backlog*

Product backlog adalah daftar urut dari semua fitur atau elemen yang diketahui harus ada dalam produk pada saat ini. *Product backlog* merupakan satu-satunya sumber kebutuhan untuk segala perubahan yang harus diterapkan pada produk [8].

2. *Sprint Planning*

Sprint planning adalah proses penentuan tugas yang akan dilaksanakan selama *sprint* yang direncanakan saat *sprint planning*. Proses ini melibatkan kolaborasi dari semua *scrum team* [8].

3. *Sprint Backlog*

Sprint Backlog adalah daftar dari *Product Backlog* yang dipilih untuk *sprint* dengan tujuan mencapai *Sprint Goal*. *Sprint Backlog* mencakup proyeksi dari *Development Team* tentang fitur-fitur yang akan dimasukkan ke dalam iterasi berikutnya dan tugas-tugas yang harus diselesaikan untuk menyelesaikan iterasi tersebut [8].

4. *Daily Scrum*

Daily scrum merupakan pertemuan harian untuk tim pengembangan yang memiliki durasi maksimal 15 menit. Pertemuan ini diadakan setiap hari selama periode *sprint*. Fokus dari pertemuan ini adalah untuk meningkatkan kolaborasi dan kinerja tim dengan

mengevaluasi pekerjaan yang telah dilakukan sejak pertemuan sebelumnya dan merencanakan langkah-langkah selanjutnya dalam *sprint* [8].

5. *Sprint Review*

Sprint review adalah kegiatan yang dilakukan pada akhir sprint untuk mengevaluasi peningkatan dan menyesuaikan *product backlog* jika diperlukan. Dalam *sprint review*, tim *scrum* dan *product owner* bekerja sama untuk meninjau apa yang telah dicapai selama sprint tersebut [8].

6. *Sprint Retrospective*

Sprint retrospective merupakan waktu yang diberikan kepada tim *scrum* untuk mengevaluasi kinerjanya sendiri dan merencanakan perbaikan yang akan dilakukan dalam *sprint* berikutnya. *Sprint retrospective* dilakukan setelah *sprint review* dan sebelum *sprint planning* berikutnya [8].

Metode scrum juga memiliki role atau bisa juga disebut scrum team. Scrum team dapat menentukan cara terbaik untuk menyelesaikan pekerjaan setiap anggota, tanpa adanya pengarahan dari pihak luar tim. Scrum team terdiri dari tiga role, yakni *product owner*, *development team*, dan *scrum master* [8].

a. *Product Owner*

Product owner adalah orang yang bertanggung jawab untuk mengoptimalkan nilai bisnis dari produk yang dikembangkan oleh *development team*. Cara pelaksanaannya dapat berbeda-beda antara organisasi, *scrum team*, dan individu. *Product owner* memiliki fleksibilitas untuk melakukan tugas-tugas tersebut sendiri atau meminta *development team* untuk melakukannya [8].

b. *Development Team*

Development team terdiri dari sekelompok ahli yang bekerja untuk menghasilkan increment "Selesai" yang dapat dirilis pada akhir setiap *sprint*. *Development team* diberi wewenang oleh organisasi untuk merencanakan dan mengelola tugas mereka sendiri. Kolaborasi dari anggota tim ini diharapkan dapat meningkatkan efisiensi dan efektivitas keseluruhan dari *development team* [8].

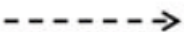
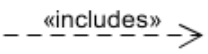
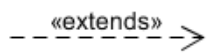
c. *Scrum Master*

Scrum master memiliki tanggung jawab untuk memperkenalkan dan mendukung penerapan *scrum*. Tugas ini dilakukan dengan membimbing *scrum team* agar memahami konsep, praktik, aturan, dan nilai-nilai dasar *scrum*. *Scrum master* bertindak sebagai pemimpin yang melayani kebutuhan *scrum team* [8].

2.13 Use Case Diagram

Use case atau diagram *use case* menggambarkan kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu [21]. Simbol - simbol *use case diagram* dapat dilihat pada Tabel 1.

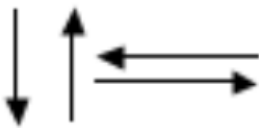
Tabel 1. Simbol *Use Case Diagram*

No	Simbol	Keterangan
1		Aktor : mewakili peran orang atau alat ketika berkomunikasi dengan <i>use case</i> .
2		<i>Use case</i> : deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
3		<i>Association</i> : Abstraksi dari penghubung antara aktor dengan <i>use case</i> .
4		<i>Generalisasi</i> : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i> .
5		Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
6		Menunjukan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

2.14 Activity Diagram

Activity Diagram merupakan gambaran *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis [11]. Simbol-simbol *activity diagram* dapat dilihat pada Tabel 2.

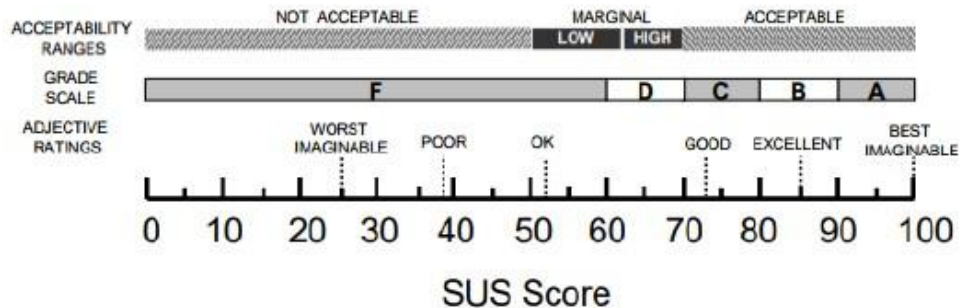
Tabel 2. Simbol *Activity Diagram*

No	Simbol	Nama	Keterangan
1		<i>Initial Node</i>	Sebuah diagram aktivitas yang memiliki status awal.
2		<i>Final Node</i>	Sebuah diagram aktivitas yang memiliki status akhir.
3		<i>Activity</i>	Aktivitas atau perilaku yang dilakukan sistem atau aktor.
4		<i>Decision</i>	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
5		<i>Line Connector</i>	Digunakan untuk menghubungkan satu simbol dengan simbol yang lain

2.15 System Usability Scale

System Usability Scale (SUS) merupakan alat penilaian yang andal untuk mengukur *usability* dari sebuah produk. SUS terdiri dari 10 item kuesioner dengan lima pilihan respons untuk responden, dari Sangat setuju sampai dengan Sangat tidak setuju. Awalnya dibuat oleh John Brooke pada tahun 1986, memungkinkan untuk mengevaluasi berbagai macam produk dan layanan, termasuk perangkat keras, perangkat lunak, perangkat seluler, situs *web* dan aplikasi [2].

Untuk menghitung skor *SUS*, pertama jumlahkan kontribusi skor dari setiap item. Kontribusi skor setiap item akan berkisar antara 0 hingga 4. Untuk item 1,3,5,7, dan 9 kontribusi skor adalah posisi skala dikurangi 1. Untuk item 2,4,6,8 dan 10, kontribusinya adalah 5 dikurangi posisi skala. Kalikan jumlah skor dengan 2,5 untuk mendapatkan nilai keseluruhan. Skor *SUS* memiliki rentang 0 hingga 100 [2].



Gambar 10. Skor Penilaian *SUS*.

2.16 Black Box Testing

Black box testing merupakan pengujian yang berfokus pada kebutuhan fungsional pada suatu sistem. Penguji dalam metode ini cukup mengetahui tentang masukan yang akan diproses oleh sistem dan keluarannya yang sesuai, tanpa harus mengetahui bagaimana di dalam sistem itu bekerja [12].

Black box testing adalah salah satu teknik pengujian perangkat lunak yang berfokus pada aspek fungsional, terutama input dan *output* dari aplikasi, tanpa memperhatikan kode program dari aplikasi tersebut. Pendekatan ini menguji apakah aplikasi sudah sesuai dengan yang diharapkan, dengan memverifikasi fungsionalitas sistem dari perspektif pengguna. Tes ini umumnya tidak menguji proses internal sistem, sehingga hasilnya didasarkan pada pengalaman pengguna aplikasi [19].

2.17 A/B Testing

A/B testing merupakan suatu metode pengujian dengan menggunakan dua jenis *website* untuk membandingkan dan mengetahui *website* yang lebih unggul. *A/B testing* biasanya digunakan untuk menentukan perbandingan nilai dari *website*. Adapun beberapa elemen yang dapat diuji dengan menggunakan *A/B testing* seperti logo, tombol, konten, layout halaman, *call-to-action*, dan lain lain.

Tujuan dilakukannya *A/B testing* adalah untuk mendapatkan informasi atau data hasil pengujian dari kedua *website* tersebut sehingga anda dapat meningkatkan atau memperbaiki *website* yang dibangun. Selain itu, *A/B testing* juga dapat digunakan untuk memahami target *audience*, melihat perilaku konsumen sesuai dengan tren atau data terkini, meningkatkan *website traffic* dan meningkatkan kualitas tampilan atau elemen dalam *website*. Dalam menganalisa pengujian ini, terdapat beberapa indikator yang dapat diterapkan seperti *conversion rate*, *UI UX* dari tampilan *website* atau seberapa lama pengguna dapat bertahan dalam *website* tersebut. [17].

2.18 Penelitian Terkait

Terdapat beberapa penelitian terkait yang menjadi perbandingan dan rujukan mengenai metode dan hasil yang dicapai. Berikut merupakan ulasan dari beberapa penelitian terkait:

2.18.1 Membangun *Frontend Website* Sistem Informasi Pemantauan Dan Peringatan Dini Bencana Tsunami Berbasis IOT Dengan Menggunakan *Library ReactJs*

Penelitian yang dilakukan oleh Ranto pada tahun 2023 bertujuan untuk mengembangkan sebuah *frontend website* sistem informasi pemantauan dan peringatan dini bencana tsunami berbasis IoT. Tujuan utama penelitian ini adalah untuk membangun sebuah sistem yang dapat memantau dan memberikan peringatan dini tentang potensi tsunami, terutama di wilayah-wilayah pantai Indonesia yang rentan

terhadap bencana alam jenis ini. Penelitian ini menggunakan *library* ReactJS sebagai dasar pengembangan *frontend website* yang responsif dan dinamis. Protokol MQTT digunakan untuk menerima data *real-time* dari server yang mengumpulkan data sensor, termasuk ketinggian air, RMS, *threshold*, *alert level*, status, dan temperatur. Selain itu, protokol HTTP digunakan untuk mengkonsumsi API yang menampilkan data dalam bentuk grafik yang mudah dipahami oleh pengguna. Dengan demikian, sistem ini dapat memberikan solusi nyata dalam meningkatkan keselamatan masyarakat di wilayah pantai Indonesia dengan menggunakan teknologi modern dan metodologi pengembangan yang efektif [3].

2.18.2 Redesain Website Pemerintah dengan Pengembangan Menggunakan Kerangka Kerja ASP.NET dan fitur Razor Pages

Penelitian yang dilakukan oleh Asgarindo Dwiki Ibrahim Adji dan tim pada tahun 2023 bertujuan untuk melakukan redesign *website* Dinas Kebudayaan dan Pariwisata Kabupaten Brebes dengan menggunakan teknologi ASP.NET dan fitur Razor Pages. Proyek ini berfokus pada pembaruan desain dan fungsionalitas *website* guna meningkatkan kepuasan pengunjung dan pengguna, mengatasi masalah terkait *User Interface (UI)* pengguna, navigasi, serta penyajian konten. Melalui pendekatan ini, diharapkan dapat menciptakan *platform* yang lebih menarik, mudah digunakan, dan informatif dalam mempromosikan potensi kebudayaan dan pariwisata daerah. Metodologi yang digunakan dalam penelitian ini adalah model *Waterfall* dari SDLC (*Software Development Life Cycle*), yang melibatkan analisis mendalam terhadap *website* yang ada, identifikasi masalah, perancangan ulang desain, implementasi menggunakan ASP.NET dan Razor Pages, serta pemeliharaan berkelanjutan. Hasil dari penelitian ini menunjukkan bahwa pembaruan desain visual dan restrukturisasi navigasi dapat meningkatkan daya tarik *website*, memberikan pengalaman pengguna yang lebih baik, serta berkontribusi positif terhadap citra Dinas Kebudayaan dan Pariwisata Kabupaten Brebes serta perekonomian local [4].

2.18.3 Pengembangan *Dashboard* Admin Untuk Bukupedia

Penelitian yang dilakukan oleh Adnan Azizi, Muhammad Naufal Al Ghazali, Ok Syahdan Khair, dan Ziyen Tsabit Saifullah Kusnandar pada tahun 2024 bertujuan untuk mengembangkan *dashboard* admin untuk Bukupedia, sebuah *platform* perdagangan buku online yang berkembang pesat. Penelitian ini berfokus pada penyediaan alat manajemen yang canggih dan intuitif untuk mengelola konten, memantau kinerja penjualan, dan mengelola pengguna secara efektif. Metode penelitian yang digunakan meliputi analisis sistem dan pengembangan perangkat lunak dengan berbagai alat dan bahasa pemrograman seperti Golang, JavaScript, Postman, XAMPP, Tailwind CSS, GoFiber, dan Visual Studio Code. Penggunaan Golang memberikan keunggulan dalam hal kecepatan dan skalabilitas, sementara JavaScript digunakan untuk meningkatkan interaktivitas halaman *web*. Hasil dari penelitian ini menunjukkan bahwa *dashboard* yang dikembangkan memberikan *User Interface (UI)* yang intuitif bagi admin untuk melakukan berbagai operasi CRUD (*Create, Read, Update, Delete*) secara efisien. Dengan demikian, diharapkan *dashboard* ini dapat meningkatkan efisiensi manajemen karya sastra dan pengelolaan penulis, serta membantu Bukupedia dalam membuat keputusan yang lebih cepat dan tepat [15].

2.18.4 Monitoring and Warning for Tsunamis Generated by Volcanoes

Laporan teknis dari UNESCO yang berjudul "Monitoring and Warning for Tsunamis Generated by Volcanoes" memberikan wawasan mendalam tentang fenomena tsunami yang dihasilkan oleh aktivitas vulkanik. Setelah kejadian tsunami merusak yang terjadi di Greenland dan Indonesia pada tahun 2017 dan 2018, laporan ini disusun untuk mengidentifikasi kebutuhan akan sistem peringatan yang efektif untuk tsunami yang dihasilkan oleh sumber non-seismik. Tsunami vulkanik, meskipun hanya menyumbang sekitar 6% dari semua tsunami yang tercatat dalam sejarah, dapat memiliki dampak yang signifikan, seperti yang terlihat dalam peristiwa-peristiwa terbaru. Laporan ini menjelaskan berbagai mekanisme yang dapat memicu tsunami vulkanik, termasuk longsoran subaerial dan bawah laut, aliran piroklastik, serta kolaps kaldera. Selain itu,

laporan ini membahas teknik pemodelan numerik yang digunakan untuk memahami proses pembentukan dan propagasi tsunami vulkanik, serta memberikan contoh sistem pemantauan dan peringatan yang ada di berbagai lokasi seperti Stromboli di Italia dan Anak Krakatau di Indonesia. Dengan menyediakan rekomendasi untuk pengembangan sistem peringatan dan peningkatan kesiapsiagaan, laporan ini berfungsi sebagai panduan penting bagi negara-negara anggota UNESCO dalam menghadapi risiko tsunami yang terkait dengan aktivitas vulkanik [6].

