



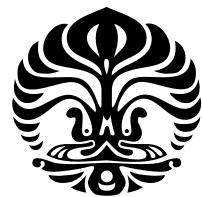
UNIVERSITAS INDONESIA

**PENGEMBANGAN SISTEM PENDETEKSI GELOMBANG AWAL,
MAGNITUDE, KEDALAMAN, DAN LOKASI SUMBER GEMPA BUMI
SECARA *REAL-TIME* MENGGUNAKAN ALGORITMA STA/LTA DAN
*MULTI-TARGET REGRESSION***

SKRIPSI

**AAN NUR WAHIDI - 1806141100
FARREL ALFARABI SALEH - 1806186622
MUHAMMAD DAHLAN YASADIPURA - 1806205193**

**FAKULTAS ILMU KOMPUTER
PROGRAM STUDI ILMU KOMPUTER
DEPOK
2022**



UNIVERSITAS INDONESIA

**PENGEMBANGAN SISTEM PENDETEKSI GELOMBANG AWAL,
MAGNITUDE, KEDALAMAN, DAN LOKASI SUMBER GEMPA BUMI
SECARA *REAL-TIME* MENGGUNAKAN ALGORITMA STA/LTA DAN
*MULTI-TARGET REGRESSION***

SKRIPSI

**Diajukan sebagai salah satu syarat untuk memperoleh gelar
Sarjana Ilmu Komputer**

**AAN NUR WAHIDI - 1806141100
FARREL ALFARABI SALEH - 1806186622
MUHAMMAD DAHLAN YASADIPURA - 1806205193**

**FAKULTAS ILMU KOMPUTER
PROGRAM STUDI ILMU KOMPUTER
DEPOK
DESEMBER 2022**

HALAMAN PERSETUJUAN

- Judul** : Pengembangan Sistem Pendeteksi Gelombang Awal, Magnitude, Kedalaman, dan Lokasi Sumber Gempa Bumi Secara *Real-time* menggunakan Algoritma STA/LTA dan *Multi-target Regression*
- Nama Penulis 1** : Aan Nur Wahidi
- NPM Penulis 1** : 1806141100
- Nama Penulis 2** : Farrel Alfarabi Saleh
- NPM Penulis 2** : 1806186622
- Nama Penulis 3** : Muhammad Dahlan Yasadipura
- NPM Penulis 3** : 1806205193

Laporan Skripsi ini telah diperiksa dan disetujui.

12 Desember 2022

Ari Wibisono, S.Kom., M.Kom Pembimbing 1 Skripsi

HALAMAN PERNYATAAN ORISINALITAS

**Skripsi ini adalah hasil karya kami sendiri,
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar.**

Nama Penulis 1	:	Aan Nur Wahidi
NPM Penulis 1	:	1806141100
Nama Penulis 2	:	Farrel Alfarabi Saleh
NPM Penulis 2	:	1806186622
Nama Penulis 3	:	Muhammad Dahlan Yasadipura
NPM Penulis 3	:	1806205193

Tanda Tangan Penulis 1 : 



Tanda Tangan Penulis 2 :



Tanda Tangan Penulis 3 :

Tanggal : **12 Desember 2022**

HALAMAN PENGESAHAN

Tugas Akhir ini diajukan oleh :

Nama/NPM Mahasiswa 1 : Aan Nur Wahidi/1806141100
Nama/NPM Mahasiswa 2 : Farrel Alfarabi Saleh/1806186622
Nama/NPM Mahasiswa 3 : Muhammad Dahlan Yasadipura/1806205193
Program Studi : Sarjana Ilmu Komputer
Judul : Pengembangan Sistem Pendekripsi Gelombang Awal, Magnitude, Kedalaman, dan Lokasi Sumber Gempa Bumi Secara *Real-time* menggunakan Algoritma STA/LTA dan *Multi-target Regression*

Telah berhasil dipertahankan di hadapan Dewan Pengaji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Ilmu Komputer pada Program Studi Sarjana Ilmu Komputer, Fakultas Ilmu Komputer, Universitas Indonesia

DEWAN PENGUJI

Pembimbing 1 :

Ari Wibisono,
S.Kom., M.Kom

(Nilai telah diberikan melalui SISIDANG pada 07-01-2023, 14:48:18) (Revisi telah disetujui melalui SISIDANG pada 10-01-2023, 06:29:38)

Pengaji :

Dr. Indra Budi
S.Kom., M.Kom

(Nilai telah diberikan melalui SISIDANG pada 22-12-2022, 11:00:45) (Revisi telah disetujui melalui SISIDANG pada 06-01-2023, 08:21:21)

Pengaji :

Ir. Wahyu Catur
Wibowo M.Sc.,
Ph.D.

(Nilai telah diberikan melalui SISIDANG pada 22-12-2022, 11:04:51) (Revisi telah disetujui melalui SISIDANG pada 06-01-2023, 13:51:37)

Ditetapkan di : Depok, Jawa Barat

Tanggal : 09 Januari 2023

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, penulis dapat menyelesaikan penelitian ini. Penulisan penelitian ini ditujukan untuk memenuhi salah satu syarat memperoleh gelar Sarjana Ilmu Komputer pada Fakultas Ilmu Komputer Universitas Indonesia. Penulis dapat menyelesaikan penelitian ini karena bantuan, saran, dan juga dukungan dari berbagai pihak baik secara langsung maupun tidak langsung. Oleh karena itu, penulis ingin mengucapkan terima kasih kepada:

1. Allah SWT, Tuhan yang Maha Esa, yang telah memberikan kesehatan, kemampuan, dan juga nikmat-nikmat lainnya untuk penulis sehingga dapat menyelesaikan penelitian ini.
2. Keluarga penulis 1 yaitu Mama Murtingah dan Almarhum Bapak D. Suyono, Mas Asghar Nur Asworo, dan Mbak Widya Nur Pangestu yang telah mendukung penulis 1 dalam penggerjaan tugas akhir dan menjalani hidup.
3. Keluarga penulis 2 yaitu Ibu Yulinda dan Bapak Noerrachman Saleh, yang telah mendukung penulis 2 dalam penggerjaan tugas akhir dan menjalani hidup.
4. Keluarga penulis 3 yaitu Ibu Susi Sumarsiah, Bapak Moh Wahyudin, dan Adik Widya Aurellia yang selalu mendukung penulis 3 dalam penggerjaan tugas akhir dan menjalani hidup.
5. Bapak Ari Wibisono, S.Kom., M.Kom selaku dosen pembimbing dari penelitian yang telah membantu penulis dengan waktu, tenaga, dan pikirannya sehingga penelitian ini dapat diselesaikan.
6. Bapak Drs. Lim Yohanes Stefanus, M.Math., Ph.D., Ibu Adila Alfa Krisnadhi, S.Kom., M.Sc., Ph.D dan Bapak Amril Syalim, S.Kom., M.Eng., Ph.D. selaku pembimbing akademik dari penulis yang telah memberikan bimbingan selama masa perkuliahan di Fakultas Ilmu Komputer Universitas Indonesia.
7. Bapak Dr.Eng. Adi Wibowo, S.Si., M.Kom., dan Bapak Satriawan Rasyid Purnama selaku narasumber dari peneltian kami yang telah membantu kami dalam melanjutkan penelitian ini.

8. Aan Nur Wahidi, selaku penulis 1 dalam penelitian ini yang mengerjakan penelitian bersama, sering membantu penulis 2 dan 3 di kala kesulitan, dan menjamu penulis 2 dan 3 dengan makan pagi, siang, dan malam di rumah sendiri selama pengerjaan tugas akhir.
9. Farrel Alfarabi Saleh, selaku penulis 2 dalam penelitian ini yang mengerjakan tugas akhir sebaik mungkin, menjadi teman seperjuangan penulis 1 dan 3 saat masa-masa kuliah,
10. Muhammad Dahlan Yasadipura, selaku penulis 3 dalam penelitian ini yang mengerjakan tugas akhir sebaik mungkin, menjadi teman seperjuangan penulis 1 dan 2 saat masa-masa kuliah,
11. Qadzif Kamil Zahari, sebagai teman penulis yang membantu penulis dalam masalah-masalah pemrograman yang dialami selama pelaksanaan penelitian ini.
12. Rekan dekat penulis 1 dan 3 Abi Fajri Abdillah, Abdurrahman Luqmanul Hakim, Adriel Gian Ananta, Kevin Asyraf Putra Priyatna, Maulana Wildan Abdurrahman,Muhammad Glend Angga Dwitama, dan Reynard Adha Ryanda serta teman-teman satu grup penulis, yang telah menemani penulis dalam menjalani perkuliahan di Fakultas Ilmu Komputer Universitas Indonesia.
13. Rekan dekat penulis 2 Anggardha Febriano, Ahmad Irfan Luthfi Tumbuan, Carlo Tupa Indriauan, Roy Godsend, Akmal Farhan Raiyan, Stephen Handiar Christian, Nanda Ryaas Absar serta teman-teman satu grup penulis, yang telah menemani penulis dalam menjalani perkuliahan di Fakultas Ilmu Komputer Universitas Indonesia.
14. Rekan dekat penulis 1 Fadullah dan Thariq Zarwal Harrison selaku teman rumah yang telah menghibur penulis dalam melakukan penelitian ini.
15. Berbagai pihak lain yang tidak mungkin dapat disebutkan satu per satu yang membantu dalam proses perkuliahan maupun penyelesaian penelitian.

Penulis menyadari bahwa penelitian ini masih belum sempurna dan memiliki banyak kekurangan. Oleh karena itu, penulis meminta kritik dan saran agar penelitian ini dapat menjadi lebih baik. Penulis berharap bahwa penelitian ini dapat memberikan manfaat bagi banyak pihak.

Depok,



(Aan Nur Wahidi) (Farrel Alfarabi Saleh) (Muhammad Dahlan Yasadipura)

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama Penulis 1	:	Aan Nur Wahidi
NPM Penulis 1	:	1806141100
Nama Penulis 2	:	Farrel Alfarabi Saleh
NPM Penulis 2	:	1806186622
Nama Penulis 3	:	Muhammad Dahlan Yasadipura
NPM Penulis 3	:	1806205193
Program Studi	:	Ilmu Komputer
Fakultas	:	Ilmu Komputer
Jenis Karya	:	Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (Non-exclusive Royalty Free Right)** atas karya ilmiah saya yang berjudul:

Pengembangan Sistem Pendekripsi Gelombang Awal, Magnitude, Kedalaman, dan Lokasi Sumber Gempa Bumi Secara *Real-time* menggunakan Algoritma STA/LTA
dan *Multi-target Regression*

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok
Pada tanggal : 12 Desember 2022
Yang menyatakan



(Aan Nur Wahidi) (Farrel Alfarabi Saleh) (Muhammad Dahlan Yasadipura)

ABSTRAK

Nama Penulis : Aan Nur Wahidi, Farrel Alfarabi Saleh, Muhammad Dahlan
Yasadipura

Program Studi : Ilmu Komputer

Judul : Pengembangan Sistem Pendekripsi Gelombang Awal, Magnitude, Kedalaman, dan Lokasi Sumber Gempa Bumi Secara *Real-time* menggunakan Algoritma STA/LTA dan *Multi-target Regression*

Pembimbing 1 : Ari Wibisono, S.Kom., M.Kom

Kata kunci:

machine learning, visualisasi data, pemrograman *back-end*, gempa bumi

Indonesia merupakan negara rawan gempa karena secara geografis indonesia terletak pada pertemuan empat lempeng tektonik. Karena ini, pengembangan sebuah sistem prediksi *real-time* gempa bumi yang mencakup wilayah yang luas dengan gempa bumi besar sangat dibutuhkan untuk mengurangi korban jiwa. Penelitian ini mengusulkan pembuatan sistem pendekripsi cepat kedatangan gelombang-p dan penentuan hiposenter dan magnitudo gempa menggunakan *deep-learning*. Pengembangan sistem berbasis web ini bertujuan untuk memperingati masyarakat agar dapat lebih dini untuk melindungi diri sebelum gempa terjadi. Menggunakan data dari BMKG, data yang kami gunakan mencakupi 1892 set data gempa pada tahun 2009–2017 dan 26 set data gempa dari Katalog BMKG Januari 2019, penelitian ini menggunakan algoritma STA/LTA dalam menemukan *P-Arrival* dan membandingkan tiga model pembelajaran mesin untuk memprediksi hiposenter gempa dimana model Conv1d digabung dengan LSTM dengan interval waktu 20 detik merupakan skenario model terbaik dengan memiliki *mean absolute error* sebesar 0.470. Selain itu, penelitian ini berhasil mengimplementasi sistem berbasis web yang dapat menampilkan visualisasi data dengan menggunakan websocket berdasarkan data seismik yang dikumpulkan oleh BMKG. Visualisasi data seismik ini ditampilkan menggunakan *dynamic line chart* dan peta web interaktif.

ABSTRACT

Name : Aan Nur Wahidi, Farrel Alfarabi Saleh, Muhammad Dahlan
Yasadipura

Study Program : Ilmu Komputer

Title : Development of a *Real-time* Initial Wave, Magnitude, Depth, and Location of Earthquake Sources Detection System using STA/LTA Algorithm and *Multi-target Regression*

Counsellor 1 : Ari Wibisono, S.Kom., M.Kom

Counsellor 2 :

Key words:

Machine learning, data visualization, back-end programming, earthquake

Indonesia is an earthquake-prone country because geographically Indonesia is located at the confluence of four tectonic plates. Therefore, the development of a real-time earthquake prediction system that covers large areas with large earthquakes is urgently needed to reduce fatalities. This study proposes the creation of a rapid detection system for the arrival of p-waves, hypocenters and earthquake magnitudes using deep-learning. The development of this web-based system is aimed at warning people so that they can protect themselves before an earthquake occurs. Using data from BMKG, we used 1892 earthquake data sets in 2009–2017 and 26 earthquake data sets from January 2019 BMKG Catalog, this research uses the STA/LTA algorithm to find P-Arrival and compares three machine learning models to predict the earthquake hypocenter where Conv1d model is combined with LSTM with a time interval of 20 seconds is the best model scenario with a mean absolute error of 0.470. In addition, this research succeeded in implementing a web-based system that can display data visualization using websocket based on seismic data collected by BMKG. This seismic data visualization is displayed using dynamic line charts and an interactive web map.

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PERSETUJUAN	ii
LEMBAR PERNYATAAN ORISINALITAS	iii
LEMBAR PENGESAHAN	iv
KATA PENGANTAR	v
LEMBAR PERSETUJUAN PUBLIKASI ILMIAH	vii
ABSTRAK	x
DAFTAR ISI	xii
DAFTAR GAMBAR	xvi
DAFTAR TABEL	xviii
DAFTAR LAMPIRAN	1
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Permasalahan	2
1.2.1 Rumusan Permasalahan	2
1.2.2 Batasan Permasalahan	3
1.3 Tujuan Penelitian	3
1.4 Posisi Penelitian	4
1.5 Sistematika Penulisan	5
2 LANDASAN TEORI	7
2.1 <i>Gempa Bumi</i>	7
2.2 Metode dalam Sains Data	8
2.2.1 Format Data Mseed dan Pick	8
2.2.2 <i>Short Time Average over Long Time Average</i>	8

2.2.3	Pemrosesan Data	9
2.2.3.1	Normalisasi <i>Min-Max</i>	9
2.2.3.2	<i>Butter Bandpass</i>	9
2.2.3.3	Imputasi <i>Interpolasi</i>	10
2.2.4	<i>Multi Target Regression</i> dalam Pembelajaran Mesin	10
2.2.4.1	<i>Convolution Neural Network</i>	10
2.2.4.2	<i>Convolution Neural Network</i> untuk data 1 dimensi	12
2.2.4.3	<i>Long short-term memory</i>	13
2.2.5	Metode Validasi	13
2.2.6	Metode Evaluasi	14
2.3	Desain <i>User Interface</i>	15
2.4	Desain <i>User Experience</i>	15
2.5	Visualisasi Data	16
2.5.1	Peta Interaktif	16
2.5.2	<i>Dynamic Line Chart</i>	17
2.6	Pemrograman <i>Back-End</i>	18
2.6.1	API	18
2.6.2	REST API	18
2.6.3	JSON	19
2.6.4	Response Time	19
2.6.5	Websocket	20
2.6.6	Websocket API	20
3	METODOLOGI	21
3.1	Tahapan Penelitian	21
3.2	Deskripsi Data	27
4	IMPLEMENTASI MULTI TARGET REGRESSION	29
4.1	Perolehan Fitur	29
4.2	Pemrosesan Data	30
4.3	Pemodelan	32
4.4	Analisis Hasil	33
4.4.1	Skenario Conv1d	33
4.4.2	Skenario LSTM	34
4.4.3	Skenario Conv1d dan LSTM	35
4.4.4	Perbandingan Semua Skenario	35
5	IMPLEMENTASI PEMROGRAMAN BACK-END	39

5.1	Pemrograman <i>Back-End</i>	39
5.1.1	Perancangan Endpoint	40
5.1.2	Inisialisasi Firebase SDK dan Vertex AI	41
5.1.3	Implementasi REST API	41
5.1.3.1	Ekstraksi Informasi Data Mseed	42
5.1.3.2	Butter Bandpass Filter, Normalisasi, dan Interpolasi Data	43
5.1.3.3	Penyelarasan Data Seismik Berdasarkan Waktu Mulai Stasiun	43
5.1.4	Implementasi Websocket API	44
5.1.4.1	Mengambil Data Seismik pada Basis Data	44
5.1.4.2	Penyelarasan Data Seismik Berdasarkan Waktu Mulai Sensor dan Pembuatan Waktu Data Seismik	44
5.1.4.3	Pengiriman Data Seismik ke <i>Frontend</i> dan ke Machine Learning	45
5.1.5	<i>Deployment</i> Websocket API	45
5.1.6	Implementasi Evaluasi Pemrograman Back-End	46
5.2	Hasil Uji Coba Pemrograman <i>Back-end</i>	46
5.2.1	Websocket API	46
5.2.1.1	<i>Response Code</i>	47
5.2.1.2	Komputasi Websocket API	47
5.2.2	Evaluasi Seluruh Proses <i>Machine Learning</i> melalui <i>Backend</i>	48
5.2.3	Evaluasi Seluruh Komputasi Websocket API	49
6	IMPLEMENTASI PEMROGRAMAN <i>FRONT-END</i> DAN VISUALISASI DATA	51
6.1	Sistem Prediksi	51
6.1.1	Pembuatan <i>User Interface</i> Halaman Admin	52
6.1.2	Pembuatan <i>User Interface</i> Halaman Pengguna	53
6.1.3	Pembuatan <i>User Interface</i> Halaman <i>Login Admin</i>	53
6.2	Visualisasi Data	54
6.2.1	Alur Pengolahan Data Hingga Visualisasi	55
6.2.2	Pengambilan Data dari Back-end	59
6.2.3	Pemetaan Data dalam Bentuk Visualisasi	60
6.3	Tampilan Hasil Prediksi	68
6.3.1	Tampilan Halaman Admin	68
6.3.2	Tampilan Halaman Pengguna	69
6.4	Hasil Pengujian GUI	70

7 KESIMPULAN DAN SARAN	72
7.1 Kesimpulan	72
7.2 Saran	73
DAFTAR REFERENSI	74

DAFTAR GAMBAR

Gambar 1.1	Diagram Posisi Penelitian	4
Gambar 2.1	Ilustrasi CNN	11
Gambar 2.2	Ilustrasi arsitektur CNN	11
Gambar 2.3	Ilustrasi input CNN 1 dimensi	12
Gambar 2.4	Ilustrasi train split validation	14
Gambar 2.5	Peta web dan desain GUI terpadu menurut (1) OpenStreetMap dan (2) Google (Cybulski and Horbiński, 2020)	17
Gambar 2.6	Visualisasi waktu nyata dari data seismik yang direkam pada node yang berbeda. (Wang et al., 2021b)	18
Gambar 2.7	Contoh data JSON	19
Gambar 3.1	Tahapan Penelitian	21
Gambar 3.2	Pembagian Tugas Penelitian	24
Gambar 4.1	Alur teknis pemrosesan data seismik sebagai fitur	30
Gambar 4.2	Alur teknis pemrosesan data target	31
Gambar 4.3	Alur teknis sistem	32
Gambar 4.4	Alur teknis evaluasi	33
Gambar 4.5	Perbandingan MAE latitudo	36
Gambar 4.6	Perbandingan MAE longitudo	36
Gambar 4.7	Perbandingan MAE magnitudo	37
Gambar 4.8	Perbandingan MAE depth	37
Gambar 4.9	Perbandingan MAE time	38
Gambar 5.1	Alur teknis sistem <i>back-end</i>	39
Gambar 5.2	Proses Ekstraksi Data Mseed untuk disimpan pada Basis Data	42
Gambar 5.3	Struktur data mseed pada Firebase Realtime Database . . .	42
Gambar 5.4	Isi trace yang memuat informasi data mseed	43
Gambar 5.5	Alur proses pengiriman data ke frontend dan <i>machine learning</i>	44
Gambar 6.1	Alur teknis pemrograman <i>front-end</i>	52

Gambar 6.2	Alur penggunaan halaman admin	53
Gambar 6.3	Alur teknis visualisasi data	55
Gambar 6.4	ilustrasi data <i>timeseries</i>	56
Gambar 6.5	ilustrasi data <i>timeseries</i> setelah di <i>Butterworth bandpass filter</i>	57
Gambar 6.6	ilustrasi data <i>timeseries</i> setelah di dinormalisasi	57
Gambar 6.7	ilustrasi data <i>timeseries</i> setelah di diinterpolasi	58
Gambar 6.8	Alur pengambilan data	59
Gambar 6.9	Alur pemetaan data	60
Gambar 6.10	Pemetaan data <i>dynamic line chart</i>	61
Gambar 6.11	Tampilan <i>dynamic line chart</i>	63
Gambar 6.12	Tabel Parameter Gempa	64
Gambar 6.13	Tampilan Peta Interaktif	65
Gambar 6.14	Tampilan Panel Kiri Halaman Pengguna	67
Gambar 6.15	Tampilan Hasil Prediksi pada Halaman Admin	68
Gambar 6.16	Tampilan Hasil Prediksi Tabel Parameter Gempa	69
Gambar 6.17	Tampilan Hasil Prediksi pada Halaman Pengguna	70
Gambar 1	Papan Kanban	78
Gambar 2	Timeline Scrum	79

DAFTAR TABEL

Tabel 4.1	hasil fitur	30
Tabel 4.2	Hasil skenario conv1d	34
Tabel 4.3	Hasil skenario conv1d	34
Tabel 4.4	Hasil skenario conv1d	35
Tabel 5.1	Rancangan <i>endpoint</i>	41
Tabel 5.2	Data yang dikirimkan ke <i>frontend</i>	45
Tabel 5.3	Hasil Evaluasi Websocket API	47
Tabel 5.4	Hasil Evaluasi Komputasi Websocket API Sebelum Proses <i>Machine Learning</i>	48
Tabel 5.5	Hasil Evaluasi Pada Proses Deteksi <i>p-arrival</i>	48
Tabel 5.6	Hasil Evaluasi Pada Proses <i>Machine Learning</i>	49
Tabel 5.7	Hasil Evaluasi Seluruh Komputasi Websocket API	49
Tabel 6.1	hasil fitur	56
Tabel 6.2	Data yang dikirimkan ke <i>frontend</i>	58
Tabel 6.3	Hasil <i>Functionality Testing</i>	71

BAB 1

PENDAHULUAN

Pada bab pendahuluan dijelaskan mengenai latar belakang dilakukannya penelitian sebagai topik penelitian, permasalahan yang ingin diselesaikan, batasan permasalahan, tujuan akhir yang ingin dicapai dari penelitian yang dilakukan, posisi penelitian, serta sistematika penulisan pada laporan ini.

1.1 Latar Belakang

Secara geografis, Indonesia berada di wilayah lingkaran api pasifik atau cincin api pasifik dimana merupakan pertemuan tiga lempeng tektonik dunia seperti Lempeng Indo-Australia, Lempeng Eurasia dan Lempeng Pasifik. Oleh sebab itu, Indonesia merupakan negara yang rawan dengan bencana seperti gempa bumi, letusan gunung berapi hingga tsunami. Menurut data sejarah gempa bumi Jawa Timur, gempa bumi besar terjadi pada tahun 1994 dengan gaya dorong subduksi masif yang bersifat tsunamigenik. Maka, sebuah pengembangan sistem peringatan dini gempa bumi(*Earthquake Early Warning System*) yang mencakup wilayah yang luas diperlukan untuk mengetahui apakah suatu getaran dari gempa bumi seperti kejadian pada jawa timur dapat menyebabkan peristiwa *megathrust* di masa depan. Berdasarkan data dari USGS, Indonesia memiliki lebih dari 150 gempa bumi berkekuatan 7.0+ dalam skala magnitudo dari periode 1900–2022. Berdasarkan catatan Badan Pusat Statistik (BPS), terdapat 10.519 frequensi gempa bumi di Tanah Air sepanjang 2021. Jumlah tersebut naik 25,7% dibandingkan tahun sebelumnya yang sebanyak 8.368 frequensi gempa bumi. Pada tanggal 21 November 2022 terjadi bencana gempa bumi yang melanda daerah Cianjur, Jawa Barat. Pemerintah Kabupaten Cianjur, Jawa Barat, mencatat jumlah korban meninggal akibat bencana gempa bumi Cianjur 5,6 Magnitudo sebanyak 600 orang. Kejadian ini merupakan salah satu alasan mengapa indonesia memerlukan suatu sistem yang dapat memperingatkan masyarakat terhadap gempa bumi sebelum gempa tersebut melanda. Tujuan dari penelitian ini bermaksud untuk mempercepat proses peringatan masyarakat terhadap gempa bumi. Untuk mendapatkan data seismik, penelitian ini mengambil data rekaman gempa bumi BMKG yang direkam menggunakan stasiun-stasiun seismik. Namun, karena stasiun seismik BMKG jauh dari satu sama lain tetapi dekat ke daerah perkotaan, menyebabkan stasiun

jugaberikan sinyal seismik bising yang mengganggu pengembangan sistem peringatan dini gempa bumi. Penulis melanjutkan penelitian untuk membuat aplikasi berdasarkan model dan sistem deep learning yang telah dibuat sebelumnya (Wibowo et al., 2022). Pada penelitian sebelumnya, data yang dilakukan terhadap prediksi membutuhkan data *time-series* yang harus diubah dahulu menjadi gambar. Hal ini menyebabkan waktu pengolahan data dan prediksi secara realtime membutuhkan waktu yang cukup lama. Maka dari itu penulis mencoba mencari model lain agar input data tidak memerlukan proses untuk mengubah data menjadi foto, sehingga waktu prediksi dapat dijalankan secara *real-time*. Pada penelitian ini skema model kami bagi 3, yaitu dengan Conv1d, LSTM, dan Conv1d + LSTM. Model ini kami gunakan karena LSTM merupakan model yang cocok untuk data berupa timeseries karena LSTM dapat memprediksi data sintetik untuk mensimulasikan deret waktu dengan karakteristik yang berbeda dan conv1d kami gunakan karena pemrosesan data pada dalam bentuk dua dimensi membutuhkan waktu yang lebih lama dalam prediksinya, sehingga tidak memungkinkan untuk dipakai secara *real-time*. Maka kami menggunakan hasil modifikasi dari CNN, yaitu *Conv1d*. Data yang digunakan pada penelitian ini didapatkan dari kerja sama dengan Badan Meteorologi, Klimatologi, dan Geofisika (BMKG). Data dikumpulkan oleh BMKG guna melaksanakan penelitian sebelumnya, sehingga kita dapat menggunakan data tersebut untuk keberlangsungan penelitian ini. Isi data berupa seluruh data berformat mseed dan pick. Data yang penulis gunakan mencakupi 1892 set data gempa pada tahun 2009–2017. Gempa-gempa pada data tersebut memiliki magnitudo 3–6,5 M dan kedalaman 1,16–588,42 km di area dari 111,5° E Bujur 115° E dan 6,6° S Lintang 11,5° S. Penulis diizinkan kembali oleh BMKG untuk menggunakan data tersebut demi melanjutkan dan mengembangkan sistem dari penelitian sebelumnya.

