

**MEMBANGUN FRONTEND WEBSITE SISTEM INFOMASI
PEMANTAUAN DAN PERINGATAN DINI BENCANA TSUNAMI
BERBASIS IOT DENGAN MENGGUNAKAN LIBRARY REACTJS**

(Skripsi)

Oleh

RANTO



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
2023**

**MEMBANGUN FRONTEND WEBSITE SISTEM INFOMASI
PEMANTAUAN DAN PERINGATAN DINI BENCANA TSUNAMI
BERBASIS IOT DENGAN MENGGUNAKAN LIBRARY REACTJS**

Oleh

RANTO

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

Pada

**Program Studi Teknik Informatika
Fakultas Teknik**



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
2023**

ABSTRAK

MEMBANGUN FRONTEND WEBSITE SISTEM INFORMASI PEMANTAUAN DAN PERINGATAN DINI BENCANA TSUNAMI BERBASIS IOT DENGAN MENGGUNAKAN LIBRARY REACTJS

Oleh

RANTO

Tsunami adalah salah satu bencana alam yang kerap terjadi di Indonesia. Salah satunya pada 22 Desember 2018 terjadi Tsunami di daerah pesisir Lampung dan Banten dengan menewaskan sedikitnya 426 orang meninggal, 7.202 luka-luka dan 23 menghilang. Berdasarkan peristiwa tersebut Indonesia membutuhkan sebuah sistem pemantauan dan peringatan dini bencana tsunami untuk mempercepat informasi mengenai bencana kepada masyarakat. Dalam penelitian ini akan dikembangkan website dashboard sistem informasi pemantauan dan peringatan dini bencana tsunami berbasis IoT menggunakan library ReactJS. Dengan memanfaatkan protocol Message Queuing Telemetry Transport (MQTT) yang digunakan untuk menerima pesan realtime yang dikirimkan oleh server. Data yang dikirimkan merupakan data sensor dan data yang telah dilakukan perhitungan disisi server. Data-data tersebut berupa data ketinggian air, forecasting 30 data, forecasting 300 data, rms, threshold, alert level, status, temperatur. Selain menggunakan protokol MQTT penelitian ini menggunakan protokol http yang digunakan untuk mengkonsumsi API untuk yang mengirimkan data record dari database yang akan ditampilkan menjadi grafik yang mudah dipahami oleh pengguna. Dengan memanfaatkan reactjs sebagai library, tailwindcss sebagai framework styling, axios library untuk melakukan http request.

Kata kunci — ReactJS, Internet of Things (IoT), MQTT, Tailwindcss, EWS, Rest API, Axios.

ABSTRACT

BUILDING A FRONTEND WEBSITE FOR AN IOT-BASED EARLY WARNING AND MONITORING SYSTEM FOR TSUNAMI DISASTERS USING THE REACTJS LIBRARY

By

RANTO

Tsunami is one of the natural disasters that often occurs in Indonesia. One of them was on December 22, 2018 a Tsunami occurred in the coastal areas of Lampung and Banten with at least 426 people dead, 7,202 injured and 23 missing. Based on this event, Indonesia needs a tsunami disaster monitoring and early warning system to accelerate information about disasters to the public. In this research, an IoT-based tsunami disaster monitoring and early warning information system dashboard website will be developed using the ReactJS library. By utilizing the Message Queuing Telemetry Transport (MQTT) protocol which is used to receive real time messages sent by the server. The data sent is sensor data and data that has been calculated on the server side. These data are in the form of water level data, 30 data forecasting, 300 data forecasting, rms, threshold, alert level, status, temperature. In addition to using the MQTT protocol, this research uses the http protocol which is used to consume APIs to send data records from the database which will be displayed into graphs that are easily understood by users. By utilizing reactjs as a library, tailwind css as a styling framework, axios library to do http requests.

Keywords — ReactJS, Internet of Things (IoT), MQTT, Tailwindcss, EWS, Rest API, Axios.

Judul Skripsi : **MEMBANGUN FRONTEND WEBSITE
SISTEM INFORMASI PEMANTAUAN DAN
PERINGATAN DINI BENCANA TSUNAMI
BERBASIS IOT DENGAN MENGGUNAKAN
LIBRARY REACTJS**

Nama Mahasiswa : **Ranto**

Nomor Pokok Mahasiswa : **1915061036**

Program Studi : **Teknik Informatika**

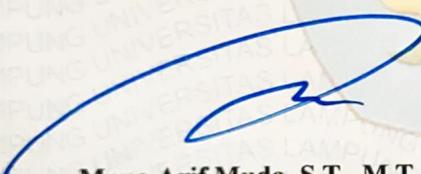
Fakultas : **Teknik**

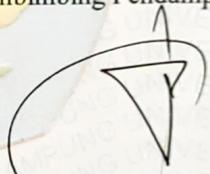


1. Komisi Pembimbing

Pembimbing Utama

Pembimbing Pendamping

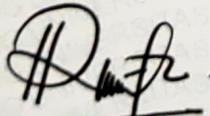

Mona Arif Muda, S.T., M.T.
NIP. 197111122000031002

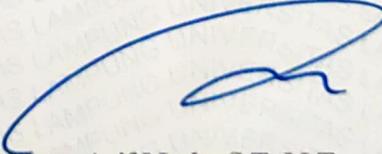

Denny Budiyanto, S.Kom, M.T.
NIP. 19912082019031011

2. Mengetahui

Ketua Jurusan
Teknik Elektro

Ketua Program Studi
Teknik Informatika


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


Mona Arif Muda, S.T., M.T.
NIP. 197111122000031002

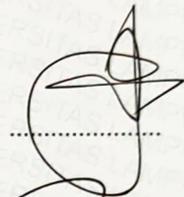
MENGESAHKAN

1. Tim Penguji

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



Sekretaris : **Deny Budiyanto, S.Kom, M.T**



Penguji : **Ir. Gigih Forda Nama, S.T., M.T.I, IPM**



Dekan Fakultas Teknik



Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc. 
NIP. 19750928 200112 1 002

Tanggal Lulus Ujian Skripsi : **8 Agustus 2023**

SURAT PERNYATAAN

Saya yang bertandatangan di bawah ini , menyatakan bahwa skripsi saya dengan judul “Membangun Frontend Website Sistem Informasi Pemantauan Dan Peringatan Dini Bencana Tsunami Berbasis IoT Dengan Menggunakan Library React JS” 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, 8 Agustus 2023

Pembuat pernyataan,



Ranto

NPM. 1915061036

RIWAYAT HIDUP



Penulis lahir di Baktirasa, 23 Mei 2001. Penulis merupakan anak pertama dari dua bersaudara dari pasangan Bapak Iskandar dan Ibu Wagiah. Pendidikan penulis SDN 2 Kedaung Lampung selatan pada tahun 2007 hingga 2013, MTs. Kesuma Sumbersari pada tahun 2013 hingga 2016, dan SMKN 1 Ketapang pada tahun 2016 hingga 2019.

Penulis menjadi mahasiswa Jurusan Teknik Elektro Universitas Lampung pada tahun 2019 melalui jalur SBMPTN (Seleksi Bersama Masuk Perguruan Tinggi Negeri). Selama menjadi mahasiswa, penulis pernah menjadi anggota Himpunan Mahasiswa Teknik Elektro (HIMATRO) sebagai anggota divisi Media Informasi pada periode 2020 dan periode 2021. Penulis juga menjadi anggota Lembaga riset Unila Robotika dan Otomasi (URO) sejak 2019 sebagai anggota robotika dan sebagai programmer. Penulis tertarik dalam bidang teknologi khususnya web dan otomasi.

Pada tahun 2022 penulis mengikuti program kampus merdeka belajar dengan mengambil belajar di sebuah instansi dengan fokus pembelajaran reactjs. Di mulai dari program tersebut terciptalah percikan untuk menyukai di bidang programmer. Pada semester selanjutnya penulis mengikuti program merdeka belajar dengan mengambil fokus ke reactJS kembali dan menjadikan penulis sudah bisa dikatakan bisa membuat sebuah website secara mandiri.

MOTTO

“Minimal Praktek bukan wacana”
(Penulis)

“Allah akan meninggikan orang-orang yang beriman diantaramu dan orang-orang yang diberi ilmu pengetahuan beberapa derajat.”

(Q.S. Al-Mujadalah : 11)

“Apapun yang menjadi takdirmu, akan mencari jalannya menemukanmu.”

(Ali Bin Abi Thalib)

“Orang bijak belajar ketika mereka bisa. Orang bodoh belajar ketika mereka terpaksa.”

(Arthur Wellesley)

“Fokuslah menjadi produktif, bukan sekadar sibuk saja.”

(Tim Ferris)

PERSEMBAHAN

Dengan mengucapkan Alhamdulillah dan syukur atas nikmat dan hidayahnya yang diberikan Allah SWT sehingga penulis bisa menyelesaikan skripsi ini dengan baik:

Rasa syukur dan Bahagia saya persembahkan karya ini kepada:

Kedua Orang Tua

Terima kasih kepada kedua orang tuaku tercinta, Ibu Wagiah dan alm Bapak Iskandar atas segala pengorbanan, doa dan ridho kalian serta dukungannya selama ini. Terima kasih telah memberikan pelajaran berharga tentang makna perjalanan hidup yang sebenarnya sehingga kelak bisa menjadi orang yang bermanfaat bagi semua orang.

Dosen Pembimbing dan Pembahas

Terima kasih kepada dosen pembimbing dan pembahas yang sudah sangat membantu, memberikan motivasi, memberikan arahan serta ilmu yang berharga.

Keluarga dan Teman-temanku

Terimakasih kepada semua orang-orang baik yang telah memberikan pengalaman, semangat, motivasinya, serta doa-doanya dan senantiasa memberikan dukungan dalam hal apapun.

Almamater Tercinta

Universitas Lampung

SANWACANA

Segala puji bagi Allah, atas limpahan nikmat-Nya yang diberikan kepada penulis sehingga dapat menyelesaikan Tugas Akhir ini. Shalawat dan salam senantiasa dicurahkan kepada Nabi Muhammad SAW, suri teladan yang mampu membuka sesuatu yang terkunci, penutup dari semua yang terdahulu, penolong kebenaran dengan jalan yang benar, dan petunjuk kepada jalanMu yang lurus. Tugas Akhir dengan judul “*Membangun Frontend Website Sistem Informasi Pemantauan Dan Peringatan Dini Bencana Tsunami Berbasis Iot Dengan Menggunakan Library Reactjs*” ini merupakan salah satu syarat untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Informatika Jurusan Teknik Elektro, Fakultas Teknik, Universitas Lampung.

Pada kesempatan ini, penulis mengucapkan terimakasih kepada:

1. Orang tua yang sangat saya cintai, Ibu Wagiah dan Alm. Bapak Iskandar yang nasihat-nasihatnya yang selalu saya ingat, serta saudara - saudara saya, adek Irwanda, dan kekasih saya Enjel Violani yang sangat penulis banggakan dan sayangi yang telah mendukung penuh sehingga penulis dapat menyelesaikan laporan tugas akhir ini.
2. Bapak Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc. selaku Dekan Fakultas Teknik Universitas Lampung.
3. Ibu Herlinawati, S.T., M.T. selaku Kepala Jurusan Teknik Elektro Universitas Lampung.
4. Bapak Mona Arif Muda, S.T., M.T. selaku Kepala Prodi Teknik Informatika Universitas Lampung dan selaku dosen pembimbing utama yang telah memberikan nasihat, arahan, dan bimbingan rutin, motivasi, dan pandangan

kehidupan kepada penulis di setiap kesempatan dengan baik dan ramah dalam mempersiapkan diri menjadi seorang Sarjana Teknik.