2.18.5 STROMBOLI VOLCANIC TSUNAMI WARNING SYSTEM (ITALY)

Sistem Peringatan Tsunami Vulkanik Stromboli di Italia merupakan inisiatif penting yang dirancang untuk mengatasi risiko tsunami yang dihasilkan oleh aktivitas vulkanik, khususnya di pulau Stromboli yang terkenal dengan aktivitas vulkaniknya yang tinggi. Setelah serangkaian tsunami merusak yang disebabkan oleh letusan dan longsoran, sistem ini dibangun untuk memberikan peringatan dini kepada penduduk dan wisatawan mengenai potensi tsunami. Sistem ini mengandalkan berbagai teknologi pemantauan, termasuk seismometer untuk mendeteksi gempa bumi, pengukuran deformasi tanah untuk mengidentifikasi perubahan pada gunung berapi, serta pemantauan satelit untuk mengawasi morfologi pulau dan sekitarnya.

Komponen utama dari sistem ini mencakup pemodelan numerik yang canggih untuk memprediksi bagaimana tsunami dapat terbentuk dan menyebar setelah peristiwa vulkanik. Dengan menggunakan data dari berbagai sumber, sistem dapat memberikan informasi yang akurat mengenai waktu, lokasi, dan dampak potensial dari tsunami. Hal ini sangat penting mengingat pulau Stromboli sering kali menjadi tujuan wisata, sehingga keselamatan pengunjung serta penduduk lokal menjadi prioritas utama.

Pentingnya sistem peringatan ini tidak hanya terletak pada kemampuannya untuk mendeteksi dan memberikan peringatan terhadap tsunami, tetapi juga dalam meningkatkan kesadaran masyarakat tentang risiko yang terkait dengan aktivitas vulkanik. Dengan adanya sistem ini, diharapkan dapat meminimalkan dampak bencana dan meningkatkan kesiapsiagaan masyarakat dalam menghadapi kemungkinan

terjadinya tsunami akibat letusan gunung berapi. Melalui upaya kolaboratif antara berbagai lembaga dan penggunaan teknologi modern, sistem peringatan ini menjadi contoh nyata dari mitigasi risiko bencana yang efektif di daerah rawan bencana vulkanik.

3.2 Alat dan Bahan Penelitian

3.2.1 Alat Penelitian

Adapun alat yang digunakan dalam penelitian ini adalah sebagai berikut:

Tabel 4. Alat Penelitian

No	Nama	Spesifikasi	Deskripsi
1	Laptop	Prosesor AMD Ryzen 3 4300U with Radeon Graphics, RAM 8GB, Storage 475GB, Windows 11 64-bit	Perangkat keras yang digunakan untuk mengembangkan aplikasi dan menulis laporan penelitian.
2	Visual Studio Code	Versi 1.85.1	<i>Tools</i> yang digunakan sebagai <i>code editor</i> untuk membuat dan mengedit <i>source code</i> dalam mengembangkan aplikasi
3	Figma	Versi 116.10.9	<i>Tools</i> yang digunakan untuk merancang desain <i>User Interface (UI)</i>
4	Trello	Online	<i>Tools</i> yang digunakan untuk memvisualkan alur kerja
5	Draw.io	Versi 15.8.7	<i>Tools</i> yang digunakan untuk membuat <i>flowchart</i> , <i>activity diagram</i>
6	Umlet	Versi 14.3.0	<i>Tools</i> yang digunakan untuk membuat <i>use case diagram</i>

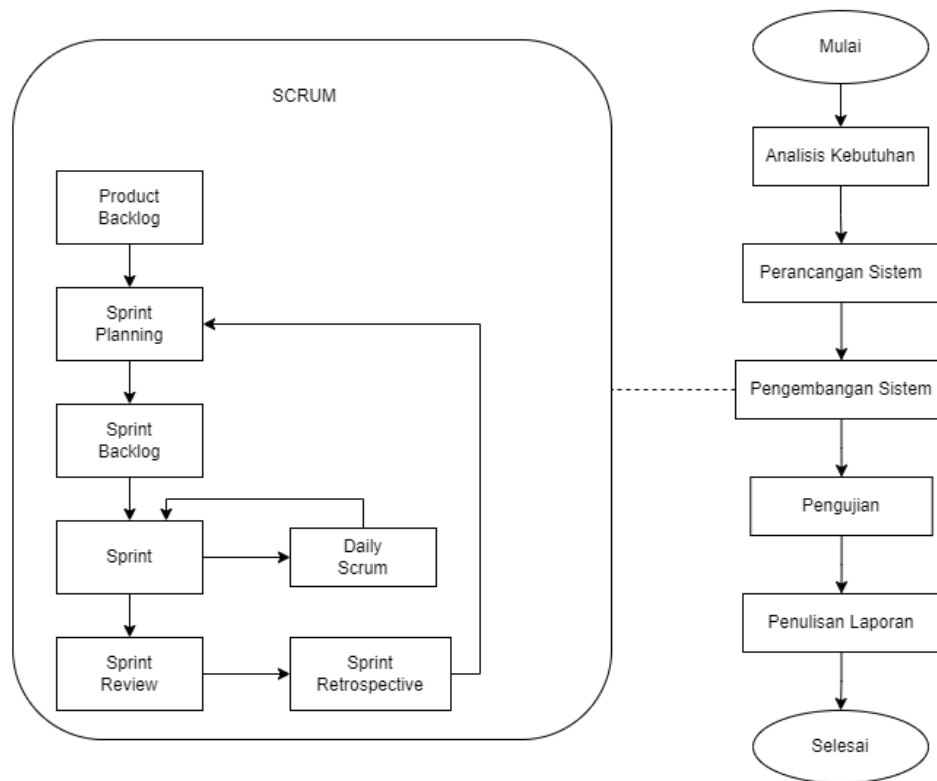
3.2.2 Bahan Penelitian

Bahan yang digunakan dalam pengerjaan penelitian dan pengembangan sistem ini berupa data yang didapatkan dari backend yang mengirimkan data yang berasal dari REST API

Tabel 5. Bahan berupa data sensor dari raspberry pi yang dikirimkan oleh server menggunakan *REST API*

NO	Data yang diterima	Keterangan
1	<i>Detail</i>	Data yang JSON yang menunjukkan status yang tersiri dari sensor, lokasi, provider, ketinggian air terakhir, waktu pengiriman data terakhir, waktu pengiriman data dari sensor ke <i>raspberry pi</i> .
2	<i>Datetime</i>	Data yang menunjukkan tanggal.
3	<i>Waterlevel</i>	Data ketinggian air laut.
4	<i>Voltage</i>	Data yang menunjukkan tegangan baterai perangkat <i>Raspberry pi</i>
5	<i>Temperature</i>	Data yang menunjukkan suhu perangkat <i>Raspberry pi</i>
6	<i>Image</i>	Data image kondisi dalam waktu 5 menit terakhir.
7	<i>forecast30</i>	Data <i>forecast</i> dari 30 data terakhir.
8	<i>Forcast300</i>	Data <i>forecast</i> dari 300 data terakhir.
9	<i>Rms</i>	Data rata ² dari <i>alertlevel</i> .
10	<i>Threshold</i>	Data sebagai ambang batas peringatan.
11	<i>Alertlevel</i>	Data selisih antara <i>Waterlevel</i> dengan <i>forecast</i> .
12	<i>Status</i>	Data yang jika <i>alertlevel</i> melebihi <i>threshold</i> maka menjadi <i>warning</i> jika tidak menjadi <i>safe</i> .

3.3 Tahapan Penelitian



Gambar 11. Tahapan Penelitian

Gambar 11 mengilustrasikan proses penelitian yang terdiri dari dua komponen utama: alur penelitian umum dan metodologi Scrum untuk pengembangan sistem. Alur penelitian dimulai dengan analisis kebutuhan melalui wawancara untuk mengidentifikasi kebutuhan fungsional dan non-fungsional proyek. Selanjutnya, tahap perancangan sistem mengolah informasi tersebut menjadi *use case diagram*, *activity diagram*, dan *user interface (UI)*. Tahap pengembangan sistem mengadopsi metodologi Scrum, yang mencakup elemen-elemen seperti *Product Backlog*, *Sprint Planning*, *Sprint Backlog*, *Sprint*, *Daily Scrum*, *Sprint Review*, dan *Sprint Retrospective*. Pendekatan ini memungkinkan pengembangan yang dilakukan secara bertahap dan terus-menerus, dengan penekanan pada kerja sama tim dan kemampuan untuk beradaptasi. Setelah pengembangan selesai, dilakukan pengujian menggunakan metode

blackbox testing dan *system usability scale*. Proses penelitian diakhiri dengan penulisan laporan, menghasilkan siklus pengembangan yang komprehensif dan terstruktur.

3.3.1 Analisis Kebutuhan

Tahap ini merupakan tahap awal dalam penelitian yang bertujuan untuk mengumpulkan informasi dasar proyek, yang kemudian akan dikembangkan lebih lanjut pada tahap berikutnya. Analisis kebutuhan dilakukan melalui wawancara langsung dengan Bapak Mona Arif Muda, pembimbing grup penelitian di Universitas Lampung, yang menjadi sumber utama informasi untuk mengembangkan proyek penelitian ini. Pertanyaan-pertanyaan yang diajukan berkaitan dengan proyek pengembangan aplikasi, termasuk tujuan, target pengguna, jenis-jenis pengguna, kebutuhan yang diperlukan, alur proses, fitur-fitur, dan batasan proyek. Berikut merupakan tabel kebutuhan sistem:

Tabel 6. Kebutuhan Sistem

No	Kebutuhan Sistem	Detail
1	Pengelolaan Informasi dari dashboard Buoy	Dalam dashboard buoy terdapat 2 device : FC (Floating Craft) buoy dan EWS buoy. FC buoy : lokasi GPS, orientasi (<i>roll, pitch, yaw</i>), dan arah kompas. EWS buoy : suhu, tingkat air, arah dan kecepatan angin, serta parameter sistem tenaga seperti tegangan dan arus dari panel surya serta status baterai.

3.3.2 Perancangan Sistem

Pada tahap ini, informasi yang telah diperoleh pada tahap sebelumnya dikembangkan lebih lanjut untuk menciptakan rancangan sistem yang jelas. Hal ini bertujuan agar pada saat memasuki tahap pengembangan sistem, semua aspek proyek telah terencana dengan baik dan sesuai dengan rancangan awal. Tahap perancangan sistem dalam

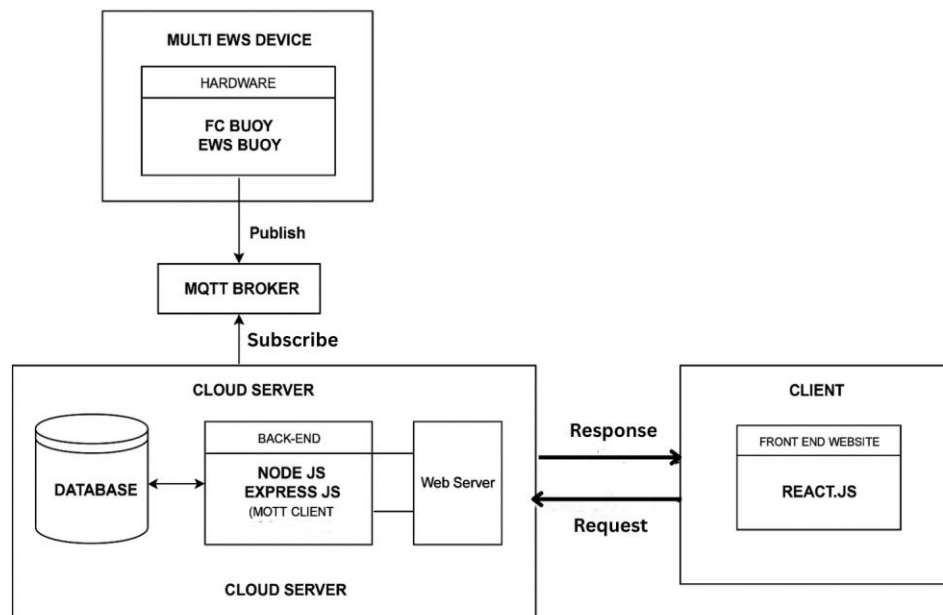
penelitian ini terbagi menjadi beberapa proses, mulai dari menentukan *environment capstone project*, menentukan *product backlog*, membuat *use case diagram*, membuat *activity diagram*, dan akhirnya merancang *user interface (UI)* aplikasi. Berikut merupakan tabel kebutuhan fungsional sistem:

Tabel 7. Kebutuhan Fungsional

No	Kebutuhan Fungsional	Detail
1	Mendapatkan informasi dari dashboard	Sistem harus dapat menyediakan informasi dari device EWS yang diperbaharui secara realtime.

3.3.2.1 Environment Capstone Project

Proyek penelitian yang akan dikembangkan merupakan sebuah *capstone project* yang terbagi menjadi beberapa bagian tergantung dengan kebutuhan proyek yang telah didapatkan pada tahap analisis kebutuhan. Berikut *environment capstone project* dalam penelitian ini.



Gambar 12. *Environment Project*

Pada Gambar 12 merupakan *Environment project* yang digunakan dalam pengembangan *Early Warning System (EWS)* ini dirancang untuk mengintegrasikan perangkat keras (*multi EWS device*), protokol komunikasi *MQTT*, server berbasis *cloud*, serta antarmuka pengguna berbasis *web*. Sistem diawali dari perangkat keras yang terdiri dari dua bagian buoy, yaitu *FC Buoy* dan *EWS Buoy*. Kedua perangkat Buoy ini bertugas mengumpulkan data lingkungan seperti tekanan air, suhu, dan getaran untuk mendeteksi potensi bencana alam tsunami. Data yang dikumpulkan oleh perangkat kemudian dikirimkan menggunakan protokol *MQTT*, di mana perangkat bertindak sebagai publisher yang mengirimkan data ke topik tertentu pada *MQTT Broker*.