1.2 Permasalahan

Pada bagian ini akan dijelaskan mengenai definisi permasalahan yang penulis hadapi dan ingin diselesaikan serta asumsi dan batasan yang digunakan dalam menyelesaiannya.

1.2.1 Rumusan Permasalahan

Berdasarkan latar belakang yang telah dijelaskan, penulis menyimpulkan beberapa permasalahan yang akan ditelaah dalam penelitian ini, diantaranya:

1. Apa metode yang dapat digunakan untuk mengidentifikasi gelombang awal suatu kejadian gempa berdasarkan data seismik dari BMKG?
2. Apa metode yang dapat digunakan untuk mendeteksi lokasi sumber, besar, dan kedalaman suatu kejadian gempa secara *real-time*?
3. Bagaimana data seismik dapat diolah untuk memvisualisasi aliran data dari sensor getaran gempa, titik terdeteksinya gelombang awal, dan lokasi hiposenter gempa?
4. Bagaimana sistem ini dapat mendeteksi gelombang awal, lokasi sumber, magnitude, dan kedalaman gempa bumi berdasarkan data dari BMKG?

1.2.2 Batasan Permasalahan

Ruang lingkup dari penelitian ini adalah sebagai berikut:

1. Eksplorasi, analisis, pengolahan, dan pemrosesan data dilakukan setelah tanggal 19 Agustus 2022 sampai 21 Desember 2022.
2. Data yang digunakan untuk pemodelan pembelajaran mesin dan visualisasi data merupakan 1892 set data gempa pada tahun 2009–2017.
3. Penulis memilih 618 kejadian yang direkam sepenuhnya di tiga stasiun (IA.GMJI, GE.JAGI, dan IA.PWJI). Tiga stasiun ini dipilih karena jarak antar stasiunnya berdekatan sehingga data diantara stasiun tersebut memiliki waktu datangnya gempa yang selaras, dan mereka mencatat kejadian gempa terbanyak di wilayah tersebut.
4. Sistem yang dibuat pada penelitian ini merupakan *Proof of Concept* (PoC) dari sebuah sistem pendeteksi gelombang awal, magnitude, kedalaman, dan lokasi sumber gempa bumi secara *real-time*.

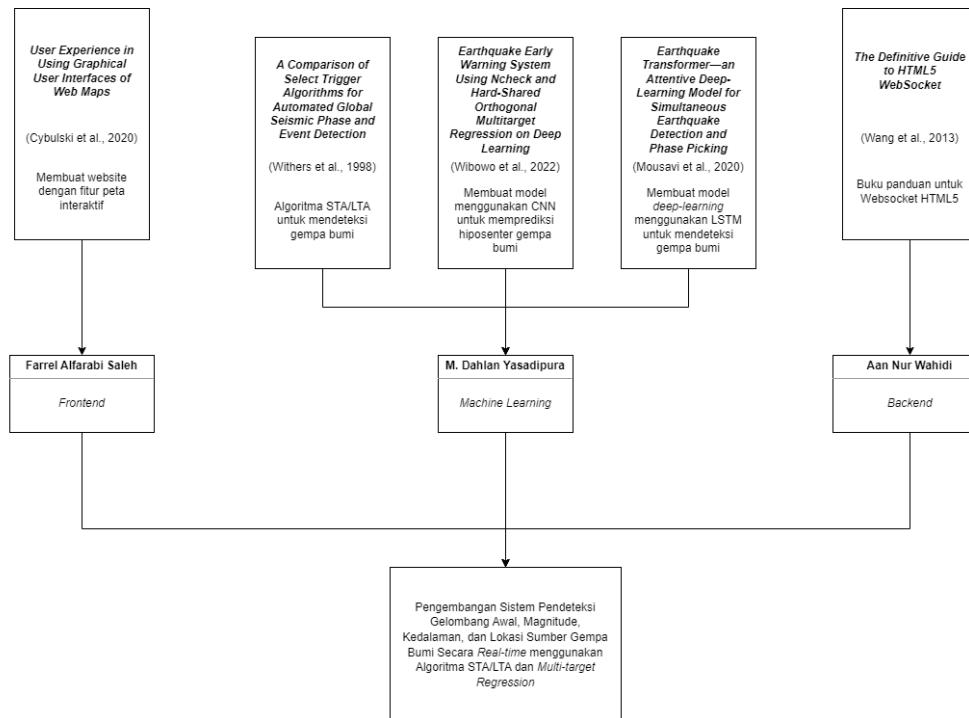
1.3 Tujuan Penelitian

Dilakukannya penelitian tugas akhir ini memiliki tujuan akhir sebagai berikut:

1. Memodelkan data seismik untuk mendeteksi gelombang awal, hiposenter, magnitude, dan kedalaman berdasarkan data dari Badan Meteorologi, Klimatologi, dan Geofisika(BMKG).

2. Memvisualisasi dan memberikan informasi yang dapat membantu Badan Meteorologi, Klimatologi, dan Geofisika(BMKG) mengenai mendeteksi gelombang awal, hiposenter, magnitude, dan kedalaman yang mudah dipahami, informatif, dan interaktif.
3. Menyajikan *application programming interface* untuk data dari Badan Meteorologi, Klimatologi, dan Geofisika(BMKG) terkait prediksi.
4. Menyambungkan model dengan data dari Badan Meteorologi, Klimatologi, dan Geofisika(BMKG) melalui pemrograman *back-end* dan menampilkan hasil prediksi dengan pemrograman *front-end*.

1.4 Posisi Penelitian



Gambar 1.1: Diagram Posisi Penelitian

Sumber: Olahan Penulis 2022

Basis dari penelitian ini adalah penelitian yang dilakukan oleh Wibowo et al. (2022) dengan judul penelitian *Earthquake Early Warning System Using Ncheck and Hard-Shared Orthogonal Multitarget Regression on Deep Learning* yang membuat suatu model sains data untuk memprediksi gelombang awal dan hiposenter gempa bumi menggunakan CNN. Dan kami juga mengacu dari penelitian dari penelitian yang dilakukan oleh (Mousavi et al., 2020) dengan judul *Earthquake*

transformer—an attentive deep-learning model for simultaneous earthquake detection and phase picking yang mana penilitian ini membahas bagaimana mengimplementasikan model LSTM pada data seismik. Namun untuk mendeteksi gelombang awal acuan utama penulis adalah penelitian yang dilakukan oleh Withers et al. (1998) dengan judul penelitian *A Comparison of Select Trigger Algorithms for Automated Global Seismic Phase and Event Detection* yang menggunakan *Short Time Average over Long Time Average* (STA/LTA). Penulis melanjutkan penelitian tersebut dan membangun *prototype* untuk merealisasikan model yang telah dibuat serta melakukan penelitian dengan menggunakan model lainnya. Selain itu terdapat buku panduan yang menjelaskan mengenai WebSocket pada HTML5 yang merupakan media untuk menghubungkan sistem secara *realtime* termasuk ide pembuatan WebSocket API yaitu pada buku yang berjudul *The Definitive Guide to HTML5 WebSocket* yang ditulis oleh Wang et al. (2013). Untuk membuat antarmuka pengguna, penulis mengacu pada penelitian (Cybulski and Horbiński, 2020) agar antarmuka pengguna sistem ini efisien dan efektif. Penilitian ini membahas perbandingan website Google Maps dan Openstreetmap yang memiliki fitur utama peta interktif dan menemukan kelebihan dan kekurangan dari masing-masing antarmuka pengguna website tersebut.

1.5 Sistematika Penulisan

Sistematika penulisan laporan adalah sebagai berikut:

- Bab 1 PENDAHULUAN

Pada bab ini mencakup latar belakang dilakukannya penelitian, permasalahan yang ingin diselesaikan serta asumsi dan batasan yang digunakan dalam penyelesaian masalah, tujuan penelitian, posisi penulis dalam penelitian, dan sistematika penulisan.

- Bab 2 LANDASAN TEORI

Pada bab ini mencakup landasan teori dan terminologi yang digunakan dalam penelitian.

- Bab 3 METODOLOGI

Pada bab ini mencakup tahap-tahap penelitian yang dilakukan dalam penelitian serta rincian mengenai tahapan tersebut.

- Bab 4 IMPLEMENTASI MULTI TARGET REGRESSION

Pada bab ini mencakup penjelasan lebih detail mengenai implementasi klasifikasi yang dilakukan dalam penelitian.

- Bab 5 IMPLEMENTASI PEMROGRAMAN *BACK-END*

Pada bab ini mencakup penjelasan lebih detail mengenai implementasi pemrograman *back-end* yang dilakukan dalam penelitian.

- Bab 6 IMPLEMENTASI PEMROGRAMAN *FRONT-END* DAN VISUALISASI DATA

Pada bab ini mencakup penjelasan lebih detail mengenai implementasi pemrograman *front-end* dan visualisasi data yang dilakukan dalam penelitian.

- Bab 7 KESIMPULAN DAN SARAN

Pada bab ini mencakup kesimpulan akhir penelitian dan saran untuk pengembangan berikutnya.

BAB 2

LANDASAN TEORI

Untuk memulai penelitian diperlukan landasan teori yang sesuai dengan masalah yang akan dipecahkan. Untuk membentuk dasar teori yang sesuai, perlu dikaitkan dengan hasil studi literatur yang dilakukan. Oleh karena itu, bab ini menjelaskan hasil tinjauan pustaka yang terkait dengan ruang lingkup penelitian ini.

2.1 *Gempa Bumi*

Menurut Bayong (2006:12) gempa bumi merupakan suatu gerakan atau getaran yang terjadi pada kulit bumi yang dihasilkan dari tenaga endogen. Tenaga endogen yaitu tenaga atau kekuatan perut bumi yang terjadi karena adanya perubahan pada kulit bumi. Sifat tenaga endogen ini dapat membentuk bumi menjadi tidak rata. Gempa bumi yang besar sering ditampakkan dengan pergetaran tanah sehingga dapat menyebabkan tsunami, longsor, dan aktivitas gunung vulkanik (Ghobarah et al., 2006). Pemahaman yang lebih baik tentang perilaku gempa bumi dapat membantu membuat kebijakan prabencana, menyelamatkan nyawa manusia dan meringankan beban ekonomi yang terlibat dalam pembentukan tim darurat, mengumpulkan persediaan medis dan makanan dan membangun kembali daerah yang terkena dampaknya (Tenreiro Machado and Lopes, 2013). Hiposenter adalah pusat titik gempa yang ada di dalam bumi. Hiposentrum adalah pusat gempa, sehingga lokasi hiposentrum adalah lokasi awal dari gempa bumi itu terjadi. Gempa bumi yang terjadi akibat dari tektonisme dan vulkanisme pasti memiliki hiposentrum. Skala Magnitudo merupakan ukuran kekuatan gempa bumi yang menggambarkan besarnya energi seismik yang dipancarkan oleh sumber gempa dan merupakan hasil pengamatan seismograf. Artinya, semakin besar gempa, maka semakin tinggi pula angka magnitudo.

Pada gempa bumi terdapat 2 gelombang utama yang mana 2 gelombang itu merupakan gelombang *P(p-arrival)*, dan gelombang *S(s-arrival)*. Gelombang *P* atau biasa disebut gelombang Utama atau gelombang kompresi (*compressional wave*), yaitu merupakan gelombang longitudinal dan gelombang tercepat, dengan kecepatan ± 6 km/detik. Pada umumnya hanya beberapa makhluk hidup dapat merasakan datang gelombang primer ini sendiri. Gelombang *S* biasa disebut gelombang kedua atau *shear wave*, merupakan gelombang transversal yang

bergerak dengan kecepatan ± 4 km/detik. Sehingga dari perbedaan rambat tersebut, *p-arrival* dapat ditangkap oleh sensor terlebih dahulu sebelum datangnya *s-arrival*.

2.2 Metode dalam Sains Data

Bagian ini menjelaskan mengenai proses sains data yang dilakukan di penelitian untuk memodelkan data seismik dari BMKG sehingga dapat memprediksi *Earthquake location*. Secara umum, proses sains data dalam penelitian berupa pemrosesan dan transformasi data, pemodelan *machine learning*, dan evaluasi. Proses sains data dilakukan secara iteratif untuk menghasilkan model yang memuaskan.

2.2.1 Format Data Mseed dan Pick

SEED (Standard for the Exchange of Earthquake Data) format standar internasional untuk pertukaran data seismologi secara digital. Format data ini dikembangkan oleh anggota "Federation of Digital Seismographic Networks" (FDSN) terutama untuk pertukaran data seismologi yang belum diproses antara institusi dan agensi yang berbeda. mseed atau miniSEED adalah bagian dari standar SEED yang digunakan untuk *time series* data. Metadata yang disediakan sangat terbatas dengan data utamanya adalah data seismik dengan informasi singkat stasiunnya. Sedangkan Pick merupakan format data yang digunakan BMKG setelah dilakukannya analisis secara manual pada data seismik yang ditangkap pada mseed.

2.2.2 Short Time Average over Long Time Average

Short Time Average over Long Time Average(STA/LTA) adalah algoritma pemicu yang digunakan untuk deteksi seismik. Ini banyak digunakan dalam banyak perangkat lunak pemrosesan jaringan seismik gerak lemah, serta dalam perekam seismik portabel, mungkin juga berguna dalam banyak aplikasi gerak kuat. itu meningkatkan deteksi gempa bumi lemah dan mengurangi jumlah deteksi palsu yang dipicu oleh kebisingan (Choubik et al., 2020). Penurunan jumlah *alarm* palsu dan selektivitas pemicu meminimalkan pekerjaan analis. Cara kerja algoritma ini didasarkan pada perbandingan amplitudo rata-rata jangka pendek (STA) dan rata-rata jangka panjang (LTA) dari sinyal di penampil sinyal aktif.

2.2.3 Pemrosesan Data

Pemrosesan data merupakan sekumpulan teknik yang diterapkan pada *database* untuk menghapus noise, *missing value*, dan data yang tidak konsisten. Hal ini juga dapat dilakukan dengan berbagai teknik. Inisiatif dalam menentukan teknik ini diperlukan karena data mentah seringkali tidak lengkap dan memiliki format yang tidak konsisten, sehingga kualitas data sendiri memiliki korelasi langsung dengan keberhasilan setiap proyek yang melibatkan analisis data.

Preprocessing sendiri meliputi validasi dan imputasi data. Tujuan validasi adalah untuk menilai kelengkapan dan keakuratan data yang disaring. Di sisi lain, tujuan imputasi adalah untuk memperbaiki kesalahan dan mengisi nilai yang hilang baik secara manual maupun otomatis. Pemrosesan data pada penelitian ini bertujuan untuk mempersiapkan data sebagai masukan algoritma *machine learning*. Metode ini dilakukan karena data yang diterima pada penelitian ini merupakan data lengkap meliputi informasi letak stasiun dan juga data seismik, sehingga data yang diterima perlu diekstraksi terlebih dahulu dan dilakukan beberapa teknik transformasi agar mereduksi tingkat *noise* dan melaraskan data yang diterima agar dapat dproses oleh *machine learning*.

2.2.3.1 Normalisasi *Min-Max*

Normalisasi min-maks melakukan transformasi linier pada data asli. Min-Max normalization merupakan metode normalisasi dengan melakukan transformasi linier terhadap data asli sehingga menghasilkan keseimbangan nilai perbandingan antar data saat sebelum dan sesudah proses (T. T. Hanifa and Telkom, 2017). Teknik ini bertujuan untuk mengubah semua data, lalu diskalakan dalam rentang (0, 1).

$$x_{scaled} = \left(\frac{x - x_{min}}{x_{max} - x_{min}} \right)$$

2.2.3.2 Butter Bandpass

Filter Butterworth adalah jenis filter pemrosesan sinyal yang dirancang untuk memiliki respons frekuensi sedatar mungkin dalam band pass. Ini juga disebut sebagai filter magnitudo datar maksimal(Hussin et al., 2016). Filter ini memiliki respon frekuensi maksimum datar (tidak ada ripple) dari frekuensi 0 Hz hingga akhir daerah pass band yaitu frekuensi cut off yang mengalami pelemahan -3dB. Proses ini dilakukan dalam pemrosesan data dimaksudkan noise pada data seismik dapat terfilter dan tidak menyebabkan *false alarm*.

2.2.3.3 Imputasi *Interpolasi*

Interpolasi merupakan metode yang menghasilkan titik-titik data baru dalam suatu jangkauan dari suatu set diskrit data-data yang diketahui. hal ini bertujuan untuk mengisi data-data yang kosong dikarenakan oleh penambahan jangkauan data. Interpolasi adalah proses inputasi nilai data dengan memperhatikan data lain yang kita miliki. Interpolasi merupakan teknik untuk mencari nilai suatu variabel yang hilang pada rentang data yang diketahui (Sukatis et al., 2019).

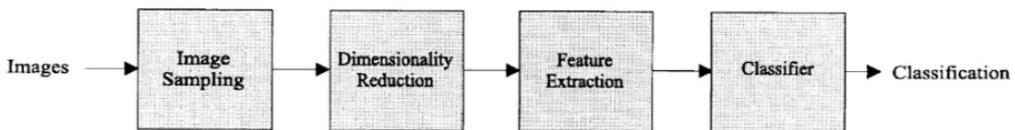
2.2.4 *Multi Target Regression* dalam Pembelajaran Mesin

Pembelajaran mesin adalah kemampuan algoritma untuk mengekstrak informasi dari data tanpa memprogram informasi tersebut. Berdasarkan dari informasi tersebut, pembelajaran mesin dapat memprediksi data berdasarkan pengetahuan tentang data masa lalu. Prediksi tersebut dapat dievaluasi lebih lanjut untuk mengetahui seberapa efektif model pembelajaran mesin tersebut. Pembelajaran mesin menerima data tambahan yang diproses sesuai kebutuhan untuk membuat prediksi dengan hasil yang lebih memuaskan. Proses ini dilakukan secara iteratif hingga hasil machine learning telah membawa hasil yang diinginkan. Secara umum, proses pembelajaran mesin untuk memperoleh pengetahuan disebut *training*, dan mengevaluasinya terhadap prediksi disebut *testing*.

Regresi adalah metode analisis statistik yang digunakan untuk melihat pengaruh dua variabel atau lebih. Hubungan variabel yang dimaksud adalah fungsional, yang tampak dalam bentuk model matematis. Analisis regresi digunakan dalam pembelajaran mesin dengan *supervised learning*. Regresi hanya memprediksi variabel yang sifat datanya interval atau rasio.

2.2.4.1 *Convolution Neural Network*

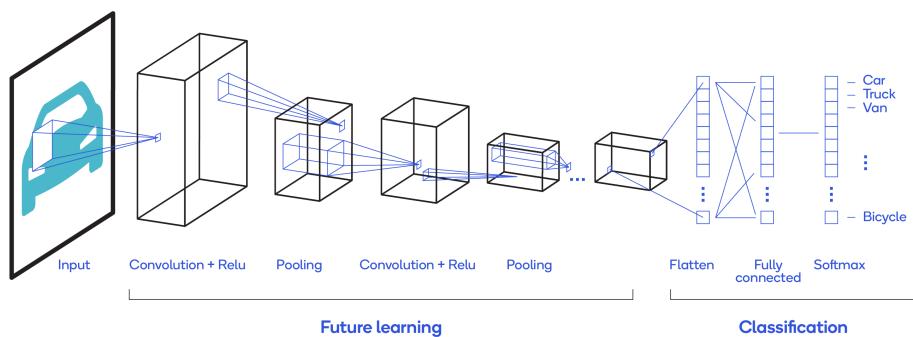
CNN adalah model yang mendapat perhatian karena kemampuan klasifikasinya berdasarkan informasi kontekstual. Model umum CNN telah dijelaskan di bawah ini pada gambar 1. Model umum CNN terdiri dari empat komponen yaitu (a) convolution layer, (b) pooling layer, (c) activation function, dan (d) fully connected layer Indolia et al. (2018).



Gambar 2.1: Ilustrasi CNN

Sumber: Indolia et al. (2018)

Convolutional Neural Network (CNN) adalah salah satu jenis neural network yang biasa digunakan pada data image. CNN bisa digunakan untuk mendeteksi dan mengenali object pada sebuah image. CNN adalah sebuah teknik yang terinspirasi dari cara manusia, menghasilkan persepsi visual seperti contoh di atasIndolia et al. (2018). CNN juga salah satu jenis algoritma pembelajaran mendalam yang dapat mengambil input dalam bentuk gambar dan menentukan aspek atau objek mana dalam gambar yang dapat digunakan mesin untuk "mempelajari" cara mengenali gambar dan membedakan gambar. Alasan utama mengapa CNN dianggap di atas model klasik lainnya adalah sebagai berikut. Pertama, minat utama untuk menerapkan CNN terletak pada gagasan menggunakan konsep pembagian bobot, karena jumlah parameter yang membutuhkan pelatihan berkangur secara substansial, menghasilkan generalisasi yang lebih baik. Karena parameter yang lebih rendah, CNN dapat dilatih dengan lancar dan tidak mengalami overfittingIndolia et al. (2018). Berikut merupakan gambaran dari arsitektur CNN dalam memecahkan suatu problem identifikasi sebuah gambar:



Gambar 2.2: Ilustrasi arsitektur CNN

Sumber: <https://medium.com/@16611110/apa-itu-convolutional-neural-network-836f70b193a4>

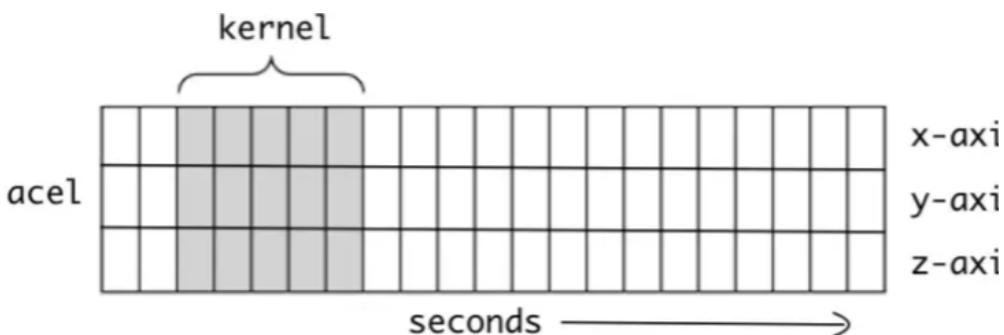
Proses yang terjadi pada bagian ini adalah melakukan “*encoding*” dari sebuah image menjadi *features* yang berupa angka-angka yang merepresentasikan gambar

tersebut. *Feature extraction layer* terdiri dari dua bagian yaitu *Convolutional Layer* dan *Pooling Layer*.

2.2.4.2 Convolution Neural Network untuk data 1 dimensi

CNN sendiri pada umumnya menggunakan input sebuah gambar dalam pengaplikasiannya. Namun, karena pada pengembangan sistem ini penulis menggunakan data *time series* dalam penginputan data. Karena pemrosesan data pada dalam bentuk dua dimensi membutuhkan waktu yang lebih lama dalam prediksinya, sehingga tidak memungkinkan untuk dipakai secara *realtime*. Maka penulis menggunakan hasil modifikasi dari CNN, yaitu *Conv1d*. Sehingga hasil input yang merupakan data time series dapat memprediksi sebuah hasil yang berupa sebuah angka numerik.

Pada pengaplikasianya 1D CNN dapat melakukan tugas pengenalan aktivitas dari data accelerometer, seperti jika orang tersebut berdiri, berjalan, melompat dll. Data ini memiliki 2 dimensi. Dimensi pertama adalah waktu dan lainnya adalah nilai percepatan dalam 3 *axis*. Plot berikut mengilustrasikan bagaimana kernel akan bergerak pada data akselerometer. Setiap baris mewakili akselerasi deret waktu untuk beberapa sumbu. Kernel hanya dapat bergerak dalam satu dimensi sepanjang sumbu waktu.



Gambar 2.3: Ilustrasi input CNN 1 dimensi

Ada perbedaan yang signifikan dalam hal kompleksitas komputasi konvolusi 1D dan 2D, yaitu gambar dengan dimensi $N \times N$ yang berkonvolusi dengan kernel $K \times K$ akan memiliki kompleksitas komputasi $O(N^2K^2)$. Sementara dalam konvolusi 1D yang sesuai (dengan dimensi yang sama, N dan K) ini memiliki nilai kompleksitas komputasi sebesar $O(NK)$. Ini berarti bahwa dalam kondisi yang ekuivalen kompleksitas komputasi CNN 1D secara signifikan lebih rendah daripada 2D CNN(Kiranyaz et al., 2021).