5. Ibu Deny Budiyanto, S.Kom, M.T. selaku dosen pembimbing pendamping yang telah memberikan bimbingan, arahan, dan nilai-nilai kehidupan kepada penulis dengan baik dan ramah.
6. Bapak Ir. Gigih Forda Nama, S.T.,M.T.I., IPM selaku dosen penguji yang telah memberikan kritik dan saran yang membangun kepada penulis dalam mengerjakan skripsi ini.
7. Bapak Wahyu Eko Sulistiono, S.T., M.Sc. selaku dosen pembimbing akademik yang telah memberikan bimbingan, arahan, pandangan hidup, dan motivasi kepada penulis untuk menjadi pribadi yang lebih baik di masa depan.
8. Segenap Dosen di Jurusan Teknik Elektro yang telah memberikan ilmu yang bermanfaat, wawasan, dan pengalaman bagi penulis.
9. Mbak Rika dan Segenap Staff di Jurusan Teknik Elektro dan Fakultas Teknik yang telah sangat membantu penulis baik dalam hal administrasi dan hal-hal lainnya.
10. Angkatan tercinta Eternity dan teman teman PSTI 2019 terima kasih sudah menjadi keluarga selama awal kuliah sampai saat ini.
11. Seluruh Tim Project PUMMA U-TEWS yang telah memberikan bantuan serta dorongan penulis dalam penyelesaian project yang dibuat.
12. Teman-teman satu perjuangan yang berasal dari SMKN 1 KETAPANG yang meberikan dukungan penuh untuk menyelesaikan tugas skripsi dengan baik.
13. Seluruh teman-teman yang terlibat langsung maupun tidak langsung dalam penyelesaian skripsi.

Penulis menyadari bahwa masih banyak kekurangan dalam penulisan skripsi ini. Penulis mengharapkan kritik dan saran konstruktif dari semua pihak demi kemajuan bersama. Penulis berharap skripsi ini dapat bermanfaat bagi kita semua.

Bandar Lampung, 8 Agustus 2023

Penulis

Ranto

DAFTAR ISI

	Halaman
SANWACANA	i
DAFTAR ISI	iv
DAFTAR GAMBAR	viii
DAFTAR TABEL	xi
I. PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Tujuan Penelitian.....	3
1.3. Rumusan Masalah	3
1.4. Batasan Masalah.....	3
1.5. Manfaat Penelitian.....	3
1.6. Sistematika Penulisan.....	4
II. TINJAUAN PUSTAKA	6
2.1. Penelitian Terkait	6
2.2. Kondisi Saat ini	8
2.3. <i>Website</i>	10

2.4.	<i>Early Warning System (EWS)</i>	10
2.5.	<i>Internet of Things (IoT)</i>	11
2.6.	<i>Message Queue Telemetry Transport (MQTT)</i>	12
2.7.	<i>ReactJS</i>	13
2.8.	<i>Chart.JS</i>	15
2.9.	<i>REST-API</i>	16
2.10.	<i>MQTT.js</i>	16
2.11.	<i>Kanban</i>	17
2.12.	<i>Axios</i>	19
2.13.	<i>TailwindCSS</i>	20
2.14.	<i>Visual Studio Code</i>	20
2.15.	<i>Trello</i>	21
2.16.	<i>Webhoster</i>	22
2.17.	<i>Black-box Testing</i>	23
III.	METODOLOGI PENELITIAN	24
3.1.	Waktu dan Tempat Penelitian	24
3.2.	Jadwal Penelitian	24
3.3.	Alat dan Bahan	25
3.3.1.	Alat Penelitian.....	25
3.3.2.	Bahan Penelitian.....	26
3.4.	Tahapan Penelitian	27
3.4.1.	Target Pengguna.....	28
3.4.2.	Tahap Penentuan <i>User story</i>	28

3.4.3.	Tahap Analisis.....	30
3.4.4.	Tahap Pengembangan Sistem	34
3.5.	Tahap Testing	36
3.6.	Tahap Pelaporan	37
IV.	PEMBAHASAN	38
4.1.	<i>User story</i>	38
4.1.1.	<i>Use case Diagram</i>	40
4.1.2.	<i>Activity Diagram</i>	41
4.2.	Tahap Analisis	45
4.3.	Pengembangan.....	47
4.3.1.	<i>Flow pertama</i>	47
4.3.2.	<i>Flow kedua</i>	48
4.3.3.	<i>Flow ketiga</i>	50
4.3.4.	<i>Flow keempat</i>	53
4.3.5.	<i>Flow kelima</i>	55
4.3.6.	<i>Flow keenam</i>	58
4.3.7.	<i>Flow ketujuh</i>	61
4.3.8.	<i>Flow kedelapan</i>	63
4.3.9.	<i>Flow kesembilan</i>	65
4.4.	<i>Testing</i>	67
4.5.	<i>Release</i>	71
4.6.	Analisis hasil	72

V. KESIMPULAN	75
5.1. Kesimpulan.....	75
5.2. Saran.....	76
DAFTAR PUSTAKA	77

DAFTAR GAMBAR

Gambar	Halaman
1. Perangkat sistem pemantauan dan peringatan bencana yang ditempatkan di Pantai Canti, Kalianda, Lampung Selatan.....	8
2. Biaya langganan dan batasan akun pada ThingsBoard	9
3. Cara Kerja MQTT	13
4. Visualisasi papan kerja metode kanban	17
5. Black-box Testing	23
6. Tahapan pada metode Kanban	27
7. Format user story yang digunakan dalam penelitian.....	28
8. Environment keseluruhan sistem	29
9. Task card	30
10. Kanban board pada penelitian yang dilakukan	34
11. Kanban board pada penelitian yang dilakukan	35
12. Kanban board pada penelitian yang dilakukan	35
13. Use Case Dashboard Early Warning System (EWS).....	40
14. Activity Diagram masuk halaman utama melihat list alat yang terpasang	41
15. Activity Diagram Memilih list alat yang ingin dibuka	41
16. Activity Diagram Melihat seluruh detail data alat	42
17. Activity Diagram menampilkan Grafik berdasarkan hari (maksimal 7 hari)	42
18. Activity Diagram melihat titik lokasi alat.....	43
19. Activity Diagram melihat gambar terkini	44
20. Activity Diagram melihat gambar terkini	44

21. Flow pertama pengembangan sistem	47
22. Desain list dan layout website.....	48
23. Flow kedua pengembangan sistem.....	48
24. Tampilan kode program komponen layout	49
25. Tampilan kode program integrasi halaman list alat	49
26. Hasil dari fetching list alat berupa data log.....	50
27. Flow ketiga pengembangan sistem	50
28. Desain halaman utama	51
29. Koding program tampilan desain list alat	52
30. Tampilan halaman HomePage	52
31. Flow keempat pengembangan sistem.....	53
32. Koding integrasi mqtt.....	53
33. Koding integrasi api berdasarkan waktu	54
34. Hasil dari fetching data api berdasarkan waktu berupa data log.....	55
35. Flow kelima pengembangan sistem	55
36. Koding untuk menampilkan data grafik.....	56
37. Koding pengontrol select pemilihan data berdasarkan waktu.....	56
38. Tampilan chart ketika server mengirimkan data berdasarkan waktu.....	57
39. Tampilan chart dengan waktu bawaan.....	57
40. Tampilan chart dengan waktu 3 hari	58
41. Flow keenam pengembangan sistem.....	58
42. Tampilan kode program konfigurasi mqtt.....	59
43. Tampilan kode program menampilkan data.....	60
44. Hasil dari konfigurasi data mqtt berupa data log ... Error! Bookmark not defined.	
45. Tampilan card data mqtt.....	61
46. Flow ketujuh pengembangan sistem	61

47. Tampilan kode program integrasi data gambar.....	61
48. Tampilan kode program integrasi api dengan 10 data	62
49. Tampilan kode dari integrasi gambar.....	62
50. Tampilan data hasil integrasi data 10 terakhir	63
51. Flow kedelapan pengembangan sistem.....	63
52. Flow kedelapan pengembangan sistem.....	63
53. Tampilan kode Maps.....	64
54. Tampilan tabel data 10 terkahir.....	64
55. Tampilan peta dari titik lokasi terdekat.....	65
56. Flow kesembilan pengembangan sistem.....	65
57. Tampilan kode countdown.....	66
58. Tampilan kode responsive website	66
59. Tampilan countdown setiap lokasi alat.....	67
60. Tampilan responsive mobile	67
61. perintah build project	71
62. create direktori pada cpanel	71
63. Akun FTP.....	72
64. Upload file dengan filezila.....	72
65. Link website realease	72

DAFTAR TABEL

Tabel	Halaman
1. Jadwal Penelitian.....	24
2. Alat yang digunakan dalam penelitian.....	25
3. Bahan berupa data sensor dari raspberry pi yang dikirimkan oleh server menggunakan MQTT dan REST API.....	26
4. Tim pengembangan project.....	29
5. Daftar user story pada system	38
6. Penentuan level pengerjaan backlog	45
7. Hasil pengujian blackbox testing pada Monitoring List Status Alat.....	69
8. Hasil pengujian blackbox testing pada laman homepage	70
9. Waktu yang delay mqtt (satu data).....	73
10. Waktu yang delay http (satu data).....	73

I. PENDAHULUAN

1.1. Latar Belakang

Bencana adalah fenomena alam yang dapat terjadi kapan saja dan tidak dapat dihindari oleh siapapun. Bencana alam yang umum terjadi di Indonesia antara lain air pasang dan tsunami. Bencana alam ini sering terjadi di Indonesia, terutama di wilayah pesisir dan laut karena permukaan air laut naik secara tiba-tiba dan air naik di daratan sehingga menyebabkan bencana alam seperti banjir dan tsunami.

Menurut data BNPB, antara tahun 2021 hingga 2023 akan terjadi 76 kali kejadian gelombang pasang/goresan di Indonesia. Tanpa adanya sistem untuk memantau dan meminimalisir terjadinya gelombang pasang, dapat menjadi bencana alam yang serius ketika gelombang laut melebihi batas normal, selain membahayakan lautan dan daratan yang bersebelahan, terutama di wilayah pesisir atau pesisir [1].

Tsunami di Pandeglang, Banten pada 22 Desember 2018 menewaskan sedikitnya 426 orang tewas, 7.202 luka-luka dan 23 hilang akibat kejadian ini. Secara fisik Kerusakan akibat tsunami Selat Sunda sebanyak 2752 rumah rusak, 92 sarana akomodasi dan warung rusak, 510 kapal rusak, 147 kendaraan roda dua dan empat rusak dan 1 dermaga rusak. Rusaknya bangunan pasca bencana memaksa pemerintah mengambil kebijakan strategis penanggulangan bencana berupa rekonstruksi [1].

Provinsi Lampung merupakan salah satu provinsi yang sebagian besar dikelilingi oleh laut dan pantai. Letak geografis provinsi Lampung terletak di antara Samudera Hindia di sebelah barat, Laut Jawa di sebelah timur, dan Selat Sunda di

sebelah selatan. Hal ini menjadikan Provinsi Lampung sebagai daerah yang harus diwaspadai tsunami dan gelombang pasang karena ketiga sisinya berbatasan dengan laut [2].

Pada 22 Desember 2018, tsunami melanda masyarakat pesisir di provinsi Lampung dan Banten. Terletak di sepanjang Selat Sunda antara Jawa dan Sumatera, Indonesia, 24 menit kemudian Letusan gunung berapi di pulau itu sekitar pukul 09.03 waktu setempat. Menurut Badan Meteorologi Indonesia, Badan Iklim dan Geologi mengatakan tsunami kemungkinan disebabkan oleh tanah longsor di dasar laut. merupakan hasil runtuh sebagian gunung berapi Anak Krakatau dan dapat diperparah pada musim semi pasang surut, di mana tingkat pasang surut adalah yang tertinggi. Gambar satelit dan udara mengkonfirmasi keruntuhan barat daya gunung berapi keesokan harinya. Gunung berapi ini pernah mengalami erupsi pada Desember 2018 yang mengakibatkan bencana tsunami yang menewaskan 108 Orang [3].