MQTT Broker berfungsi sebagai penghubung komunikasi antara perangkat dan server dengan menerapkan arsitektur *publish-subscribe*. Dalam sistem ini, broker menerima data dari perangkat yang melakukan *publish*, kemudian meneruskannya ke *cloud server* yang telah berperan sebagai *subscriber*. Di dalam *cloud server*, terdapat komponen *back-end* yang dibangun menggunakan Node.js dan Express.js. *Back-end* ini juga dilengkapi dengan *MQTT client* (misalnya menggunakan *mqtt.js*) yang secara aktif

menerima data dari *broker* dan menyimpannya ke dalam *database* untuk kebutuhan *monitoring* dan analisis data.

Selain itu, *cloud server* juga memiliki web server yang berfungsi menangani permintaan (*request*) dari pengguna dan mengembalikan data yang dibutuhkan dalam bentuk (*respons*). Komponen *client* dalam sistem ini merupakan antarmuka pengguna berbasis *React.js* yang menampilkan data dalam bentuk grafik atau tabel secara interaktif dan *real-time*. *Website* ini memungkinkan pengguna untuk memantau kondisi lingkungan secara langsung berdasarkan data yang dikirimkan dari buoy.

Secara keseluruhan, sistem ini memiliki beberapa keunggulan, seperti kemampuan komunikasi data secara *real-time* berkat penggunaan protokol MQTT, arsitektur yang fleksibel dan mudah dikembangkan, serta tampilan pemantauan yang dapat diakses dari mana saja melalui website. Dengan pemisahan komponen yang jelas antara perangkat, komunikasi, server, dan antarmuka pengguna, sistem menjadi modular, mudah dipelihara, dan scalable untuk pengembangan ke depannya.

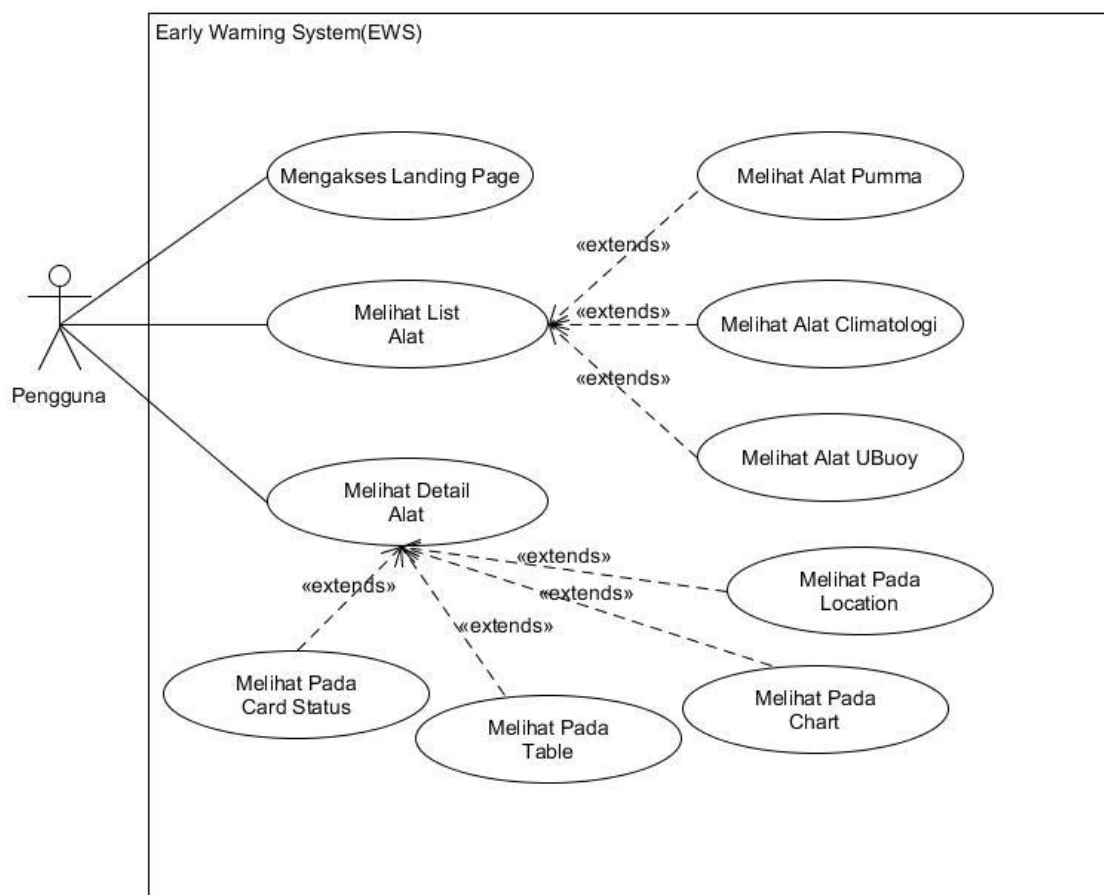
Tabel 8. *Environment Project*

No	Nama	Bagian
1	Akbar Andika Khadafi	<i>Front End Developer</i>
2	Muhammad Rafsanjani	<i>Back End Developer</i>
3	Ridho Tegar Pradana	<i>Hardware EWS</i>

Pada penelitian ini akan berfokus pada pengembangan *front end* dan menggunakan *React JS* dan *Tailwind CSS*.

3.3.2.2 Use Case Diagram

Pembuatan *use case diagram* dalam pemodelan sistem dilakukan dengan memperhatikan kebutuhan fungsional yang teridentifikasi dari tahap perancangan sistem. Diagram *use case* digunakan untuk menggambarkan sistem dengan jelas berdasarkan peran aktor dan interaksinya dengan setiap *use case* dalam sistem tersebut. *Use case* dapat memberikan gambaran umum kepada tim pengembang tentang fungsionalitas sistem, sehingga memudahkan proses implementasinya.



Gambar 13. Use Case Diagram Pengguna

Pada gambar 13 merupakan *Use Case Diagram* pada sistem *Early Warning System (EWS)* menggambarkan interaksi antara pengguna dengan fitur-fitur utama yang tersedia pada sistem. Pengguna merupakan aktor utama yang dapat mengakses berbagai fungsi, dimulai dari mengakses *landing page* untuk melihat informasi awal terkait sistem. Setelah itu, pengguna dapat melihat informasi alat yang tersedia, termasuk alat PUMMA, Climatologi, dan UBuoy. Proses pemilihan alat ini merupakan bagian yang dapat dilakukan dalam tahapan melihat informasi alat, sehingga hubungan antar *use case* digambarkan menggunakan relasi <<extend>>.

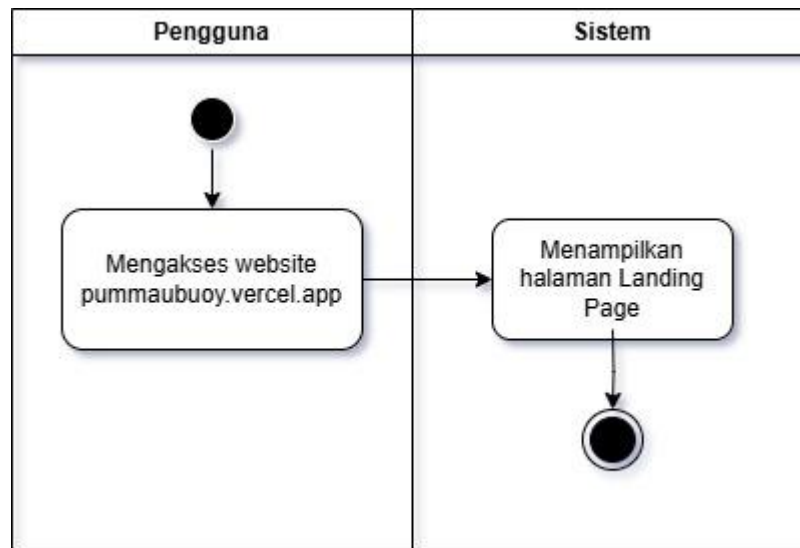
Setelah memilih alat tertentu, pengguna memiliki opsi untuk melihat detail dari alat tersebut. Aksi ini direpresentasikan oleh *use case* “Melihat detail alat”, yang memiliki hubungan <<extend>> terhadap beberapa *use case* tambahan, seperti melihat informasi pada *card status*, melihat data dalam bentuk tabel, grafik, serta lokasi dari alat. Relasi <<extend>> digunakan karena aktivitas-aktivitas tersebut bersifat opsional dan tidak selalu dijalankan oleh semua pengguna. Dengan kata lain, pengguna dapat memilih untuk hanya melihat salah satu jenis tampilan detail sesuai kebutuhannya.

Penggunaan relasi <<extend>> pada diagram ini bertujuan untuk menunjukkan proses yang opsional secara terstruktur, serta membantu memahami alur interaksi antara pengguna dan sistem dalam konteks *monitoring* data peringatan dini. Diagram ini sekaligus menggambarkan modularisasi fitur dan fleksibilitas antarmuka yang disediakan dalam sistem.

3.3.2.3 Activity Diagram

Activity diagram dibuat untuk memperjelas rancangan alur sistem dari setiap *use case* yang didalam *use case diagram*. Dalam *activity diagram* akan menggambarkan secara jelas aktivitas-aktivitas yang terjadi dan interaksi antara aktor dan sistem. Adanya *activity diagram* berfungsi untuk membantu tim pengembang dalam memahami alur sistem secara keseluruhan. Berikut adalah *activity diagram* sistem:

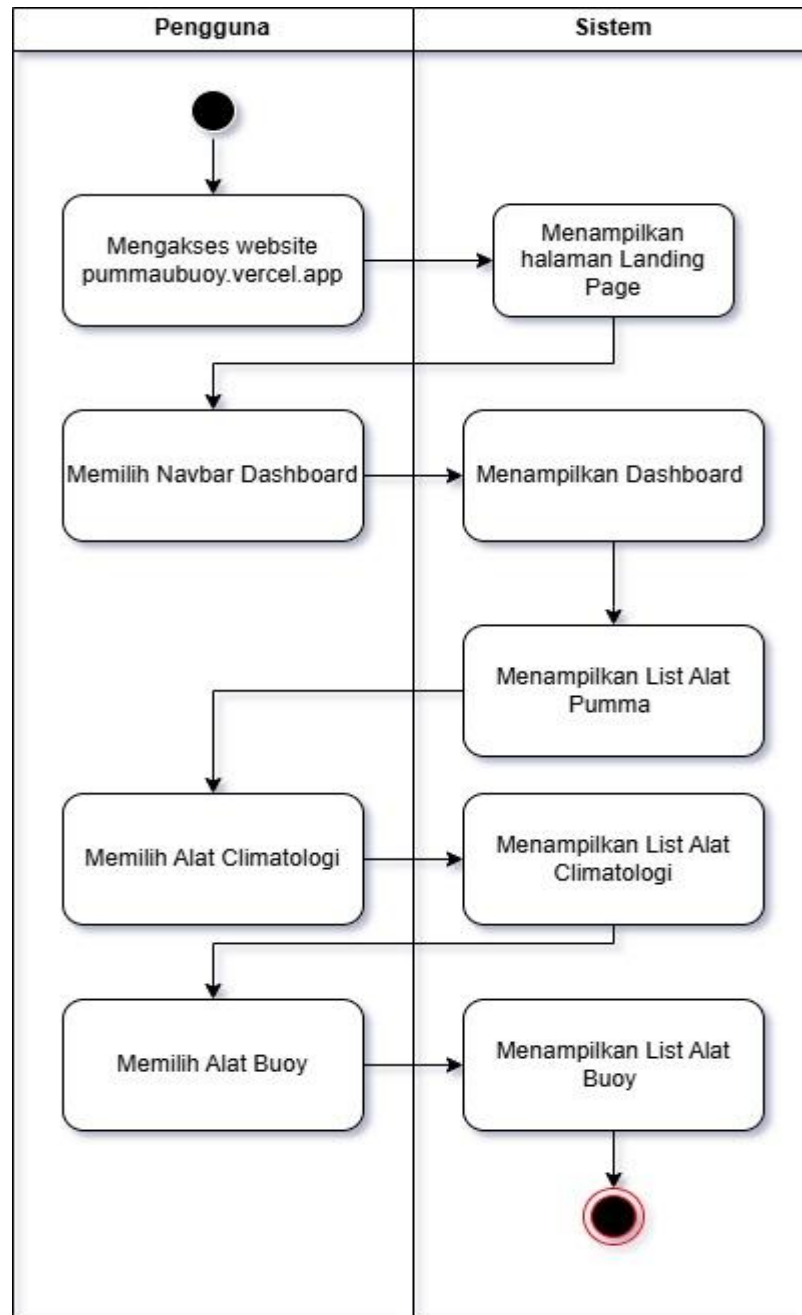
1. Mengakses *Landing Page*



Gambar 14. AD Mengakses *Landing Page*

Gambar 14 merupakan *activity diagram* mengakses halaman *Landing Page*. *Activity diagram* tersebut menggambarkan interaksi antara Pengguna dan Sistem dalam mengakses halaman *Landing Page*. Alur diagram dimulai saat pengguna mengakses *website* dengan membuka url “pummaubuoy.vercel.app”, lalu sistem akan merespon dengan menampilkan *Landing Page*.