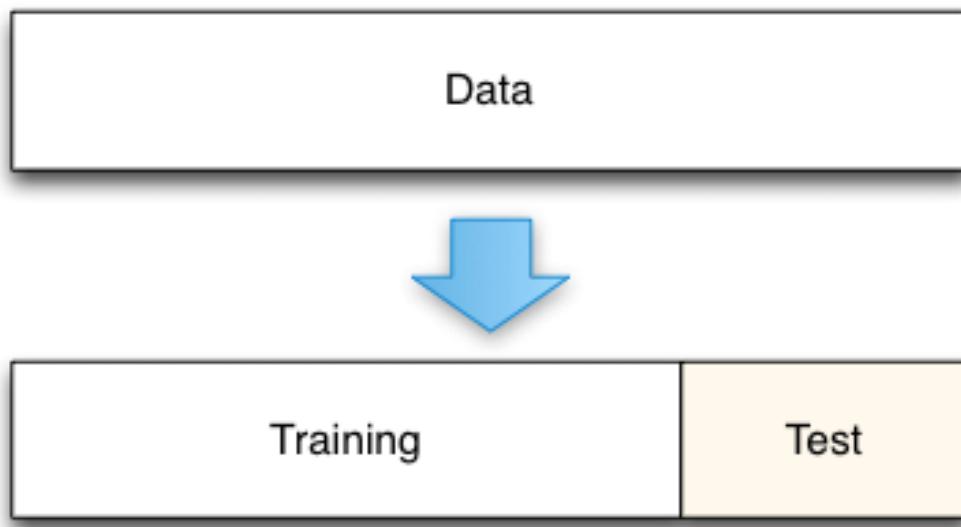
2.2.4.3 Long short-term memory

Long short term memory network (LSTM) adalah salah satu modifikasi dari recurrent neural network atau RNN. Banyak modifikasi dari RNN, tetapi LSTM merupakan salah satu yang populer di antaranya. LSTM diusulkan sebagai solusi untuk mengatasi terjadinya vanishing gradient pada RNN saat memproses data sequential yang panjang. LSTM hadir untuk menutupi kekurangan RNN, karena tidak dapat memprediksi kata berdasarkan data masa lalu yang disimpan dalam waktu lama. Dengan demikian, LSTM mampu mengingat kumpulan informasi yang sudah lama disimpan, dan juga menghapus informasi yang sudah tidak relevan lagi. LSTM lebih efisien dalam mengolah, memprediksi, dan mengklasifikasikan data berdasarkan periode waktu tertentu. Jaringan LSTM hierarkinya memecahkan masalah multidimensi dengan membagi keseluruhan masalah menjadi sub-masalah dan mengurnya dalam struktur hierarkis. Ini memiliki keuntungan bahwa fokus dapat ditempatkan pada satu atau beberapa sub-masalah yang terhubung(Lindemann et al., 2021).

Gerbang multiplikatif memungkinkan sel memori LSTM untuk menyimpan dan mengakses informasi dalam jangka waktu yang lama, sehingga mengurangi masalah *vanishing gradient*. Misalnya, selama gerbang input tetap tertutup (yaitu memiliki aktivasi mendekati 0), aktivasi sel tidak akan ditimpak oleh input baru yang tiba di jaringan, dan oleh karena itu dapat tersedia untuk jaringan jauh di kemudian hari dalam urutan, dengan membuka gerbang output(Graves, 2012).

2.2.5 Metode Validasi

Pada pengembangan machine learning ini, penulis menggunakan metode validasi yang paling dasar yaitu *train/test split*. Manfaat dari pendekatan ini adalah kita dapat melihat bagaimana model bereaksi terhadap data yang sebelumnya tidak terlihat. Dalam penelitian ini penulis melanjutkan jumlah validasi seperti penelitian sebelumnya. penulis menggunakan 90% data untuk pelatihan (556 poin data) dan 10% untuk validasi data secara simultan (62 titik data). Validasi data dilakukan secara acak dan digunakan untuk menyetel jaringan parameter(Wibowo et al., 2022).



Gambar 2.4: Ilustrasi train split validation

2.2.6 Metode Evaluasi

Tujuan utama regresi linier adalah untuk membuat perkiraan nilai suatu variabel (variabel dependen) jika nilai variabel yang lain yang berhubungan dengannya (variabel lainnya) sudah ditentukan. Pada pengembangan sistem ini, penulis menggunakan metode evaluasi *Mean Absolute Error(MAE)*. Hal ini dilakukan guna mengukur tingkat keakuratan model peramalan. Ini juga dikenal sebagai akurasi yang bergantung pada skala karena menghitung kesalahan dalam pengamatan yang dilakukan pada skala yang sama. Ini digunakan sebagai metrik evaluasi untuk model regresi dalam *machine learning*. MAE adalah selisih mutlak rata-rata antara nilai sebenarnya (aktual) dengan nilai prediksi (perkiraan). MAE digunakan untuk mengukur keakuratan model statistik dalam membuat peramalan atau prediksi. Secara statistik, *Mean Absolute Error* (MAE) mengacu pada hasil pengukuran selisih antara dua variabel kontinu.

$$MAE = \left(\frac{1}{n} \right) \sum_{i=1}^n |y_i - x_i|$$

n = total number of data points
 y_i = prediction
 x_i = true value

2.3 Desain *User Interface*

User Interface (UI) atau antarmuka pengguna adalah interaksi antara pengguna dan sistem yang menggunakan perintah, grafik, tampilan, konten, dan lain-lain. Penulis juga dapat mengatakan bahwa antarmuka pengguna merupakan grafis representasi struktur tata letak. Sekarang, antarmuka pengguna memiliki berbagai format ukuran dan sistem seperti PC, ponsel, game hingga konten seperti blog dan masih banyak lagi (Goel et al., 2022).

Selama orang telah merancang sistem komputer interaktif, beberapa telah berusaha untuk mempromosikan desain yang baik dengan menerbitkan panduan desain antarmuka pengguna (Johnson, 2014). Terdapat beberapa prinsip dari pengembangan antarmuka pengguna seperti fokus pada pengguna dan tugasnya dan bukan pada teknologinya, pertimbangkan fungsi terlebih dahulu dan presentasi kemudian, sesuaikan dengan pandangan pengguna tentang tugas tersebut, desain untuk kasus umum, jangan memperumit tugas pengguna, memfasilitasi pembelajaran, desain untuk responsif, sampaikan informasi bukan hanya data, cobalah pada pengguna lalu perbaiki (Johnson, 2007).

2.4 Desain *User Experience*

User Experience (UX) atau pengalaman pengguna adalah proses keseluruhan dan pengalaman dari pengguna dengan aplikasi atau semua jenis konten (Goel et al., 2022). UX adalah seni perencanaan desain produk jadi interaksi dengan produk jadi akan sehebat mungkin (Kristiadi et al., 2017). UX adalah reaksi atau perilaku pengguna pada dasar-dasar penggunaan benda itu. Pengalaman pengguna terhadap aplikasi ditentukan oleh cara mereka berinteraksi dengannya (Goel et al., 2022). *User experience* secara murni merupakan perasaan subjektif yang dirasakan seorang pengguna saat menggunakan suatu produk atau servis (Wang et al., 2021a)

Dari perspektif pengguna, UX adalah representasi suatu produk dan bagaimana penggunaan produk tersebut pada dunia nyata, menembus semua aspek interaksi pengguna dengan produk (Norman, 2004). Untuk mencapai suatu UX yang bagus, diperlukan sebuah pengertian dalam terhadap pengguna dan kepentingan pengguna, yang pada akhirnya berbicara kepada pengalaman yang akan dimiliki pengguna saat menggunakan produk (Pillay and Wing, 2019). UX harus diperkenalkan pada tahap awal dengan mengidentifikasi pengguna dan peran yang akan menggunakan perangkat lunak, dan menggunakan pemikiran itu untuk sebagai dasar UX dan perangkat lunak secara keseluruhan. (Pillay and Wing, 2019).

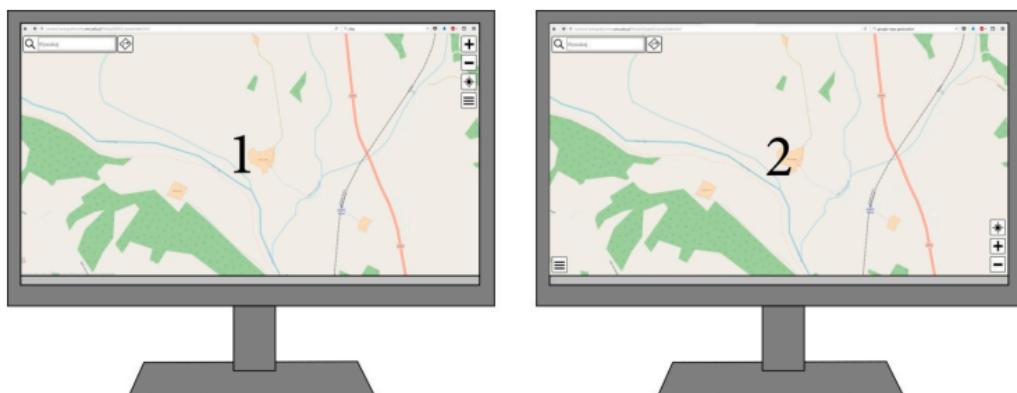
2.5 Visualisasi Data

Visualisasi data secara umum merepresentasikan grafik dan bagan yang menampilkan secara visual sebuah representasi dari data. Grafik-grafik ini merupakan cara yang ampuh untuk memberi rangkuman serta tampilan dari sebuah data yang dapat dimengerti oleh kebanyakan orang. Grafik-grafik ini juga membantu menemukan sebuah bentuk dan pola dari sebuah data yang dapat membantu dalam analisis data. Sebuah pengertian dari grafik-grafik ini dapat menjelaskan konsep utama dari sebuah data secara visual (Sahay, 2016). Dalam penelitian ini penulis menggunakan peta interaktif serta *realtime chart* sebagai alat visualisasi data.

2.5.1 Peta Interaktif

Pada kartografi internet, salah satu faktor penting adalah interaktivitas. Interaksi merupakan sistem komunikasi antar peta dan pengguna. Ini adalah cara pengguna mengontrol peta seperti mengubah skala, memindah lokasi yang ditampilkan, dan mencari lokasi yang diinginkan (Roth, 2013). Interaksi pada peta berada pada *graphical user interface* (GUI) yang mengandung tombol-tombol yang memiliki suatu fungsi dan simbol-simbol yang merepresentasikan kegunaan dari tombol-tombol tersebut (Roth, 2015). Terdapat perbedaan dari *user interface* (UI) dan *user experience* (UX) pada sebuah peta interaktif. *User interface* merupakan alat-alat seperti tombol textitgraphical user interface yang membantu user mengontrol serta mengubah apa yang ada pada peta (Haklay, 2010). Design UX berhubungan dengan bagaimana pengguna berkomunikasi menggunakan alat-alat ini sebagaimana pengguna dapat merasa nyaman dan puas (Garrett, 2011). Sedangkan desain UI merupakan sebuah proses yang memiliki tujuan untuk mengimplementasi alat-alat interaktif seperti *Leaflet.js library* atau *library*

peta interaktif lain (Horbiński and Lorek, 2020). Tombol-tombol *graphical user interface* (GUI) memiliki dampak besar terhadap efisiensi sebuah peta interaktif. Penelitian GUI mempertimbangkan penempatan, angka, dan grafik tombol pada tampilan pertama sebagai faktor yang berpengaruh (Roth, 2013).



Gambar 2.5: Peta web dan desain GUI terpadu menurut (1) OpenStreetMap dan (2) Google (Cybulski and Horbiński, 2020)

Terdapat sebuah eksperimen (Cybulski and Horbiński, 2020) yang menggunakan eye-tracking sebagai alat pendekripsi efisiensi GUI untuk membandingkan efisiensi antara Google Maps dan Openstreetmap. Ekperimen ini mengadopsi 2 kelompok berisi 20 orang untuk menguji GUI dari kedua website lalu menganalisis efisiensi GUI dan pengalaman visual pengguna. Penelitian ini memiliki kesimpulan bahwa salah satu pemandu untuk design UI/UX sebuah peta interaktif adalah menyusun tombol dengan fungsi interaktif serupa secara bersamaan (pencarian, perutean), susun tombol di jumlah sudut layar paling sedikit, dan gunakan ikon sebagai alat pemandu pengguna terhadap fungsi GUI.

2.5.2 *Dynamic Line Chart*

Seorang Ilmuwan bersejarah bernama Playfair dan seorang ahli matematika bernama Lamber merupakan pencipta grafik visual, mereka percaya bahwa merepresentasikan data kompleks pada sebuah grafik dapat membantu masyarakat mengerti data (Simkin and Hastie, 1987). Berdasarkan arah pengembangan visualisasi informasi saat ini. *Dynamic chart* merupakan sebuah bentuk ekspresi data *time-series* (Li and Qin, 2020). Karena data seismik yang didapatkan dari stasiun sensor BMKG merupakan data time-series maka sebuah dynamic line chart adalah bentuk visualisasi data yang cocok untuk merepresentasikan data ini.



Gambar 2.6: Visualisasi waktu nyata dari data seismik yang direkam pada node yang berbeda. (Wang et al., 2021b)

2.6 Pemrograman *Back-End*

back-end merupakan ruang logis dengan fungsional dan operasi dari aplikasi perangkat lunak atau sistem informasi (Adam et al., 2019). *back-end* merupakan program yang berhubungan dengan *server* yang bekerja dibalik layar untuk memenuhi kebutuhan yang diperlukan *front-end* agar suatu aplikasi atau web aplikasi dapat berjalan sebagaimana mestinya. Sehingga *backend* sangat memengaruhi dengan fungsionalitas suatu sistem.

2.6.1 API

API (Application Programming Interface) memaparkan layanan atau data yang disediakan melalui serangkaian *resources* yang telah ditentukan, seperti *methods*, *objects* atau URI (Stylos et al., 2009). Dengan menggunakan *resources* ini, sistem lain dapat mengakses data atau layanan tanpa harus mengimplementasikan objek dan prosedur yang mendasarinya (Meng et al., 2018).

2.6.2 REST API

REST (Representational State Transfer) API merupakan API yang menggunakan HTTP *request* untuk mengakses dan menggunakan data yaitu POST, GET, PUT, dan DELETE yang merupakan fundamental fungsi pada database yaitu Create, Read, Update, dan Delete (Poulter et al., 2015). Layanan ini kemudian dapat dikonsumsi oleh semua jenis klien yang diautentifikasi perangkat (Poulter et al., 2015).

2.6.3 JSON

JSON (JavaScript Object Notation) merupakan salah satu format data yang saat ini populer digunakan pada web atau aplikasi sebagai format pertukaran data. Hal ini dikarenakan JSON merupakan format data yang ringan, mudah dibaca dan ditulis oleh manusia, dan mudah bagi mesin untuk mengurai dan menghasilkannya (Abd El-Aziz and Kannan, 2014).



Gambar 2.7: Contoh data JSON

Sumber: Olahan Penulis (2022)

JSON diawali dan diakhiri dengan kurung kurawal sebagai pembungkusan *object* seperti tipe data *dictionary*. JSON memiliki dua fitur utama yaitu *key* dan *value*. *key* harus berupa *string* sedangkan *value* harus berupa tipe data yang kompatibel dengan JSON seperti *string*, *number*, *array*, *boolean*, *null* dan *object* data JSON lainnya.

2.6.4 Response Time

Response time adalah jeda waktu antara inisiasi *client* dari perintah sistem sampai dengan penyelesaian dan penampilan hasilnya (Miller, 1968). *Response time* merupakan salah satu metrik untuk *performance testing* dari suatu sistem. Dari perspektif pengguna, *response time* sistem yang ditunjukkan sebagai waktu tunggu pengguna saat mengakses suatu sistem, sangat menentukan kesan pengguna terhadap kinerja sistem sehingga panjang waktu respon adalah salah satu faktor penting dari *user experience* (Zhou et al., 2016).

2.6.5 Websocket

Websocket merupakan protokol komunikasi komputer yang dapat digunakan untuk komunikasi secara *realtime* atau *full-duplex* antara aplikasi berbasis web dan webserver (Rachmat et al., 2016). Websocket mendukung pengiriman menggunakan data *binary* atau *text strings* sehingga pengiriman dapat menggunakan JSON yang sangat populer sebagai media pengiriman data.

2.6.6 Websocket API

Websocket API merupakan API yang menggunakan layanan websocket untuk dapat menghubungkan dua sistem secara *realtime* dengan tujuan data dikirimkan secara terus menerus tanpa harus melakukan *request* berulang kali (Wang et al., 2013).

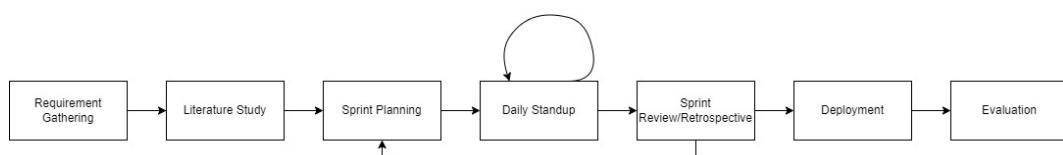
BAB 3

METODOLOGI

Bab ini membahas metodologi penelitian yang diterapkan pada skripsi ini yang mencakup metode penelitian dan tahapan penelitian. Karena penelitian ini mencakupi *software development* penulis menerapkan scrum *agile framework* sebagai metode penyelesaian penelitian ini. Tahapan penelitian dibagi berdasarkan tiga bidang, yaitu *machine learning*, pemrograman *back-end* dan *front-end* dan visualisasi data.

3.1 Tahapan Penelitian

Dalam pelaksanaan penelitian ini penulis merancang sebuah rangkaian tahapan-tahapan berdasarkan kerangka *agile* bernama scrum. Kerangka ini penulis gunakan karena memiliki kegiatan-kegiatan yang pantas dilaksanakan demi menyelesaikan masalah-masalah yang penulis sebutkan pada rumusan permasalahan. Terdapat 7 kegiatan yang penulis adopsi dari kerangka *agile* scrum yaitu pengumpulan informasi kebutuhan, studi literatur, perencanaan *sprint*, pertemuan harian, *sprint review/retrospective*, *deployment*, dan evaluasi. Tahapan-tahapan ini serta urutan pelaksanaan dapat dilihat pada Gambar 3.1.



Gambar 3.1: Tahapan Penelitian

Sumber: (Olahan Penulis, 2022)

Berikut adalah penjelasan singkat mengenai tahap-tahap penelitian tersebut:

1. Pengumpulan Kebutuhan

Penulis melaksanakan wawancara dengan Bapak Ariska Rudyanto, S.Si, Dipl.Tsu, M.Sc selaku perwakilan dari BMKG. Dalam wawancara ini menanyakan beberapa pertanyaan terkait sistem peringatan dini gempa bumi yang sedang dikembangkan oleh BMKG serta mengumpulkan informasi terkait siapa saja target pengguna dan fitur-fitur apa yang mereka inginkan pada suatu sistem peringatan dini gempa bumi. BMKG memerlukan dua

jenis aplikasi yaitu aplikasi admin yang akan digunakan oleh operator BMKG sebagai alat pengawasan serta aplikasi user yang akan digunakan oleh masyarakat. Pada tahap wawancara dimana penulis mempresentasikan progres visualisasi data yang sudah penulis terapkan, Bapak Ariska berkomentar bahwa aplikasi admin perlu memiliki beberapa fitur tambahan selain yang penulis sudah rencanakan yaitu fitur *freeze* yaitu fitur dimana komponen *dynamic line chart* berhenti sebagian saat mendeteksi *p-arrival* agar dapat terlihat lebih jelas posisi mendeteksi *p-arrival* pada grafik, fitur yang memperbesar dan memfokuskan peta pada daerah sumber gempa saat mendeteksi oleh algoritma prediksi, penanda stasiun perlu memiliki nama stasiun ditampilkan pada peta interaktif, dan hasil dari algoritma prediksi yakni waktu mendeteksi *p-arrival*, kedalaman gempa, magnitudo gempa, serta lokasi sumber gempa perlu terlihat lebih menonjol karena ini merupakan informasi yang utama untuk ditunjukkan kepada pengguna atau operator.

2. Studi Literatur

Penulis mempelajari konsep dan teori yang diperlukan sebagai landasan untuk melakukan penelitian melalui studi literatur. Penulis mempelajari dan mengumpulkan literatur yang berhubungan dengan penelitian. Beberapa literatur yang dikumpulkan diperoleh dari portal *database online IEEE Xplore* dan *ScienceDirect*.

3. *Sprint Planning*

Pada penelitian ini kami mendefinisikan *sprint* sebagai waktu jangka kerja 2 minggu atau 14 hari. Setiap awal pelaksanaan sebuah *sprint* kami mengadakan diskusi yang menentukan tugas-tugas yang ditargetkan untuk selesai pada *sprint* tersebut dan menentukan bobot setiap tugas serta kepentingannya terhadap keseluruhan proyek. Tugas-tugas ini didapatkan pada tahap pengumpulan kebutuhan yang lalu dikumpulkan menjadi suatu daftar *backlog*. Pada metodologi *agile Scrum*, *backlog* merupakan daftar semua tugas-tugas yang perlu dilaksanakan untuk menyelesaikan suatu proyek. Pada scrum, terdapat sebuah papan kanban yang merupakan tabel berisi backlog, tugas yang dikerjakan, dan tugas yang selesai. Kanban dan linimasa Scrum dapat dilihat pada Lampiran 1.

4. *Daily Standup*

Pada *agile framework scrum*, *daily standup* merupakan sebuah pertemuan harian yang ditujukan untuk membantu koordinasi tim. Putaran umpan balik yang cepat ini membantu tim menyelaraskan dan tetap di jalur, yang

serupa dengan ngerumpi dalam sepak bola. Jika muncul masalah, Anda dapat mengatasinya dengan cepat dan menjaga proyek tetap pada jalurnya. Pada penelitian ini penulis menjalankan *daily standup* setiap hari melalui pertemuan daring.

5. *Sprint Retrospective*

Sprint retrospective adalah cara untuk menyatukan tim scrum dan merencanakan cara terbaik untuk meningkatkan kualitas dan efektivitas terkait dengan orang, interaksi, proses, alat, dan definisi selesai setiap orang. Kegiatan ini kami dilaksanakan setiap akhir sebuah sprint namun bergantian dengan *Sprint Review*.

6. *Sprint Review*

Sprint review adalah aktivitas inspeksi yang dilakukan terhadap tujuan sprint atau dalam hal ini produk. Dalam sprint review akan dilihat apakah aktivitas yang dilakukan sudah memenuhi definition of done atau belum. Pada akhir *sprint* apabila tidak dijadwalkan *Sprint Retrospective*, penulis bertemu secara daring untuk membahas tugas apa saja yang sudah dilaksanakan dan yang akan dilaksanakan pada *sprint* selanjutnya.

7. *Deployment*

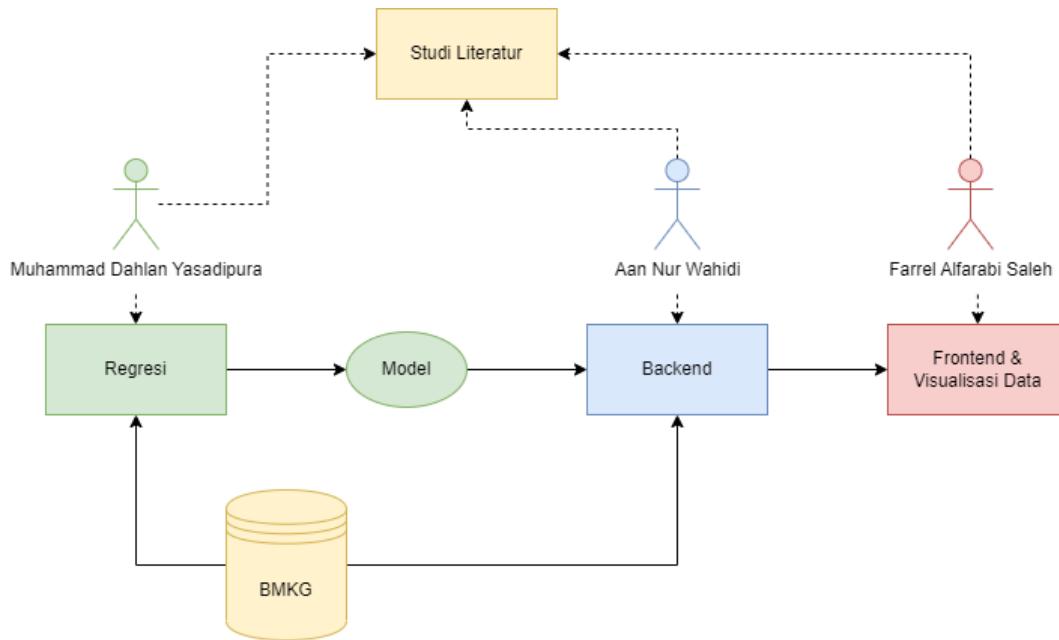
Pada tahap ini penulis mempublikasi aplikasi menggunakan Firebase Hosting untuk *frontend*, Google Cloud Run untuk *backend*, dan Vertex AI untuk model *machine learning*.

8. Evaluasi

Pada tahap ini kami mengevaluasi kinerja sistem yang sudah dipublikasi dengan menguji performa algoritma prediksi, menguji koneksi dengan server dan menguji user interface menggunakan pengujian fungsional.

Studi ini dilaksanakan oleh tiga peneliti yang memiliki bidang pekerjaan masing-masing. Masing-masing bidang ini merupakan cabang ilmu pengetahuan komputer. Bidang-bidang ilmu ini termasuk *machine learning*, pemrograman *back-end* dan *front-end* dan visualisasi data. Meskipun setiap penulis memiliki fokus yang berbeda-beda, penelitian ini diselesaikan melalui kolaborasi satu sama lain. Salah satu kolaborasi ini merupakan studi literatur. Bidang *machine learning* akan dipelajari dan dipraktikkan oleh Muhammad Dahlan Yasadipura, pemrograman *back-end* akan dipelajari dan dipraktikkan oleh Aan Nur Wahidi, dan pemrograman *front-end* dan visualisasi data akan dipelajari dalam literatur dan

dipraktekkan oleh Farrel Alfarabi Saleh. Rangkaian ilmu serta pembagian tugas dalam penelitian ini dapat dilihat pada Gambar 3.2



Gambar 3.2: Pembagian Tugas Penelitian

Sumber: (Olahan Penulis, 2022)

Berikut adalah penjelasan singkat mengenai tugas-tugas tersebut:

1. Regresi

Tahapan dari klasifikasi secara garis besar dibagi menjadi empat tahap. Pada tahap pertama data dikumpulkan berdasarkan kebutuhan penelitian. Selanjutnya, pada tahap kedua data akan dibersihkan, diproses, dan dipersiapkan sebagai masukan untuk model *machine learning*. Setelah itu, pemodelan berdasarkan data dari proses sebelumnya dilakukan. Terakhir, pada tahap analisis hasil model akan dievaluasi berdasarkan metrik yang sesuai. Penjelasan masing-masing tahap adalah sebagai berikut

(a) Perolehan Data

Penelitian ini menggunakan data yang diperoleh dari kerja sama dengan Badan Meteorologi, Klimatologi, dan Geofisika (BMKG). Data dikumpulkan oleh BMKG guna melaksanakan penelitian sebelumnya, sehingga kita dapat menggunakan data tersebut untuk keberlangsungan penelitian ini. Isi data berupa seluruh data berformat mseed dan pick. Data yang penulis gunakan mencakupi 1892 set data gempa pada tahun 2009–2017. Gempa-gempa pada data tersebut memiliki

magnitudo 3–6-5 M dan kedalaman 1,16–588,42 km di area dari 111,5° E Bujur 115° E dan 6,6° S Lintang 11.5° S. penulis sebagai penulis diizinkan kembali oleh BMKG untuk menggunakan data tersebut demi melanjutkan dan mengembangkan sistem dari penelitian sebelumnya.