Berdasarkan kondisi tersebut, maka diperlukan sistem peringatan dini dan pemantauan bencana untuk mempercepat pemberian informasi tentang bencana yang akan datang sehingga evakuasi dapat dilakukan dengan cepat. Saat ini telah dibangun sistem peringatan dini dan pemantauan yang disebut juga Early Warning System (EWS) dan ditempatkan di beberapa titik di pantai pesisir lampung untuk memantau situasi laut, namun saat ini sistem tersebut masih menggunakan sistem http sebagai transfer data dan terhubung dengan pihak ketiga, khususnya ThingsBoard, yang memiliki beberapa kelemahan. Diantaranya adalah waktu pengiriman data yang sedikit lebih lama, lalu lintas data yang terbatas di setiap akun, biaya berlangganan, dan database yang terletak di server pihak ketiga untuk data atribut, tidak menjadi data pribadi dan tidak dapat menyimpan data historis yang terekam dalam waktu yang lama. Selain itu, keamanan sistem kurang terjamin karena data sensor yang dikirimkan disimpan oleh pihak ketiga. Ini akan berpengaruh pada tampilan di dashboard ThingsBoard. Yang menyebabkan tampilan akan berhenti ketika data penuh dan harus menghapus data tersebut.

Dalam penelitian ini akan dikembangkan sistem pemantauan dan peringatan dini yang mengimplementasikan data yang diberikan oleh backend kedalam bentuk

tampilan website berupa dashboard pemantauan seperti ThingsBoard. Kelebihan sistem ini adalah berdiri sendiri dan bisa diatur untuk tampilan yang ingin ditampilkan.

1.2. Tujuan Penelitian

Tujuan dari penelitian ini adalah membangun sebuah sistem monitoring Early Warning System tsunami yang berisi data real-time dan data tidak real-time serta dan memberikan informasi tentang data adanya potensi bahaya tsunami dalam bentuk grafik dan card status warning tsunami.

1.3. Rumusan Masalah

Adapun rumusan masalah pada penelitian ini adalah bagaimana cara membuat sistem yang informatif untuk menyampaikan informasi data realtime dan data non real time pemantau gelombang air laut yang berpotensi tsunami.

1.4. Batasan Masalah

Adapun batasan masalah pada penelitian ini adalah sebagai berikut.

1. Pengembangan aplikasi hanya berfokus pada pengembangan Front-end dan konfigurasi data pada website desktop.
2. Pengembang aplikasi melakukan publikasi dengan memanfaatkan layanan cpanel dari web hosting.

1.5. Manfaat Penelitian

Manfaat Penelitian ini adalah sebagai berikut:

1. Membantu Instansi resmi dan Instansi terkait dalam melakukan memonitoring dan menganalisis data bencana gelombang pasang dan tsunami yang berpotensi terjadi.
2. Memudahkan masyarakat dalam memonitoring akan terjadinya bencana alam gelombang pasang dan tsunami yang akan terjadi.

1.6. Sistematika Penulisan

Sistematika penulisan yang digunakan pada skripsi ini adalah pembagian menjadi 5 bab sebagai berikut:

BAB I : PENDAHULUAN

Bab ini menguraikan secara umum mengenai latar belakang penelitian, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan penelitian, dan sistematika penulisan.

BAB II : TINJAUAN PUSTAKA

Pada Bab ini membahas mengenai penelitian terkait dengan protokol *MQTT* pada sistem *iot* dan *REST API*, kondisi saat ini pada alat sensor di salah satu tempat, *Early Warning System (EWS)*, *ReactJS*, *ChartJS*, *REST API*, *MQTT.JS*, *Kanban Method*, *Axios*, *TailwindCSS*, *Visual Studio Code*, *Trello*, *Webhoster*, *Blackbox-testing*.

BAB III : METODE PENELITIAN

Pada Bab ini membahas mengenai metode *kanban* yang terdiri dari *requirement gathering*, penentuan *user story*, *breakdown user story* menjadi *backlog (WOW)*, tahap *Work in Progress (WIP)*, *testing* dan evaluasi yang digunakan dalam pengembangan *Website Sistem Informasi Pemantauan Dan Peringatan Dini Bencana Tsunami Berbasis Iot Dengan Menggunakan library Reactjs*.

BAB IV : HASIL DAN PEMBAHASAN

Pada Bab ini berisi tentang hasil serta pembahasan yang diperoleh dalam penelitian.

BAB V : KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari hasil penelitian dan saran-saran sebagai masukan untuk penelitian lanjutan di masa mendatang.

DAFTAR PUSTAKA**LAMPIRAN**

II. TINJAUAN PUSTAKA

2.1. Penelitian Terkait

Pada penelitian yang dilakukan oleh Nasi Tantitharanukul pada tahun 2017 menjelaskan berbagi data antar kota antaranya data pemantauan banjir, data gempa, dan data lalu lintas. Pada penelitian ini menggunakan *protocol MQTT*, atau *protocol*, protokol perpesanan ringan berdasarkan konsep terbitkan-berlangganan. Pemanfaatan protokol *MQTT* menggunakan dua elemen pemesanan untuk berbagi data yaitu topik dan pesan. Yang pertama topik digunakan untuk menentukan pelanggan mana yang menerima data mana dan data yang perlu kita bagikan disajikan dalam elemen pesan. Penelitian ini menggunakan *MQTT-Topic Naming Criteria of Open Data for Smart Cities (MTNC)* untuk berbagi data menggunakan protokol MQTT. *MQTT-Topics Management System (MTMS)* digunakan untuk mengelola topik pesan MQTT yang dibagikan oleh penerbit mana pun berdasarkan MTNC [4].

Penelitian Choopan Rattanapoka tentang pemantauan kualitas udara berbasis *MQTT* dengan *NodeMCU* dan *Node-RED* mendemonstrasikan pemantauan kualitas udara berbasis *MQTT*. Menggunakan *NodeMCU ESP8266* yang dikombinasikan dengan sensor yang mengukur suhu, kelembaban, karbon monoksida (CO), gas ozon (O3), dan PM2.5. Broker MQTT digunakan untuk menerima data yang dikirim oleh pembacaan sensor. Peneliti kemudian menggunakan *node red* untuk membuat aliran untuk mengelola dan mengatur data yang diterima. Data yang terkumpul kemudian dikirim ke Air Quality Monitor, aplikasi web responsif yang menampilkan data dalam alat pengukur, teks, dan .

grafik pada antarmuka pengguna. Jika nilai data melebihi rentang yang ditentukan, Node-RED akan mengirimkan pesan peringatan melalui *LINE Notify* untuk memberitahu pengguna [5].

Terdapat juga penelitian dari Rohan Banda dan Atul Oak yang meneliti tentang pengelolaan identifikasi lokasi dan sms berantai untuk sistem transportasi sekolah cerdas menggunakan Iot. Secara tradisional, sekolah telah menyediakan layanan transportasi kepada siswa dan staf yang tidak dapat diandalkan, kurangnya komunikasi dengan orang tua dan layanan antar jemput siswa tidak terjamin. Dengan berkembangnya teknologi, sistem GPS digunakan untuk melacak bus sekolah, tetapi sistemnya kurang aman. Maka dari itu peneliti menggunakan IoT untuk melakukan pelacakan terhadap bus yang dinaiki oleh anak-anak sekolah agar bisa dipantau oleh orang tua secara langsung. Dengan memanfaatkan GPS dan alat yang deteksi yang di pasang di bus. Area dipindai diambil dari server dan disimpan dalam database. Segmen perangkat yang akan digunakan ditentukan nama Raspberry Pi. Itu termasuk dalam cloud dan memiliki kapasitas akses melalui perangkat lunak seluler. Tujuannya adalah untuk menunda waktu pelanggan dapat dibatasi. Metode sederhana dan komunikatif sebagai bagian dari pelacakan bus teguh Aplikasi ini dapat diperluas ke kerangka pemantauan pusat semua bus Perhatian yang jelas dan panduan rute yang andal mudah dimungkinkan dengan sistem server pusat. Juga dimungkinkan melalui *MQTT* daripada menggunakan Hypertext Transfer Protocol (*HTTP*) yang digunakan secara tradisional akan ringan, hemat data, dan dapat diskalakan. Melalui *MQTT* seperti halnya *HTTP* (*Hypertext Transfer Protocol*) yang digunakan secara tradisional. Ringan, hemat data, dan dapat diskalakan [6].

Selanjutnya tentang penelitian yang terkait dengan reactjs yang menggunakan protokol mqtt untuk visualisasi sinyal lampu dan notifikasi secara waktu nyata pada sistem pemonitor APILL di kota Pekanbaru. Pada penelitian ini peneliti bermula pada masalah pengontrol simpang bersinyal yang masih menggunakan controller yang bersifat analog. Yang mana jika terjadi kerusakan pada pada controller APILL akan menjadikan kemacetan pada jalan. Untuk itu difokuskan pada penelitian ini pemanfaatan reactJs dan *MQTT*. *ReactJS*

dimanfaatkan untuk menampilkan data-data sinya lampu *APILL* serta memberikan notifikasi secara real time ke dalam browser web. Sedangkan MQTT digunakan sebagai media untuk mengelola komunikasi data antara modul deteksi yang dipasang pada controller APILL dilapangan ke server Back-End. Data inilah yang kemudian ditampilkan ke browser web melalui React JS. Dengan adanya solusi pada penelitian ini, maka diharapkan Dinas Perhubungan kota Pekanbaru dapat mengetahui permasalahan lampu APILL di lapangan secara cepat, sehingga bisa segera mengambil tindakan relevan yang diperlukan[7].

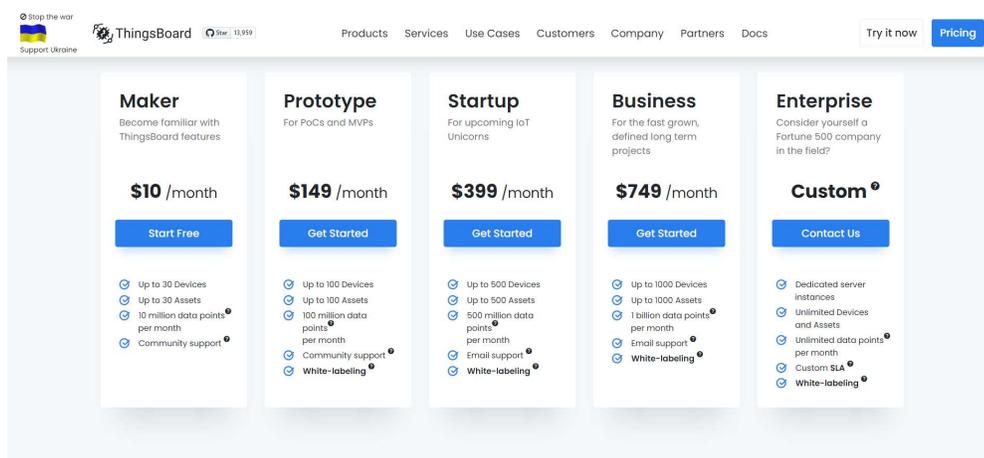
2.2. Kondisi Saat ini

Penelitian ini dilaksanakan berdasarkan dari sistem pemantauan dan peringatan dini bencana tsunami yang sudah berjalan. Pada sistem yang berjalan ini, terdapat beberapa perangkat yang salah satunya menggunakan *Raspberry pi* sebagai piranti utama untuk mengontrol sensor pengukur ketinggian permukaan air laut yang dipasang pada alat yang terletak di Pantai Canti Kalianda, Lampung Selatan. Sistem akan bekerja dengan mengukur ketinggian air laut secara periodik dan mengirimkan data tersebut ke aplikasi dashboard pihak ketiga yaitu *ThingsBoard*. Protokol yang digunakan untuk mengirimkan data tersebut masih menggunakan *Hypertext Transfer Protocol (HTTP)*. Data yang diterima oleh sensor akan berlangsung dikirimkan ke server milik *ThingsBoard* menggunakan *HTTP* dan data tersebut akan disimpan ke *database* milik *ThingsBoard*.



Gambar 1 Perangkat sistem pemantauan dan peringatan bencana yang ditempatkan di Pantai Canti, Kalianda, Lampung Selatan.

Sistem yang berjalan saat ini menggunakan aplikasi pihak ketiga dalam mengirim, menyimpan dan menampilkan data. Namun ini semua mempunyai beberapa kelemahan yang akan menjadi kendala untuk keberlangsungan dari informasi yang akan diterima oleh umum. Yang pertama, untuk melakukan pengiriman, penyimpanan dan menampilkan data di dashboard membutuhkan akun, dimaka akun tersebut akan dikenakan biaya langganan dalam penggunaannya. Selain itu, dalam setiap akun dibatasi dengan jumlah data points / traffic yang dikirimkan setiap bulannya. Kelemahan selanjutnya adalah keamanan data yang dikirimkan tidak dapat bersifat milik pribadi dikarenakan data tersebut tersimpan di database milik ThingsBoard dan akan dihapus apabila telah melampaui traffic data yang telah ditentukan. Selain itu kelemahan terakhir adalah dalam desain untuk website yang akan ditampilkan harus mengikuti desain yang disediakan oleh *ThingsBoard* yang terbatas.