2. Memilih *List Alat*



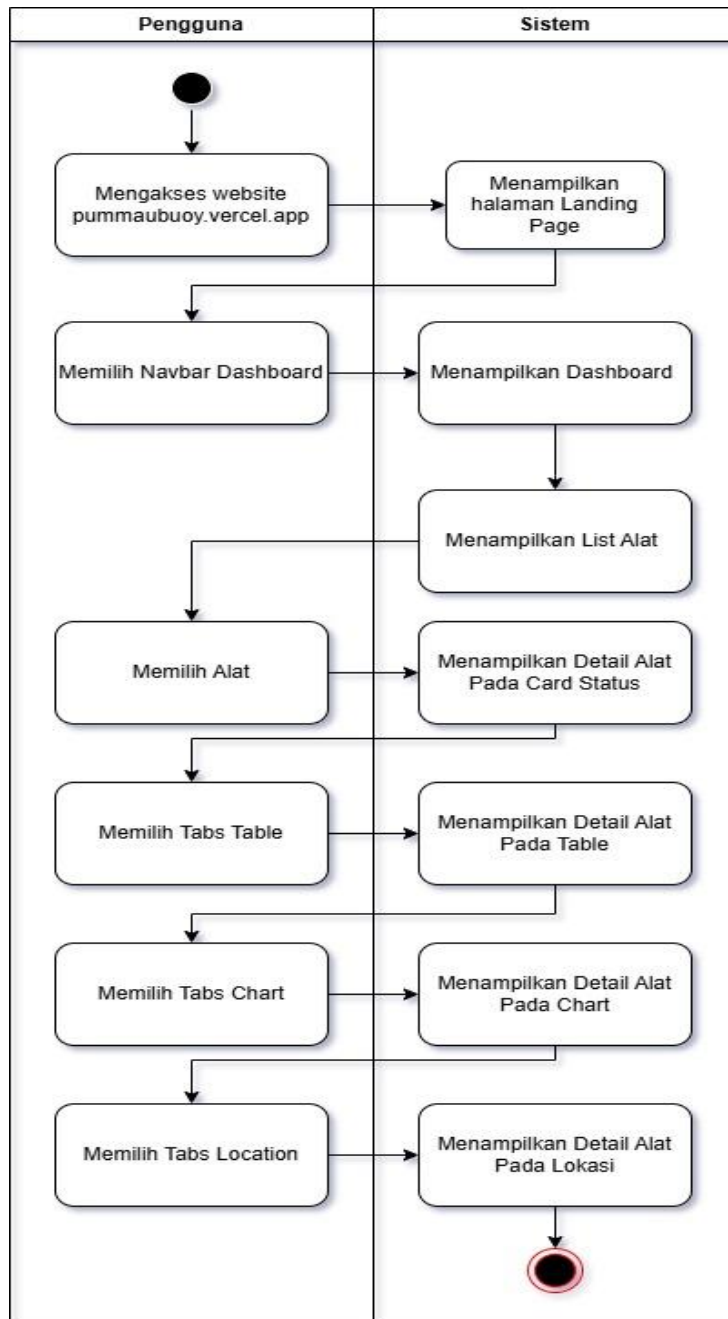
Gambar 15. AD Memilih *list device* yang ingin dibuka

Gambar 15 merupakan *activity diagram* menampilkan *device* pada *website PUMMA*. *Activity diagram* tersebut menggambarkan interaksi antara Pengguna dan Sistem dalam

mengelola data *device*. Alur diagram dimulai saat pengguna mengakses website `pummaubuoy.vercel.app`, kemudian sistem akan merespons dengan menampilkan halaman *Landing Page*. Selanjutnya, pengguna memilih navigasi *Dashboard*, yang kemudian direspons sistem dengan menampilkan halaman *Dashboard* beserta daftar alat (*device*) yang terpasang.

Pengguna kemudian dapat memilih salah satu kategori alat, yaitu *Climatologi*, sehingga sistem menampilkan daftar alat yang termasuk dalam kategori tersebut. Setelah itu, pengguna memilih alat *Buoy*, dan sistem akan menampilkan halaman yang berisi daftar *device Buoy* secara lebih rinci. Interaksi ini menunjukkan alur bagaimana sistem menampilkan data berdasarkan pilihan pengguna secara bertahap hingga pada kategori alat yang lebih spesifik.

3. Melihat Detail Pada Alat



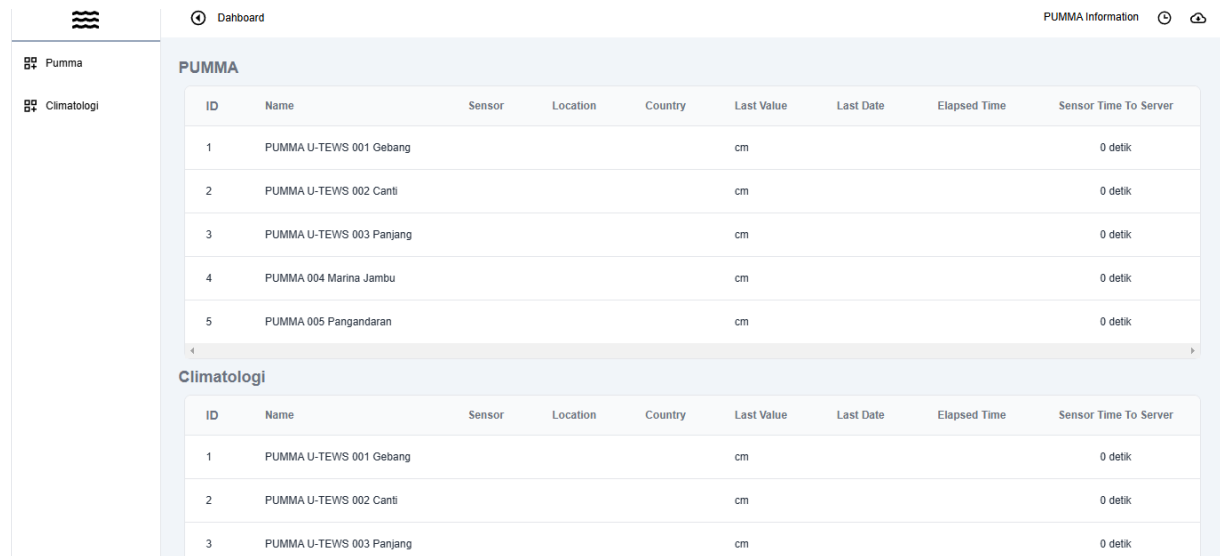
Gambar 16. AD Melihat Detail Pada Alat

Gambar 16 merupakan *activity diagram* yang menggambarkan proses interaksi antara pengguna dan sistem dalam menampilkan detail informasi alat (*device*) pada website

PUMMA. Proses diawali ketika pengguna mengakses website pummaubuooy.vercel.app, kemudian sistem merespons dengan menampilkan halaman *Landing Page*. Selanjutnya, pengguna memilih menu *Navbar Dashboard*, dan sistem menampilkan halaman *Dashboard* yang berisi daftar alat yang tersedia. Setelah itu, pengguna memilih salah satu alat dari daftar, dan sistem akan menampilkan informasi awal alat dalam bentuk *Card Status*. Untuk melihat data lebih rinci, pengguna dapat memilih beberapa *tabs* yang tersedia, seperti *Tabs Table* yang menampilkan data dalam bentuk tabel, *Tabs Chart* yang menyajikan data dalam bentuk grafik, serta *Tabs Location* yang menunjukkan informasi lokasi dari alat tersebut. Setiap tindakan yang dilakukan oleh pengguna akan direspons secara langsung oleh sistem dengan menampilkan data sesuai dengan bentuk visualisasi yang dipilih. Diagram ini menggambarkan bagaimana sistem memberikan fleksibilitas bagi pengguna untuk menelusuri informasi alat secara lebih detail melalui berbagai tampilan yang tersedia.

3.3.2.4 User Interface (UI) Website

User interface (UI) aplikasi dibuat berdasarkan pemodelan sistem yang telah dibuat yaitu *use case diagram* dan *activity diagram*. Desain UI yang dibuat adalah hasil akhir dari tahapan perancangan sistem, hasil ini akan menjadi acuan utama untuk membuat tampilan aplikasi pada tahap pengembangan. Desain UI berupa *mockup* yang dibuat menggunakan *tool figma*. Berikut beberapa *mockup* aplikasi:

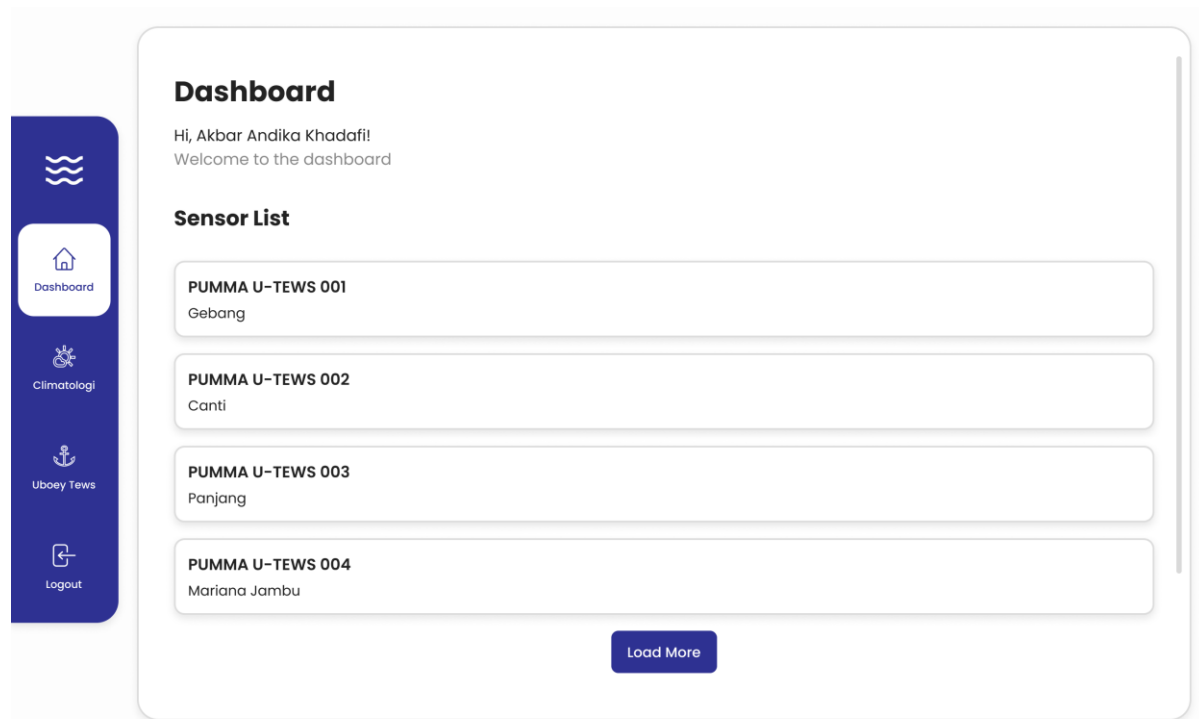


ID	Name	Sensor	Location	Country	Last Value	Last Date	Elapsed Time	Sensor Time To Server
1	PUMMA U-TEWS 001 Gebang				cm			0 detik
2	PUMMA U-TEWS 002 Canti				cm			0 detik
3	PUMMA U-TEWS 003 Panjang				cm			0 detik
4	PUMMA 004 Marina Jambu				cm			0 detik
5	PUMMA 005 Pangandaran				cm			0 detik

ID	Name	Sensor	Location	Country	Last Value	Last Date	Elapsed Time	Sensor Time To Server
1	PUMMA U-TEWS 001 Gebang				cm			0 detik
2	PUMMA U-TEWS 002 Canti				cm			0 detik
3	PUMMA U-TEWS 003 Panjang				cm			0 detik

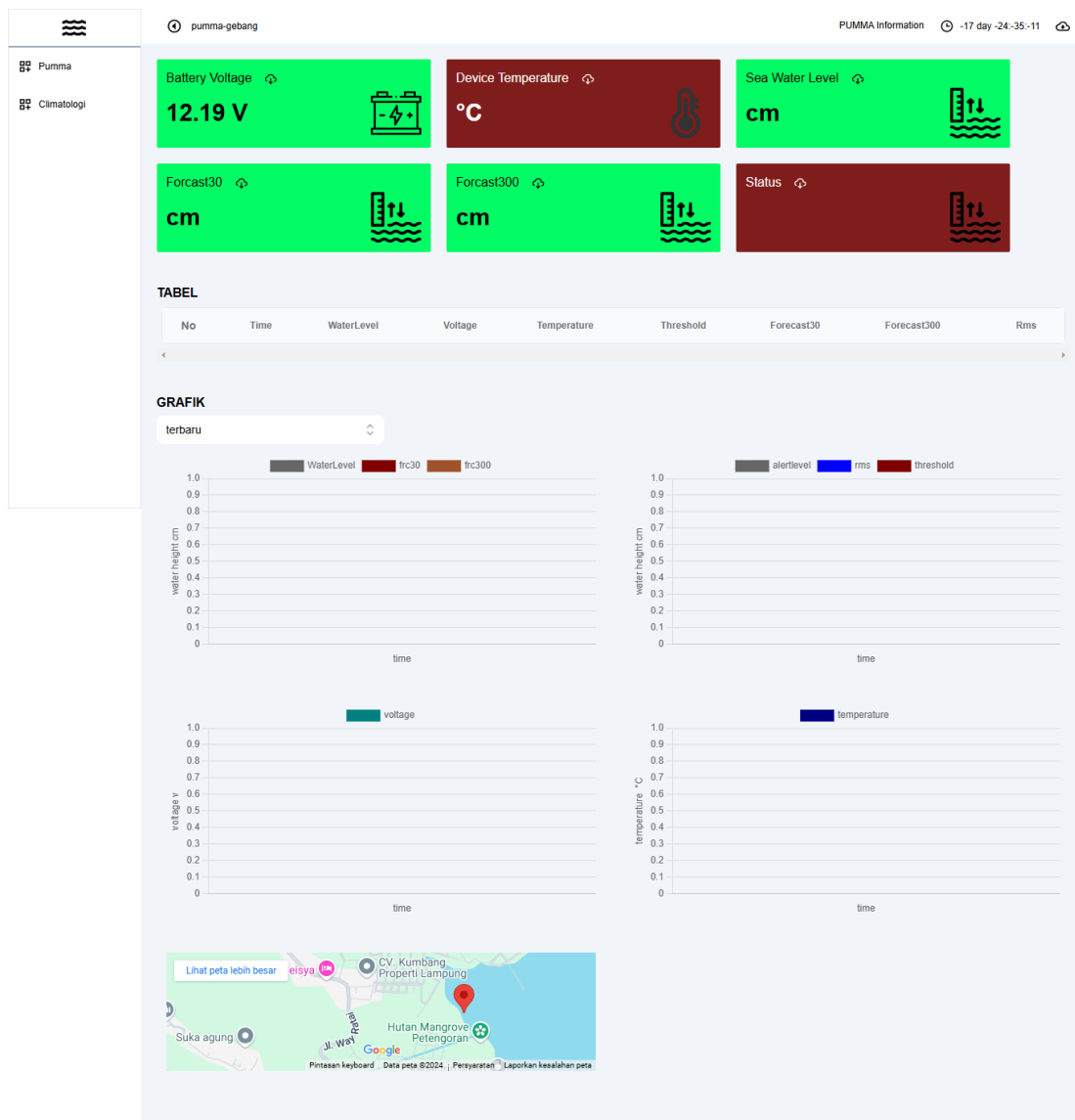
Gambar 17. Halaman *Dashboard* Sebelum Redesain

Gambar 17 merupakan halaman *website* sebelum redesain yang menampilkan pilihan nama alat yang berfungsi sebagai navigasi untuk mengakses halaman informasi dari alat yang dipilih. Pada sistem PUMMA, terdapat lima alat yang terpasang di berbagai lokasi, yaitu Gebang, Canti, Panjang, Marina Jambu, dan Pangandaran.