(b) Pembersihan dan Pemrosesan Data

Selanjutnya data kemudian dibersihkan dan diproses. Pada penelitian ini digunakan 3 stasiun dan 3 *channel* sensor. Tiga stasiun ini dipilih karena jarak antar stasiunnya berdekatan sehingga data diantara stasiun tersebut memiliki waktu datangnya gempa yang selaras, dan mereka mencatat kegempaan terbanyak di wilayah tersebut(Wibowo et al., 2022). Setiap data pada mseed diekstrasi menjadi sebuah data frame dan dicocokan dengan pick file yang tersedia dan dimasukan kedalam beberapa fitur. lalu data seismik yang di peroleh diolah kembali agar sesuai dengan *input* yang diinginkan.

(c) Pemodelan

Fitur akan dimasukkan ke dalam algoritma *machine learning* untuk menghasilkan prediksi *multi output regression*. Parameter pada algoritma *machine learning* disetel dengan melatih algoritma dengan parameter-parameter yang berbeda. Hal ini untuk menghasilkan model dengan metrik evaluasi tertinggi untuk algoritma tersebut. Selain parameter input yang berbeda, parameter dan layer pada model penulis setel agar juga dapat menghasil metrik evaluasi tertinggi.

(d) Analisis Hasil

Algoritma *machine learning* dievaluasi berdasarkan metrik regresi yang sudah dimaksud pada subbab 2.2.6. Untuk menyesuaikan tujuan model yaitu memprediksi lokasi gempa, dan besar *magnitude* gempa. Hasil *evaluation matrix* yang memiliki error terendah, kedepannya penulis implementasikan dan penulis hubungkan dengan *backend* dan *frontend*.

2. Pemrograman *Back-end*

Tahapan dari pemrograman *back-end* terdiri dari tiga tahap. Tahap pertama adalah perancangan *endpoint* yang dibutuhkan untuk visualisasi data secara *realtime* pada data seismik mseed. Tahap kedua adalah implementasi pemrograman *back-end* berdasarkan rancangan sebelumnya. Tahap terakhir adalah melakukan evaluasi dari hasil implementasi.

(a) Perancangan Endpoint

Pada tahapan ini, penulis merancang *endpoint* yang diperlukan

berdasarkan kebutuhan visualisasi data secara *realtime* oleh *frontend* dan input data berdasarkan model *machine learning*. Perancangan *endpoint* meliputi URL berdasarkan nama mseed dan stasiun yang diperlukan yaitu GMJI, JAGI, dan PWJI.

(b) Implementasi

Pada tahapan ini, penulis melakukan implementasi pemrograman *back-end* berdasarkan rancangan yang telah dibuat sebelumnya. Hasil implementasi berupa sebuah URL WebSocket dan REST API yang dapat diakses oleh *front-end* untuk kemudian melakukan *request* agar mendapatkan data *static* dan secara *realtime* dalam format JSON untuk visualisasi data dan juga hasil prediksi gempa pada data mseed.

(c) Evaluasi

Pada tahapan ini, evaluasi dilakukan berdasarkan metrik dari implementasi *back-end* yang sudah dibuat.

3. Pemrograman *Front-end* dan Visualisasi Data

Pemrograman terbagi menjadi dua bagian yaitu Pembuatan *User Interface* Aplikasi Admin dan Pembuatan *User Interface* Aplikasi User

(a) Pembuatan *User Interface* Halaman Admin

Pada aplikasi ini penulis membuat halaman yang berisi peta interaktif yang menandakan lokasi stasiun penerima sinyal gempa dan lokasi sumber gempa, *Dynamic Line Chart* yang menampilkan data *realtime* yang berasal dari tiga sensor dari stasiun PWJI, GMJI, dan JAGI, serta tabel yang menampilkan informasi hasil prediksi dari model *machine learning* seperti magnitude, lokasi sumber gempa, kedalaman, lokasi stasiun pendekripsi, dan waktu penemuan *p-arrival*. Aplikasi ini ditujukan untuk tim operator dari BMKG sebagai alat pengawasan terhadap aplikasi user serta stasiun-stasiun gempa. Penulis juga membuat halaman log-in yang digunakan sebagai sistem autentikasi pengguna. Pengguna hanya dapat registrasi melalui sistem backend dikarenakan hanya pengguna tertentu yang akan diberi akses sebagai admin.

(b) Pembuatan *User Interface* Halaman Pengguna

Aplikasi ini memiliki halaman yang serupa dengan aplikasi admin namun tidak memiliki GUI *Dynamic Line Chart*. Interface aplikasi ini lebih mengutamakan tampilan informasi yang penting bagi pengguna umum seperti sumber gempa serta waktu hingga gempa. Aplikasi ini

juga memiliki halaman yang menjelaskan cara menggunakan aplikasi ini serta halaman yang berisi informasi developer aplikasi. Aplikasi ini ditujukan kepada pengguna umum sebagai alat peringatan gempa.

Pada setiap aplikasi terdapat visualisasi data yang diterima oleh stasiun BMKG. Tahap pertama adalah penentuan tipe visualisasi data yang ingin dibuat. Tahap kedua adalah implementasi dan analisis hasil penelitian. Tahap terakhir merupakan pengambilan kesimpulan dari hasil implementasi pada visualisasi data.

(a) Penentuan Tipe Visualisasi Data

Pada tahapan Studi Literatur, penulis memperoleh konsep tipe visualisasi data. Dari pengetahuan ini, terdapat tipe visualisasi data yang kemudian diimplementasi sebagai tujuan untuk menjawab rumusan permasalahan. Tipe visualisasi data tersebut merupakan *dynamic line chart* dan peta interaktif. Penulis menggunakan *dynamic line chart* untuk menampilkan data seismik secara *real-time* dan peta interaktif sebagai visualisasi lokasi sumber gempa dan stasiun perekam kejadian.

(b) Implementasi dan Analisis Hasil Uji

Pada tahapan ini, penulis membuat visualisasi data berdasarkan tipe visualisasi yang diperoleh pada tahap sebelumnya. Data yang divisualisasi berasal dari *back-end* yang didapatkan melalui beberapa *endpoint*. Kemudian, penulis menjelaskan hasil uji coba visualisasi data untuk dynamic line chart serta peta interaktif.

3.2 Deskripsi Data

Data yang digunakan pada penelitian ini adalah data mseed dan pick. Data yang penulis gunakan mencakupi 1892 set data gempa pada tahun 2009–2017 dan 26 set data gempa dari Katalog BMKG Januari 2019. Gempa-gempa pada data tersebut memiliki magnitudo 3–6,5 M dan kedalaman 1,16–588,42 km di area dari 111,5° E Bujur 115° E dan 6,6° S Lintang 11,5° S. Penulis memilih 618 kejadian yang direkam sepenuhnya di tiga stasiun (IA.GMJI, GE.JAGI, dan IA.PWJI). Tiga stasiun ini dipilih karena jarak antar stasiunnya yang kecil, dan mereka mencatat kejadian gempa terbanyak di wilayah tersebut (Wibowo et al., 2022). Data dari masing-masing mseed berisi informasi mengenai nama stasiun penerima, nama saluran dari setiap stasiun, *sampling-rate*, waktu perekaman data dimulai dan diakhiri, serta data amplitudo getaran gempa bumi dalam bentuk

time-series. Setiap data pick berkolerasi dengan suatu data msseed, dari berkas ini penulis mengambil data waktu *p-arrival*, waktu *s-arrival*, dan koordinat sumber gempa sebagai parameter pengujian pada hasil algoritma prediksi.

BAB 4

IMPLEMENTASI MULTI TARGET REGRESSION

Bab ini menguraikan lebih detail terkait implementasi yang dilakukan dari tahap klasifikasi yang dikerjakan oleh Muhammad Dahlan Yasadipura.

4.1 Perolehan Fitur

Pada penelitian ini digunakan 3 stasiun dan 3 *channel* sensor. Tiga stasiun ini dipilih karena jarak antar stasiunnya berdekatan sehingga data diantara stasiun tersebut memiliki waktu datangnya gempa yang selaras, dan mereka mencatat kejadian gempa terbanyak di wilayah tersebut(Wibowo et al., 2022).

Skema model kami bagi 3 dalam penelitian ini, yaitu dengan Conv1d, LSTM, dan Conv1d + LSTM. Model ini kami gunakan karena LSTM merupakan model yang cocok untuk data berupa timeseries karena LSTM dapat memprediksi data sintetik untuk mensimulasikan deret waktu dengan karakteristik yang berbeda dan conv1d kami gunakan Karena pemrosesan data pada dalam bentuk dua dimensi membutuhkan waktu yang lebih lama dalam prediksinya, sehingga tidak memungkinkan untuk dipakai secara *realtime*.Maka kami menggunakan hasil modifikasi dari CNN, yaitu *Conv1d*.

Data yang kami olah merupakan dan berformat mseed dan pick. Setiap data pada mseed diekstrasi menjadi sebuah data frame yang terdiri dari tiga stasiun sehingga setiap mseed masing masing memiliki keterangan keterangan kolom berupa tanggal dan waktu kejadian, waktu terjadinya *p-arrival*, latitudo, longitudo, dan magnitudo.lalu kami membuat dataframe baru yang memiliki fitur kolom berupa starttime, endtime, npts, data seismik pada 3 sensor setiap stasiunnya, waktu *p-arrival*, waktu *s-arrival*, lokasi stasiun, dan sampling rate. Data tersebut diambil dari data seismik dan sampling rate pada mseed, dan file pick yang dicocokan dengan ID pada mseed. berikut merupakan tampilan fitur hasil ekstrasi yang didapatkan, hasil disajikan dalam bentuk tabel berikut.

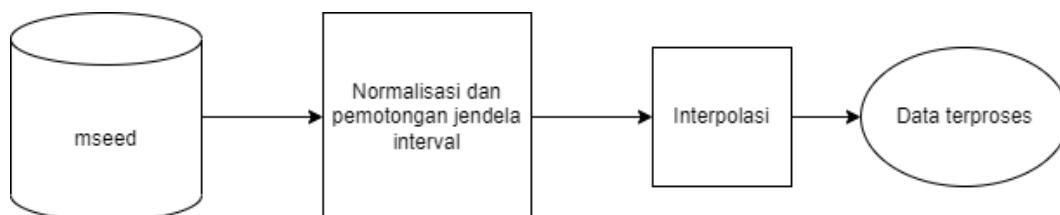
Tabel 4.1: hasil fitur

No	Fitur	deskripsi
1	id	Nama file mseed
2	starttime	Waktu dimulainya rekaman seismik
3	endtime	Waktu selesainya rekaman seismik
4	npts	Jumlah banyaknya data seismik
5	<i>E_WAVEFORM</i>	Data seismik pada sensor E yang berupa <i>timeseries</i>
6	<i>N_waveform</i>	Data seismik pada sensor N yang berupa <i>timeseries</i>
7	<i>Z_waveform</i>	Data seismik pada sensor Z yang berupa <i>timeseries</i>
8	<i>p_arrival_time</i>	Waktu <i>p-arrival</i> tiba dalam bentuk data ke-n
9	<i>s_arrival_time</i>	Waktu <i>s-arrival</i> tiba dalam bentuk data ke-n
8	station	Stasiun gempa yang beroperasi
9	<i>sampling_rate</i>	Banyak data perdetiknya

Untuk selanjutnya data *E_WAVEFORM*, *N_WAVEFORM*, dan *Z_WAVEFORM* harus kami proses terlebih dahulu agar sesuai dengan *input* pada model *machine learning* yang kami kembangkan.

4.2 Pemrosesan Data

Data mseed yang kami terima sebanyak 618 yang kita saring lagi menjadi 3 stasiun stasiun ini dipilih karena jarak antar stasiunnya yang kecil, sehingga total stasiun yang kami gunakan sebanyak 1854 stasiun dengan masing-masing terdapat 3 *channel* sensor pada stasiunnya. Data diproses sesuai dengan gambaran besar pada gambar 4.3

**Gambar 4.1:** Alur teknis pemrosesan data seismik sebagai fitur

Sumber: (Olahan Penulis, 2022)

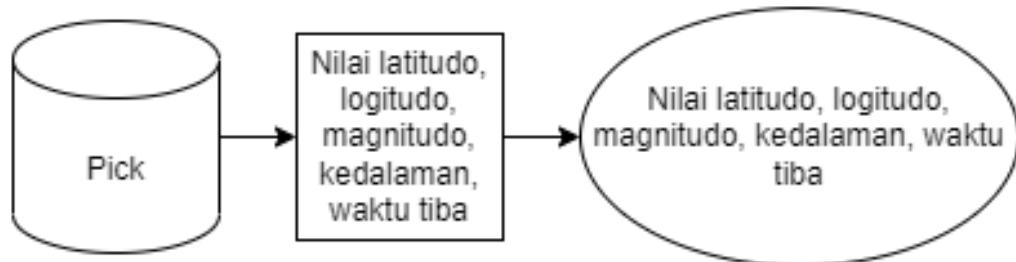
Pertama data akan di *filter* dengan menggunakan *filter butter bandpass*, hal ini bertujuan untuk meminimalkan *noise* pada data seismik sehingga menimbulkan false alarm pada proses penentuan *p_arrival*. Setelah itu dilakukan *normalization*

pada setiap *timeseries* dengan *min-max scaler normalization* dimana *min-max scaler* akan mengolah data dengan nilai minimum dan maksimum data teknik ini bertujuan untuk mengubah semua data, lalu diskalakan dalam rentang (0, 1).

Selanjutnya, data gelombang seismik akan dipotong dengan panjang masing-masing jendela dengan interval 15 detik sebelum dan sesudah *p-arrival*, 10 detik sebelum dan sesudah *p-arrival*, dan 5 detik sebelum dan sesudah *p-arrival*. hal ini dilakukan untuk mengetahui parameter output manakah yang memiliki nilai error terendah sehingga nilai terbaik akan diimplementasi pada program.

Perbedaan *sampling rate* mempengaruhi banyaknya data perdetik, sehingga jumlah data setelah dipotong kedalam jendela berbeda di setiap stasiunnya. Maka dari itu dilakukanlah proses interpolasi pada data. Laju pengambilan sampel adalah 20-25Hz yang diinterpolasi sesuai preferensi *input* model menjadi 100 Hz, sehingga series input berukuran 3000×3 untuk interval 30 detik, 2000×3 untuk interval 20 detik, dan 1000×3 untuk interval 10 detik. Dalam pengaplikasianya secara realtime, *p-arrival* didapatkan berdasarkan algoritma STA LTA pada backend. lalu dilakukanlah pemotongan jendela interval dan interpolasi menjadi 100hz.

Setelah itu kita memproses data dari mseeed dengan tujuan memperoleh data yang ingin diprediksi, yaitu logitudo, latitudo, magnitudo, kedalaman, dan waktu tiba gempa. Maka berikut adalah gambaran besar bagaimana data dapat diperoleh menjadi target output yang diinginkan.



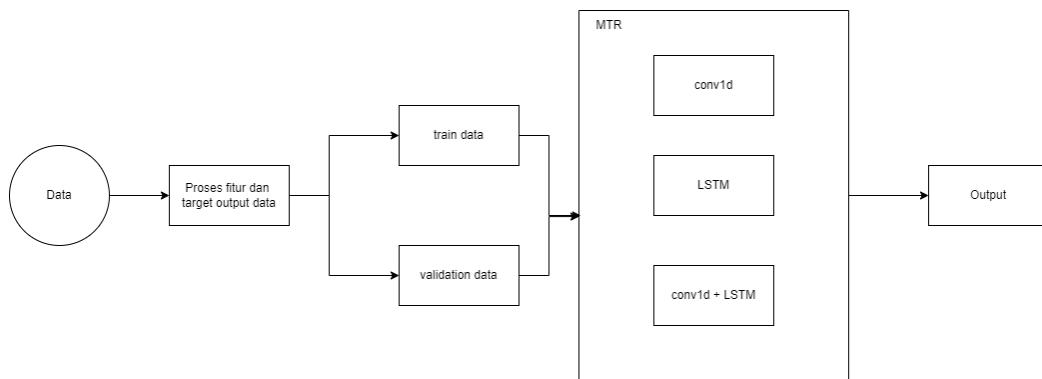
Gambar 4.2: Alur teknis pemrosesan data target

Sumber: (Olahan Penulis, 2022)

Setelah data selesai diolah, data dipisah untuk dijadikan test set dan training set. Train set merupakan 90% dan test set merupakan 10% dari dataset. Data perlu dipisah karena diakhir pemodelan diperlukan cara untuk mengukur performa model untuk tahap analisis.

4.3 Pemodelan

Pemodelan secara garis besar dilakukan dalam beberapa tahap. Tahap pertama merupakan memisahkan train set menjadi dua yaitu *train set* dan *test set*. Tahap kedua melatih model *Multi Target Regression*. Tahap terakhir model sudah dapat di *evaluate* untuk dilihat hasilnya. Untuk alur Fitur-fitur yang dipakai untuk pemodelan dapat dilihat pada Lampiran 4.3



Gambar 4.3: Alur teknis sistem

Sumber: (Olahan Penulis, 2022)

Pertama data diproses seperti pada pemrosesan data, yang mana fitur merupakan data seismik yang tersedia dan *target output* merupakan data latitudo, longitudo, magnitudo, dan kedalaman. Setelah itu data dipisah data *train* dan data *validation* sebagaimana *train set* merupakan 90% dan *validation set* merupakan 10% dari dataset. Pada tahap kedua data setelah data dibagi *train* dan validasinya, data lalu di training melalui model yang kami buat. Skema model kami bagi 3 dalam penelitian ini, yaitu dengan Conv1d, LSTM, dan Conv1d + LSTM. Model ini kami gunakan karena LSTM merupakan model yang cocok untuk data berupa *time-series* karena LSTM dapat memprediksi data sintetik untuk mensimulasikan deret waktu dengan karakteristik yang berbeda dan conv1d kami gunakan Karena pemrosesan data pada dalam bentuk dua dimensi membutuhkan waktu yang lebih lama dalam prediksinya, sehingga tidak memungkinkan untuk dipakai secara *real-time*. Maka kami menggunakan hasil modifikasi dari CNN, yaitu *Conv1d* sehingga hasil input yang merupakan data time series dapat memprediksikan sebuah hasil yang berupa sebuah angka numerik.



Gambar 4.4: Alur teknis evaluasi

Sumber: (Olahan Penulis, 2022)

Pemodelan akan menghasilkan model *machine learning* terbaik berdasarkan MAE untuk masing-masing algoritma dan skenario. Setiap model akan memprediksi *test set* dan hasil prediksinya tersebut akan dianalisis lebih lanjut menggunakan MAE. Model terbaik sesuai hasil analisis akan disimpan di *back-end* untuk sistem prediksi. Alur proses ini dapat dilihat pada Gambar 4.4

4.4 Analisis Hasil

Setiap model terbaik dari masing-masing skenario dan algoritma machine learning akan dianalisis. Pada pengembangan sistem ini, kami menggunakan metode evaluasi *Mean Absolute Error*(MAE). Hal ini dilakukan guna mengukur tingkat keakuratan model prediksi hipotesenter antara prediksi terhadap kejadian asli. Ini juga dikenal sebagai akurasi yang bergantung pada skala karena menghitung kesalahan dalam pengamatan yang dilakukan pada skala yang sama. Model dipilih berdasarkan MAE dari validation set 182 stasiun gempa. Skenarionya sendiri dibagi menjadi tiga, yaitu *Conv1d*, *LSTM*, dan *Conv1d* dengan *LSTM*. Dimana setiap skenario diuji dengan tiga input yaitu jendela interval 30 detik, 20 detik, dan 10 detik. Sebagai catatan, hasil MAE merupakan data yang dihitung merupakan data sebelum di-denormalisasi sehingga rentang data pada hasil metode evaluasi MAE adalah (0-1).

4.4.1 Skenario Conv1d

Skenario pertama menggunakan model dari conv1d. Teknik ini akan diimplementasi untuk 182 stasiun sebagai data validasi. Hasil evaluasi model klasifikasi menggunakan *validation set* tersebut dapat dilihat pada

Tabel 4.2: Hasil skenario conv1d

interval <i>input</i>	MAE lat	MAE long	MAE depth	MAE magnitude	MAE time
30 detik	0.123	0.184	0.060	0.097	0.074
20 detik	0.162	0.188	0.063	0.101	0.091
10 detik	0.135	0.165	0.073	0.092	0.088

Pada skenario menggunakan model Conv1d. Pada input dengan interval 30 detik memiliki MAE latitudo sebesar 0.123, MAE longitudo sebesar 0,184, MAE depth sebesar 0.060, MAE magnitudo sebesar 0.097, dan MAE time sebesar 0,074. input dengan interval 20 memiliki MAE latitudo sebesar 0.162, MAE longitudo sebesar 0.188, MAE depth sebesar 0.063, MAE magnitudo sebesar 0.101, dan MAE time sebesar 0,091. Terakhir input dengan interval 10 memiliki MAE latitudo sebesar 0.135, MAE longitudo sebesar 0,165, MAE depth sebesar 0.073, MAE magnitudo sebesar 0.092,dan MAE time sebesar 0,088.

4.4.2 Skenario LSTM

Skenario kedua menggunakan model dari LSTM.Teknik ini akan diimplementasi untuk 182 stasiun sebagai data validasi. Hasil evaluasi model klasifikasi menggunakan *validation set* tersebut dapat dilihat pada

Tabel 4.3: Hasil skenario conv1d

interval <i>input</i>	MAE lat	MAE long	MAE depth	MAE magnitude	MAE time
30 detik	0.122	0.169	0.060	0.100	0.069
20 detik	0.113	0.163	0.063	0.092	0.085
10 detik	0.141	0.173	0.059	0.109	0.081

Pada skenario menggunakan model LSTM. Pada input dengan interval 30 detik memiliki MAE latitudo sebesar 0.122, MAE longitudo sebesar 0,169, MAE depth sebesar 0.060, MAE magnitudo sebesar 0.100, dan MAE time sebesar 0,069. input dengan interval 20 memiliki MAE latitudo sebesar 0.113, MAE longitudo sebesar

0,163, MAE depth sebesar 0.063, MAE magnitudo sebesar 0.092, dan MAE time sebesar 0,085. Terakhir input dengan interval 10 memiliki MAE latitudo sebesar 0.141, MAE longitudo sebesar 0,173, MAE depth sebesar 0.059, MAE magnitudo sebesar 0.109,dan MAE time sebesar 0,081.

4.4.3 Skenario Conv1d dan LSTM

Skenario pertama menggunakan model dari Conv1d dan LSTM.Teknik ini akan diimplementasi untuk 182 stasiun sebagai data validasi. Hasil evaluasi model klasifikasi menggunakan *validation set* tersebut dapat dilihat pada

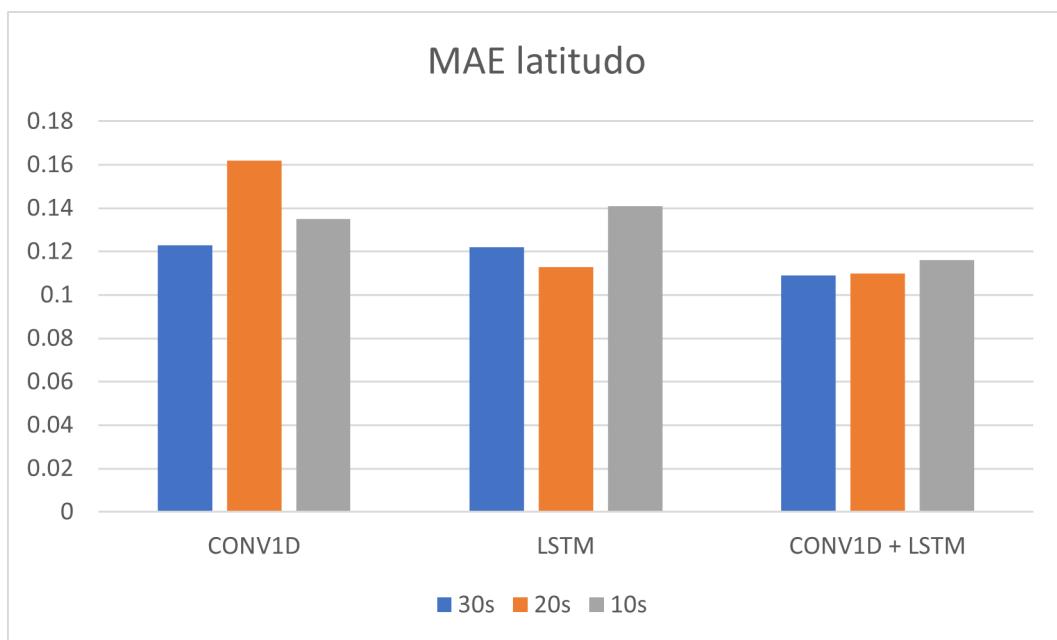
Tabel 4.4: Hasil skenario conv1d

interval <i>input</i>	MAE lat	MAE long	MAE depth	MAE magnitude	MAE time
30 detik	0.109	0.158	0.055	0.096	0.074
20 detik	0.110	0.140	0.061	0.088	0.069
10 detik	0.116	0.140	0.063	0.090	0.074

Pada skenario menggunakan model Conv1d dan LSTM. Pada input dengan interval 30 detik memiliki MAE latitudo sebesar 0.109, MAE longitudo sebesar 0,158, MAE depth sebesar 0.055, MAE magnitudo sebesar 0.096, dan MAE time sebesar 0,074. input dengan interval 20 memiliki MAE latitudo sebesar 0.110, MAE longitudo sebesar 0,140, MAE depth sebesar 0.061, MAE magnitudo sebesar 0.088, dan MAE time sebesar 0,069. Terakhir input dengan interval 10 memiliki MAE latitudo sebesar 0.116, MAE longitudo sebesar 0,140, MAE depth sebesar 0.063, MAE magnitudo sebesar 0.090,dan MAE time sebesar 0,074.