Gambar 2. Biaya langganan dan batasan akun pada ThingsBoard

Berdasarkan kondisi tersebut, penelitian ini dilakukan untuk mengatasi kelemahan yang terjadi ketika menggunakan pihak ketiga sebagai media pemantauan dan peringatan bencana alam yang terjadi saat ini berjalan masih menggunakan protokol pengiriman dan penyimpanan data dari aplikasi ketiga.

Untuk mengatasi kelemahan tersebut, maka telah dibuat sebuah protocol pengiriman data *MQTT* untuk mengirimkan data realtime dan tetap menggunakan HTTP untuk data bukan real-time oleh peneliti sebelumnya. Yang pada pada penelitian ini berfokus pada tampilan dashboard yang memanfaatkan data yang

dikirim oleh backend yang berupa data real time dan data bukan real-time yang berupa rest-api yang akan ditampilkan di dashboard pumma.isi-net.org. Website ini akan menggantikan peran dashboard *ThingsBoard* yang berbayar dan hanya menyediakan tampilan yang tidak bisa kita sesuaikan dengan keinginan. Dengan menggunakan ReactJS akan mewujudkan tujuan dari penelitian ini yaitu untuk mempermudah akses masyarakat umum dan peneliti untuk mengontrol keadaan ketinggian permukaan air laut yang diletakkan alat pengukur tersebut.

2.3. Website

Website sebuah media informasi yang ada di internet. *Website* tidak hanya dapat digunakan untuk penyebaran informasi saja melainkan bisa digunakan untuk membuat toko online. *Website* adalah kumpulan dari halaman-halaman situs, yang biasanya terangkum dalam sebuah domain atau subdomain, yang tempatnya berada di dalam *World Wide Web (WWW)* di Internet. Sebuah halaman *web* adalah dokumen yang ditulis dalam format *HTML (Hyper Text Markup Language)*, yang hampir selalu bisa diakses melalui *HTTP*, yaitu protokol yang menyampaikan informasi dari *server website* untuk ditampilkan kepada para pemakai melalui *web browser*. Semua publikasi dari *website-website* tersebut dapat membentuk sebuah jaringan informasi yang sangat besar [8].

2.4. 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.

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.

Early Warning System (EWS) merupakan suatu sistem yang dibentuk dengan rancangan konsep deteksi dini sehingga dapat mencari manfaat dan risiko dengan lebih cepat atau bisa disebut dengan sistem peringatan dini. Tujuan dari pengelolaan risiko teridentifikasi ini adalah agar dapat menurunkan tingkat kerugian. Terdapat beberapa tipe komponen yang melandasi *early warning system* yaitu pedoman fungsional, Pedoman organisasi pada level departemen, Petunjuk kemajuan dan data promosi, Data pembantu, dan Pemeriksaan. Alert atau jenis sistem peringatan dini terdiri atas beberapa bentuk, antara lain *Database Alert, Email Alert, SMS Alert, dan Sounds Alert* [9].

2.5. 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 [10].

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.

Contoh aplikasi IoT meliputi:

1. *Smart Home*

Memungkinkan pengendalian perangkat-perangkat di rumah seperti lampu, suhu, keamanan, dan perangkat elektronik lainnya melalui jaringan internet.

2. *Smart City*

Menggunakan sensor dan koneksi internet untuk mengelola dan memantau infrastruktur kota seperti pencahayaan, parkir, pengelolaan limbah, dan lalu lintas.

3. *Smart Healthcare*

Memungkinkan pemantauan kesehatan pasien secara real-time melalui perangkat medis yang terhubung, serta pengelolaan inventaris obat dan perangkat medis.

2.6. *Message Queue Telemetry Transport (MQTT)*

Message Queue Telemetry Transport (MQTT) adalah salah satu protokol paling luas di *Internet of Things (IoT)*. Namun, protokol ini tidak menerapkan skema keamanan yang kuat secara default, yang tidak memungkinkan mekanisme otentikasi yang aman antara peserta komunikasi. Selain itu, kami tidak dapat mempercayai kerahasiaan dan integritas data. Perangkat *IoT* yang ringan mengirimkan data yang semakin masuk akal di area *Smart Building*, *Smart City*, *Smart House*, *Smart Car*, *Connected Car*, *Health Care*, *Smart Retail*, *Industrial IoT (IIoT)*, dll. Hal ini membuat tantangan keamanan dalam protokol yang digunakan di *IoT* sangat penting. Standar protokol *MQTT* sangat menyarankan untuk mengimplementasikannya melalui *Transport Layer Security (TLS)* daripada *TCP* biasa. Meskipun begitu, opsi ini tidak dimungkinkan di sebagian besar perangkat ringan yang membentuk ekosistem *IoT*. Cukup sering, sumber daya perangkat *IoT* yang terbatas mencegah penggunaan algoritma kriptografi asimetris aman yang diterapkan sendiri.

MQTT dibagi menjadi tiga:

1. *BROKER*

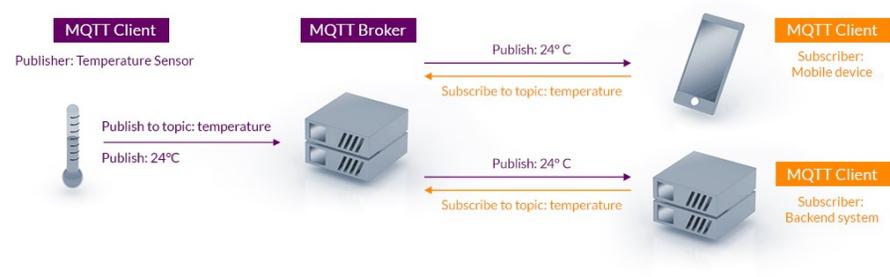
merupakan pusat bintang dalam protokol *MQTT* dan bertugas dalam pertukaran pesan antar peserta lainnya. Semua peserta lain terhubung dengannya dan hanya dengannya, jadi dia juga bertanggung jawab atas otentikasi semua peserta dalam jaringan.

2. *PUBLISHER*

adalah elemen yang mengirim data ke broker sehingga mengirimkan data ini ke satu atau lebih subscribe.

3. *SUBSCRIBER*

adalah elemen yang menerima data ke *broker*. Data yang mereka terima adalah data yang dikirim oleh penerbit.



Gambar 3. Cara Kerja *MQTT*

Cara kerja dari *MQTT* adalah dengan menggunakan metode *publish / subscribe*. Metode ini bekerja dengan cara satu perangkat mengirimkan sebuah pesan (*publish*), kemudian perangkat lain yang bertindak sebagai penerima (*subscribe*) dapat menerima pesan yang telah dikirimkan tadi sesuai dengan topic yang sesuai. Seluruh pengiriman dan penerimaan pesan ini diatur oleh *MQTT* Broker [11].

2.7. *ReactJS*

ReactJS adalah salah satu kerangka kerja web populer yang lebih penting daripada kerangka kerja lain seperti Angular, Vue, dll. Ini karena penerapan Virtual

DOM, yang tujuan utamanya adalah untuk meningkatkan kinerja aplikasi secara keseluruhan.

ReactJS adalah kerangka kerja web yang terutama dirancang untuk mengatasi masalah kinerja dalam aplikasi web. React menggunakan DOM virtual yang memutuskan apakah komponen harus dimuat ulang atau tidak berdasarkan status komponen saat ini dan perubahan yang telah terjadi. Ini mencegah aplikasi merender ulang yang tidak perlu. Selain itu, React juga memperkenalkan aliran data satu arah yang membantu mengontrol aliran data dalam aplikasi yang membuat pelacakan kejadian menjadi lebih mudah dan juga menyederhanakan propagasi dan stabilitas [12].

ReactJS adalah pustaka JavaScript yang populer untuk membangun antarmuka pengguna (*UI*) interaktif. Dikembangkan oleh *Facebook*, *ReactJS* digunakan untuk membuat komponen *UI* yang dapat digunakan ulang dan dapat dikelola dengan mudah.

Beberapa fitur utama dari ReactJS meliputi:

1. Komponen

ReactJS memandang UI sebagai kumpulan komponen yang dapat digabungkan bersama untuk membentuk tampilan yang lengkap. Setiap komponen berfungsi sebagai unit terpisah yang dapat memiliki logika, properti (*properties*), dan keadaan (*state*) mereka sendiri.

2. *Virtual DOM*

ReactJS menggunakan *Virtual DOM (Document Object Model)* untuk mempercepat manipulasi UI. *Virtual DOM* adalah representasi virtual dari struktur *DOM* aktual, dan *ReactJS* mengoptimalkan perubahan pada *Virtual DOM* untuk meminimalkan pengubahan langsung pada *DOM* aktual, yang dapat memakan waktu dan mempengaruhi kinerja.

3. Reaktivitas

ReactJS memperhatikan perubahan keadaan (*state*) dan properti (*props*) komponen. Ketika ada perubahan dalam keadaan atau properti, *ReactJS* secara otomatis memperbarui tampilan yang terkait dengan efisien melalui

rekonsiliasi *Virtual DOM*.

4. *JSX*

ReactJS memperkenalkan *JSX (JavaScript XML)* yang memungkinkan penulisan kode *JavaScript* dan *HTML* secara bersamaan. *JSX* mempermudah penulisan dan pemahaman komponen *ReactJS* dengan sintaks yang dekat dengan *HTML*.

5. Ekosistem yang Kuat:

ReactJS memiliki ekosistem yang kaya dengan banyak pustaka dan alat pendukung seperti *React Router* untuk *routing*, *Redux* untuk manajemen keadaan, dan banyak lagi. Ini memudahkan pengembangan aplikasi yang kompleks dengan memanfaatkan komunitas yang aktif.

SPA (Single Page Application) merupakan aplikasi yang bekerja di dalam browser yang tidak membutuhkan reload page saat digunakan. Dengan kata lain, pengguna atau user tidak akan berpindah halaman dengan melakukan request kepada server setiap kali terjadi interaksi pada aplikasi. Yang membedakan *SPA* dengan non-*SPA* adalah single page application hanya akan melakukan load terhadap satu halaman dari server kemudian mekanisme routing yang biasanya dihandle oleh server kini dibebankan pada client. Akibatnya, website yang menggunakan *SPA* memiliki performa yang lebih cepat tanpa harus merequest kembali halaman secara terus menerus [13].

2.8. *Chart.JS*

ChartJS adalah salah satu *library* yang memanfaatkan element canvas untuk membuat grafik yang akan ditampilkan melalui *web browser*. *Library ChartJS* dapat dimanfaatkan untuk menampilkan informasi laporan dalam bentuk grafik sehingga lebih mudah dipahami oleh orang lain dari pada menampilkan data dalam bentuk tabel. Antarmuka SIMORI (Sistem Informasi Mobilitas Sivitas Akademika IPB dan Publikasi Ilmiah) [14].

Chart.js adalah *library* grafik berbasis *Javascript* yang gratis dan *open source* untuk visualisasi data yang mendukung 8 jenis tipe grafik, yaitu: bar, line, area, pie,

bubble, radar dan scatter. *Chart.js* dibuat oleh pengembang web berbasis London, Nick Downia dan sekarang telah dikembangkan lebih lanjut oleh komunitas Github [15].

2.9. *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 [16].

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.