Gambar 18. Halaman *Dashboard* Sesudah Redesain

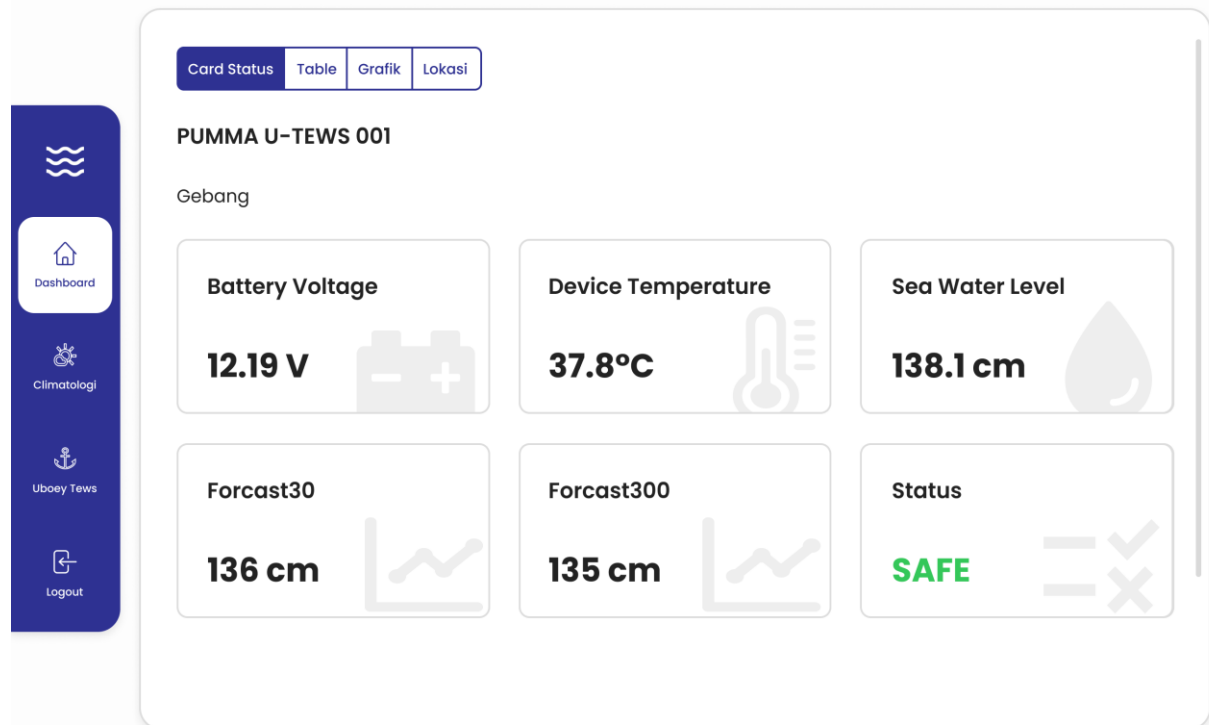
Gambar 18 merupakan halaman *website* setelah redesign yang menampilkan pilihan nama alat yang berfungsi sebagai navigasi untuk mengakses halaman informasi dari alat yang dipilih. Pada sistem PUMMA, terdapat lima alat yang terpasang di berbagai lokasi, yaitu Gebang, Canti, Panjang, Marina Jambu, dan Pangandaran.



Gambar 19. Rancangan Halaman Detail *Device* Sebelum Redesain

Gambar 19 merupakan halaman detail alat sebelum redesain yang menampilkan informasi lengkap terkait alat yang dipilih, dalam hal ini alat yang berada di Gebang. Halaman ini menyediakan berbagai parameter penting yang divisualisasikan dalam bentuk card status, tabel data, grafik, dan peta lokasi. Informasi yang disajikan meliputi tegangan baterai (*Battery Voltage*), suhu perangkat (*Device Temperature*), ketinggian air laut (*Sea Water Level*), prediksi kondisi dalam 30 menit (*Forecast30*) dan 300 menit

(*Forecast300*), serta status alat. Selain itu, data ditampilkan secara lebih rinci dalam tabel dan grafik interaktif yang memudahkan analisis tren, ditambah dengan peta untuk menunjukkan lokasi alat secara geografis. Halaman ini dirancang agar informasi dapat diakses dengan lebih mudah dan informatif bagi pengguna.

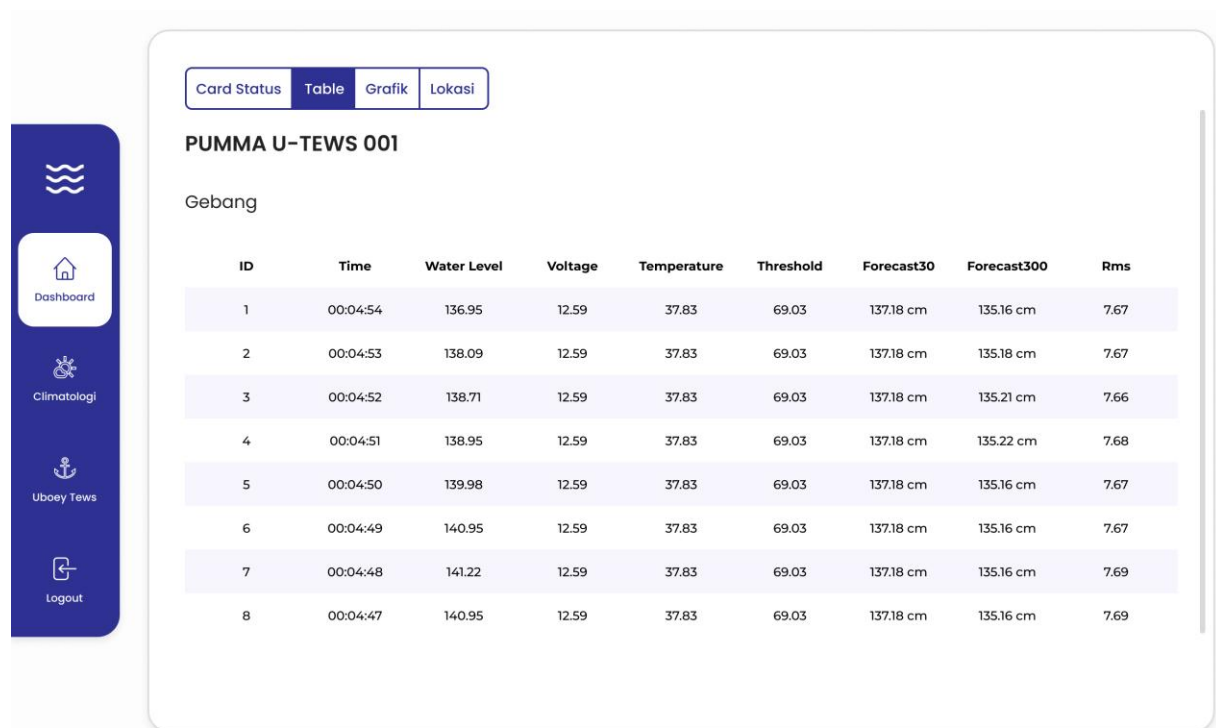


Gambar 20. Halaman Detail *Device* Pada *Card Status*

Gambar 20 merupakan halaman *website* setelah redesain yang menampilkan informasi alat PUMMA U-TEWS 001 yang berada di Gebang. Halaman ini dirancang dengan navigasi berbasis *tab* yang mencakup *Card Status*, *Table*, *Chart*, dan *Location*, untuk mempermudah pengguna dalam mengakses berbagai jenis data. Informasi penting yang ditampilkan dalam bentuk *card* meliputi tegangan baterai (*Battery Voltage*), suhu perangkat (*Device Temperature*), ketinggian air laut (*Sea Water Level*), prediksi kondisi dalam 30 data terakhir (*Forecast30*) dan 300 data terakhir (*Forecast300*), serta status

alat yang menunjukkan kondisi terkini, seperti "SAFE" Desain ini mengutamakan kemudahan akses dan penyajian data yang lebih visual agar lebih *user-friendly*.

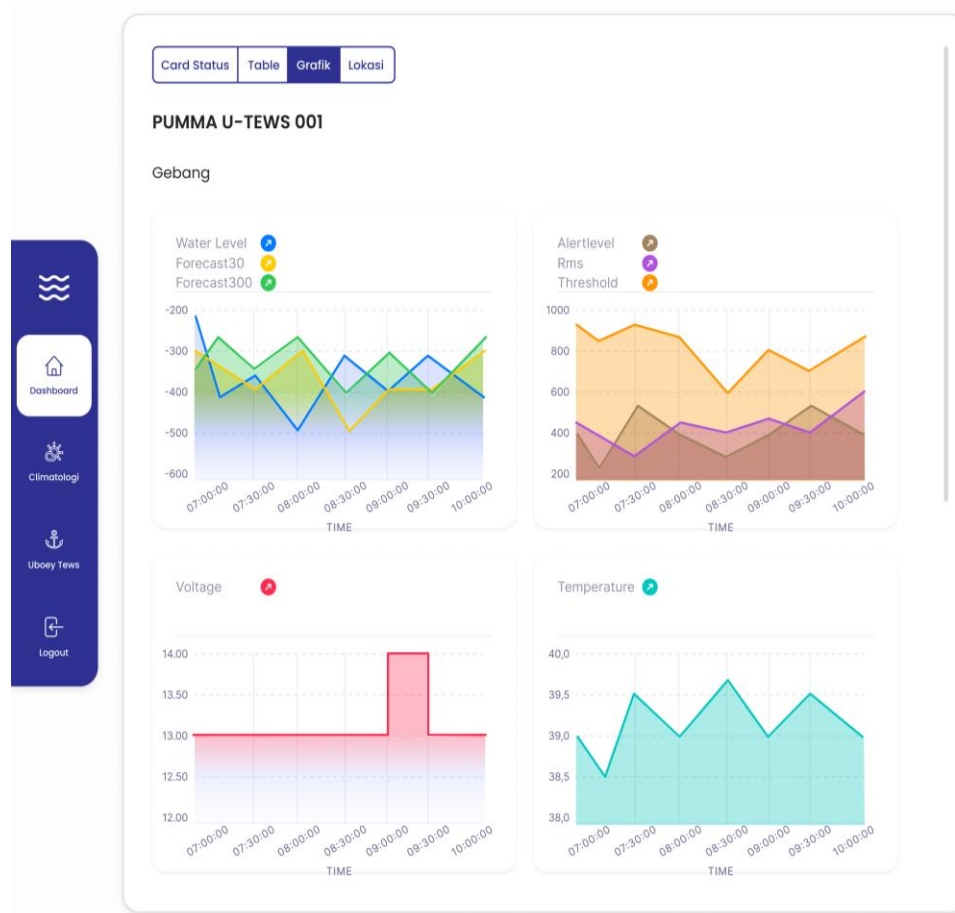
Perbedaan desain sebelumnya terletak pada penggabungan halaman informasi pada gambar 20, yang mencakup *card status*, *tabel*, *chart*, dan *location*, menjadi satu halaman tunggal. Penggabungan seluruh data ini menyebabkan website menjadi lebih berat dan mengalami *lag*, dikarenakan adanya tumpang tindih data dari sensor yang sangat banyak. Oleh karena itu, peneliti melakukan redesain halaman tersebut dengan memberikan navigasi berbasis *tab* bagi pengguna untuk memilih *card status*, *tabel*, *chart*, dan *location* yang ingin ditampilkan secara terpisah, sesuai dengan pilihan yang dipilih. Hal ini bertujuan untuk mengurangi beban pada *website* dan meningkatkan kinerjanya.



ID	Time	Water Level	Voltage	Temperature	Threshold	Forecast30	Forecast300	Rms
1	00:04:54	136.95	12.59	37.83	69.03	137.18 cm	135.16 cm	7.67
2	00:04:53	138.09	12.59	37.83	69.03	137.18 cm	135.18 cm	7.67
3	00:04:52	138.71	12.59	37.83	69.03	137.18 cm	135.21 cm	7.66
4	00:04:51	138.95	12.59	37.83	69.03	137.18 cm	135.22 cm	7.68
5	00:04:50	139.98	12.59	37.83	69.03	137.18 cm	135.16 cm	7.67
6	00:04:49	140.95	12.59	37.83	69.03	137.18 cm	135.16 cm	7.67
7	00:04:48	141.22	12.59	37.83	69.03	137.18 cm	135.16 cm	7.69
8	00:04:47	140.95	12.59	37.83	69.03	137.18 cm	135.16 cm	7.69

Gambar 21. Halaman Detail *Device* Pada *Table*

Gambar 21 merupakan halaman *website* setelah redesain yang menampilkan informasi alat PUMMA U-TEWS 001 yang berada di Gebang. Halaman ini dirancang dengan navigasi berbasis *tab* yang mencakup *Card Status*, *Table*, *Chart*, dan *Location*, untuk mempermudah pengguna dalam mengakses berbagai jenis data. Informasi penting yang ditampilkan dalam tabel meliputi ketinggian air laut (*Water Level*), tegangan baterai (*Voltage*), suhu perangkat (*Temperature*), nilai ambang batas (*Threshold*), prediksi kondisi dalam 30 data terakhir (*Forecast30*) dan 300 data terakhir (*Forecast300*), serta nilai RMS (*Root Mean Square*). Desain ini mengutamakan kemudahan akses dan penyajian data yang lebih visual agar lebih *user-friendly*.



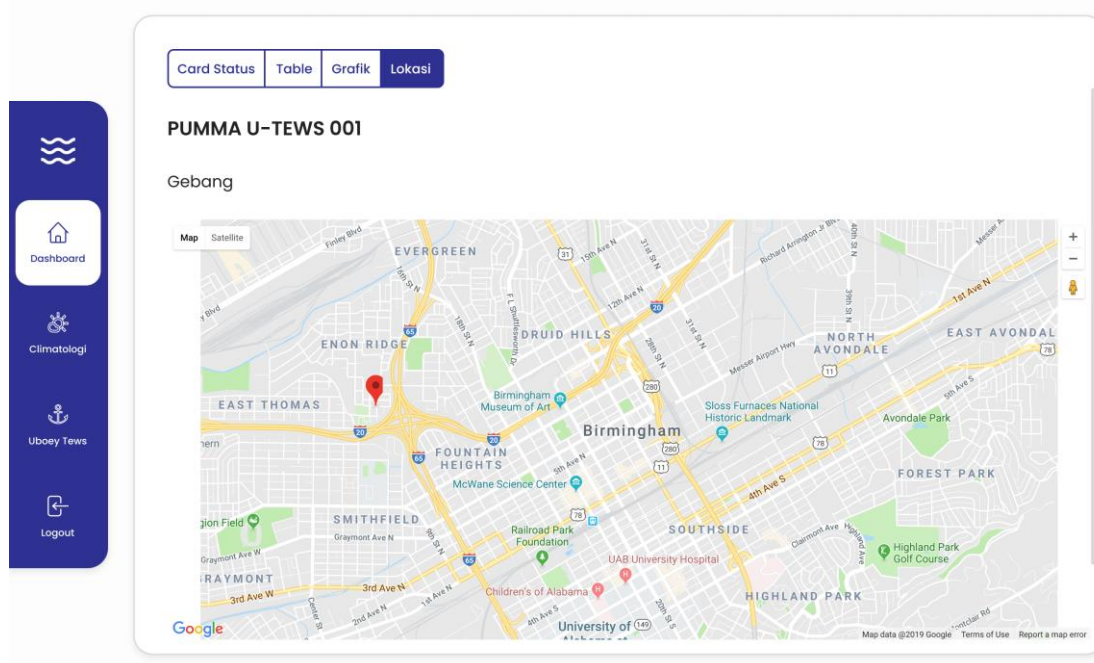
Gambar 22. Halaman Detail *Device* Pada Grafik

Gambar 22 merupakan halaman *website* setelah redesain yang menampilkan informasi alat PUMMA U-TEWS 001 yang berada di Gebang. Halaman ini dirancang dengan navigasi berbasis tab yang mencakup *card status*, *tabel*, *chart*, dan *location*, untuk mempermudah pengguna dalam mengakses berbagai jenis data.