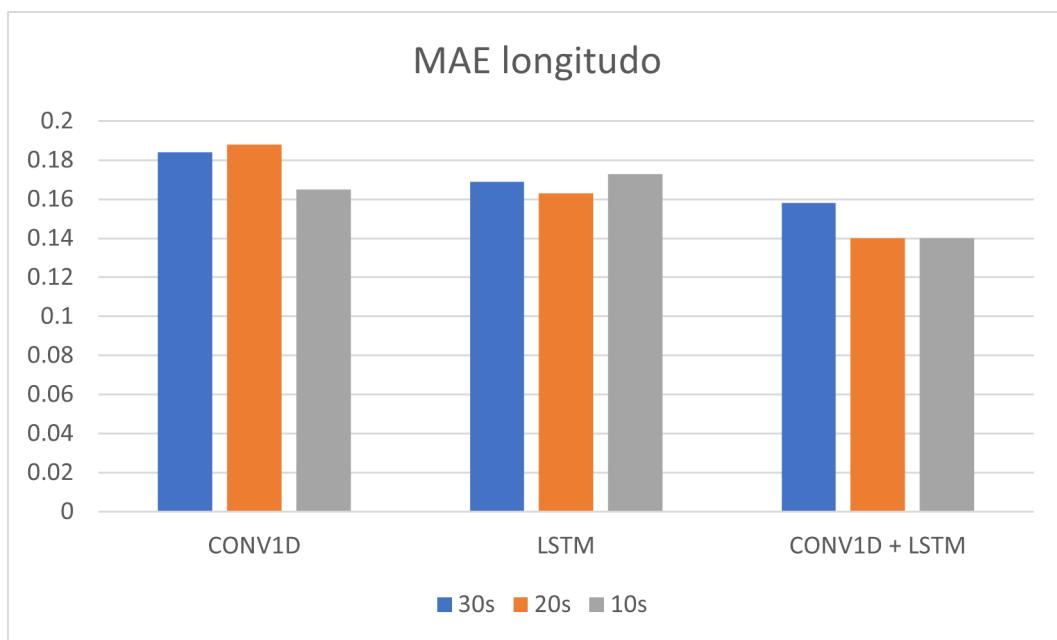
4.4.4 Perbandingan Semua Skenario

Analisis perbandingan dilakukan untuk metrik MAE. Analisis dilakukan berdasarkan model yang diimplementasi pada masing-masing skenario. Hasil perbandingan MAE setiap skenario dapat dilihat pada gambar berikut.



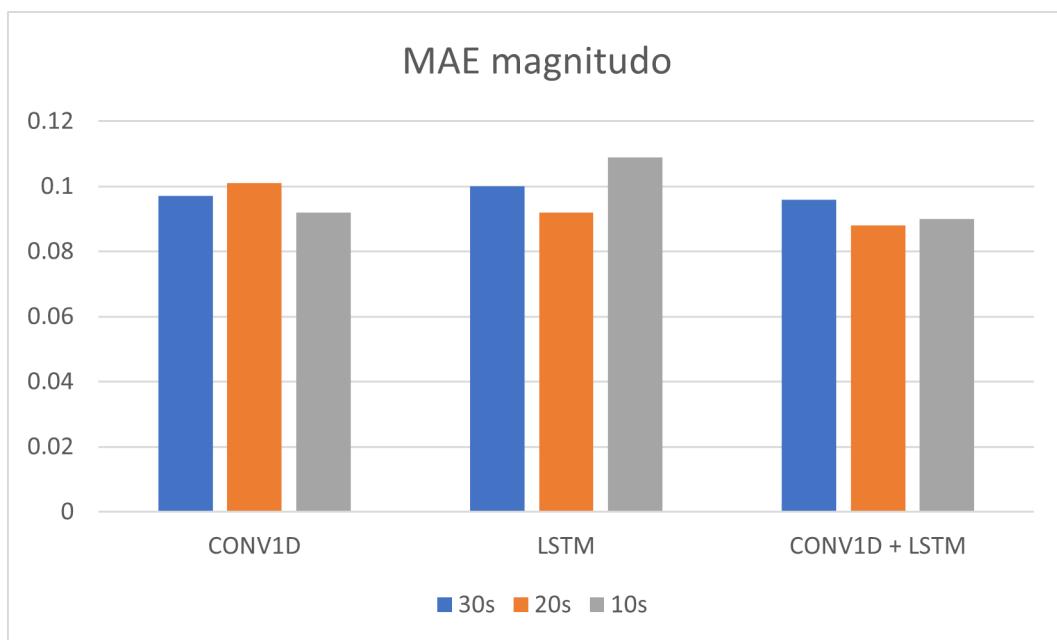
Gambar 4.5: Perbandingan MAE latitudo

Sumber: (Olahan Penulis, 2022)



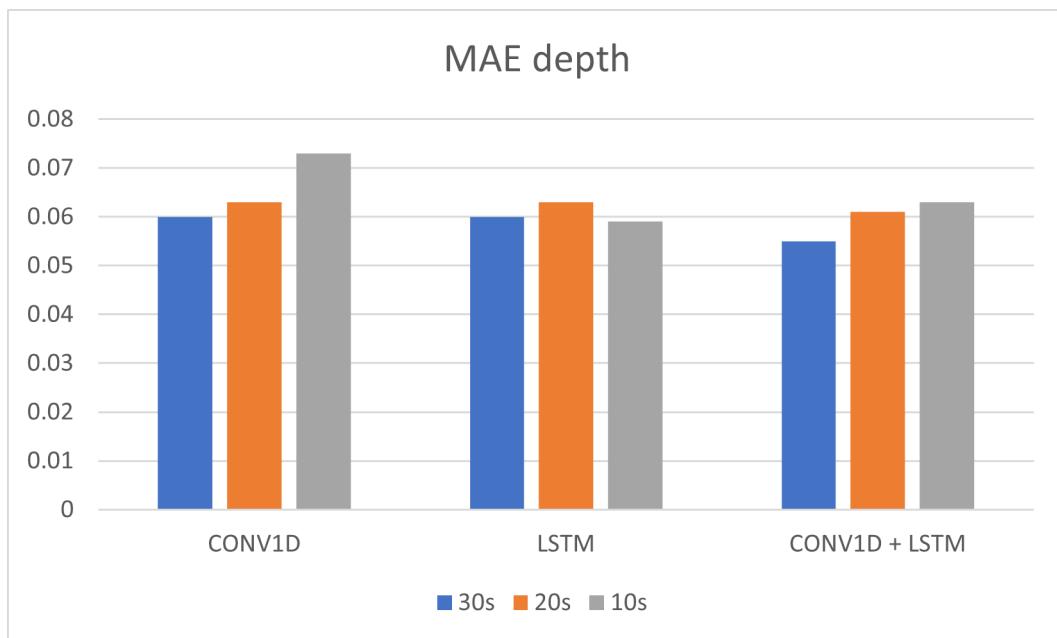
Gambar 4.6: Perbandingan MAE longitudo

Sumber: (Olahan Penulis, 2022)



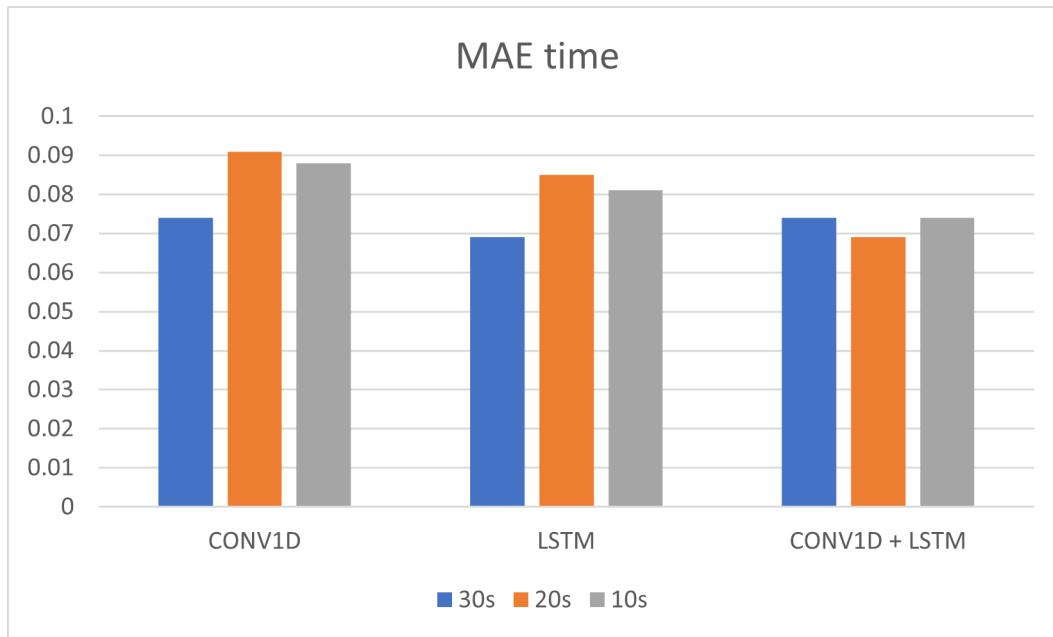
Gambar 4.7: Perbandingan MAE magnitudo

Sumber: (Olahan Penulis, 2022)



Gambar 4.8: Perbandingan MAE depth

Sumber: (Olahan Penulis, 2022)



Gambar 4.9: Perbandingan MAE time

Sumber: (Olahan Penulis, 2022)

Dari hasil berbagai skenario pada gambar sebelumnya. Terbukti bahwa interval waktu pada input tidak mempengaruhi hasil MAE, sehingga kita dapat menarik kesimpulan model dengan layer Conv1d dengan LSTM lebih baik dibanding dengan model lainnya.

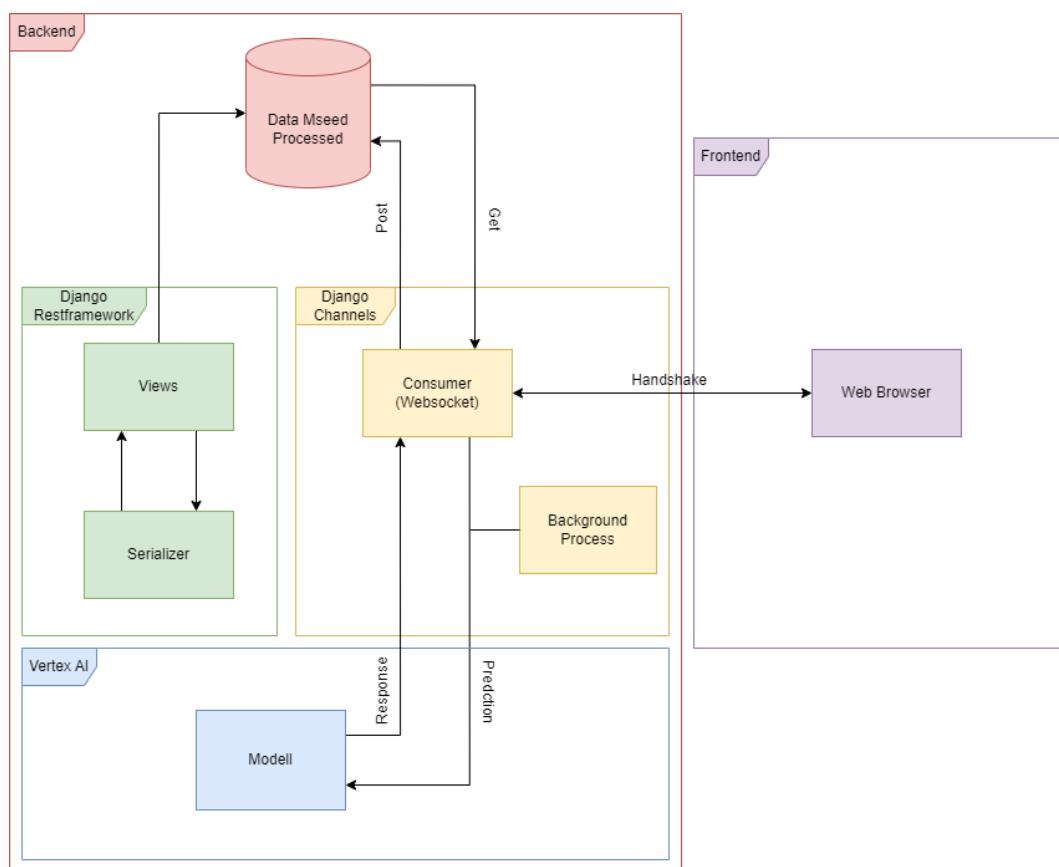
BAB 5

IMPLEMENTASI PEMROGRAMAN *BACK-END*

Bab ini menguraikan lebih detail terkait implementasi Pemrograman *Back-end* yang dilakukan oleh Aan Nur Wahidi.

5.1 Pemrograman *Back-End*

Pada bagian ini akan dijelaskan tahapan-tahapan pembuatan sistem *back-end* dalam menyediakan Websocket API dengan beberapa *endpoint* yang dapat di-request dengan parameter tertentu yang berfungsi untuk melakukan prediksi menggunakan model *machine learning* dalam proses simulasi sistem prediksi gempa menggunakan data mseed.



Gambar 5.1: Alur teknis sistem *back-end*

Sumber: (Olahan Penulis, 2022)

Alur teknis dari sistem back-end dijelaskan pada Gambar 5.1. Sebelumnya

data mseed yang mentah akan melalui proses ekstraksi data untuk disimpan ke basis data Firebase Realtime Database melalui Django Restframework agar dapat merepresentasikan data dalam bentuk JSON. Setelah data tersedia pada basis data, barulah proses sistem dimulai. Pertama *Request* akan diterima oleh *Consumer* menggunakan *connect* pada websocket yang kemudian akan memproses *handshaking* dari React.Js dengan parameter apa yang diakses seperti nama mseed dan stasiunnya. Saat mulai *handshaking* *Consumer* akan mulai pemanggilan data ke basis data dan akan dilakukan beberapa proses data pada *Background Process* termasuk dilakukannya prediksi *p-arrival*, kemudian saat memenuhi syarat input model baru akan mulai masuk pada prediksi besaran gempa pada model. Hasil prediksi akan dikirim dalam bentuk JSON oleh *Consumer* ke *frontend* dan disimpan pada basis data Firebase Realtime Database.

Sistem *backend* dibuat menggunakan Django Channels yang akan menggunakan sistem *asynchronous* dan Django Restframework sebagai pengekstraksi data mseed. Model *di-deploy* menggunakan Vertex AI dan Firebase Realtime Database sebagai basis data dari sistem *back-end*. Django digunakan sebagai *web framework* untuk menyediakan API karena terdapat library yang harus digunakan untuk dapat memproses data mseed yaitu obspy dengan bahasa pemrograman Python. Selain itu Django menyediakan berbagai macam *tools* dan *packages* seperti *templates* yang dapat langsung digunakan sehingga memudahkan dalam pembuatan *backend*. Basis data yang digunakan adalah Firebase Realtime Database karena membutuhkan basis data luar yang dapat menghapus atau menuliskan dengan mudah dan cepat secara langsung untuk penulis bereksperimen apakah data sudah dikemas dengan benar.

5.1.1 Perancangan Endpoint

Penulis merancang beberapa *endpoint* seperti Tabel 5.1. berdasarkan data yang dibutuhkan *frontend*. Terdapat tiga *endpoint* yang digunakan untuk visualisasi data seismik sedangkan prediksi akan dilakukan secara otomatis pada saat memenuhi kondisi tertentu. *endpoint* tersebut adalah GET /get_gmji/{nama mseed}, /get_jagi/{nama mseed}, dan /get_pwji/{nama mseed} merupakan pengambilan data seismik bergantung stasiun dan nama mseednya. Begitu pula dengan Django Restframework terdapat tiga yaitu GET /get_GMJI/{nama mseed}, /get_JAGI/{nama mseed}, dan /get_PWJI/{nama mseed} digunakan untuk melihat informasi utama dari mseednya.

Tabel 5.1: Rancangan *endpoint*

Method	Endpoint	Deskripsi
GET	/get_gmji_data/{nama mseed}	Mendapatkan data seismik mseed, kebutuhan frontend, dan hasil prediksi berdasarkan stasiun GMJI dan nama mseed
GET	/get_jagi_data/{nama mseed}	Mendapatkan data seismik mseed, kebutuhan frontend, dan hasil prediksi berdasarkan stasiun JAGI dan nama mseed
GET	/get_pwji_data/{nama mseed}	Mendapatkan data seismik mseed, kebutuhan frontend, dan hasil prediksi berdasarkan stasiun PWJI dan nama mseed
GET	/get_GMJI/{nama mseed}	Mendapatkan informasi mseed untuk stasiun GMJI berdasarkan nama mseednya
GET	/get_JAGI/{nama mseed}	Mendapatkan informasi mseed untuk stasiun JAGI berdasarkan nama mseednya
GET	/get_PWJI/{nama mseed}	Mendapatkan informasi mseed untuk stasiun PWJI berdasarkan nama mseednya

5.1.2 Inisialisasi Firebase SDK dan Vertex AI

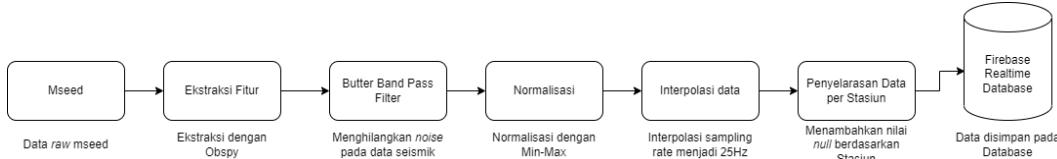
Firebase SDK dan Vertex AI perlu dihubungkan dahulu sebelum dapat digunakan oleh *backend*. Pada Firebase SDK dilakukan inisialisasi aplikasi yang bertujuan untuk menghubungkan *backend* dengan Firebase nya. Inisialisasi menggunakan Firebase SDK dengan nama *library* firebase_admin. Inisialisasi membutuhkan *pair key* dan *database url* untuk menghubungkan ke Firebase Realtime Database. Setelah inisialisasi selesai kemudian menentukan *database reference* yang mengacu pada nama tabel yang dibuat.

Sama halnya dengan Firebase SDK, Vertex AI juga diperlukan inisialisasi namun bedanya Vertex AI membutuhkan nama proyek, lokasi, *storage bucket*, dan *credential* dari Google Cloud. Lalu setelah inisialisasi selesai, selanjutnya menentukan alamat *endpoint* yang sudah ter-deploy model di dalamnya.

5.1.3 Implementasi REST API

Implementasi REST API dibuat menggunakan Django Restframework dalam penyimpanan informasi mseed ke basis data. REST API ini tidak digunakan aktif pada aplikasi melainkan hanya untuk ekstraksi data mseed dan digunakan

membantu penulis dalam berbagai uji coba pengemasan informasi mseed ke basis data. REST API terdiri dari tiga *endpoint* dengan proses komputasi yang sama yaitu mengembalikan informasi mseed berdasarkan stasiun dan nama mseed nya. Semua menggunakan *dictionary* Python yang kemudian diterima basis data dan mengembalikan JSON.



Gambar 5.2: Proses Ekstraksi Data Mseed untuk disimpan pada Basis Data

Sumber: (Olahan Penulis, 2022)

5.1.3.1 Ekstraksi Informasi Data Mseed

Terdapat proses lain sebelum disimpannya data ke basis data yaitu pertama mendapatkan informasi mseed dengan bantuan *library* Python obspy. Obspy memiliki semua fitur untuk mengekstraksi data mseed sehingga dapat dibaca dengan mudah. Mseed dibaca dengan *method* *read* dengan parameter nama *file*-nya. Karena mseed merupakan kumpulan data seperti csv, obspy akan mengembalikan data *array* dari *dictionary* sehingga harus diiterasikan terlebih dahulu. Kemudian untuk ekstraksi data dapat memanggil *key* dari data *trace* setiap mseednya dan mengambil *value*-nya. Pengemasan menggunakan nama mseed sebagai *key* utama lalu *trace* sebagai *array* dari informasi data dalam mseed tersebut. Berikut adalah struktur data mseed yang disimpan pada Firebase Realtime Database.



Gambar 5.3: Struktur data mseed pada Firebase Realtime Database

Sumber: (Olahan Penulis, 2022)



Gambar 5.4: Isi trace yang memuat informasi data mseed

Sumber: (Olahan Penulis, 2022)

5.1.3.2 Butter Bandpass Filter, Normalisasi, dan Interpolasi Data

Tahap selanjutnya adalah melakukan *bandpass filter*, normalisasi data, dan interpolasi data. *Bandpass filter* dilakukan untuk mengurangi *noise* pada data seismik mseed, kemudian normalisasi menggunakan *min-max* agar data seismik terbatas dari -1 hingga 1, hasil normalisasi dibulatkan menjadi 10 angka dibelakang koma untuk mengurangi angka yang berlebihan.

Interpolasi dilakukan dikarenakan data mseed memiliki *sampling rate* yang berbeda yaitu 20Hz dan 25Hz begitu pula dengan jumlah data seismiknya yang berbeda-beda. Interpolasi dilakukan dengan tujuan agar *sampling rate* setiap data *trace* sama yaitu 25Hz. Alasan tidak disamakan berdasarkan jumlah data seismik dari *trace*-nya karena data *trace* memiliki *starttime* dan *endtime* yang berbeda sehingga jumlah data seismiknya juga akan berbeda-beda meskipun *sampling rate*-nya sama.

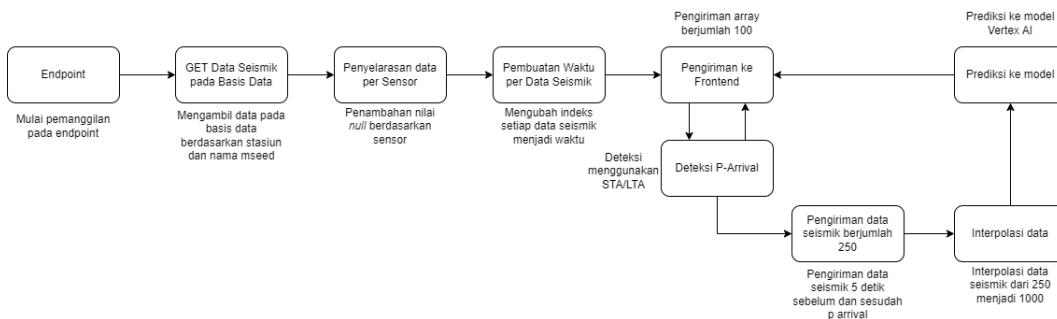
5.1.3.3 Penyalarasan Data Seismik Berdasarkan Waktu Mulai Stasiun

Proses ini merupakan kebutuhan *frontend* untuk menyelaraskan data seismik mseed untuk di-*stream* secara *realtime*. Setiap *trace* memiliki *starttime* yang berbeda sehingga untuk memudahkan visualiasi akan dilakukan penambahan data *null* sejumlah perbedaan *starttime* yang lebih dahulu dari setiap stasiun. *Starttime* akan diubah menjadi *timestamp* lalu akan diambil perbedaan waktu dari setiap stasiun. Kemudian perbedaan waktu tersebut akan dikalikan dengan *sampling rate* yang kini telah berubah menjadi 25Hz agar menjadi jumlah data poin yang harus ditambahkan. *Starttime* yang duluan akan disimpan pada basis data sebagai acuan pembuatan waktu dari setiap data poin seismik pada mseed. Setelah dilakukan

semua proses tersebut data kemudian akan disimpan ke basis data.

5.1.4 Implementasi Websocket API

Implementasi Websocket API dibuat dengan menggunakan Django untuk frameworknya dan Django Channels untuk websocketnya. Websocket API terdiri dari tiga *endpoint* yaitu mengembalikan data seismik mseed dan hasil prediksi dengan proses komputasi yang sama.



Gambar 5.5: Alur proses pengiriman data ke frontend dan *machine learning*

Sumber: (Olahan Penulis, 2022)

5.1.4.1 Mengambil Data Seismik pada Basis Data

Pengambilan Data Seismik pada Basis Data menggunakan referensi yang sudah dibuat sebelumnya pada inisialisasi Firebase SDK. Pengambilan dengan mengiterasikan *trace* pada mseed dengan nama yang dipakai sebagai parameter. Selanjutnya diambil data seismik mseed berdasarkan stasiun setiap sensor yaitu BHE, BHN, dan BHZ yang kemudian mengembalikan array dari ketiga data tersebut.

5.1.4.2 Penyelarasan Data Seismik Berdasarkan Waktu Mulai Sensor dan Pembuatan Waktu Data Seismik

Penyelarasan data seismik antara sensor (BHE, BHN, BHZ) merupakan proses yang sama pada setiap stasiun namun menggunakan *starttime* dari sensor yang mulai lebih dahulu. Kemudian untuk keperluan visualisasi diperlukannya perubahan *array index* dari data seismik menjadi waktu berdasarkan *starttime* yang disimpan sebelumnya. Proses tersebut dilakukan dengan menggunakan jumlah data sensor yang terbesar kemudian *starttime* akan ditambahkan dengan kecepatan setiap data poinnya yaitu $\frac{1}{25}$, setelah selesai barulah mulai pengiriman data seismiknya.

5.1.4.3 Pengiriman Data Seismik ke *Frontend* dan ke Machine Learning

Data yang dikirimkan ke *frontend* berupa *array* dengan *index* sebesar 100 menyesuaikan permintaan *frontend*, dengan ketentuan data akan dikirim dengan pertambahan satu *index* saja, sehingga skenario pengiriman akan menjadi *array* dari *index* 0 sampai 99, kemudian 1 sampai 100, lalu 2 sampai 101 dan seterusnya hingga data selesai. Sebelum data berjumlah 100, akan dikirimkan *array* dengan nilai *null* untuk setiap *key*-nya agar pengiriman sesuai dengan waktunya.

Selanjutnya pada saat data terkirim sejumlah 750 data atau sama dengan 30 detik sejak websocket mengirimkan data, data akan mulai dilakukan pendekripsi *p-arrival* dengan *library* Python obspy yang menggunakan algoritma STA/LTA. Lalu saat *p-arrival* ditemukan akan diberikan input model ke *machine learning* dengan ketentuan lima detik sebelum dan lima detik sesudah *p-arrival*. Terdapat interpolasi data saat mengirim data ke *machine learning* yaitu mengubah data yang berjumlah 250 menjadi 1000 data. Seluruh data tersebut akan dikirimkan secara bersamaan yaitu data seismik, *p-arrival*, dan hasil prediksi model. Apabila belum ditemukan atau dikomputasi, akan diberikan nilai *null* terlebih dahulu.