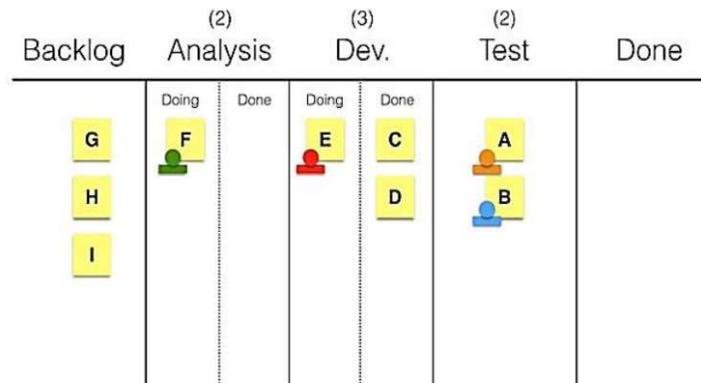
2.10. *MQTT.js*

MQTT.js adalah pustaka JavaScript yang mengimplementasikan protokol *MQTT (Message Queuing Telemetry Transport)* untuk komunikasi berbasis pesan antar perangkat. *MQTT.js* memungkinkan aplikasi atau perangkat yang ditulis dalam bahasa JavaScript untuk berkomunikasi dengan *broker MQTT* [17].

MQTT.js memungkinkan aplikasi JavaScript untuk berkomunikasi dengan perangkat lain melalui protokol *MQTT* yang efisien dan ringan. Hal ini berguna untuk berbagai skenario seperti *Internet of Things (IoT)*, sensor monitoring, komunikasi mesin ke mesin (M2M), dan banyak lagi.

2.11. Kanban

Kanban adalah sebuah teknik manajemen proyek yang menggabungkan *work in progress (WIP)*, memvisualisasikan alur kerja, mengukur dan mengelola alur, membuat kebijakan proses secara eksplisit, dan menggunakan model untuk mengenali peluang peningkatan. Metode Kanban dapat meningkatkan pemahaman, visibilitas, dan pengendalian alur kerja, serta mendukung manajemen melalui dua prinsip inti, yaitu membatasi *WIP* dan memvisualisasikan alur kerja dengan menggunakan papan Kanban. Selain itu, metode Kanban memiliki papan yang digunakan untuk memvisualisasikan alur kerja dan memantau kemajuan proyek. Selanjutnya, papan Kanban memvisualisasikan aktivitas proses pengembangan dan menjaga *WIP* tetap terkendali [18].



Gambar 4. Visualisasi papan kerja metode kanban

Dari gambar 4 tentang visualisasi papan kerja Kanban yang menjelaskan proses pengerjaan dimulai dari sebuah requirement gathering. Dimana ini akan mempertemukan pihak-pihak yang akan melakukan kesepakatan akan rencana kedepannya. Untuk hasil tersebut akan dilanjutkan pada bagian user story, pada proses ini akan menampilkan apa saja yang diinginkan oleh pemangku kepentingan dan ini akan menjadi tujuan dari pembuatan aplikasi ini. Untuk selanjutnya dilakukan proses perancangan awal yaitu membuat use case dan diagram activity. Lalu jika sudah melakukan proses perancangan maka dilanjutkan dengan melakukan pengerjaan. Pengerjaan dilakukan dengan meletakkan semua task yang sudah di drop down dan diletakkan pada papan, lalu task masuk pada wow dan akan masuk ke dalam wip. Jika di dalam proses wip sudah Selesai maka akan dilanjutkan

dengan pengujian. Jika pengujian sesuai maka akan masuk kedalam bagian done dan wip akan tersisih oleh task yang ada dibagian wow. Namun jika pada saat testing masih ada bug maka akan kembali lagi ke dalam proses wip. Dan menunggu antrian selanjutnya. Dan jika semua selesai maka akan dilakukan release.

Kanban membatasi pekerjaan yang sedang berjalan sesuai dengan kapasitas tim, yang menyeimbangkan permintaan terhadap keluaran pekerjaan yang disampaikan oleh tim. Ini membantu untuk memvisualisasikan masalah proses, mengurangi cacat dan mempertahankan aliran yang stabil. Manfaat utama menggunakan Kanban adalah pemahaman yang lebih baik tentang keseluruhan proses, peningkatan kualitas, dan cakupan kepuasan pelanggan yang lebih baik. Karena sifatnya yang terperinci dan prinsip visualisasi, ini membantu dalam memahami keseluruhan pekerjaan tim dan mengurangi ukuran batch.

Kanban memiliki beberapa kelebihan dibandingkan metode pengembangan proyek yang lain yaitu Fleksibilitas. Kanban dapat disesuaikan dengan kebutuhan dan dinamika proyek pengembangan perangkat lunak yang sering berubah-ubah. Tim pengembang dapat dengan mudah menyesuaikan kapasitas dan prioritas pekerjaan sesuai dengan perubahan yang terjadi.

Kanban memiliki tiga prinsip dasar dalam penggunaannya yaitu:

1. Visualisasikan pekerjaan.

Kanban Board adalah sebuah model visual yang digunakan dalam Kanban untuk merepresentasikan alur kerja secara grafis, sehingga memudahkan dalam memantau dan mengevaluasi jalannya proses dari awal hingga akhir. Model Kanban Board ini dibuat sesuai dengan tahapan yang harus dilalui dalam pengembangan perangkat lunak.

2. Batasi pekerjaan yang sedang berlangsung.

Pada tahap awal, tim akan menetapkan batasan pekerjaan untuk setiap alur pada Kanban Board yang disebut "*Work In Progress (WIP)*". Tujuan dari WIP ini adalah untuk mengurangi pemborosan dan membantu tim untuk fokus menyelesaikan pekerjaan yang sedang dalam proses, sebelum memulai pekerjaan baru setelahnya.

3. Fokus pada alur kerja

Manfaat yang efektif dari penggunaan Kanban adalah tim dapat berfokus pada aliran kerja proyek dari awal hingga selesai. Dalam mencapai fokus ini, tim harus mengikuti dua prinsip yang telah dijelaskan sebelumnya. Fokus pada aliran kerja proyek ini membantu tim untuk mengidentifikasi hambatan yang mungkin terjadi dan mengambil tindakan yang diperlukan untuk memastikan aliran kerja proyek tetap lancar dan terus berjalan.

Pemvisualisasian ini dilakukan menggunakan sebuah *kanban board* yang berisi proses-proses dalam pengembangan seperti perencanaan, perancangan, persetujuan, pengembangan, pengujian, pengintegrasian, dan penerapan. Membatasi pekerjaan atau *limit WIP* dilakukan dengan tujuan agar pekerjaan dilakukan lebih optimal. Pada dasarnya sebuah tim memiliki batasan maksimal pekerjaan yang dilakukan agar mencapai hasil yang optimal. Melakukan peningkatan aliran kerja dilakukan untuk mencapai estimasi waktu yang sesuai. Menemukan dan mengaplikasikan matriks pekerjaan yang sesuai merupakan langkah untuk melakukan pengestimasian pekerjaan dengan tepat [19].

2.12. *Axios*

Axios adalah salah satu sintaks yang sangat sederhana di *reactJS* untuk berinteraksi dengan *API* dan secara otomatis mengubah data *JSON*. Beranjak dari keunggulan dari *axios*, penelitian ini bermaksud untuk meningkatkan kinerja sistem pada saat melakukan permintaan dan respon data melalui *REST APIs*. *Axios* adalah salah satu sintaks yang sangat sederhana di *reactJS* untuk berinteraksi dengan *API* dan secara otomatis mengubah data *JSON*. library *axios* dapat digunakan untuk melakukan permintaan dengan metode *axios get* melalui *URLs* dengan kinerja yang baik dari *Fetch API* [20].

Library Axios bisa mengatasi berbagai masalah yang akan terjadi pada saat pengiriman data tanpa memerlukan perintah khusus untuk mengatasi pesan error tersebut. Mengingat banyak kelebihan yang ditawarkan *ReactJS*, maka penelitian

ini merasa perlu melakukan pendalaman terhadap cara kerja *React.JS* terutama penggunaan *Axios* dalam pengambilan dan menampilkan data melalui *REST API*.

2.13. *TailwindCSS*

Tailwind adalah sekumpulan kelas *utilitas* dan *preprocessor CSS* yang menghasilkan kelas-kelas CSS tersebut dan memungkinkan penggunaan alat CSS tambahan seperti *@apply* pengarah. Kita perlu menginstal framework itu sendiri dan juga menambalnya ke dalam kita Rantai alat pemrosesan CSS [21].

Visual Studio Code merupakan sebuah text editor yang dibuat oleh Microsoft. Visual Studio code ini mendukung banyak bahasa pemrograman seperti bahasa javascript, c, c#, c++, python, dan sebagainya. Dengan plugin yang di akses di dalam Marketplace Visual Studio Code.

2.14. *Visual Studio Code*

Visual Studio Code adalah *editor source code* yang dikembangkan oleh *Microsoft* untuk *Windows*, *Linux* dan *MacOS*. Ini termasuk dukungan untuk *debugging*, *GIT Control* Yang disematkan, penyorotan sintaks, penyelesaian kode cerdas, cuplikan, dan kode *refactoring*. Hal ini juga dapat disesuaikan, sehingga pengguna dapat mengubah tema editor, *shortcut keyboard*, dan preferensi. *Visual Studio Code* gratis dan *open-source*, meskipun unduhan resmi berada di bawah lisensi proprietary. *Visual Studio Code* memungkinkan pengguna untuk menulis kode pada berbagai bahasa pemrograman, termasuk *HTML*, *CSS*, *JavaScript*, *PHP*, *Python*, dan lain-lain. *Visual Studio Code* berjalan pada sistem operasi *Windows*, *Mac*, dan *Linux*. Editor teks ini dilengkapi dengan fitur-fitur seperti fitur *debugging*, *Git integration*, *autocomplete*, *snippet*, *syntax highlighting*, dan lain-lain [22].

Kode *Visual Studio* didasarkan pada *Elektron*, kerangka kerja yang digunakan untuk menyebarkan aplikasi *Node.js* untuk desktop yang berjalan pada *Blinklayout*. Meskipun menggunakan kerangka *Elektron*, *Visual Studio Code* tidak

menggunakan Atom dan menggunakan komponen editor yang sama (diberi kode nama "Monaco") yang digunakan dalam *Visual Studio Team Services* yang sebelumnya disebut *Visual Studio Online* (Lardinois, 2015).

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

- Gratis dan *open-source*
- Ringan dan cepat
- Mudah digunakan
- Mendukung banyak bahasa pemrograman
- Memiliki banyak ekstensi dan plugin

2.15. *Trello*

Aplikasi *Trello* merupakan sebuah aplikasi kolaborasi yang memungkinkan seseorang atau sekelompok orang mengatur berbagai macam tugas atau proyek dalam satu tempat yang berbasis *cloudbase tool* sebagai penerapan manajemen proyek dengan metode Kanban dengan konsep menampilkan seluruh aktivitas tugas dalam satu bentangan dan dapat dilihat oleh setiap anggota dari tim, serta dapat mengatur tugas menjadi *boards* dan membagi group menjadi task [23].

Trello memiliki beberapa kelebihan yang membedakannya dari aplikasi manajemen proyek dan kolaborasi lainnya yaitu:

1. Antarmuka pengguna yang intuitif:

Trello memiliki antarmuka pengguna yang sangat mudah digunakan dan intuitif. Pengguna dapat membuat dan mengatur papan tugas, daftar, dan kartu dengan mudah, bahkan bagi mereka yang tidak memiliki pengalaman dengan aplikasi manajemen proyek sebelumnya.

2. Fleksibilitas dan kustomisasi:

Trello memberikan fleksibilitas dan kustomisasi yang tinggi untuk pengguna.

Pengguna dapat menyesuaikan papan tugas dan kartu sesuai dengan kebutuhan mereka, menambahkan label, *member*, daftar, dan catatan pada kartu, serta mengatur *deadline* dan pengingat untuk tugas tertentu.

3. Integrasi dengan aplikasi lain:

Trello memiliki integrasi dengan banyak aplikasi lain seperti Google Drive, Slack, Dropbox, dan Evernote. Integrasi ini memungkinkan pengguna untuk menghubungkan tugas dan informasi dari berbagai platform dalam satu tempat yang mudah diakses.

4. Mudah diakses dari berbagai perangkat:

Trello dapat diakses dari berbagai perangkat, termasuk desktop, tablet, dan smartphone. Aplikasi Trello mobile yang tersedia untuk Android dan iOS juga memungkinkan pengguna untuk mengakses papan tugas mereka dari mana saja dan kapan saja.