Pada bagian visualisasi data, terdapat empat grafik utama yang menampilkan parameter-parameter penting. Grafik pertama menunjukkan *Water Level* atau ketinggian air laut yang berfluktuasi antara 200 hingga 600, dengan tiga garis berbeda yang merepresentasikan data aktual serta prediksi untuk 30 dan 300 data terakhir. Di sebelahnya, terdapat grafik *Alert Level* yang memvisualisasikan nilai RMS dan *Threshold* dalam rentang 400-1000, memberikan indikasi yang jelas tentang tingkat kewaspadaan.

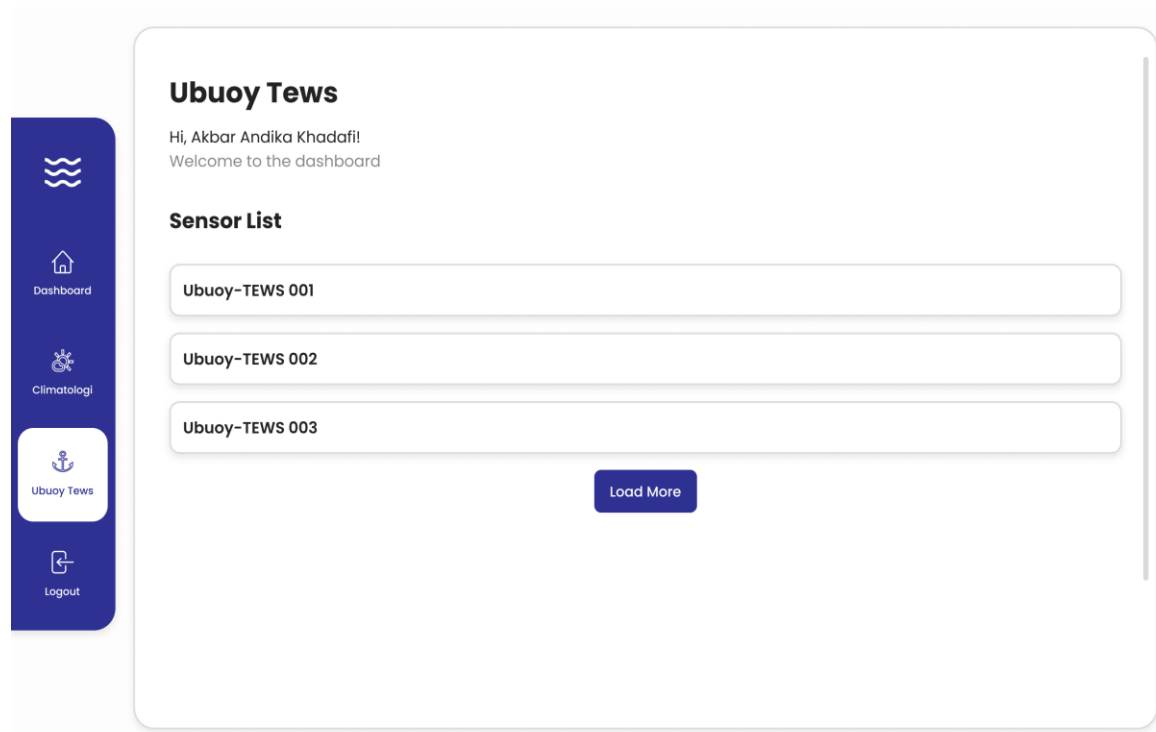
Dua grafik lainnya menampilkan parameter teknis perangkat yang tak kalah penting. Grafik *Voltage* memperlihatkan tegangan baterai yang mayoritas stabil di sekitar 13.00 volt, dengan satu lonjakan singkat yang terekam. Sementara itu, grafik *Temperature* menunjukkan suhu perangkat yang bergerak relatif stabil di kisaran 38.5-39.5°C, mengindikasikan kondisi operasional yang normal.

Keseluruhan data ini ditampilkan dalam rentang waktu dari pukul 07:00:00 hingga 10:00:00, memberikan gambaran yang komprehensif tentang performa sistem dalam periode tersebut. Desain User Interface (UI) yang mengutamakan kemudahan akses dan penyajian data visual ini membuat informasi lebih mudah dipahami dan *user-friendly*, memungkinkan pengguna untuk memantau kondisi perangkat dan lingkungan secara efektif.



Gambar 23. Halaman Detail *Device* Pada Lokasi

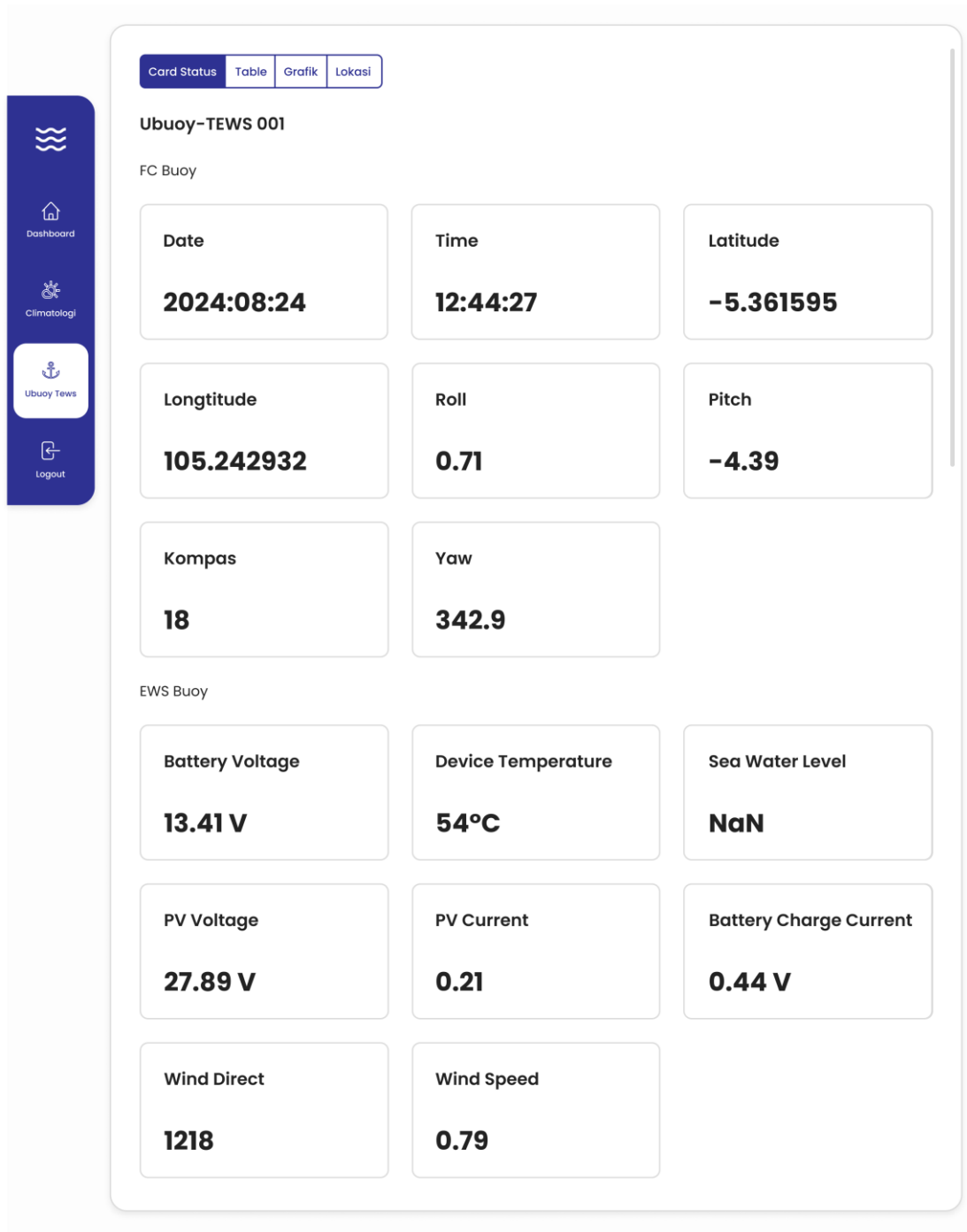
Gambar 23 di atas menampilkan tab Lokasi dari halaman monitoring PUMMA U-TEWS 001 yang berada di Gebang. Pada tampilan ini, terdapat peta interaktif dari Google Maps yang menunjukkan lokasi perangkat yang ditandai dengan penanda merah. Interface peta ini terintegrasi dengan sempurna dalam desain keseluruhan website, dengan tetap mempertahankan navigasi tab yang konsisten *card status*, *tabel*, *chart*, dan *location*. Desain ini memungkinkan pengguna untuk dengan mudah memahami posisi geografis dari perangkat PUMMA U-TEWS 001 dalam konteks lingkungan sekitarnya.



Gambar 24. Halaman *List Device Buoy*

Gambar 24 menampilkan halaman *dashboard* Ubuoy Tews yang menyajikan User Interface (UI) pengguna yang bersih dan terorganisir. Di bagian atas halaman, terdapat sapaan personal "Hi, Akbar Andika Khadafi!" diikuti dengan pesan selamat datang "Welcome to the dashboard" yang memberikan nuansa ramah bagi pengguna. Bagian utama halaman menampilkan "Sensor List" yang berisi daftar tiga sensor yang tersedia: BUOY U-TEWS 001 dan BUOY U-TEWS 002.

Setiap sensor ditampilkan dalam kotak tersendiri dengan latar belakang putih, memberikan tampilan yang rapi dan mudah dibaca. Navigasi di sisi kiri layar tetap konsisten dengan halaman lainnya, menampilkan menu utama yang terdiri dari *card status*, *tabel*, *chart*, dan *location*. Desain keseluruhan menggunakan skema warna yang profesional dengan dominasi putih dan aksen biru tua, menciptakan tampilan yang bersih dan profesional.



Gambar 25. Halaman Detail Buoy Pada *Card Status*

Gambar 25 menampilkan halaman *Card Status* untuk perangkat Ubuoy-TEWS 001 yang berlokasi di FC Buoy. Halaman ini menampilkan berbagai parameter pengukuran yang diorganisir dalam kartu-kartu individual yang rapi dan mudah dibaca. Pada bagian FC Buoy, terdapat informasi dasar seperti waktu dan lokasi yang tercatat pada tanggal 2024:08:24 pukul 12:44:27. Koordinat lokasi ditunjukkan dengan *Latitude* -5.361595 dan *Longitude* 105.242932. Data orientasi perangkat ditampilkan melalui parameter *Roll* (0.71), *Pitch* (-4.39), Kompas (18), dan *Yaw* (342.9).

Bagian EWS Buoy menyajikan informasi teknis perangkat yang mencakup berbagai pengukuran daya dan kondisi lingkungan. *Battery Voltage* tercatat 13.41V dengan *Device Temperature* mencapai 54°C. *Sea Water Level* menunjukkan nilai NaN yang mengindikasikan data tidak tersedia saat itu. Parameter daya lainnya meliputi *PV Voltage* sebesar 27.89V, *PV Current* 0.21, dan *Battery Charge Current* 0.44V. Informasi tambahan tentang kondisi angin juga tercatat, dengan *Wind Direct* menunjukkan nilai 1218 dan *Wind Speed* 0.79. Semua data ini ditampilkan dalam format yang konsisten dengan navigasi berbasis tab di bagian atas *card status*, *tabel*, *chart*, dan *location*.

Dashboard

Climatologi

Ubuoy TEWS

Logout

Card Status

Table

Grafik

Lokasi

Ubuoy-TEWS 001

FC Buoy

ID	Date	Time	Latitude	Longitude	Roll	Pitch	Kompas	Yaw
1	2024:08:24	12:44:27	-5.361595	105.242932	0.71	-4.39	18	342.9
2	2024:08:24	12:44:27	-5.361595	105.242932	0.71	-4.39	18	342.9
3	2024:08:24	12:44:27	-5.361595	105.242932	0.71	-4.39	18	342.9
4	2024:08:24	12:44:27	-5.361595	105.242932	0.71	-4.39	18	342.9
5	2024:08:24	12:44:27	-5.361595	105.242932	0.71	-4.39	18	342.9
6	2024:08:24	12:44:27	-5.361595	105.242932	0.71	-4.39	18	342.9
7	2024:08:24	12:44:27	-5.361595	105.242932	0.71	-4.39	18	342.9
8	2024:08:24	12:44:27	-5.361595	105.242932	0.71	-4.39	18	342.9
9	2024:08:24	12:44:27	-5.361595	105.242932	0.71	-4.39	18	342.9
10	2024:08:24	12:44:27	-5.361595	105.242932	0.71	-4.39	18	342.9
11	2024:08:24	12:44:27	-5.361595	105.242932	0.71	-4.39	18	342.9
12	2024:08:24	12:44:27	-5.361595	105.242932	0.71	-4.39	18	342.9