Tabel 5.2: Data yang dikirimkan ke *frontend*

Key	Value	Tipe Data
E_Data	Data seismik untuk sensor BHE	Float
N_Data	Data seismik untuk sensor BHN	Float
Z_Data	Data seismik untuk sensor BHZ	Float
data_prediction	Data prediksi terdiri dari latitude, longitude, depth, magnitude, dan time	<i>object</i> JSON
p_Arrival	<i>p-arrival</i> pengembalian angka dari <i>index</i> data seismik	Integer
sampling_rate	<i>sampling rate</i> dari data mseed	Integer
time	Waktu yang ditunjukkan berdasarkan starttime pada setiap data seismik	String
x	<i>Index array</i> yang menunjukkan pada setiap data seismik	Integer

5.1.5 Deployment Websocket API

Websocket API akan di-deploy ke Google Cloud dengan menggunakan konfigurasi pada Dockerfile. Sebelum di-deploy penulis membuat *container image* pada lokal docker untuk mengurangi masalah sebelum di-deploy ke Google Cloud. Penulis

juga sudah menyiapkan *project* yang sudah teraktivasi *billing* sehingga dapat memanfaatkan berbagai macam fitur yang disediakan Google Cloud. Kemudian penulis menyiapkan file docker-compose.yml yang berisi untuk melakukan *build* lalu mengeksekusi Dockerfile dan menjalankan perintah untuk menjalankan Django. Karena Channels menggunakan Daphne yang merupakan *protocol server* untuk menjalankan WebSocket, penulis mengganti *command* menjalankan Django menggunakan Daphne untuk uji coba sebelum melakukan *deployment*.

Setelah semua sudah berjalan dengan baik yaitu *build container image* hingga dapat berjalan pada lokal, lalu dilakukannya *deployment* ke Cloud Run menggunakan Google Cloud CLI pada terminal lokal. Sebelumnya penulis harus memiliki Google Cloud CLI terlebih dahulu untuk dapat menjalankannya dan konfigurasi akun serta *project* yang digunakan. Lalu melakukan perintah *deploy* secara otomatis menggunakan perintah "gcloud run deploy" pada *directory* lokal programnya.

5.1.6 Implementasi Evaluasi Pemrograman Back-End

Pada sistem *back-end* yang dibuat penulis akan melakukan evaluasi secara manual mencatat waktu dan *response code* dengan menuliskannya pada program *backend*. Uji coba dilakukan ketika *backend* sudah di-*deploy* pada Cloud Run dengan menyimpan hasilnya pada Google Cloud Storage Bucket. Pengujian yang dilakukan adalah pengukuran waktu yang dibutuhkan pada komputasi-komputasi WebSocket API dan waktu prediksi STA/LTA dan Multi Target Regression (MTR) yang dibutuhkan *machine learning* pada Vertex AI sampai hasil prediksi dikembalikan ke *backend*.

5.2 Hasil Uji Coba Pemrograman *Back-end*

Pada bagian ini, penulis akan melakukan evaluasi terhadap websocket apakah dapat terhubung tanpa ada masalah, kemudian pengukuran komputasi sebelum websocket mengirimkan data ke *frontend*. Terakhir akan dihitung waktu yang dibutuhkan *machine learning* dari *backend* untuk memprediksi data seismik termasuk interpolasi datanya.

5.2.1 Websocket API

Penulis akan menguji setiap *endpoint* yang sudah dibuat sebelumnya pada tahapan implementasi. Pengujian pertama adalah pengecekan bahwa websocket dapat

terhubung tanpa ada masalah dan kedua adalah menghitung komputasi sebelum websocket mengirim data ke *frontend*.

5.2.1.1 Response Code

Pengujian pertama adalah uji coba *response code* yang didapat pada tiga endpoint berdasarkan tiga stasiun yaitu GMJI, JAGI, dan PWJI.

Tabel 5.3: Hasil Evaluasi Websocket API

Endpoint	Code Request	Response
GET /get_gmji_data/{nama_mseed}	200 Success	[{"E_data": -3.2506e-06, "N_data": -5.04093e-05, "Z_data": 0, "p_Arrival": 0, "sampling_rate": 25, "time": "30:05", "x": 11, "data_prediction": {"latitude": null, "longitude": null, "depth": null, "magnitude": null, "time": null}}, ...]
GET /get_jagi_data/{nama_mseed}	200 Success	[{"E_data": 0, "N_data": -8.10429e-05, "Z_data": 1.0887e-06, "p_Arrival": 0, "sampling_rate": 25, "time": "30:11", "x": 148, "data_prediction": {"latitude": null, "longitude": null, "depth": null, "magnitude": null, "time": null}}, ...]
GET /get_pwji_data/{nama_mseed}	200 Successss	[{"E_data": 0, "N_data": 9.90971e-05, "Z_data": 0, "p_Arrival": 0, "sampling_rate": 25, "time": "30:07", "x": 67, "data_prediction": {"latitude": null, "longitude": null, "depth": null, "magnitude": null, "time": null}}, ...]

Hasi evaluasi pada Tabel 5.3. menunjukkan bahwa masing-masing endpoint tidak memiliki kendala dan mengembalikan *response code* 200 yaitu *success* dengan nama mseed yang tersedia.

5.2.1.2 Komputasi Websocket API

Pengujian komputasi Websocket API dilakukan sebelum mengirim data ke *frontend* sehingga ketika *frontend* memulai untuk menjalankan suatu mseed, akan ada beberapa jeda waktu hingga data terkirim. Pengujian dilakukan dengan ketentuan melakukan *request* pada setiap *endpoint* dengan empat mseed yang berbeda.

Tabel 5.4: Hasil Evaluasi Komputasi Websocket API Sebelum Proses *Machine Learning*

Nama Mseed	Nama Stasiun	Waktu Komputasi
20090118_064750	GMJI	0.22 s
20090118_064750	JAGI	0.27 s
20090118_064750	PWJI	0.26 s
20090119_211141	GMJI	0.30 s
20090119_211141	JAGI	0.26 s
20090119_211141	PWJI	0.24 s
20120915_163224	GMJI	0.23 s
20120915_163224	JAGI	0.26 s
20120915_163224	PWJI	0.36 s
20100208_112154	GMJI	0.30 s
20100208_112154	JAGI	0.25 s
20100208_112154	PWJI	0.27 s

Hasil evaluasi komputasi Websocket API pada Tabel 5.4. dengan rata-rata dari seluruh tabel adalah 0.26 detik (menggunakan angka asli) yaitu jeda waktu sebelum *backend* benar-benar mengirimkan data ke *frontend*.

5.2.2 Evaluasi Seluruh Proses *Machine Learning* melalui *Backend*

Pengujian terhadap *p-arrival* dilakukan terus menerus dari awal data dikirimkan ke *frontend* hingga ditemukan karena menggunakan *method* pada *library* obspy yaitu STA/LTA, namun untuk prediksi MTR dilakukan satu kali saja pada setiap stasiun.

Tabel 5.5: Hasil Evaluasi Pada Proses Deteksi *p-arrival*

Nama Komputasi	Waktu
STA/LTA	0.002 s

Selanjutnya pengujian pada waktu prediksi model *machine learning* pada Vertex AI. Pengujian dilakukan pada saat *backend* telah menemukan *p-arrival* pada data seismik. *Backend* akan memulai *request* ke *machine learning* untuk melakukan proses prediksi MTR. Berikut adalah hasil dari seluruh pengujian.

Tabel 5.6: Hasil Evaluasi Pada Proses *Machine Learning*

Nama Mseed	Nama Stasiun	Waktu Prediksi
20090118_064750	GMJI	1.21 s
20090118_064750	JAGI	1.25 s
20090118_064750	PWJI	1.18 s
20090119_211141	GMJI	1.40 s
20090119_211141	JAGI	1.19 s
20090119_211141	PWJI	1.22 s
20120915_163224	GMJI	1.26 s
20120915_163224	JAGI	1.03 s
20120915_163224	PWJI	1.24 s
20100208_112154	GMJI	1.26 s
20100208_112154	JAGI	1.21 s
20100208_112154	PWJI	1.24 s

Berdasarkan Tabel 5.6. rata-rata dari percobaan menggunakan empat mseed yang berbeda adalah 1.23 s. Sehingga waktu yang dibutuhkan pada backend untuk komputasi *machine learning* adalah 1.232 detik hingga hasil dapat ditampilkan pada *frontend*.

5.2.3 Evaluasi Seluruh Komputasi Websocket API

Hasil evaluasi dari seluruh komputasi Websocket API dapat dilihat pada tabel 5.7. Berdasarkan tabel tersebut bahwa total waktu komputasi pada Websocket API yang digunakan untuk pengiriman data adalah 6.492 detik hingga hasil prediksi dikirim ke *frontend*.

Tabel 5.7: Hasil Evaluasi Seluruh Komputasi Websocket API

Parameter	Waktu Rata-rata
Waktu Komputasi untuk Visualisasi	0.26 s
Waktu Algoritma STA/LTA	0.002 s
Waktu Prediksi MTR	1.23 s
Waktu Mulai Prediksi MTR	5 s
Total	6.492 s

Menurut referensi paper utama kami yaitu Wibowo et al. (2022) rata-rata jarak waktu *p-arrival* hingga *s-arrival* adalah 21.53 detik. Sehingga apabila kita lakukan pengurangan dari total yaitu 6.5 detik (hasil dibulatkan) akan menghasilkan kurang lebih 15 detik. Waktu ini merupakan waktu yang dibutuhkan sistem *backend* untuk dapat mengirim hasil prediksi hiposenter gempa bumi ke *frontend* dari mulai ditemukannya *p-arrival* pada sistem.

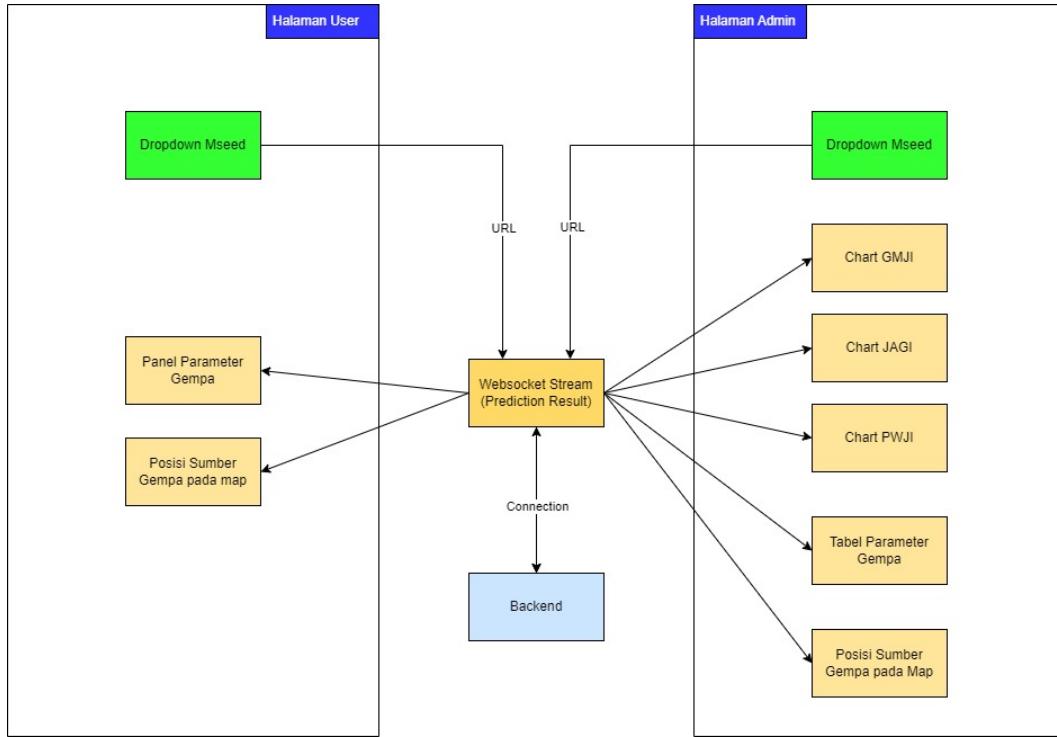
BAB 6

IMPLEMENTASI PEMROGRAMAN *FRONT-END* DAN VISUALISASI DATA

Bab ini menguraikan lebih detail terkait implementasi, hasil, dan evaluasi Pemrograman *Front-end* dan Visualisasi Data yang dilakukan oleh Farrel Alfarabi Saleh

6.1 Sistem Prediksi

Bagian ini menjelaskan teknik pemrograman front-end yang digunakan dalam penelitian ini. Gambar 6.1 menggambarkan alur teknis implementasi sistem dari sisi pemrograman front-end. Penulis menggunakan Javascript sebagai bahasa pemrograman, React.js sebagai framework pemrograman front-end, HTML untuk menstruktur halaman website, dan CSS sebagai alat *styling*. Penulis menggunakan React.js sebagai framework karena membangun UI dan aplikasi web menggunakan React.js lebih cepat dan memerlukan lebih sedikit kode dibanding vanilla javascript. Sistem prediksi ini memiliki dua halaman utama yakni halaman pengguna dan halaman admin.



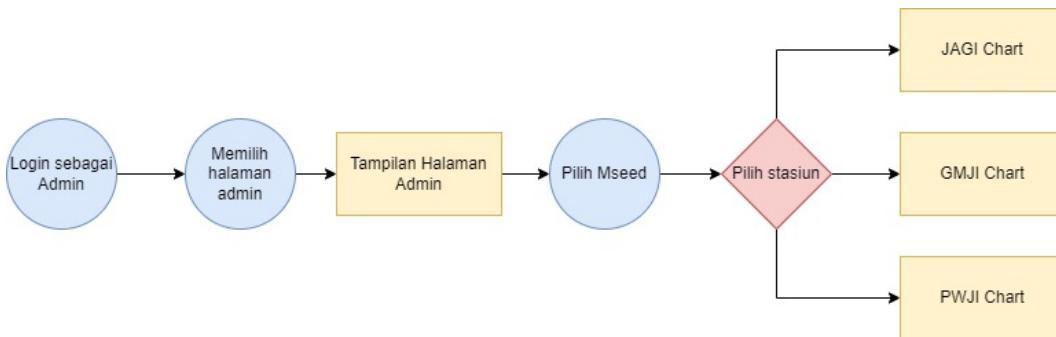
Gambar 6.1: Alur teknis pemrograman *front-end*

Sumber: (Olahan Penulis, 2022)

6.1.1 Pembuatan *User Interface* Halaman Admin

Penulis membuat halaman ini dengan tujuan untuk memberi admin pengawasan terhadap stasiun gempa dan visualisasi data *realtime* dalam bentuk *dynamic line chart* yang didapat dari stasiun sensor seismik, namun karena stasiun-stasiun ini belum memiliki API yang dapat mengirim data secara *realtime*, data yang ditunjukkan berasal dari sebuah database yang dikirim secara periodik melalui websocket. Dalam penggunaan *dynamic line chart* terdapat beberapa tahap yang dapat dijalankan pengguna yaitu memilih data mseed yang ingin disimulasikan lalu memilih grafik dari stasiun yang ingin ditampilkan. Selain *dynamic line chart*, halaman ini juga menampilkan lokasi dari stasiun pendekripsi getaran serta lokasi sumber gempa secara *realtime* setelah mendapat data dari backend pada sebuah peta interaktif. Untuk mempermudah pengguna, penulis membuat sebuah tabel parameter gempa yang berisi tabel informasi dari hasil algoritma prediksi yang termasuk waktu terdeteksinya *p-arrival*, magnitude gempa, kedalaman, nama stasiun pendekripsi *p-arrival*, dan lokasi stasiun tersebut. Halaman ini juga menampilkan performa dari algoritma prediksi dengan menghitung waktu yang diperlukan data untuk berjalan dari database hingga terlihat pada *user interface*. Halaman ini hanya dapat diakses oleh admin karena terdapat informasi yang

hanya berguna untuk admin atau operator untuk memonitor sistem prediksi. Alur penggunaan halaman ini dapat dilihat pada gambar 6.2



Gambar 6.2: Alur penggunaan halaman admin

Sumber: (Olahan Penulis, 2022)

6.1.2 Pembuatan *User Interface* Halaman Pengguna

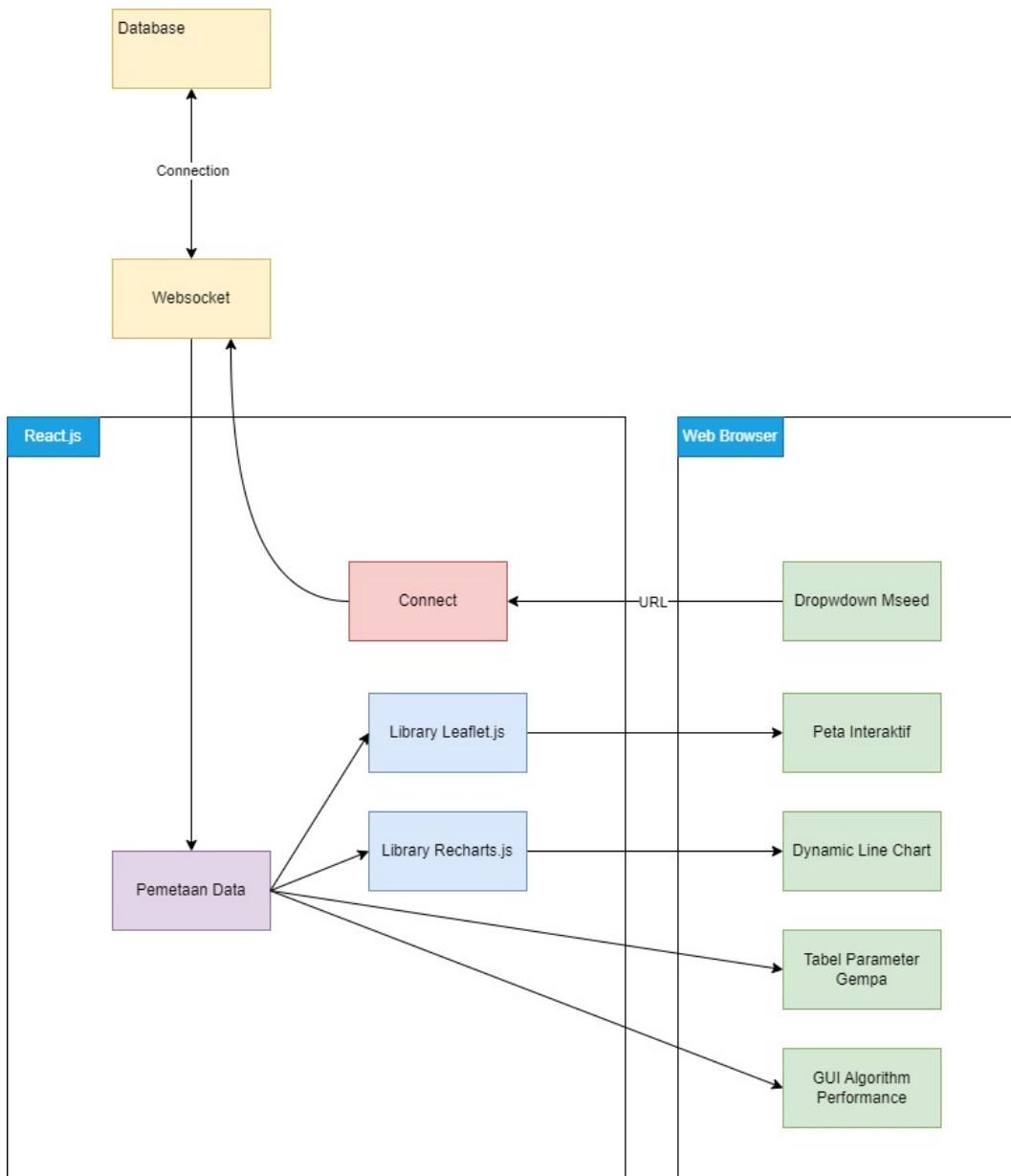
Halaman ini merupakan halaman awal yang akan ditampilkan website. Sebagai halaman utama dari website, halaman ini ditujukan untuk pengguna umum sebagai sebuah wadah peringatan dini gempa bumi. Halaman ini memiliki antarmuka yang serupa dengan dengan halaman admin namun hanya menampilkan informasi yang esensial untuk menginformasikan pengguna terhadap bencana gempa bumi. Halaman ini memiliki alur sederhana yaitu memilih data mseed yang ingin disimulasikan lalu sistem akan menunjukkan lokasi sumber gempa pada peta interaktif dan memberikan informasi dari hasil algortima prediksi sesaat terdeteksinya *p-arrival*. Halaman ini juga menampilkan beberapa informasi tambahan seperti hitungan mundur hingga kejadian gempa, area yang dapat terpengaruh oleh gempa bumi, serta sejarah kejadian gempa bumi dalam waktu dekat.

6.1.3 Pembuatan *User Interface* Halaman Login Admin

Halaman ini bertujuan untuk mengautentifikasi pengguna admin agar dapat mengakses halaman admin. Halaman ini tidak diperlukan bagi pengguna yang tidak memiliki peran admin. Untuk menjadi sebuah admin, pengguna diperlukan untuk mengontak pengelola website agar dapat didaftarkan sebagai admin pada database user secara manual.

6.2 Visualisasi Data

Subbab ini akan menjelaskan secara teknis implementasi dari visualisasi data pada penelitian yang direpresentasikan pada Gambar 6.3. Pada pemrograman visualisasi data pengguna menggunakan *library* React.js untuk membangun *user interface* aplikasi, *library* recharts.js untuk membangun visualisasi *dynamic line chart*, dan *library* leaflet.js untuk menampilkan peta interaktif. Data yang ditampilkan berasal dari *endpoint back-end*. Pada penelitian ini visualisasi data ditampilkan menggunakan peta interaktif dan *dynamic line chart*. Pembangunan visualisasi data dapat dibagi menjadi 2 tahap yaitu pengambilan data dan pemetaan data dalam bentuk visualisasi.



Gambar 6.3: Alur teknis visualisasi data

Sumber: (Olahan Penulis, 2022)

6.2.1 Alur Pengolahan Data Hingga Visualisasi

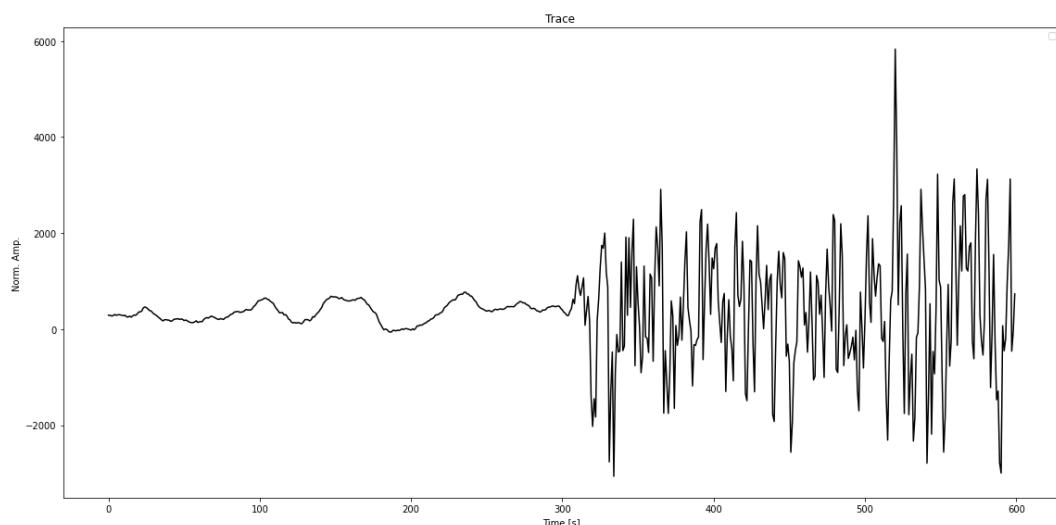
Data didapat dari BMKG dari seluruh stasiun di Pulau Jawa dalam rentang waktu kejadian tahun 2009 sampai 2017. Terdapat dua format data yang diberikan pertama mseed (miniSEED) yang merupakan bagian dari format data SEED yang utamanya berisi data seismik dan informasi singkat dari stasiun dan kedua Pick merupakan format data yang digunakan BMKG setelah dilakukan analisis secara manual terhadap mseed. Ekstraksi data mseed dilakukan dengan bantuan library Python obspy. Obspy memiliki semua fitur untuk mengekstraksi data mseed berikut

merupakan hasil ekstrasi dari mseed.

Tabel 6.1: hasil fitur

No	Fitur	deskripsi
1	id	Nama file mseed
2	starttime	Waktu dimulainya rekaman seismik
3	endtime	Waktu selesainya rekaman seismik
4	npts	Jumlah banyaknya data seismik
5	<i>E_WAVEFORM</i>	Data seismik pada sensor E yang berupa <i>timeseries</i>
6	<i>N_waveform</i>	Data seismik pada sensor N yang berupa <i>timeseries</i>
7	<i>Z_waveform</i>	Data seismik pada sensor Z yang berupa <i>timeseries</i>
8	<i>p-arrival_time</i>	Waktu <i>p-arrival</i> tiba dalam bentuk data ke-n
9	<i>s_arrival_time</i>	Waktu <i>p-arrival</i> tiba dalam bentuk data ke-n
8	<i>station</i>	Stasiun gempa yang beroperasi
9	<i>sampling_rate</i>	Banyak data perdetiknya

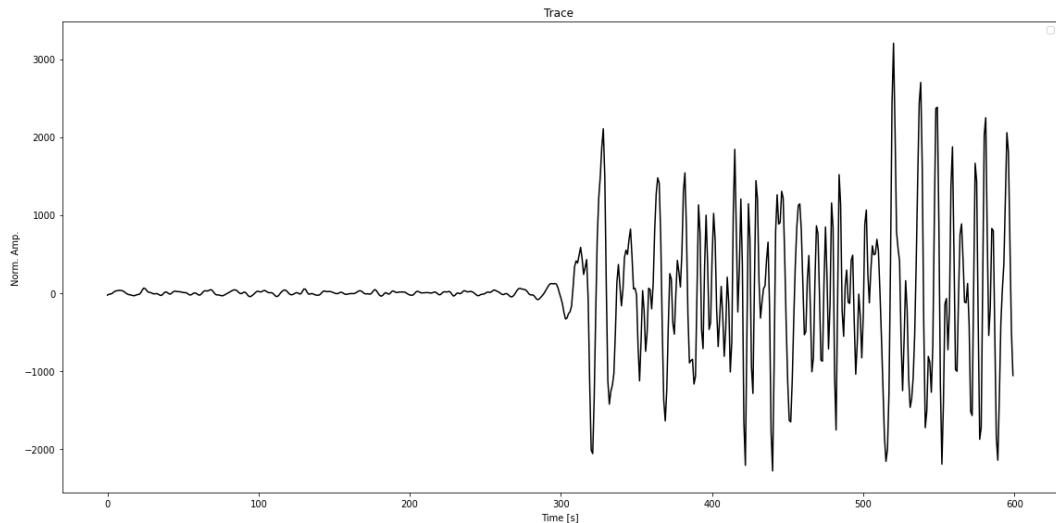
Setelah dilakukan ekstrasi data, data pada sensor E,N,dan Z akan ditampung kedalam 3 variabel, berikut merupakan ilustrasi data dalam bentuk *time series*.