5. *Trello Business Class*

Versi berbayar dari *Trello* yang dirancang khusus untuk tim dan organisasi. Versi ini memiliki fitur tambahan seperti kontrol akses, integrasi dengan aplikasi lain, dan dukungan pelanggan 24/7.

Kelebihan-kelebihan tersebut membuat Trello menjadi pilihan yang tepat bagi tim atau organisasi yang ingin meningkatkan produktivitas dan kolaborasi dalam pekerjaan. Trello sangat mudah digunakan, fleksibel, dan dapat diakses dari berbagai perangkat, serta memiliki berbagai fitur yang dapat membantu pengguna mengelola proyek dan tugas dengan lebih efektif [23].

2.16. *Webhoster*

Hosting merupakan sebuah space atau tempat penyimpanan data sebuah *blog* atau *website* di internet yang memungkinkan data tersebut diakses secara online dan dilihat oleh semua orang dari seluruh penjuru dunia.

Webhoster menyediakan sebuah layanan hosting yang *unlimited* dimana pengguna bisa dengan bebas mengupload konten website baik itu berupa gambar, video maupun jenis file lain tanpa harus takut kehabisan ruang atau space tersebut. Dengan memanfaatkan fitur *Cpanel Hosting* tampilan website bisa di upload dan diakses oleh banyak orang didunia. Dalam *cpanel* bisa membuat folder baru dengan nama website dan domain yang akan digunakan. Dan dengan mengupload build dari project yang sudah dibuat maka *website* akan otomatis akan dipublikasi oleh *webhoster*.

Web hosting adalah layanan jasa atau penyewaan tempat untuk menyimpan file atau bentuk script yang berada Internet dan memungkinkan untuk perorangan atau pun organisasi guna menampilkan layanan jasa atau produk di web atau bahkan situs Internet seperti web portal, web pribadi dan banyak lagi.

2.17. *Black-box Testing*



Gambar 5. Black-box Testing

Black Box testing ini merupakan metode untuk melakukan pengujian fungsional pada sebuah aplikasi. Pengujian ini berfokus pada pengujian atas kelengkapan menu, modul, dan proses operasionalnya. Pada prinsipnya metode *black box* atau biasa disebut dengan pengujian fungsional adalah suatu cara untuk menguji perangkat lunak tanpa mengetahui struktur atau isi kode pemrograman [24].

III. METODOLOGI PENELITIAN

3.1. Waktu dan Tempat Penelitian

Penelitian dan pembuatan Tugas Akhir ini dilakukan pada:

Waktu : Desember 2022 – juli 2023

Tempat :

- Laboratorium Teknik Komputer, Teknik Elektro, Universitas Lampung
- Pantai Canti, Kalianda, Lampung Selatan
- Hutan Mangrove Petengoran, Desa Gebang, Kabupaten Pesawaran
- Pulau Panjang Krakatau, Kabupaten Lampung Selatan, Provinsi Lampung

3.2. Jadwal Penelitian

Jadwal pada penelitian ini adalah sebagai berikut:

Tabel 1. Jadwal Penelitian

No	Aktivitas	Des 2022	Jan 2023	Feb 2023	Mar 2023	Apr 2023	Mei 2023	Jun 2023	Juli 2023
1	Penentuan Kebutuhan	■	■						
2	Penentuan <i>User story</i>	■	■						
3	Perancangan			■					
4	<i>Work On Wait (WOW)</i>			■	■				
5	<i>Work in Progress (WIP)</i>			■	■	■			
6	Pengujian					■	■		
7	<i>Done</i>					■	■		
8	Pelaporan						■	■	■

3.3. Alat dan Bahan

3.3.1. Alat Penelitian

Tabel 2. Alat yang digunakan dalam penelitian

No	Nama Alat	Spesifikasi	Deskripsi
1	Laptop	<i>Processor Intel Core i7-10750U, RAM 16GB, SSD 512GB, Sistem Operasi windows 10</i>	Perangkat keras yang digunakan sebagai <i>compiler</i> dalam pemrograman
2	<i>Trello</i>	<i>Online Trello</i>	<i>Tools</i> yang digunakan sebagai <i>kanban board</i> untuk memvisualkan alur kerja
3	<i>Visual Studio Code</i>	Versi 1.80.1	Kode editor untuk membuat aplikasi <i>website</i> dan lain-lain
4	<i>Niaga Hoster</i>	<i>Website Hosting</i>	Tools untuk melakukan hosting <i>Website</i>
5	<i>Figma</i>	Versi 28.4.12.3	Perangkat lunak untuk melakukan perancangan antarmuka
6	<i>Node.js dan yarn</i>	Node.js versi v18.12.1 yarn 1.22.19	Aplikasi yang digunakan untuk menjalankan kode program dan mengelola dependensi proyek.
7	MQTTX	MQTTX Versi 1.91	Software yang digunakan untuk pengujian broker mqtt dalam menerima data

3.3.2. Bahan Penelitian

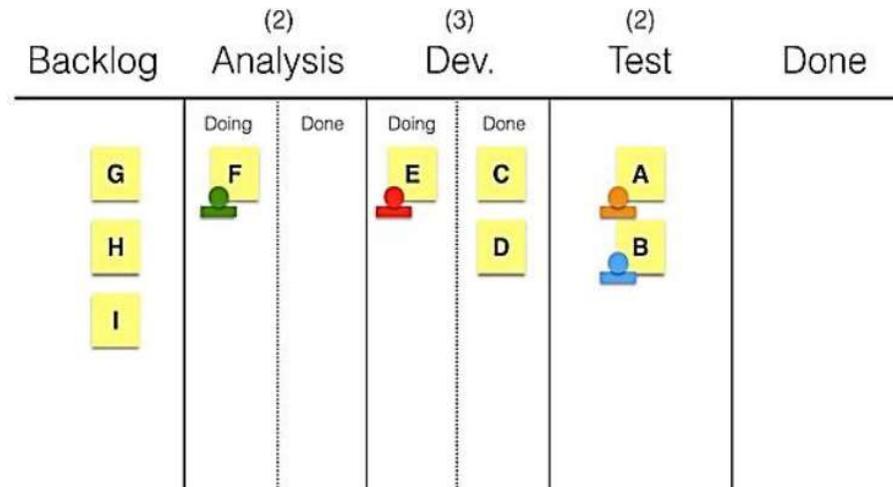
Bahan yang digunakan dalam pengerjaan penelitian dan pengembangan sistem ini berupa data yang didapatkan dari backend yang mengirimkan data yang berasal dari REST API dan data MQTT:

Tabel 3. Bahan berupa data sensor dari raspberry pi yang dikirimkan oleh server menggunakan *MQTT* dan *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>Forecast300</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> .

3.4. Tahapan Penelitian

Tahapan yang dilakukan dalam penelitian ini mengikuti model pengembangan perangkat lunak Kanban. Berikut merupakan gambar dari tahapan pengembangan menggunakan metode Kanban:



Gambar 6. Tahapan pada metode Kanban

Gambar 6. menggambarkan tahap-tahap dalam metode *Kanban*, dimulai dengan adanya pengumpulan informasi sebelum *requirement gathering*. Lalu dilanjutkan dengan tahap pengembangan sistem yang dimulai dengan menentukan *user story*, latar belakang masalah, dan motivasi dalam penelitian untuk menentukan aktivitas dan fitur pada aplikasi. Setelah menentukan *user story*, mereka akan dibagi menjadi beberapa *backlog* yang akan digunakan sebagai *task card* yang akan dikerjakan dan diletakkan pada *Work On Wait (WOW)*. Ketika *WOW* selesai, *developer* akan memulai kerja dengan menarik *task card* dari *WOW* ke *Work in Progress (WIP)*. Setelah selesai, *task card* akan dilakukan *testing* menggunakan metode *blackbox testing*. Jika berhasil, *task card* akan dipindahkan ke bagian *Done*, namun jika gagal, akan dipindahkan kembali ke *WIP*, asalkan jumlah *task* pada *WIP* belum mencapai batas maksimal yaitu 2 *task card*. Penjabaran dari masing-masing langkah dalam metode *Kanban* adalah sebagai berikut:

3.4.1. Target Pengguna

Tahap ini dilakukan untuk mendefinisikan mengenai pengguna yang menggunakan aplikasi serta tujuan pengembangan aplikasi. Studi kasus penelitian ini dilakukan untuk pengembangan website informasi data *Early Warning System* yang menggunakan protokol *mqtt* dan *https*. Setelah mengetahui atas kebutuhan dilanjutkan dengan hal yang berhubungan dengan aktivitas yang akan dilakukan dalam pengembangan sistem. Target pengguna website ini adalah masyarakat sekitar sebagai media pemantau tsunami dan instansi yang bisa digunakan untuk melakukan penelitian lain.

3.4.2. Tahap Penentuan *User story*

Tahap ini dilakukan untuk menghasilkan sebuah daftar *user story* untuk dijadikan pedoman atas penentuan *backlog* yang digunakan dalam pengembangan aplikasi. Hal tersebut dikarenakan *backlog* merupakan hasil turunan atau breakdown yang didapatkan dari *user story* itu sendiri. Format penggunaan *user story* yang digunakan dalam penelitian ini adalah sebagai berikut:

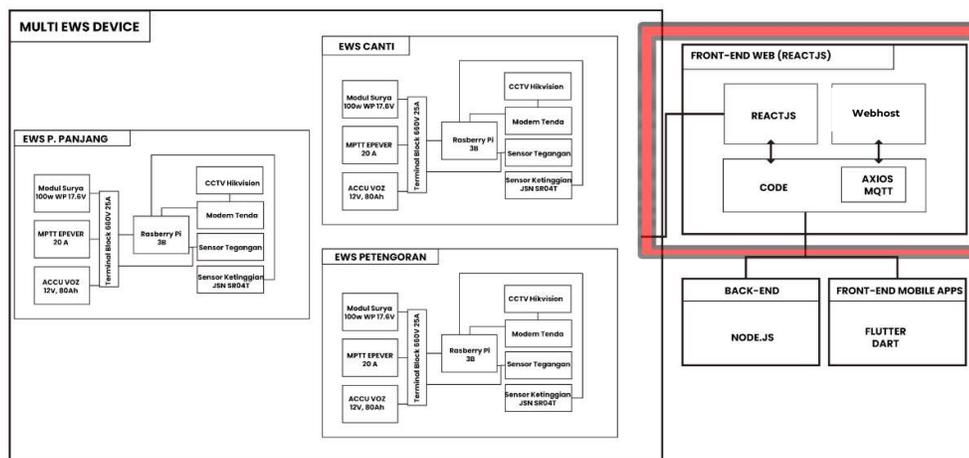
Saya [pengguna] ingin melakukan [kegiatan] agar [hasil yang diharapkan]

Gambar 7. Format *user story* yang digunakan dalam penelitian

Gambar 7. merupakan format dari *user story* yang digunakan dalam penelitian. Pengguna yang diisikan merupakan target pengguna yang telah ditentukan dalam penelitian. Kegiatan adalah hal yang dilakukan oleh pengguna agar dapat mencapai hasil yang diharapkan melalui penggunaan aplikasi yang dikembangkan.

Penentuan *environment*

Struktur *system project* ini ditentukan berdasarkan *Library* ataupun aplikasi yang digunakan sesuai dengan kebutuhan pengembangan project aplikasi. Beberapa *Library* dipilih sebagai program untuk membuat aplikasi ini, ada pula resource lain yang digunakan dalam pengembangan project. Berikut *environment* dalam mengembangkan project aplikasi ini.



Gambar 8. *Environment* keseluruhan sistem

Gambar 8. merupakan *environment* sistem untuk mengembangkan project aplikasi. Struktur tersebut sudah sesuai dibuat berdasarkan sistem yang dipilih untuk mengembangkan aplikasi ini.

Dalam project yang akan dikembangkan tersebut terdapat beberapa bagian yang akan dikerjakan secara terpisah. Beberapa bagian tersebut adalah sebagai berikut :

Tabel 4. Tim pengembangan project

No	Nama	Bagian
1	Muhammad Abdurachman Hilmy	Hardware EWS
2	M. Nur Hasanuddin	Back-end Developer
3	Ramadhan Andhika Vioamanta	Front-end Developer Mobile Apps
4	Ranto	Front-end Developer Web

Pada pengembangan ini pengembang fokus pada *frontend website* yang menggunakan *reactJS* yang sebagai *Library* utama pembuatan *website*.