EWS Buoy

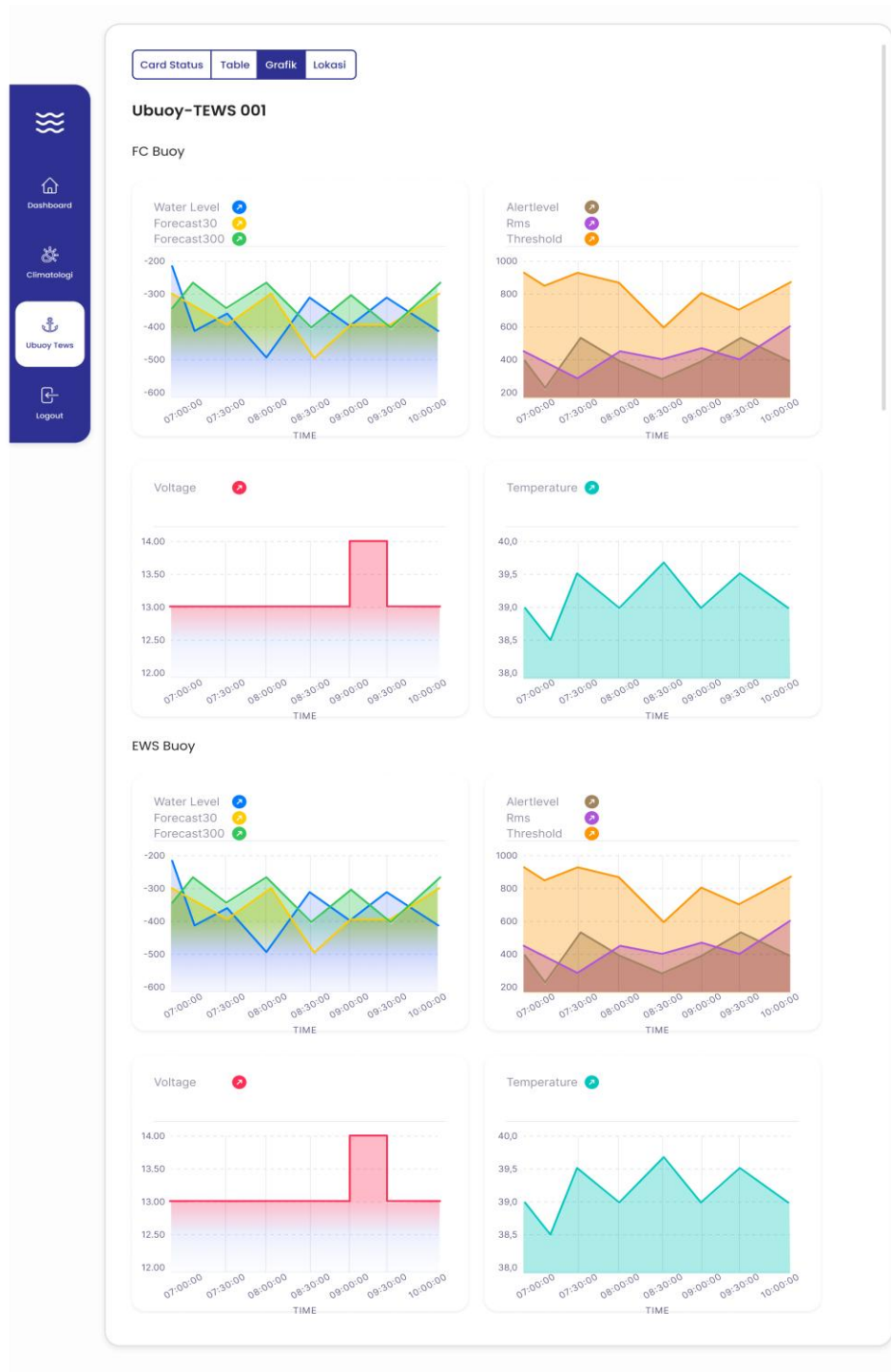
ID	Battery Voltage	Device Temperature	Sea Water Level	PV Voltage	PV Current	Battery Charge	Wind Direct	Wind Speed
1	13.41 V	54°C	NaN	27.89 V	0.21	0.44 V	1218	0.79
2	13.41 V	54°C	NaN	27.89 V	0.21	0.44 V	1218	0.79
3	13.41 V	54°C	NaN	27.89 V	0.21	0.44 V	1218	0.79
4	13.41 V	54°C	NaN	27.89 V	0.21	0.44 V	1218	0.79
5	13.41 V	54°C	NaN	27.89 V	0.21	0.44 V	1218	0.79
6	13.41 V	54°C	NaN	27.89 V	0.21	0.44 V	1218	0.79
7	13.41 V	54°C	NaN	27.89 V	0.21	0.44 V	1218	0.79
8	13.41 V	54°C	NaN	27.89 V	0.21	0.44 V	1218	0.79
9	13.41 V	54°C	NaN	27.89 V	0.21	0.44 V	1218	0.79
10	13.41 V	54°C	NaN	27.89 V	0.21	0.44 V	1218	0.79

Gambar 26. Halaman Detail Buoy Pada Table

Gambar 26 menampilkan *tab Table* dari Ubuoy-TEWS 001 yang berlokasi di FC Buoy. Halaman ini menyajikan data dalam format tabel yang terorganisir dalam dua bagian

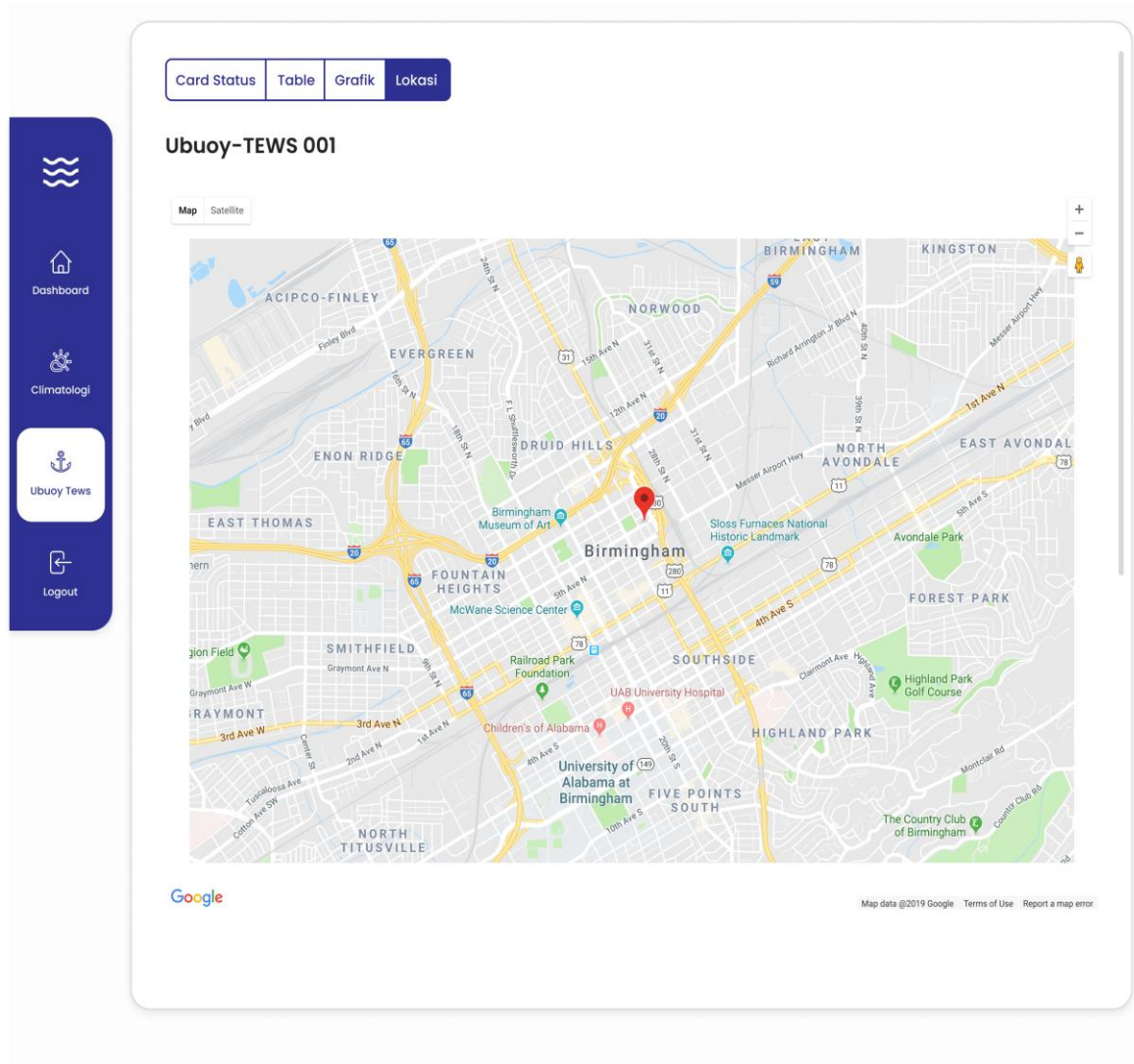
utama: *FC Buoy* dan *EWS Buoy*. Tabel *FC Buoy* menampilkan 100 baris data dengan kolom-kolom yang mencakup *ID*, *Date*, *Time*, *Latitude*, *Longitude*, *Roll*, *Pitch*, *Kompas*, dan *Yaw*. Semua baris menunjukkan data yang identik yang diambil pada 2024-08-24 pukul 12:44:27, dengan koordinat di *Latitude* -5.361595 dan *Longitude* 105.242932. Parameter orientasi juga konsisten di semua baris dengan *Roll* 0.71, *Pitch* -4.39, *Kompas* 18, dan *Yaw* 342.9.

Bagian *EWS Buoy* menampilkan 100 baris data dengan parameter yang berbeda, meliputi *Battery Voltage*, *Device Temperature*, *Sea Water Level*, *PV Voltage*, *PV Current*, *Battery Charge*, *Wind Direct*, dan *Wind Speed*. Seperti tabel sebelumnya, data di setiap baris identik dengan nilai *Battery Voltage* 13.41V, *Device Temperature* 54°C, *Sea Water Level* NaN (*Not a Number*), *PV Voltage* 27.89V, *PV Current* 0.21, *Battery Charge* 0.44V, *Wind Direct* 1218, dan *Wind Speed* 0.79. Tata letak tabel menggunakan desain yang bersih dengan latar belakang baris bergantian untuk memudahkan pembacaan, serta tetap mempertahankan konsistensi navigasi dengan *tab* dan menu *sidebar* yang ada di halaman lainnya.



Gambar 27. Halaman Detail Buoy Pada Grafik

Gambar 27 menampilkan *tab Chart* dari Ubuoy-TEWS 001 yang membagi visualisasi data menjadi dua bagian utama: FC Buoy dan EWS Buoy. Setiap bagian menampilkan empat *chart* yang identik untuk memantau parameter-parameter penting. Pada setiap bagian (FC Buoy dan EWS Buoy).



Gambar 28. Halaman Detail Lokasi Buoy

Gambar 28 menampilkan *tab Location* dari perangkat Ubuoy-TEWS 001 yang menggunakan *Google Maps* sebagai *platform* visualisasi lokasi. Tampilan peta yang interaktif ini memungkinkan pengguna untuk dengan mudah memahami konteks geografis dari lokasi perangkat Ubuoy-TEWS 001 dalam hubungannya dengan lingkungan sekitar dan infrastruktur penting di area tersebut.

3.3.3 Pengembangan Sistem

Tahap ini merupakan tahap untuk mengimplementasikan rancangan sistem yang telah dibuat menjadi kode program dengan hasil akhir berupa website Pumma. Sesuai dengan metode *scrum*, maka proses pengembangan akan terdiri dari beberapa proses yang akan dilakukan adalah sebagai berikut:

3.3.3.1 Scrum Team

Penelitian ini berdasarkan proyek yang dikerjakan oleh beberapa orang dengan tanggung jawab dan pekerjaan berbeda-beda yang tergabung dalam *scrum team*. *Scrum team* merupakan sebutan kelompok dalam proyek pengembangan yang menggunakan metode *scrum*, terdiri dari 3 peran yaitu *product owner*, *scrum master*, dan *developers*. *Product owner* dan *scrum master* masing-masing 1 orang, sedangkan untuk *developers* berjumlah lebih dari 1 orang.

Tabel 9. *Team Scrum*

Peran	Nama	Keterangan
<i>Product Owner</i>	Mona Arif Muda, S.T., M.T.	<i>Product owner</i> bertanggung jawab untuk mengelola <i>product backlog</i> .
<i>Scrum Master</i>	Akbar Andika Khadafi	<i>Scrum master</i> bertanggung jawab

		untuk memastikan proses <i>scrum</i> berjalan dengan baik
<i>Developer</i>	Akbar Andika Khadafi	<i>Frontend Developer</i>
	Muhammad Rafsanjani	<i>Backend Developer</i>
	Ridho Tegar Pradana	<i>Hardware Developer</i>

3.3.3.2 *Product Backlog*

Product backlog akan tentukan dalam bentuk daftar prioritas yang berisi seluruh fitur-fitur aplikasi yang akan dikerjakan selama tahap pengembangan sistem. Pembuatan *product backlog* pada penelitian ini akan menggunakan *tools trello* untuk memudahkan dalam manajemen proyek. Pada tahap ini *product backlog* dapat diperbarui seiring dengan proses pengembangan aplikasi berjalan untuk memastikan selalu relevan dan dapat memenuhi kebutuhan yang akan ikut berkembang.

3.3.3.3 *Sprint Planning*

Sprint Planning adalah tahap awal dalam metodologi *scrum* yang digunakan dalam pengembangan perangkat lunak. Pertemuan ini dilakukan di awal setiap *sprint*, yaitu periode waktu yang biasanya berlangsung antara dua hingga empat minggu. Dalam *Sprint Planning*, anggota tim pengembang bekerja sama untuk memahami tujuan *sprint* dan merinci tugas-tugas yang harus diselesaikan untuk mencapai tujuan tersebut. *Product Owner* berbagi pembaruan tentang kebutuhan pelanggan dan mendefinisikan elemen *product backlog* yang perlu dikerjakan. Tim pengembang kemudian melakukan estimasi waktu untuk setiap tugas, memilih tugas-tugas yang akan diselesaikan selama *sprint*, dan merancang rencana rinci untuk mencapai tujuan *sprint*. *Sprint Planning* memberikan arah yang jelas dan memastikan bahwa tim memiliki pemahaman yang seragam tentang pekerjaan yang akan dilakukan selama *sprint* berikutnya.

3.3.3.4 *Sprint Backlog*

Sprint Backlog dibuat pada tahap *Sprint Planning*, menyajikan daftar tugas yang berasal dari *Product Backlog* dan telah diprioritaskan oleh *product owner*. Setiap tugas dalam *Sprint Backlog* memiliki estimasi waktu yang diperkirakan untuk penyelesaiannya. Seiring berjalannya *sprint*, tim memantau dan memperbarui *Sprint Backlog* sesuai kebutuhan, memastikan transparansi dan pemahaman yang jelas tentang kemajuan kerja. *Sprint Backlog* membantu tim untuk tetap fokus pada tujuan *sprint* dan memastikan bahwa pekerjaan yang diperlukan untuk mencapainya dapat diselesaikan dalam batas waktu yang ditetapkan.

3.3.3.5 *Sprint*

Sprint merupakan tahap utama dalam proses pengembangan sistem penelitian ini, dimana tim *developer* akan mengerjakan tugas-tugas yang telah direncanakan di *sprint planning*. Selama *sprint* berlangsung *developer* harus berfokus untuk menyelesaikan *sprint backlog* yang ditentukan tanpa adanya banyak perubahan untuk memastikan semua pekerjaannya dapat diselesaikan dengan tepat waktu. Dalam penelitian ini *sprint* direncanakan akan berlangsung selama 4 kali dengan durasi setiap *sprint* yaitu 2 minggu, hal ini telah disepakati oleh *team scrum* dengan mempertimbangkan pembagian tugas-tugas dari setiap *sprint*.