Gambar 6.4: ilustrasi data *timeseries*

Sumber: (Olahan Penulis, 2022)

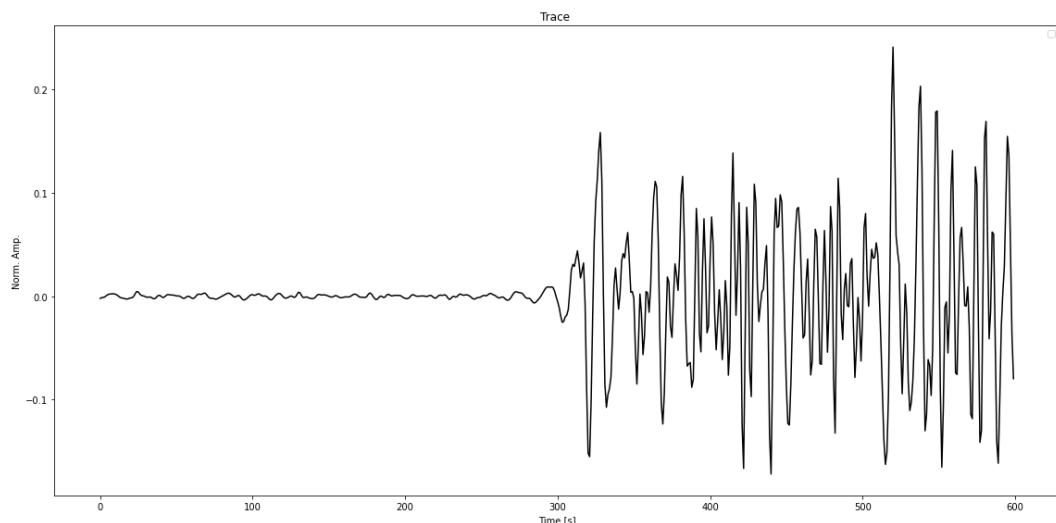
Setelah data diperoleh, data akan dilakukan filter menggunakan *butterworth bandpass* agar noise pada data dapat disamarkan sehingga tidak terjadi alarm palsu.berikut merupakan hasil ilustrasi dari data setelah difilter.



Gambar 6.5: ilustrasi data *timeseries* setelah di *Butterworth bandpass filter*

Sumber: (Olahan Penulis, 2022)

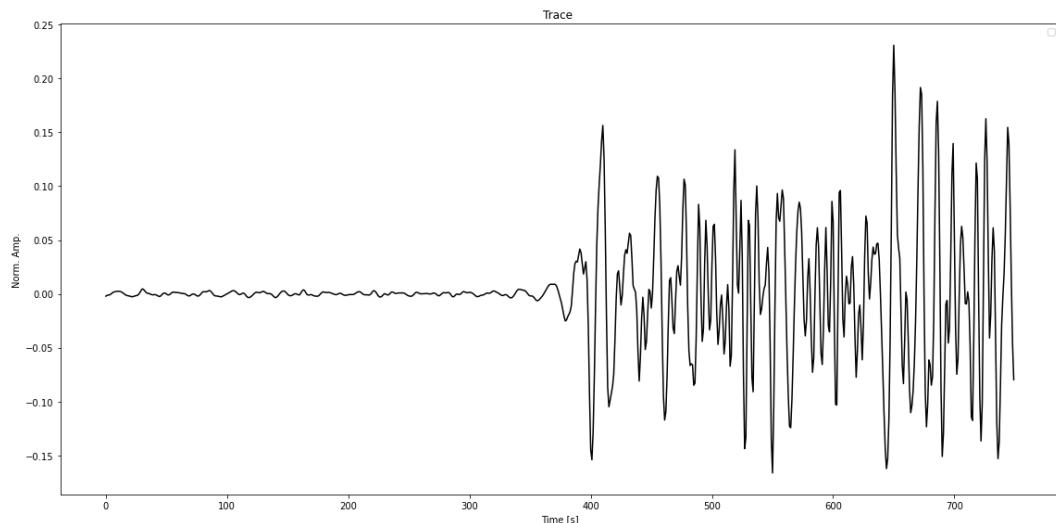
Setelah itu data dinormalisasi terhadap nilai maksimal dan nilai minimal amplitudo. Hal ini bertujuan agar data dapat disimplifikasi nilainya tanpa ada kualitas data yang berkurang. berikut merupakan hasil ilustarsi dari data setelah dinormalisasi.



Gambar 6.6: ilustrasi data *timeseries* setelah di dinormalisasi

Sumber: (Olahan Penulis, 2022)

Terakhir data lalu diinterpolasi nilai dari data *time series* ketiga sensor dapat berjalan dalam waktu yang sama, dikarenakan data ketiga sensor tersebut memiliki nilai *sampling rate* yang berbeda. Berikut merupakan hasil ilustrasi data dari *sampling rate* 20hz sampai 25hz.



Gambar 6.7: ilustrasi data *timeseries* setelah diinterpolasi

Sumber: (Olahan Penulis, 2022)

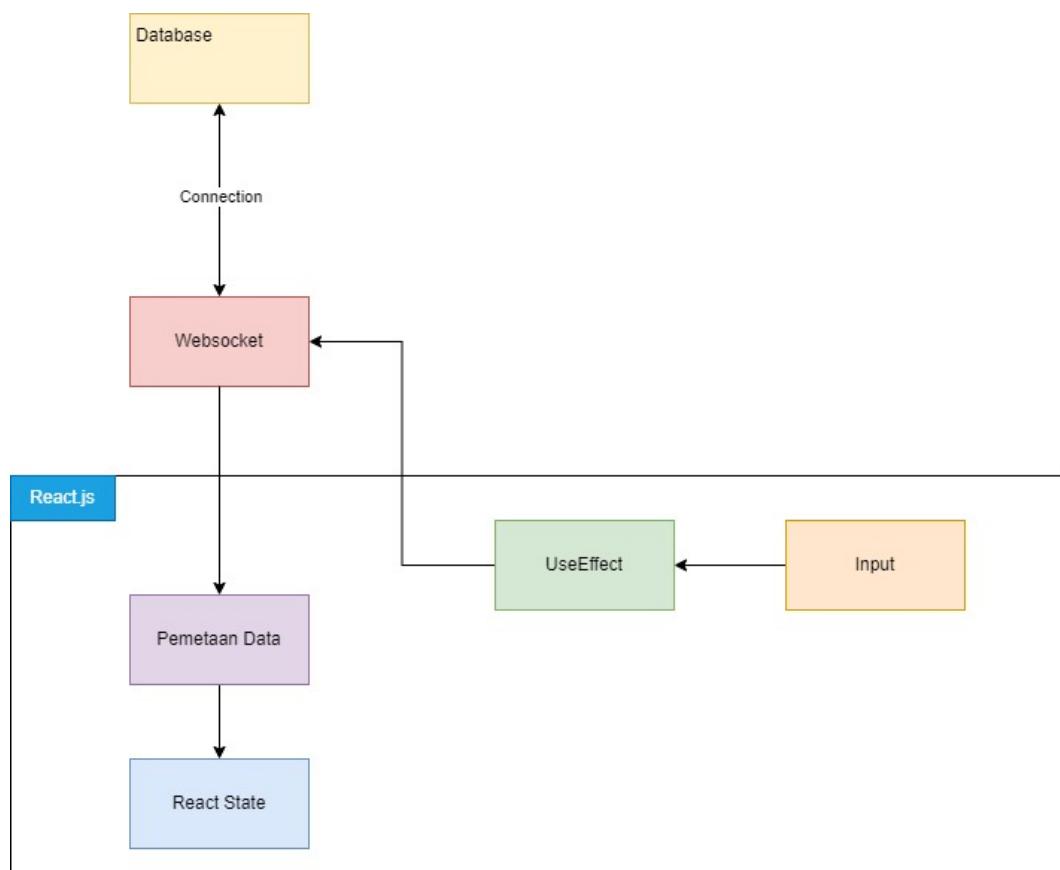
Setelah data seismik melalui semua proses tersebut data kemudian disimpan pada basis data. Pada *backend* terdapat proses untuk kebutuhan visualisasi data sebelum data dikirimkan yaitu penyelarasan data seismik dan pembuatan waktu data seismik. Lalu data dalam bentuk JSON berisi hasil fitur dikirim ke *frontend* melalui *websocket* secara *realtime*. Berikut adalah data yang dikirimkan ke *frontend* melalui *websocket*.

Tabel 6.2: Data yang dikirimkan ke *frontend*

Key	Value	Tipe Data
E_Data	Data seismik untuk sensor BHE	Float
N_Data	Data seismik untuk sensor BHN	Float
Z_Data	Data seismik untuk sensor BHZ	Float
data_prediction	Data prediksi terdiri dari latitude, longitude, depth, magnitude, dan time	object JSON
p_-arrival	<i>p-arrival</i> pengembalian angka dari <i>index</i> data seismik	Integer
sampling_rate	<i>sampling rate</i> dari data mseed	Integer
time	Waktu yang ditunjukkan berdasarkan starttime pada setiap data seismik	String
x	<i>Index array</i> yang menunjukkan pada setiap data seismik	Integer

6.2.2 Pengambilan Data dari Back-end

Pengambilan dari data yang akan dipetakan ke *dynamic line chart* dan peta interaktif berasal dari endpoint backend yang dikirim melalui websocket API. Data yang dikirim oleh backend merupakan data tipe JSON (*JavaScript Object Notation*). Lalu menggunakan input dari user nama mseed terpilih akan digunakan sebagai address URL untuk menyambungkan *front-end* pada *websocket*. Pengambilan data dilaksanakan pada sebuah *parent component* dimana komponen ini akan membuat hubungan ke 3 aliran data websocket berdasarkan 3 stasiun yang kami gunakan sebagai sumber data yaitu stasiun GMJI, JAGI, dan PWJI. Aliran data ini kemudian diolah dan dibagi ke *child component* yang akan memetakan data dalam suatu bentuk visualisasi. Alur pengambilan data dapat dilihat pada Gambar 6.4.



Gambar 6.8: Alur pengambilan data

Sumber: (Olahan Penulis, 2022)

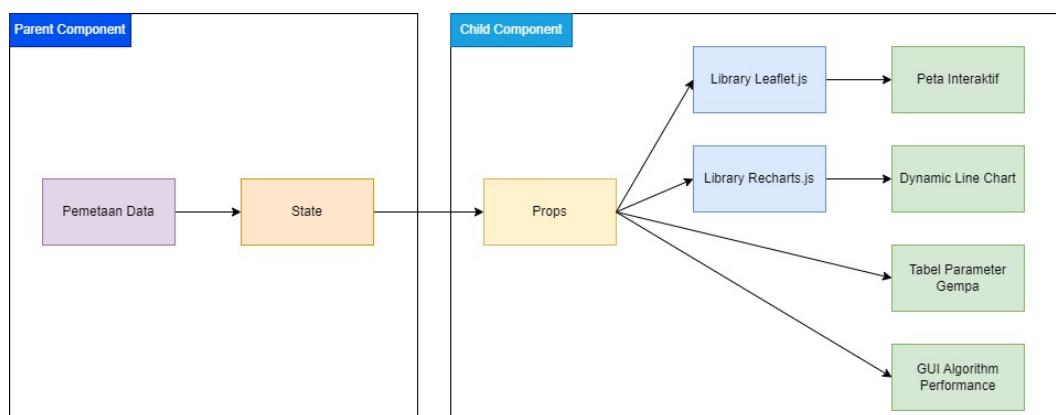
Dynamic line chart menampilkan tiga arus data seismik yang didapat melalui pengiriman *websocket* secara periodik serta titik dimana terdeteksinya p-arrival. Pada setiap saat, grafik ini akan menampilkan 100 data npts yang bergerak ke arah kiri secara periodik, maka data yang diterima oleh *frontend* merupakan *array*

berisi 100 obyek data dimana obyek pada index terakhir merupakan data terbaru. Grafik menampilkan 100 data setiap saat agar dapat terlihat lebih nyaman bagi pengguna. Pergerakan data pada grafik akan memiliki kecepatan yang sesuai dengan frekuensi setiap stasiun seismik sehingga dapat mensimulasikan menerima secara *realtime* dari stasiun-stasiun ini. *Dynamic line chart* ini juga dapat mengubah sumber mseed secara dinamis dengan memutuskan koneksi pada *socket* terdahulu lalu menyambungkan koneksi baru berdasarkan pilihan sumber file mseed.

Peta interaktif menampilkan lokasi dari tiga stasiun seismik serta lokasi sumber gempa bumi berdasarkan informasi yang dihasilkan oleh algoritma prediksi. Data lokasi dari sumber ini didapatkan melalui tiga koneksi websocket yang sama dengan digunakan oleh *dynamic line chart*. Selain lokasi sumber gempa, peta juga akan menggunakan data magnitude dan kedalaman gempa bumi untuk ditampilkan pada suatu *pop-up* yang tercantum pada lingkaran sumber gempa.

6.2.3 Pemetaan Data dalam Bentuk Visualisasi

Data yang didapatkan melalui endpoint Websocket API dalam bentuk JSON perlu dipetakan agar dapat divisualisasikan menjadi *line chart* menggunakan *library Recharts.js*. Data ini juga akan digunakan untuk memanipulasi peta interaktif *leaflet.js*. Setelah data diterima dan disimpan pada sebuah state pada *parent component*, data lalu dikirim ke *child component* untuk pemetaan. Alur pemetaan data dapat dilihat pada Gambar 6.5. Pada komponen *linechart* data yang diambil dari datastream websocket dapat diakses menggunakan obyek *props* yang merupakan sebuah obyek yang diwariskan oleh *parent component* kepada sebuah *child component*.

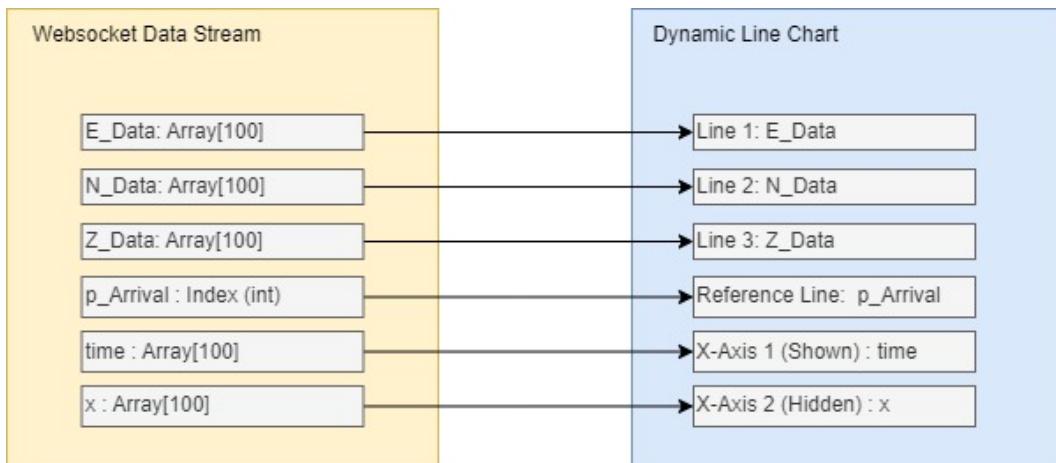


Gambar 6.9: Alur pemetaan data

Sumber: (Olahan Penulis, 2022)

Agar dapat membangun sebuah visualisasi data seismik secara *realtime* dalam

bentuk *dynamic line chart*, komponen ini membutuhkan data npts dari tiga sensor dari setiap stasiun, data waktu setiap npts terekam oleh stasiun, dan waktu saat ditemukan *p-arrival* oleh algoritma prediksi. Data ini lalu disimpan pada sebuah state pada komponen linechart dan visualisasi grafik akan *update* setiap kali state yang menyimpan data ini berubah. Pemetaan data pada *dynamic line chart* dapat dilihat pada Gambar 6.6.

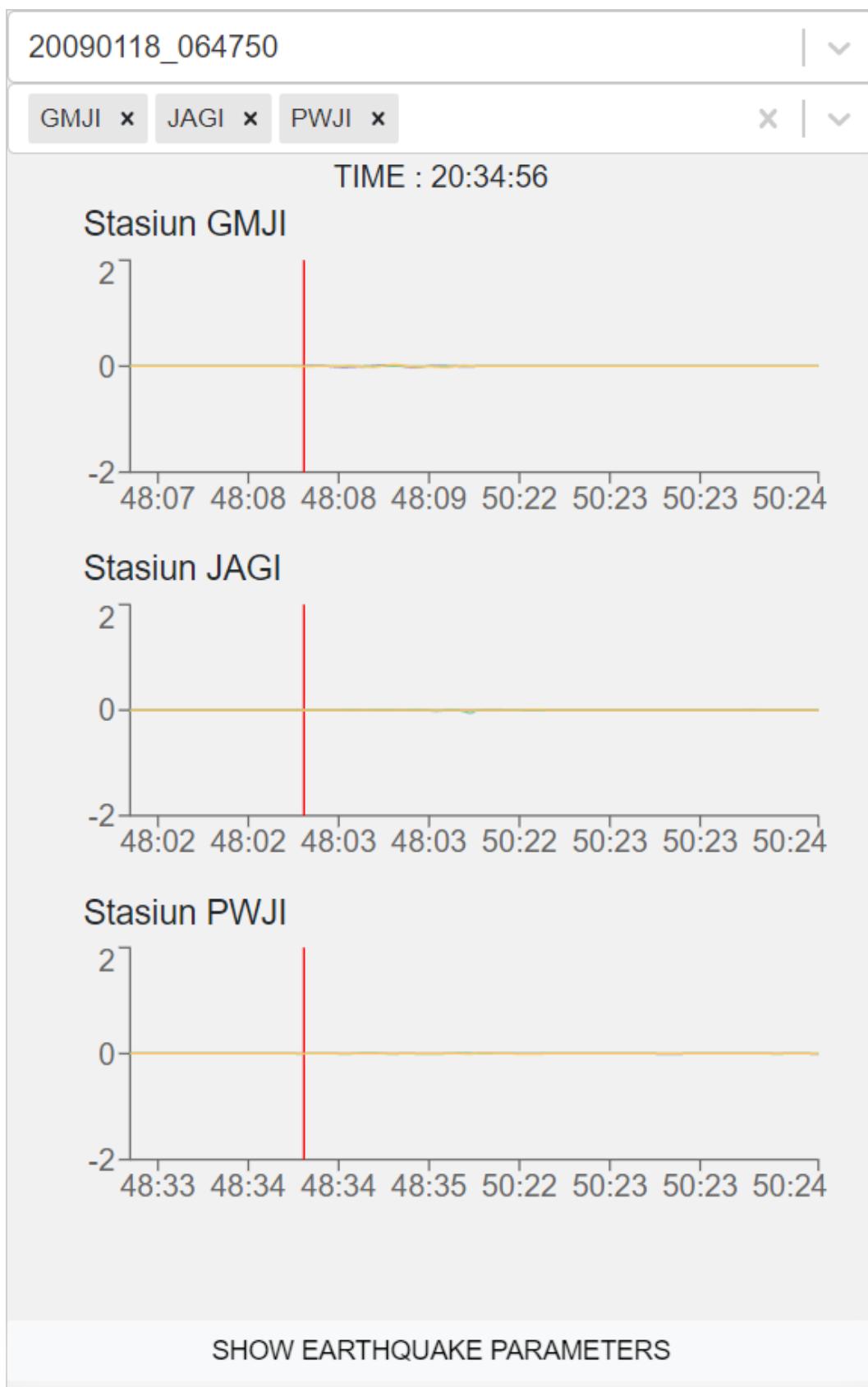


Gambar 6.10: Pemetaan data *dynamic line chart*

Sumber: (Olahan Penulis, 2022)

Untuk menampilkan grafik pada halaman admin, pengguna perlu memilih sebuah kejadian berdasarkan file mseed yang tersedia pada GUI *dropdown* yang akan disimulasikan, lalu memilih sumber stasiun menggunakan GUI *checkbox* yang terletak dibawah *dropdown* mseed. Dalam memvisualisasi *dynamic line chart* terdapat tiga endpoint. Tiga endpoint ini berkorespondensi dengan tiga stasiun seismik. Endpoint-endpoint dapat diakses melalui url seperti /get_jagi_data/mseed/ dimana mseed merupakan pilihan nama file mseed yang didapatkan melalui GUI dropdown pada panel kiri halaman admin. Setiap pemilihan mseed, *useEffect* (sebuah metode React.js) akan membuat hubungan dengan tiga *websocket* baru dan memutuskan hubungan dengan tiga *websocket* lama. Selama suatu mseed terpilih, *front-end* akan selalu berhubungan dengan tiga *websocket* yang berkorespondensi pada stasiun sensor GMJI, PWJI, dan JAGI agar pada pemilihan untuk menampilkan grafik, setiap grafik akan selalu menampilkan data yang sinkronus dengan grafik lain meskipun tidak ditampilkan semua diawal. Terdapat sebuah fitur pada *dynamic line chart* ini dimana setiap pendekripsi *p-arrival*, grafik akan menandakan posisi penemuan ini dengan sebuah garis merah. Untuk menampilkan garis ini, *useEffect* perlu menangkap *datapoint* pada arus data dimana *p_Arrival* tidak kosong. Lalu saat ditemukan posisi *p-arrival* berdasarkan

index data pada Array data stream, data ini disimpan pada komponen *recharts.js ReferenceLine*. Grafik ini juga akan menghentikan *update* dari 50% tampilan grafik untuk memberi pengguna sebuah gambar yang menunjukkan titik dimana p-arrival terdeteksi pada grafik. Tampilan dari grafik ini dapat dilihat pada Gambar 6.7. Sesaat terdeteksinya p-arrival oleh algoritma prediksi *useEffect* akan mendeteksi p-arrival lalu menunggu hingga *ReferenceLine* mencapai *index* ke-25 pada x-axis grafik lalu 50% dari grafik berhenti.



Gambar 6.11: Tampilan *dynamic line chart*

Sumber: (Olahan Penulis, 2022)

Pada panel bagian kiri dimana terdapat daftar *dynamic line chart* (Pada Gambar 6.8), terdapat juga sebuah tombol yang akan menampilkan tabel berisi informasi waktu dan tanggal terdeteksi p-arrival, magnitude, kedalaman gempa, nama stasiun yang mendeteksi p-arrival, dan lokasi stasiun tersebut. Tabel ini bertujuan agar pengguna dapat melihat informasi gempa tanpa membuka pop-up pada GUI sumber gempa pada peta interaktif.

Parameter Gempa Bumi	
P-Arrival Time	18 Januari 2009 01:04:45.425
Magnitudo	4.19
Kedalaman	29.5 KM
Receiver ID	GMJI
Receiver Lat & Long	-9.063565, 114.438963
SHOW EARTHQUAKE PARAMETERS	

Gambar 6.12: Tabel Parameter Gempa

Sumber: (Olahan Penulis, 2022)

Setelah algoritma prediksi mendeteksi *p-arrival* lalu mendapat hasil prediksi koordinat sumber gempa, magnitude, dan kedalaman gempa, peta interaktif akan memindahkan posisi pusatnya pada posisi sumber gempa tersebut dan membuat sebuah lingkaran merah. Lingkaran ini merupakan representasi visual sumber gempa bumi pada peta interaktif. Apabila lingkaran ini ditekan, akan muncul sebuah pop-up yang berisi data koordinat, magnitude, dan kedalaman yang didapatkan dari koneksi dengan tiga websocket GMJI, PWJI, dan JAGI. Hasil prediksi yang memanipulasi peta interaktif merupakan hasil yang ditemukan pertama oleh salah satu dari stasiun.



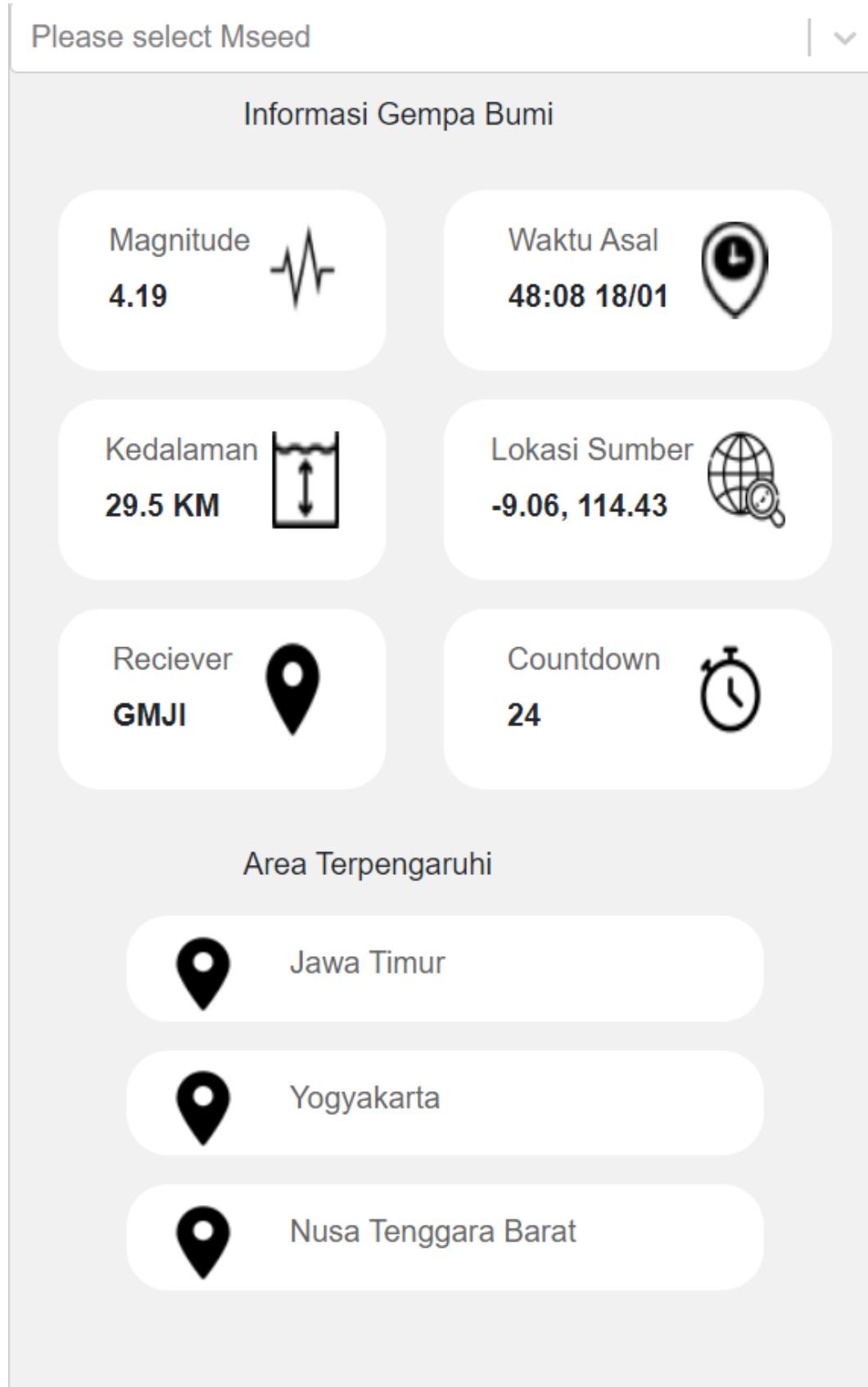
Gambar 6.13: Tampilan Peta Interaktif

Sumber: (Olahan Penulis, 2022)

Peta interaktif juga memiliki beberapa GUI yang dapat memandu pengguna memahami penggunaan sistem ini. GUI ini merupakan tombol zoom-in dan zoom-out, tombol legenda yang berisi legenda dari peta interaktif, simbol serta nama dari tiga stasiun yang kami gunakan sebagai sumber informasi, pop-up pada simbol stasiun yang menampilkan lokasi stasiun dan pop-up sumber gempa yang menampilkan informasi hasil prediksi. Halaman ini juga memiliki sebuah tampilan yang menunjukkan performa dari algoritma prediksi waktu yang diperlukan untuk mengirim hasil prediksi pada setiap pengiriman atau terdeteksinya *p-arrival* dan waktu yang diperlukan untuk algoritma menemukan hasil prediksi yang berupa koordinat latitude-longitude lokasi sumber gempa, magnitude, dan kedalaman dari suatu gempa bumi.