Penentuan *backlog*

Semua *user story* yang telah didefinisikan sebelumnya lalu dilakukan *breakdown* yang kemudian menjadi sebuah *backlog*. *Backlog* yang nantinya *backlog*. *Backlog* yang didapat kemudian diletakan sebagai *task card* pada *kanban board* menggunakan *tools Trello*. *Backlog* atau *task card* inilah yang kemudian digunakan dalam proses pengembangan.



Gambar 9. *Task card*

Gambar 9. merupakan *breakdown* dari *user story* yang telah ditentukan sebelumnya hingga menjadi beberapa *backlog*. Kemudian dari *backlog* yang didapat diletakkan sebagai *task card* pada *kanban board* menggunakan *tools Trello*.

3.4.3. Tahap Analisis

Task card

Backlog yang telah didapatkan kemudian dituliskan pada sebuah *task card* yang ada pada *trello*. Format pada *task card* yang digunakan adalah sebagai berikut:

a. Label

Label merupakan tanda yang digunakan dalam setiap *task card*. Label yang digunakan dalam pengembangan aplikasi ini adalah label status dari *task card* tersebut yang menunjukkan seberapa penting *task card* tersebut harus segera selesai.



Gambar 10. Label yang digunakan dalam *kanban board*

Gambar 10. menunjukkan label yang digunakan dalam *task card* di *kanban board* pada proses pengembangan aplikasi. Terdapat 5 buah label yang masing-masing memiliki keterangan sebagai berikut:

1. Label *slow* berwarna hijau gelap diberikan untuk menandakan bahwa *task card* memiliki *deadline* berdasarkan requirement yang dibutuhkan dan dengan rentang waktu lebih dari 5 hari.
2. Label *medium* berwarna kuning diberikan pada *task card* dengan berdasarkan requirement yang dibutuhkan dan rentang waktu antara 2 sampai dengan 5 hari.
3. Label *urgent* berwarna merah menunjukkan bahwa *task card* tersebut berdasarkan requirement yang dibutuhkan dan memiliki *deadline* kurang dari 24 jam.
4. Label *verify* berwarna biru cerah menunjukkan bahwa *task card* tersebut perlu diverifikasi apakah telah memenuhi sesuai kebutuhan atau belum.
5. Label *done* berwarna ungu diberikan pada *task card* yang telah selesai dikerjakan.
6. Label *Late* berwarna merah muda menunjukkan bahwa *taskbar* tersebut mengalami keterlambatan atau kendala pengerjaan.

b. Deskripsi

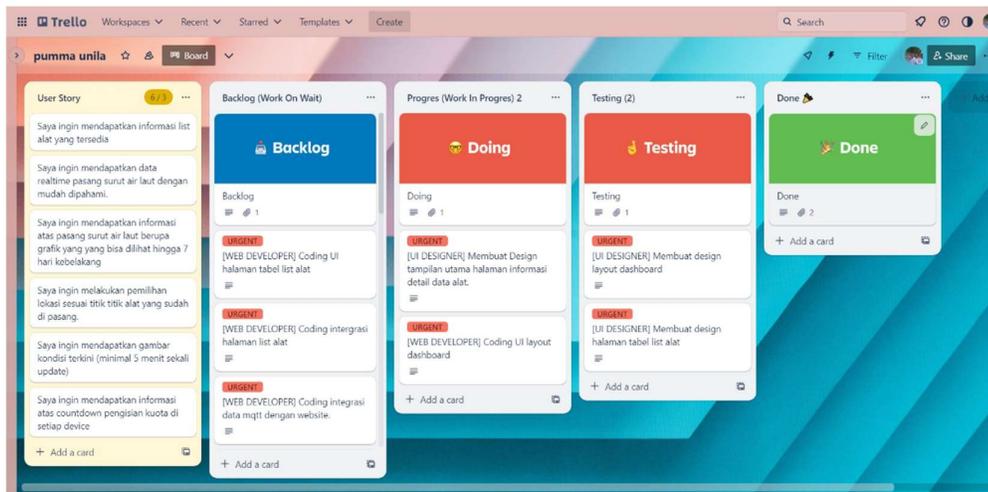
Deskripsi *task card* yaitu keterangan dari tugas/*backlog* yang harus diselesaikan serta *role* (*developer, ui designer, tester*) yang harus mengerjakannya.

c. *Deadline*

Tanggal yang tampil pada *task card* merupakan batas waktu terakhir atau *deadline* dari *task card* tersebut. Setiap *task card* yang ada diberikan batas waktu atau *deadline*.

Kanban board

Jumlah dari *backlog* dari *breakdown user story* berjumlah 13 *Backlog*. *Backlog* inilah digunakan menjadi *task card* atau tugas-tugas yang disusun pada *kanban board* seperti pada gambar 4.36 dan harus diselesaikan serta menjadi acuan dalam proses pengembangan aplikasi.



Gambar 11. *Kanban board* pada penelitian yang dilakukan

Gambar 11. menampilkan *kanban board* yang digunakan dalam penelitian. *Kanban board* pengembangan memiliki kolom yaitu sebagai berikut:

a. *User stories*

Kolom *user stories* berisi daftar dari *user story* yang telah ditentukan.

b. *Backlog*

Backlog atau WOW yang berisi *task card-task card* yang perlu dikerjakan pada pengembangan aplikasi.

c. *Progress*

Kolom *progress* atau *WIP* berisi *task card* yang sedang dikerjakan. Pengembangan aplikasi ini membatasi jumlah *task card* yang ada pada kolom *progress* sebanyak 2 buah *backlog/task card*.

d. *Testing*

Kolom *testing* berisi *task card* yang telah selesai dikerjakan namun belum dilakukan pengujian.

e. *Done*

Kolom *done* berisi *task card* yang telah selesai dikerjakan dan telah dinyatakan selesai dalam pengujian.

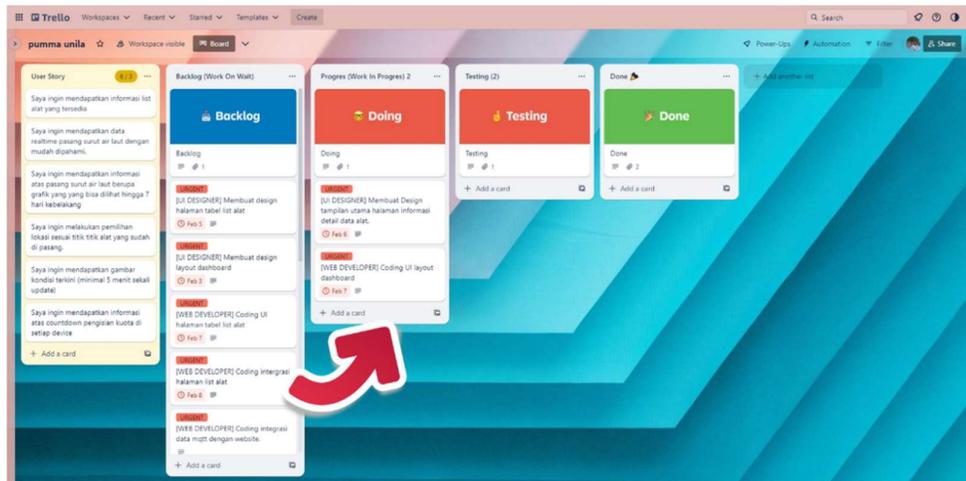
Limit WIP

Penelitian ini membatasi jumlah *Work in Progress (WIP)* sebanyak 2 buah *backlog*. Hal ini dilakukan atas prinsip Kanban dengan membatasi pekerjaan pada alur kerja. *WIP* tidak memiliki ketentuan namun pada penelitian ini ditentukan berdasarkan saran Henrik Kniberg & Mattias Skarin pada bukunya yang berjudul *Kanban and Scrum making the most of both* yang menyatakan bahwa inisiasi *WIP* yang ideal adalah bahwa setiap anggota dapat memegang setidaknya 2 buah *task card* atau *backlog*. Selain hal tersebut, *WIP* dilakukan untuk pertimbangan atas kemampuan dari pengembang dalam menyelesaikan suatu tugas / *taskbar* dalam satu alur kerja.

3.4.4. Tahap Pengembangan Sistem

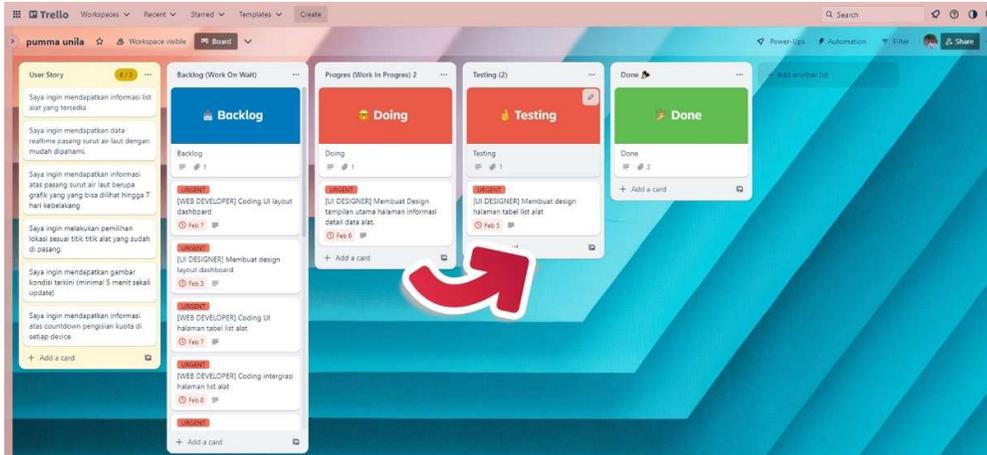
Visualisasi Alur Kerja

Pengembangan sistem dilakukan pada Kanban board dengan taskcard sebagai backlog sebagai tugas yang harus dituntaskan. Visualisasi dari pengerjaan yang dilakukan pada metode kanban adalah sebagai berikut:



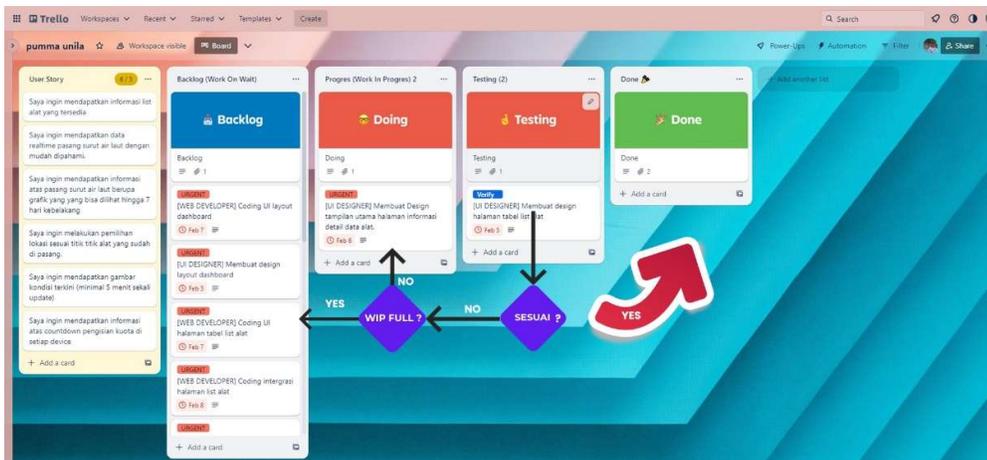
Gambar 12. Kanban board pada penelitian yang dilakukan

Gambar 12. menampilkan langkah awal yang dilakukan dalam tahap pengembangan. Pengembangan dimulai dengan menarik *task card* yang berada di *backlog (WOW)* kedalam kolom *progress (WIP)*. Jumlah *task card* yang dapat ditempatkan pada kolom *progress* berjumlah 2 buah dan penarikan atau pemindahan *task card* dilakukan berdasarkan prioritas yang ditunjukkan oleh label pada *task card* tersebut.