3.3.3.6 *Daily Scrum*

Pertemuan *daily scrum* atau *daily standup* dilakukan setiap hari selama *sprint* berlangsung dengan durasi 15 menit yang di hadiri oleh *scrum master* dan *developers*. Pertemuan ini dapat dilakukan secara daring maupun luring yang dipimpin oleh *scrum master*, selama pertemuan setiap orang dari tim *developers* akan menyampaikan pekerjaan yang telah dilakukan, pekerjaan yang akan dilakukan pada hari ini, dan hambatan yang dialami selama proses pengerjaan.

3.3.3.7 *Sprint Review*

Sprint review dilakukan pada akhir setiap *sprint* yang dihadiri oleh *team scrum*, dalam pertemuan ini tim *developers* akan mendemonstrasikan pekerjaan yang telah dilakukan selama *sprint* dan akan diberikan umpan balik oleh *product owner* terhadap demonstrasi yang telah diperlihatkan. Selama tahap *sprint review* ini berlangsung akan berfokus terhadap diskusi *team scrum* untuk memastikan proses pekerjaan yang telah dilakukan oleh tim *developers* berjalan dengan sesuai perancangan sistem. Dalam tahap ini juga *product backlog* dapat diperbarui untuk menyesuaikan dengan perubahan kebutuhan atau tambahan yang didapatkan dari hasil diskusi dan umpan balik yang diberikan oleh *product owner* sehingga proses pengembangan dapat lebih adaptif dan responsif terhadap perubahan.

3.3.3.8 *Sprint Retrospective*

Sprint retrospective merupakan pertemuan yang dihadiri oleh *team scrum* setelah *sprint* selesai untuk merefleksikan kinerja selama *sprint* berlangsung. Pertemuan ini akan mendiskusikan hambatan yang terjadi dari setiap orang dalam *team scrum* selama proses pengembangan untuk mengidentifikasinya sehingga mendapatkan solusi dan adanya perubahan kinerja yang lebih baik. *Sprint retrospective* memfasilitasi *team scrum* untuk terus meningkatkan kinerjanya selama proses pengembangan sistem berlangsung.

3.3.4 **Pengujian Sistem**

Tahapan pengujian sistem pada penelitian ini dilakukan setelah aplikasi yang dirancang telah selesai dikembangkan pada tahap pengembangan sistem. dalam tahap ini pengujian dilakukan dengan *System Usability Scale*, *Black Box Testing* dan *A/B Testing*.

3.3.4.1 System Usability Scale

System Usability Scale (SUS) dalam penelitian ini dilakukan untuk mengukur kegunaan aplikasi yang dikembangkan dengan memberikan kuisioner *SUS* kepada pengguna yang telah melakukan skenario penggunaan aplikasi. Kuisioner *SUS* terdiri dari 10 pertanyaan dan 5 pilihan jawaban, berikut tabel 10 pertanyaan dan 5 pilihan jawaban dalam kuisioner *SUS*:

Tabel 10. Pertanyaan *SUS*

No	Pernyataan	Skala
1	Saya pikir bahwa saya akan ingin lebih sering menggunakan aplikasi ini.	1-5
2	Saya menemukan bahwa aplikasi ini tidak perlu dibuat serumit ini.	1-5
3	Saya pikir aplikasi ini mudah untuk digunakan	1-5
4	Saya pikir bahwa saya akan membutuhkan bantuan dari orang teknis untuk dapat menggunakan aplikasi ini.	1-5
5	Saya menemukan berbagai fungsi di aplikasi ini terintegrasi dengan baik.	1-5
6	Saya pikir ada terlalu banyak ketidaksesuaian didalam aplikasi ini.	1-5
7	Saya bayangkan bahwa kebanyakan orang akan mudah untuk mempelajari aplikasi ini dengan sangat cepat.	1-5
8	Saya menemukan aplikasi ini sangat rumit untuk digunakan.	1-5
9	Saya merasa sangat percaya diri untuk menggunakan aplikasi ini.	1-5
10	Saya perlu belajar banyak hal sebelum saya bisa memulai menggunakan aplikasi ini.	1-5

Tabel 11. Pilihan Jawaban *SUS*

No	Jawaban
1	Sangat Tidak Setuju

2	Tidak Setuju
3	Ragu Ragu
4	Setuju
5	Sangat Setuju

3.3.4.2 A/B Testing

Dalam penelitian ini, *A/B testing* dilakukan untuk membandingkan dua versi desain *User Interface (UI)* website guna mengevaluasi efektivitas dan pengalaman pengguna dengan memberikan kuisoner *A/B Testing* kepada pengguna yang telah melakukan skenario penggunaan aplikasi. Kuisoner *A/B testing* terdiri dari 12 pertanyaan dan 2 pilihan jawaban, berikut tabel 12 pertanyaan dan 2 pilihan jawaban dalam kuisoner A/B Testing:

Tabel 12. Pertanyaan SUS

No	Pernyataan	Skala
1	Desain website mana yang menurut Anda lebih menarik secara visual?	1-2
2	Layout website mana yang lebih mudah dipahami?	1-2
3	Kombinasi warna pada website mana yang lebih nyaman dilihat?	1-2
4	Website mana yang memiliki menu navigasi yang lebih mudah digunakan?	1-2
5	Pada website mana Anda lebih mudah menemukan informasi yang dicari?	1-2
6	Website mana yang memiliki struktur konten yang lebih terorganisir?	1-2
7	Website mana yang lebih cepat dimuat?	1-2
8	Website mana yang lebih responsif saat digunakan?	1-2
9	Website mana yang menyajikan informasi dengan lebih jelas?	1-2
10	Website mana yang memiliki konten yang lebih lengkap	1-2

11	Secara keseluruhan, website mana yang lebih profesional?	1-2
12	Website mana yang akan Anda rekomendasikan kepada orang lain?	1-2

Tabel 13. Pilihan Jawaban *A/B Testing*

No	Jawaban
1	<i>Web A</i>
2	<i>Web B</i>

3.3.5 Penulisan Laporan

Tahapan penulisan laporan merupakan tahap akhir dari penelitian ini, tahap ini melakukan penulisan laporan penelitian pengembangan *frontend website* Puma pemantauan dan peringatan dini bencana tsunami berbasis *IOT* dengan menggunakan *library ReactJS*. Laporan penelitian berisi data dan informasi yang didapatkan selama proses penelitian berlangsung.

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

1. *Website front-end* PUMMA dan BUOY telah berhasil dibangun menggunakan *React JS* dan *Tailwind CSS* dengan berbagai fitur yang memungkinkan pengguna untuk melihat *landing page* dan detail alat PUMMA dan BUOY. Informasi tersebut disajikan dalam berbagai format, termasuk *card status*, *table*, *chart*, dan *location*, sehingga mempermudah pemantauan dan analisis data secara efisien.
2. Pengembangan aplikasi menggunakan metode Scrum yang dilakukan dalam empat iterasi selama dua bulan. Proses pengerjaan dimulai dengan pengembangan halaman *User Interface (UI)* sampai pengimplementasian *API* aplikasi ke halaman yang telah dibuat.
3. Berdasarkan hasil pengujian pada 6 skenario di setiap iterasi menggunakan metode *blackbox testing*, fitur yang dikembangkan berfungsi dengan baik dan memenuhi *sprint backlog* di setiap iterasi dengan tingkat keberhasilan 100%.
4. Berdasarkan hasil pengujian kebergunaan menggunakan *System Usability Scale (SUS)* memiliki *acceptance score* sebesar 83,8 masuk dalam kategori *acceptable* dengan *grade B* dan adjective rating *excellent*.
5. Berdasarkan hasil pengujian *A/B Testing* diperoleh bahwa *Website B* yang merupakan hasil pengembangan memiliki rata-rata skor sebesar 96,6%, sedangkan *Website A* yang merupakan versi sebelum pengembangan hanya memperoleh skor 3,3%. Hasil ini menunjukkan bahwa *Website B* lebih disukai oleh responden dibandingkan *Website A*, yang menandakan keberhasilan pengembangan dalam meningkatkan kualitas dan pengalaman pengguna. Dengan mayoritas responden

memilih *Website B*, dapat disimpulkan bahwa perubahan dan perbaikan yang dilakukan memberikan dampak positif terhadap kegunaan dan kenyamanan pengguna.

5.2 Saran

1. Melakukan pengembangan jumlah alat dan menambahkan alat pada *menu klimatologi*.
2. Melakukan pengembangan dengan fitur *download* data pada grafik agar bisa dikembangkan oleh pengunjung *website*.
3. Menambahkan fitur notifikasi apabila status *Danger*.
4. Melakukan *update API* dari alat yang baru hidup atau alat yang baru di tambahkan.
5. Melakukan pengembangan *landing page* yang berisikan detail tentang pumma dan buoy seperti menambahkan informasi sensor dan fungsi nya serta penjelasan terkait data data yang di tampilkan pada *dashboard*.
6. Melakukan pengembangan halaman *members*.

DAFTAR PUSTAKA

- [1] A. Dwiki *et al.*, “Redesain Website Pemerintah dengan Pengembangan Menggunakan Kerangka Kerja ASP.NET dan fitur Razor Pages,” *J. Ilmu Tek.*, vol. 1, no. 2, pp. 50–63, 2023.
- [2] A. R. Firdaus, N. H. Wardani, and L. Fanani, “Evaluasi dan Rekomendasi Perbaikan Usability pada System Autoworks pada PT. Prima Berkat Gemilang dengan menggunakan Metode Heuristic Evaluation dan System Usability Scale (SUS),” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 6, pp. 5799–5807, 2019, [Online]. Available: <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/5580>
- [3] Agustini and W. J. Kurniawan, “Sistem E-Learning Do’a dan Iqro’ dalam Peningkatan Proses Pembelajaran pada TK Amal Ikhlas,” *J. Mhs. Apl. Teknol. Komput. dan Inf.*, vol. 1, no. 3, pp. 154–159, 2019, [Online]. Available: <http://www.ejournal.pelitaindonesia.ac.id/JMApTeKsi/index.php/JOM/article/view/526>
- [4] C. N. Mazda (Chadziqatun Najilatil Mazda), D. Pelansiani (Devy Pelansiani), and Sunardi, “Reconstruction Resilience Policy of Sunda Strait Tsunami,” in *Proceedings of The 6th Annual Scientific Meeting on Disaster Research*, pp. 386–392, 2019.
- [5] de Rosal Ignatius Moses Setiadi, A. F. Najib, E. H. Rachmawanto, C. A. Sari, M. K. Sarker, and N. Rijati, “A comparative study MD5 and SHA1 algorithms to encrypt REST API authentication on mobile-based application,” *2019 Int. Conf. Inf. Commun. Technol. ICOIACT 2019*, no. January 2020, pp. 206–211, 2019, doi:

10.1109/ICOIACT46704.2019.8938570.

- [6] F. Schindel  et al., “Monitoring and Warning for Tsunamis Generated by Volcanoes,” 2023.
- [7] I. P. Sari, A. Jannah, A. M. Meuraxa, A. Syahfitri, and R. Omar, “Perancangan Sistem Informasi Penginputan Database Mahasiswa Berbasis Web,” *Hello World J. Ilmu Komput.*, vol. 1, no. 2, pp. 106–110, 2022, doi: 10.56211/helloworld.v1i2.57.
- [8] K. Schwaber and J. Sutherland, “The Scrum Guide - The Definitive Guide to Scrum: The Rules of the Game,” *Scrum. org, Oct.*, vol. 2, no. October, p. 17, 2011, [Online]. Available: <http://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-US.pdf#zoom=100>
- [9] M. A. Ir. Diah Novianti, *BENCANA ALAM DAN MITIGASI BENCANA ALAM*. Surabaya: CV.Jakad Media, 2022.
- [10] M. A. Muhyidin, M. A. Sulhan, and A. Sevtiana, “Perancangan Ui/Ux Aplikasi My Cic Layanan Informasi Akademik Mahasiswa Menggunakan Aplikasi Figma,” *J. Digit*, vol. 10, no. 2, p. 208, 2020, doi: 10.51920/jd.v10i2.171.
- [11] M. D. Irawan and S. A. Simargolang, “Implementasi E-Arsip Pada Program Studi Teknik Informatika,” *J. Teknol. Inf.*, vol. 2, no. 1, p. 67, 2018, doi: 10.36294/jurti.v2i1.411.
- [12] M. E. Khan, “DIFFERENT APPROACHES TO BLACK BOX,” vol. 2, no. 4, pp. 31–40, 2011.
- [13] M. Fariz, S. Lazuardy, and D. Anggraini, “Modern Front End Web Architectures with React.Js and Next.Js,” *Int. Res. J. Adv. Eng. Sci.*, vol. 7, no. 1, pp. 132–141, 2022.
- [14] M. Irfan, Y. Haryadi, D. Haryanto, and A. Rusdiansyah, “Tinjauan Teknis Penempatan Sistem Mooring Buoy Dan Obu Ina-Tews Di Dasar Laut,” *Oseanika*, vol. 2, no. 1, pp. 1–16, 2021, doi: 10.29122/oseanika.v2i1.4952.

- [15] M. Naufal Al Ghazali, A. Azizi, O. S. Khair, and Z. T. Saifullah Kusnandar, "Pengembangan Dashboard Admin Bukupedia," *J. Tek. Indones.*, vol. 3, no. 1, pp. 11–21, 2024, doi: 10.58860/jti.v3i1.320.
- [16] Ranto, "Membangun Frontend Website Sistem Infomasi Pemantauan Dan Peringatan Dini Bencana Tsunami Berbasis Iot Dengan Menggunakan Library Reactjs," vol. VIII, no. I, pp. 01–65, 2023.
- [17] R. A. Kurniawan, "Pembuatan Design System Menggunakan Pendekatan Atomic Design dan A/B Testing," *J. Teknol. Dan Sist. Inf. Bisnis*, vol. 6, no. 3, pp. 543–549, 2024, doi: 10.47233/jteksis.v6i3.1346.
- [18] T. J. Charity, "Smart World of Internet of Things (IoT) and It ' s Security Concerns," pp. 240–245, 2016.
- [19] V. Rahadian Fakhrizal, A. Musnansyah, and T. Mulyana, "Perancangan Prototype Aplikasi Promosi Properti Primary Berbasis Website Menggunakan CMS Wordpress," vol. 10, no. 6, p. 5316, 2023.
- [20] Y. Firmanto, "EARLY WARNING SYSTEM: SOLUSI KLAIM NEGATIF RUMAH SAKIT PROGRAM JAMINAN KESEHATAN NASIONAL," vol. 20, no. 1, p. 32, 2021.
- [21] Yuni and Sugiati, "Analisis dan Perancangan UML (Unified Modeling Language).," 2012.