Selain pada halaman admin, sistem ini juga akan menyambung pada websocket pada halaman pengguna. Pada halaman pengguna pendekripsi *p-arrival* tidak ditampilkan menggunakan *dynamic line chart* namun menampilkannya menggunakan GUI kartu pada panel kiri dari halaman. Tampilan panel ini dapat dilihat pada Gambar 6.8. Panel ini juga menunjukkan magnitude, kedalaman, waktu terdeteksi *p-arrival*, lokasi sumber, stasiun pendekripsi, lokasi yang dapat terpengaruh oleh gempa bumi dan waktu hitung mundur hingga kejadian gempa bumi tersebut. Data magnitude, kedalaman, waktu terdeteksi gempa, dan lokasi

sumber gempa didapatkan dengan cara yang serupa dengan pengambilan data pada halaman admin. Pada halaman ini peta interaktif berkerja serupa dengan pada halaman admin.



Gambar 6.14: Tampilan Panel Kiri Halaman Pengguna

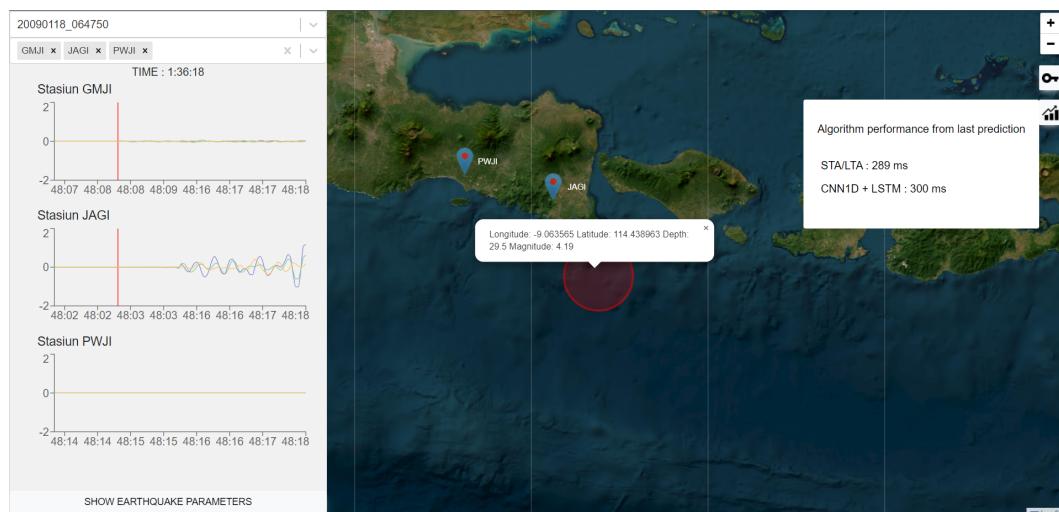
Sumber: (Olahan Penulis, 2022)

6.3 Tampilan Hasil Prediksi

Subbab ini akan menunjukkan serta menjelaskan tampilan hasil prediksi pada halaman pengguna dan halaman admin yang telah diimplementasi pada bab sebelumnya.

6.3.1 Tampilan Halaman Admin

Tampilan hasil prediksi pada halaman admin dapat dilihat pada Gambar 6.11. Contoh data yang ditampilkan merupakan hasil dari prediksi file mseed dengan *id* 20090118_064750. Data yang divisualisasikan merupakan rekaman getaran bumi pada tanggal 18 Januari 2018 yang direkam oleh stasiun GMJI, JAGI, dan PWJI.



Gambar 6.15: Tampilan Hasil Prediksi pada Halaman Admin

Sumber: (Olahan Penulis, 2022)

Dapat terlihat pada Gambar 6.11 bahwa dua dari tiga stasiun pada saat itu telah mendeteksi *p-arrival* pada simulasi kejadian gempa bumi ini. Pada grafik stasiun GMJI *p-arrival* terdeteksi pada menit ke-48:08 dan pada grafik stasiun JAGI *p-arrival* terdeteksi pada menit ke-48:02 sementara grafik stasiun PWJI belum mendeteksi *p-arrival*. Hal ini menandakan antara ketiga stasiun, stasiun JAGI merupakan stasiun pertama yang mendeteksi *p-arrival*. Pada grafik JAGI dan GMJI dapat terlihat bawah bagian kiri dari grafik sudah berhenti bergerak dan grafik bagian kanan tetap berjalan. Penghentian jalannya setengah dari grafik ini menunjukkan posisi dimana algoritma mendeteksi *p-arrival*. Setelah terdeteksi *p-arrival* dapat terlihat bahwa peta interaktif menampilkan sebuah lingkaran dipusat tampilan sebagai representasi sumber gempa bumi serta pop-up yang menunjukkan informasi koordinat sumber, magnitude, dan kedalaman gempa. Pada Gambar 6.12

dapat dilihat bahwa tabel parameter gempa juga memperbarui data sesuai dengan hasil prediksi.

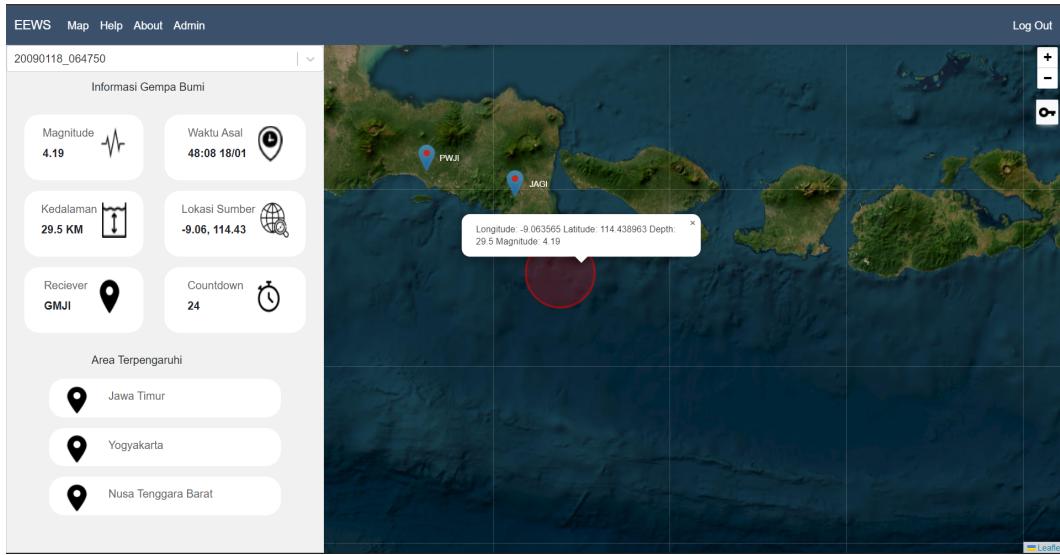
Parameter Gempa Bumi	
P-Arrival Time	18 Januari 2009 01:04:45.425
Magnitudo	4.19
Kedalaman	29.5 KM
Receiver ID	GMJI
Receiver Lat & Long	-9.063565, 114.438963
SHOW EARTHQUAKE PARAMETERS	

Gambar 6.16: Tampilan Hasil Prediksi Tabel Parameter Gempa

Sumber: (Olahan Penulis, 2022)

6.3.2 Tampilan Halaman Pengguna

Tampilan hasil prediksi pada halaman pengguna dapat dilihat pada Gambar 6.13. Data yang ditampilkan merupakan hasil dari prediksi file mseed dengan *id* 20090118_064750. Data yang divisualisasikan merupakan rekaman getaran bumi pada tanggal 18 Januari 2018 yang direkam oleh stasiun GMJI, JAGI, dan PWJI.



Gambar 6.17: Tampilan Hasil Prediksi pada Halaman Pengguna

Sumber: (Olahan Penulis, 2022)

Dapat dilihat pada Gambar 6.13 bahwa panel kiri halaman sudah memiliki informasi gempa bumi yang didapatkan dari algoritma prediksi. Tidak seperti halaman admin, halaman ini tidak menunjukkan grafik stasiun seismik dan hanya menampilkan hasil prediksi algoritma CNN1D+LSTM. Namun untuk informasi *countdown* atau hitung mundur hingga gempa, angka ini didapatkan berdasarkan rata-rata selisih waktu *p-arrival* dan *s-arrival* pada penelitian (Wibowo et al., 2022). Informasi area terpengaruh juga merupakan data sementara dikarenakan belum ada algoritma yang kami kembangkan yang dapat memprediksi informasi tersebut. Peta interaktif serupa dengan pada halaman admin akan memindahkan posisi tengah dan menampilkan lingkaran yang merepresentasikan sumber gempa.

6.4 Hasil Pengujian GUI

Subbagian ini mencakup hasil tahap pengujian GUI. Pengujian dilakukan dengan menggunakan metode pengujian fungsionalitas dengan 15 skenario yang dijalankan. Jika skenario berjalan dengan sukses dan menghasilkan hasil yang sesuai, penulis memberikan hasil tes dengan label sukses. Namun, jika skenario gagal dan tidak memberikan hasil yang sesuai, penulis menandainya sebagai gagal. Hasil uji fungsional ditunjukkan pada Tabel 6.1

Tabel 6.3: Hasil *Functionality Testing*

Skenario	Hasil Pengujian
Sistem dapat memuat <i>dynamic line chart</i> GMJI saja	Sukses
Sistem dapat memuat <i>dynamic line chart</i> JAGI saja	Sukses
Sistem dapat memuat <i>dynamic line chart</i> PWJI saja	Sukses
Sistem dapat memuat <i>dynamic line chart</i> GMJI, JAGI saja	Sukses
Sistem dapat memuat <i>dynamic line chart</i> GMJI, PWJI saja	Sukses
Sistem dapat memuat <i>dynamic line chart</i> JAGI, PWJI saja	Sukses
Sistem dapat memuat <i>dynamic line chart</i> GMJI, JAGI, PWJI	Sukses
Sistem dapat menghapus <i>dynamic line chart</i> GMJI	Sukses
Sistem dapat menghapus <i>dynamic line chart</i> JAGI	Sukses
Sistem dapat menghapus <i>dynamic line chart</i> PWJI	Sukses
Sistem dapat mengubah mseed dan <i>dynamic line chart</i> mengubah isi data secara dinamis	Sukses
sistem dapat memuat data mseed 20090118_064750 melalui websocket	Sukses
sistem dapat memuat data mseed 20090119_211141 melalui websocket	Sukses
sistem dapat memuat data mseed 20100208_112154 melalui websocket	Sukses
sistem dapat memuat data mseed 20120915_163224 melalui websocket	Sukses

BAB 7

KESIMPULAN DAN SARAN

Bab ini menjelaskan kesimpulan dari seluruh penelitian yang telah dilakukan dan saran untuk penelitian selanjutnya.

7.1 Kesimpulan

Berdasarkan eksperimen dan implementasi yang telah dilakukan, dapat diambil beberapa kesimpulan bersumber pada data yang sudah didapat. Berikut adalah beberapa kesimpulan yang dapat diambil dari penelitian ini:

1. Data seismik dapat dimodelkan menggunakan algoritma machine learning dengan mengkalibrasi *sampling rate* melalui teknik interpolasi sehingga dapat dilatih dan diprediksi oleh model. Dalam menemukan *p-arrival* model menggunakan algoritma STA/LTA dengan bantuan *library* Python Obspy.
2. Penelitian ini menemukan bahwa skenario dengan model Conv1d digabung dengan LSTM memiliki performa yang lebih baik dibandingkan Conv1d dan LSTM masing-masing untuk menemukan hiposenter, magnitude, dan kedalaman gempa bumi. Selain itu skenario dengan jendela interval 10 detik (5 detik sebelum *p-arrival* dan 5 detik setelah *p-arrival*) memiliki nilai MAE tertinggi diantara skenario dan model yang digunakan, dengan nilai MAE all sebanyak 0.470.
3. Terdapat dua bagian *backend* pada sistem ini yaitu REST API digunakan untuk menyimpan hasil ekstraksi data ke basis data dan Websocket API yang digunakan sebagai media pengiriman data secara *realtime*. Sehingga Websocket API dapat digunakan untuk membantu memvisualisasikan data secara *realtime*. Waktu komputasi pada *backend* untuk *pre-processing* data secara rata-rata adalah 0.26 detik sedangkan waktu yang dibutuhkan untuk memprediksi hiposenter gempa rata-rata adalah 1.23 detik dengan waktu mulai prediksi adalah 5 detik setelah *p-arrival*. Kemudian waktu deteksi *p-arrival* adalah 0.002 detik sehingga total keseluruhan adalah sekitar 6.5 detik. Berdasarkan paper Wibowo et al. (2022) rata-rata jarak *p-arrival* dengan *s-arrival* adalah 21.53 detik, sehingga waktu yang tersisa sebelum getaran gempa bumi dapat dirasakan adalah sekitar 15 detik.

4. Berdasarkan implementasi *multi-target regression*, implementasi pemrograman back-end dan implementasi pemrograman front-end, penelitian ini menggunakan algoritma STA/LTA untuk mendeteksi gelombang awal suatu dan model Conv1d digabung dengan LSTM untuk mendeteksi lokasi sumber, magnitude, dan kedalaman suatu gempa bumi, lalu mengirim hasil prediksi tersebut menggunakan websocket agar dapat divisualisasikan pada antarmuka pengguna secara *real-time*.

7.2 Saran

Penulis memahami bahwa masih terdapat kekurangan dalam pelaksanaan penelitian ini. Oleh karena itu, penulis memiliki beberapa saran terhadap penelitian lanjutan diantaranya:

1. Menggunakan data langsung dari stasiun-stasiun seismik dengan menghubungkan antarmuka pada suatu API yang mengirim data secara *real-time*.
2. Pembuatan antarmuka dapat memberi peringatan gempa bumi yang lebih menarik perhatian pengguna apabila sistem ingin diterapkan secara umum sebagai alarm kejadian gempa bumi.
3. Membuat aplikasi *mobile* agar pengguna dapat menerima peringatan gempa dimana saja mereka berada.
4. Data yang digunakan untuk *training* lebih banyak lebih banyak agar model dapat lebih akurat dibandingkan penelitian ini.
5. Menggunakan model *machine learning* lain untuk memprediksi *p-arrival*.

DAFTAR REFERENSI

- Abd El-Aziz, A. A. and Kannan, A. (2014). Json encryption. In *2014 International Conference on Computer Communication and Informatics*, pages 1–6.
- Adam, B. M., Rachmat Anom Besari, A., and Bachtiar, M. M. (2019). Backend server system design based on rest api for cashless payment system on retail community. In *2019 International Electronics Symposium (IES)*, pages 208–213.
- Choubik, Y., Mahmoudi, A., Himmi, M., and El Moudnib, L. (2020). Sta/lta trigger algorithm implementation on a seismological dataset using hadoop mapreduce. *IAES International Journal of Artificial Intelligence (IJ-AI)*, 9:269.
- Cybulski, P. and Horbiński, T. (2020). User experience in using graphical user interfaces of web maps. *ISPRS International Journal of Geo-Information*, 9(7):412.
- Garrett, J. J. (2011). *The elements of user experience: User-centered design for the web and beyond*. New Riders.
- Ghobarah, A., Saatcioglu, M., and Nistor, I. (2006). The impact of the 26 december 2004 earthquake and tsunami on structures and infrastructure. *Engineering Structures*, 28:312–326.
- Goel, G., Tanwar, P., and Sharma, S. (2022). UI-UX design using user centred design (UCD) method. In *2022 International Conference on Computer Communication and Informatics (ICCCI)*. IEEE.
- Graves, A. (2012). *Supervised Sequence Labelling with Recurrent Neural Networks*, volume 385.
- Haklay, M. (2010). *Interacting with Geospatial Technologies*. Wiley-Blackwell.
- Horbiński, T. and Lorek, D. (2020). The use of leaflet and GeoJSON files for creating the interactive web map of the preindustrial state of the natural environment. *Journal of Spatial Science*, 67(1):61–77.
- Hussin, S. F., Hamid, Z., and Birasamy, G. (2016). Design of butterworth band-pass filter. pages 128–2883.

- Indolia, S., Goswami, A. K., Mishra, S., and Asopa, P. (2018). Conceptual understanding of convolutional neural network- a deep learning approach. *Procedia Computer Science*, 132:679–688. International Conference on Computational Intelligence and Data Science.
- Johnson, J. (2007). *GUI Bloopers 2.0: Common User Interface Design Don'ts and Dos*.
- Johnson, J. (2014). *Designing with the Mind in Mind*. Elsevier.
- Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., and Inman, D. J. (2021). 1d convolutional neural networks and applications: A survey. *Mechanical Systems and Signal Processing*, 151:107398.
- Kristiadi, D. P., Udjaja, Y., Supangat, B., Prameswara, R. Y., Warnars, H. L. H. S., Heryadi, Y., and Kusakunniran, W. (2017). The effect of UI, UX and GX on video games. In *2017 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom)*. IEEE.
- Li, Y. and Qin, Y. (2020). Research on the construction of information visualization-dynamic chart. *E3S Web of Conferences*, 179:01020.
- Lindemann, B., Müller, T., Vietz, H., Jazdi, N., and Weyrich, M. (2021). A survey on long short-term memory networks for time series prediction. *Procedia CIRP*, 99:650–655. 14th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 15-17 July 2020.
- Meng, M., Steinhardt, S., and Schubert, A. (2018). Application programming interface documentation: What do software developers want? *Journal of Technical Writing and Communication*, 48:295–330.
- Miller, R. B. (1968). Response time in man-computer conversational transaction. In *1968 Fall Joint Computer Conference*.
- Mousavi, S., Ellsworth, W., Weiqiang, Z., Chuang, L., and Beroza, G. (2020). Earthquake transformer—an attentive deep-learning model for simultaneous earthquake detection and phase picking. *Nature Communications*, 11:3952.
- Norman, D. (2004). Emotional design. *Ubiquity*, 2004:1–1.
- Pillay, N. and Wing, J. (2019). Agile ux: Integrating good ux development practices in agile. In *2019 Conference on Information Communications Technology and Society (ICTAS)*, pages 1–6.

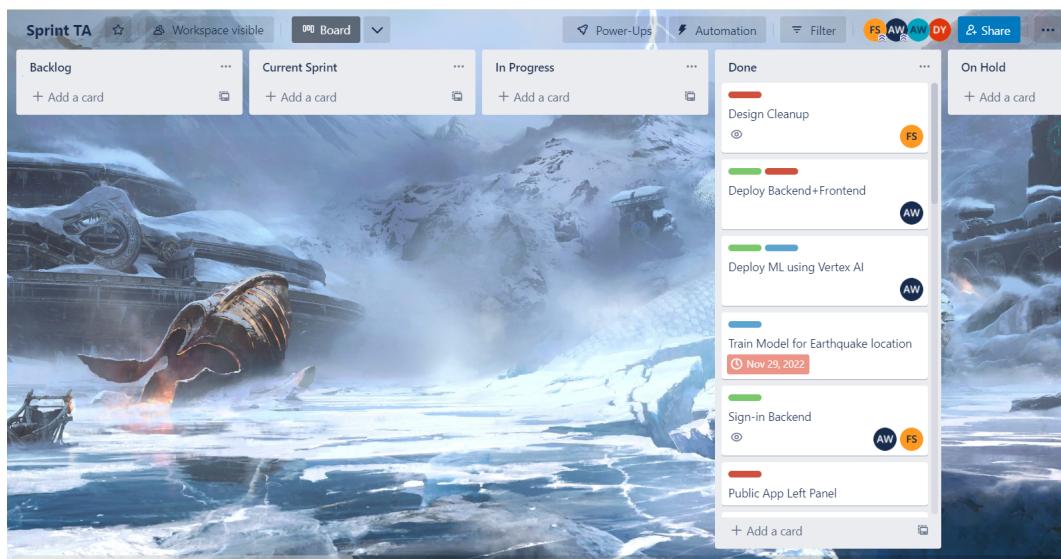
- Poulter, A. J., Johnston, S. J., and Cox, S. J. (2015). Using the mean stack to implement a restful service for an internet of things application. In *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, pages 280–285.
- Rachmat, N., Octaria, O., Tarigan, D., and Samsuryadi, S. (2016). Sistem pemanggilan antrian menggunakan websocket.
- Roth, R. E. (2013). Interactive maps: What we know and what we need to know. *Journal of Spatial Information Science*, (6).
- Roth, R. E. (2015). Interactivity and cartography: A contemporary perspective on user interface and user experience design from geospatial professionals. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 50(2):94–115.
- Sahay, A. (2016). *Data Visualization, volume I: Recent trends and applications using conventional and Big Data*. Business Expert Press.
- Simkin, D. and Hastie, R. (1987). An information-processing analysis of graph perception. *Journal of the American Statistical Association*, 82(398):454–465.
- Stylos, J., Faulring, A., Yang, Z., and Myers, B. A. (2009). Improving api documentation using api usage information. In *2009 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 119–126.
- Sukatis, F. F., Noor, N. M., Zakaria, N. A., Ul-Saufie, A. Z., and Annas, S. (2019). Estimation of missing values in air pollution dataset by using various imputation methods. *Entropy*.
- T. T. Hanifa, S. Al-faraby, F. I. and Telkom, U. (2017). Analisis churn prediction pada data pelanggan pt . telekomunikasi dengan logistic regression dan underbagging.
- Tenreiro Machado, J. and Lopes, A. (2013). Analysis and visualization of seismic data using mutual information. *Entropy*, 15:3892–3909.
- Wang, J., Zeng, X., and Tang, M. (2021a). Research on the design of smart cosmetic based on user experience. In *2021 26th International Conference on Automation and Computing (ICAC)*, pages 1–4.
- Wang, S., Panning, M. P., Vance, S. D., and Song, W. (2021b). Underground microseismic event monitoring and localization within sensor networks. *Sensors*, 21(8):2830.

- Wang, V., Salim, F., and Moskovits, P. (2013). *The Definitive Guide to HTML5 WebSocket*, pages 149–162.
- Wibowo, A., Pratama, C., Sahara, D. P., Heliani, L. S., Rasyid, S., Akbar, Z., Muttaqy, F., and Sudrajat, A. (2022). Earthquake early warning system using ncheck and hard-shared orthogonal multitarget regression on deep learning. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5.
- Withers, M., Aster, R., Young, C., Beiriger, J., Harris, M., Moore, S., and Trujillo, J. (1998). A comparison of select trigger algorithms for automated global seismic phase and event detection. *Bulletin of the Seismological Society of America*, 88:95–106.
- Zhou, R., Shao, S., Li, W., and Zhou, L. (2016). How to define the user’s tolerance of response time in using mobile applications. In *2016 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pages 281–285.

LAMPIRAN

LAMPIRAN 1: DOKUMENTASI SCRUM AGILE FRAMEWORK

Lampiran ini berisi dokumentasi penerapan metodologi *scrum agile framework* terhadap proses penelitian ini.



Gambar 1: Papan Kanban

Sumber: (Olahan Penulis, 2022)

Today < > December 2022

SUN 27	MON 28	TUE 29	WED 30	THU Dec 1	FRI 2
	<ul style="list-style-type: none"> ● 12am Implement lokasi su ● 1pm Sprint Review ● 2pm Daily Standup 	<ul style="list-style-type: none"> ● 12am About Page [Open] ● 12am Deploy ML using Ve ● 12am Design Cleanup [Op] <p>5 more</p>	<ul style="list-style-type: none"> ● 1pm Meeting Pak Ari ● 2pm Daily Standup 	<ul style="list-style-type: none"> ● 2pm Daily Standup 	<ul style="list-style-type: none"> ● 2pm Daily Standup
4	5	<ul style="list-style-type: none"> ● 1pm Sprint Planning ● 2pm Daily Standup 	<ul style="list-style-type: none"> ● 7pm Daily Standup 	<ul style="list-style-type: none"> ● 1pm Meeting Pak Ari ● 2pm Daily Standup 	<ul style="list-style-type: none"> ● 2pm Daily Standup
11	12	<ul style="list-style-type: none"> ● 1pm Sprint Review ● 2pm Daily Standup 	<ul style="list-style-type: none"> ● 12am Deploy Backend+Frc ● 7pm Daily Standup 	<ul style="list-style-type: none"> ● 1pm Meeting Pak Ari ● 2pm Daily Standup 	<ul style="list-style-type: none"> ● 2pm Daily Standup
18	19	<ul style="list-style-type: none"> ● 1pm Sprint Planning ● 2pm Daily Standup 	<ul style="list-style-type: none"> ● 7pm Daily Standup 	<ul style="list-style-type: none"> ● 1pm Meeting Pak Ari ● 2pm Daily Standup 	<ul style="list-style-type: none"> ● 2pm Daily Standup
25	26	<ul style="list-style-type: none"> ● 1pm Sprint Review ● 2pm Daily Standup 	<ul style="list-style-type: none"> ● 7pm Daily Standup 	<ul style="list-style-type: none"> ● 1pm Meeting Pak Ari ● 2pm Daily Standup 	<ul style="list-style-type: none"> ● 2pm Daily Standup

Gambar 2: Timeline Scrum**Sumber:** (Olahan Penulis, 2022)