Gambar 13. Kanban board pada penelitian yang dilakukan

Gambar 13. menggambarkan proses task card yang berada pada WIP selesai dikembangkan akan dipindahkan di kolom testing. Pemindahan ke kolom *testing* bermanfaat untuk menandakan bahwa pekerjaan atau *task card* tersebut memerlukan pengujian.



Gambar 14. Kanban board pada penelitian yang dilakukan

Gambar 14. menjelaskan kondisi apabila proses testing berhasil maka diberikan label verify dan di cek apakah sesuai kebutuhan. Jika telah sesuai maka dipindahkan ke kolom *Done* namun jika belum memenuhi kebutuhan maka harus diubah dan dikerjakan kembali sehingga dipindahkan kembali ke kolom *progress (WIP)* dengan catatan *task card* pada kolom *progress (WIP)* belum mencapai batas maksimal yang ditentukan dalam penelitian yaitu 2 buah *task card*. Jika ternyata

kolom *progress (WIP)* telah terisi penuh maka dipindahkan kedalam kolom *backlog (WOW)* kembali.

Tahapan ini dilakukan penulisan kode dari sistem yang telah dirancang. Perancangan sistem yang telah dilakukan harus diimplementasikan dalam bentuk kode. Hasil dari proses ini berupa website yang siap untuk dilakukan pengujian atau testing menggunakan metode *blackbox testing*. Sesuai dengan metode kanban, maka dalam proses pengembangan menggunakan prinsip-prinsip diantaranya adalah sebagai berikut:

1. Visualisasi Alur Kerja

Proses ini adalah alur pengembangan pada sebuah *Kanban Board*. Dalam penelitian ini *Kanban Board* digambarkan menggunakan tools *Trello*. Proses visualisasi alur merupakan penggambaran alur pengembangan pada sebuah *kanban board*. Dalam penelitian ini sendiri *kanban board* digambarkan menggunakan tools bernama *Trello*.

2. Membatasi *Work in Progress* (pekerjaan yang sedang berjalan)

Dalam proses pengembangan aplikasi ini jumlah *Work in Progress* pada *kanban board* maksimal 2 *task card*.

3. Fokus pada *WIP*

Fokus pada *WIP* disini merupakan sebuah prinsip dari metode Kanban dimana tidak boleh terdapat suatu pekerjaan baru yang dilakukan sebelum pekerjaan yang lama terselesaikan.

3.5. Tahap Testing

Tahapan testing atau pengujian dilakukan menggunakan metode Blackbox testing. Pengujian dilakukan untuk menemukan kesalahan-kesalahan yang mungkin terjadi untuk dilakukan perbaikan. Pengujian dilakukan secara langsung pada *website pumma.isi-net.org*. Teknik pengujian ini menggunakan panduan pada *decision table testing*. *Blackbox testing* atau pengujian secara langsung pada website ini dilakukan dengan cara mengecek aplikasi terhadap use case yang telah dirancang. Hasil akhir dari pengujian adalah aplikasi telah bebas dari kesalahan atau bug

sehingga diharapkan saat sampai ke tangan pengguna website sudah siap digunakan.

3.6. Tahap Pelaporan

Tahap akhir dari penelitian ini adalah pelaporan hasil dan temuan penelitian mengenai *Website Sistem Informasi Pemantauan Dan Peringatan Dini Bencana Tsunami Berbasis IoT Dengan Menggunakan Library Reactjs*. Dari data yang dihasilkan dan telah dianalisis kemudian dilakukan pengambilan kesimpulan dan saran. Hasil temuan yang ada kemudian digunakan sebagai skripsi pada Universitas Lampung.

V. KESIMPULAN

5.1. Kesimpulan

Kesimpulan yang diperoleh berdasarkan hasil dari penelitian adalah sebagai berikut:

1. Berhasil dikembangkan sebuah website Early Warning System dengan menggunakan *Library React JS* yang mempunyai fitur monitoring status dan list alat, detail informasi alat, dan countdown pengisian kuota internet pada alat.
2. Berdasarkan data yang diterima MQTT lebih unggul dalam kecepatan pengiriman data dibandingkan https yang mempunyai range kecepatan rata-rata 199 ms.
3. Berdasarkan penelitian data grafik tidak dikirimkan secara utuh namun dipangkas dengan range yang ditentukan untuk mengurangi keterlambatan data yang dikirim namun tidak mengurangi informasi yang disampaikan.
4. Proses pengembangan aplikasi menggunakan 6 *user story* yang kemudian dilakukan *breakdown* hingga menghasilkan 17 *backlog*. *Backlog* ini kemudian disusun dalam *task card* serta diletakkan pada *kanban board* dan menjadi acuan dalam proses pengembangan hingga proses pengujian aplikasi.
5. Hasil pengujian menggunakan metode *blackbox testing* dengan 8 skenario didapatkan bahwa hasil pengujian yang dihasilkan telah sesuai dengan hasil yang diharapkan.

5.2. **Saran**

Saran yang dapat diberikan untuk melanjutkan penelitian ini berdasarkan hasil penelitian yang telah dilakukan adalah sebagai berikut:

1. Melakukan pengembangan dengan fitur download data pada grafik agar bisa dikembangkan oleh pengunjung website.
2. Melakukan pengembangan dengan desain yang lebih menarik dan lebih modern supaya mengikuti gaya desain terbaru.
3. Melakukan pengembangan autentikasi untuk pengunjung dengan pembatasan fitur-fitur yang sifatnya rahasia.
4. Melakukan pengembangan jumlah alat dan menambahkan alat pada menu klimatologi.

DAFTAR PUSTAKA

- [1] C. N. Mazda and D. Pelansiani, "Reconstruction Resilience Policy Of Sunda Strait Tsunami," pp. 386–392, 2019.
- [2] Badan Nasional Penanggulangan Bencana, "Data Kejadian Bencana Gelombang Pasang atau Abrasi Tahun 2021," *Https://Gis.Bnpb.Go.Id/a9a84363-D3E8-4a4B-B4B5-24D61768De69*, 2022, [Online]. Available: <https://gis.bnpb.go.id/arcgis/apps/sites/?fromEdit=true#/public/pages/data-bencana>
- [3] G. Agency, "The December 22 , 2018 Sunda Strait tsunami," pp. 2018–2020, 2018.
- [4] N. Tantitharanukul, K. Osathanunkul, K. Hantrakul, P. Pramokchon, and P. Khoenkaw, "MQTT-Topics Management System for sharing of Open Data," *2nd Jt. Int. Conf. Digit. Arts, Media Technol. 2017 Digit. Econ. Sustain. Growth, ICDAMT 2017*, pp. 62–65, 2017, doi: 10.1109/ICDAMT.2017.7904935.
- [5] S. Chanthakit and C. Rattanapoka, "Mqtt based air quality monitoring system using node MCU and node-red," *Proceeding 2018 7th ICT Int. Student Proj. Conf. ICT-ISPC 2018*, pp. 3–7, 2018, doi: 10.1109/ICT-ISPC.2018.8523891.
- [6] R. Bandal and A. Oak, "Managing Location Identification and Chain SMS for Smart School Transport System using IoT," *2019 Int. Conf. Comput. Commun. Informatics, ICCCI 2019*, pp. 1–6, 2019, doi: 10.1109/ICCCI.2019.8821865.
- [7] A. Wibowo and M. Mahrus Zain, "Pemanfaatan ReactJS dan Protokol MQTT untuk Visualisasi Sinyal Lampu dan Notifikasi secara Waktu Nyata pada Sistem Pemonitor APILL di Kota Pekanbaru," *J. Komput. Terap.*, vol. 7, no. Vol. 7 No. 2 (2021), pp. 314–328, 2021, doi: 10.35143/jkt.v7i2.5108.
- [8] Y. Trimarsiah and M. Arafat, "Analisis dan Perancangan Website sebagai Sarana Informasi pada Lembaga Bahasa Kewirausahaan dan Komputer Akmi Baturaja," *Jurnal Ilmiah MATRIK*, vol. 19, no. 1. pp. 1–10, 2017.

- [9] Yuki Firmanto, "EARLY WARNING SYSTEM: SOLUSI KLAIM NEGATIF RUMAH SAKIT PROGRAM JAMINAN KESEHATAN NASIONAL," vol. 20, no. 1, pp. 12–32, 2021.
- [10] T. J. Charity, "Smart World of Internet of Things (IoT) and It ' s Security Concerns," pp. 240–245, 2016, doi: 10.1109/iThings-GreenCom-CPSCCom-SmartData.2016.64.
- [11] MQTT.org, "MQTT (The Standard for IoT Messaging)", [Online]. Available: <https://mqtt.org/>
- [12] A. Javeed, "Performance Optimization Techniques for ReactJS," *Proc. 2019 3rd IEEE Int. Conf. Electr. Comput. Commun. Technol. ICECCT 2019*, pp. 1–5, 2019, doi: 10.1109/ICECCT.2019.8869134.
- [13] B. H. Hayadi, "Bab 2 Landasan Teori," *Apl. dan Anal. Lit. Fasilkom UI*, vol. m, no. 1998, pp. 7–34, 2018, [Online]. Available: <http://elib.unikom.ac.id/files/disk1/655/jbptunikompp-gdl-supriadini-32740-6-12.unik-i.pdf>
- [14] Y. D. Indriani, K. B. Seminar, and H. Sukoco, "Sistem Pendukung Informasi Eksekutif Mobilitas Sivitas Akademika Dan Publikasi Ilmiah Institut Pertanian Bogor," *Jurnal Pustakawan Indonesia*, vol. 16, no. 2. pp. 1–9, 2017. [Online]. Available: <https://jurnal.ipb.ac.id/index.php/jpi/article/view/25677>
- [15] J. R. Fiqrussalam and N. H. Farmansyah, *Pembuatan Dashboard KONTRAK KINERJA UP3 Pembuatan Dashboard KONTRAK KINERJA UP3*. 2020.
- [16] 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. ICOI ACT 2019*, pp. 206–211, 2019, doi: 10.1109/ICOI ACT46704.2019.8938570.
- [17] npmjs, "library for the MQTT protocol", [Online]. Available: <https://www.npmjs.com/package/mqtt>
- [18] H. Alaidaros, M. Omar, and R. Romli, "Identifikasi kriteria yang mempengaruhi tugas pemantauan proyek perangkat lunak metode Agile Kanban." AIP, Sintok, Kedah, Malaysia.
- [19] N. Faizah, N. Santoso, and A. A. Soebroto, "Pengembangan Sistem

Aplikasi Manajemen Proyek menggunakan Kanban Framework.” <http://j-ptiik.ub.ac.id>, Malang,.

- [20] S. Hartati, A. B. Mutiara, and H. As'ari, “Fetching dan Display Data API Menggunakan Library AXIOS Pada ReactJS,” pp. 1–23, 2016.
- [21] N. Rappin, “Modern CSS with Tailwind: Flexible Styling without the Fuss,” *Modern CSS with Tailwind*. pp. 1–90, 2021.
- [22] Priyono, “TINJAUAN PUSTAKA DAN LANDASAN TEORI (Skripsi Universitas Mercubuana Yogya),” *TINJAUAN PUSTAKA DAN LANDASAN TEORI (Skripsi Universitas Mercubuana Yogya)*, vol. 9, no. 1. pp. 76–99, 2010.
- [23] F. Amsury, I. Kurniawati, Heriyanto, and M. Rizki Fahdia, “Pelatihan Pemanfaatan Aplikasi Trello Untuk Meningkatkan Efektifitas Manajemen Proyek Pada Karyawan PT Jaya Persada Indonesia,” *Abdi Teknayasa*. 2022. doi: 10.23917/abditeknayasa.v3i2.1249.
- [24] E. Sonalitha, B. Nurdewanto, A. Zubair, S. R. Asriningtias, K. Yudhistiro, and I. Mujahidin, “Blackbox Testing Model Boundary Value of Mapping Taxonomy Applications and Data Analysis of Art and Artworks,” in *2020 3rd International Seminar on Research of Information Technology and Intelligent Systems, ISRITI 2020*, Institute of Electrical and Electronics Engineers Inc., Dec. 2020, pp. 7–11. doi: 10.1109/ISRITI51436.2020.9315406.