

BAB III

METODE PENELITIAN

Penelitian ini dilakukan dengan metode eksperimen komputasional dengan pendekatan kuantitatif. Eksperimen komputasional yang dimaksud adalah untuk mencari dan memperoleh arsitektur model yang sedemikian rupa sehingga menghasilkan nilai akurasi yang optimal. Penelitian dilakukan di Laboratorium Bumi dan Antariksa Fakultas Pendidikan Matematika dan Ilmu Pengetahuan Alam, Universitas Pendidikan Indonesia menggunakan perangkat komputasi yang mendukung dalam melakukan proses pelatihan (*training*) data dalam Python sebagai bahasa pemrograman utama. Adapun pustaka-pustaka Python yang digunakan di antaranya ialah Numpy, Pandas, Tensorflow, Scikit-Learn, Matplotlib, Gwpy, Gwosc, dan PyCBC.

3.1. Sarana dan Prasarana

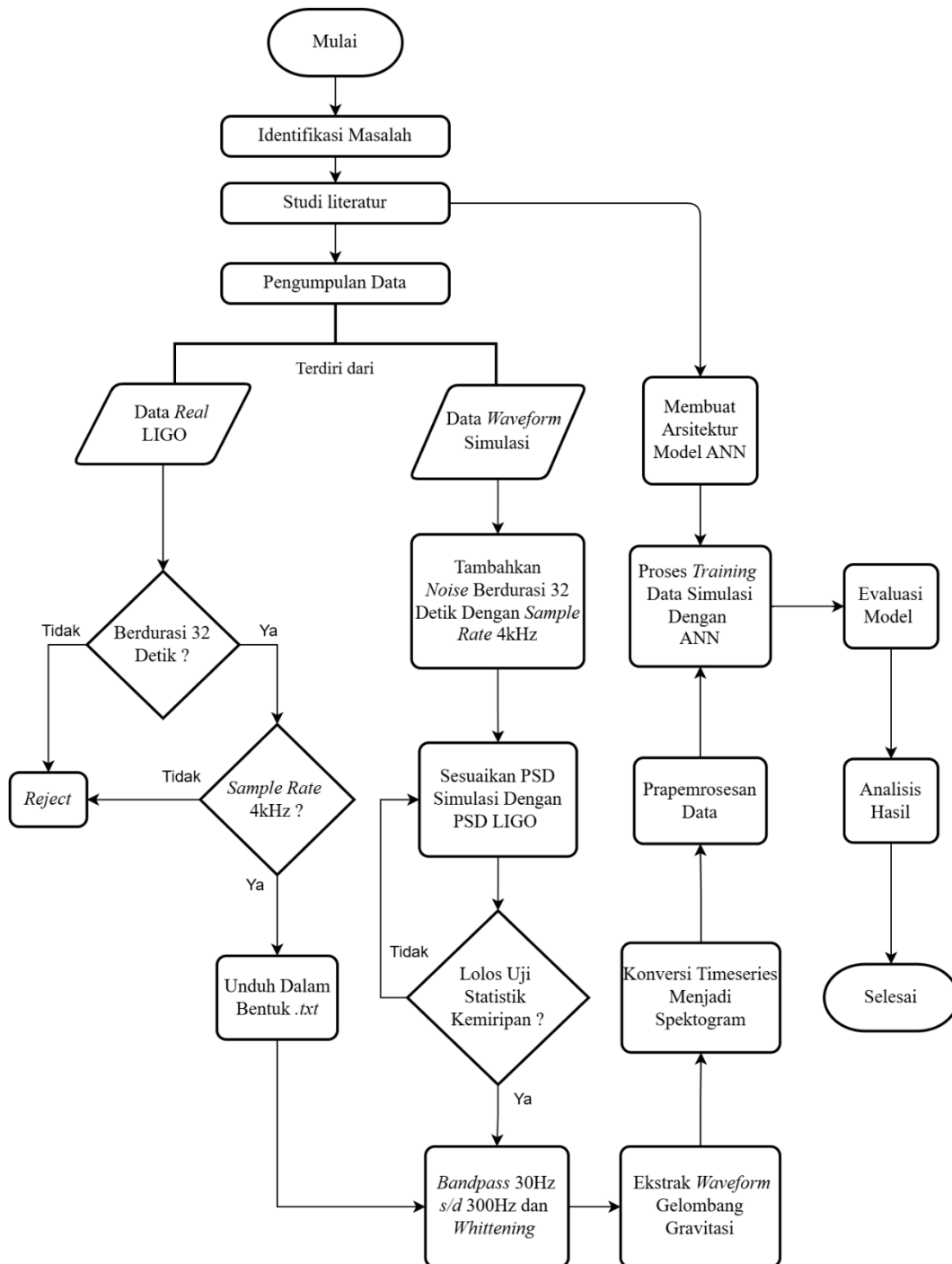
Untuk mendukung kelancaran penelitian ini beberapa alat berupa perangkat keras dan lunak yang digunakan di antaranya ialah sebagai berikut,

1. Laptop : Penelitian ini menggunakan laptop sebagai perangkat keras untuk mengerjakan semua tahapan proses mulai dari pengambilan data, hingga analisis hasil. Adapun spesifikasi laptop yang digunakan ialah prosesor Intel Core i7-7600U 2,80 GHz dengan RAM 12 GB.
2. Google Colab : Fasilitas ini digunakan dalam mengeksekusi program untuk melatih data karena google colab menyediakan GPU secara gratis dengan spesifikasi NVIDIA Tesla T4 GPU 1,59 GHz VRAM 16 GB GDDR6.
3. Python : Dalam penelitian akan digunakan bahasa pemrograman Python karena memiliki fleksibilitas yang sangat baik dan tersedianya berbagai pustaka yang mendukung semua pekerjaan terutama dalam pengembangan model CNN.
4. Numpy : Numpy (Harris dkk., 2020) digunakan dalam melakukan analisis dan transformasi data numerik terutama yang berhubungan dengan operasi matematis pada data yang berbentuk matriks.
5. Scikit-learn : Pustaka ini digunakan karena menyediakan alat untuk melakukan pra-pemrosesan seperti normalisasi, standarisasi, dan menangani data yang hilang. Pustaka ini juga digunakan untuk melakukan pemisahan data menjadi

data *training*, *test*, dan *validation*. Selain itu, pustaka ini menyediakan berbagai fungsi metrik untuk keperluan evaluasi.

6. Tensorflow : Tensorflow ini merupakan pustaka utama dalam pengembangan model CNN dalam penelitian ini. Pustaka ini menyediakan berbagai fungsi utama dalam membangun model CNN seperti *Sequential*, *Dense*, *compile*, dan *train_model*.
7. PyCBC : PyCBC digunakan terutama dalam pengambilan data simulasi karena pustaka ini menyediakan modul untuk memanggil data hasil simulasi numerik yang telah dilakukan oleh berbagai penelitian. Selain itu, pustaka ini juga digunakan dalam proses analisis dan transformasi data simulasi, sedemikian sehingga dapat digunakan pada proses pelatihan data.

3.2. Prosedur Penelitian



Gambar 3.1 Alur Penelitian

3.3. Teknik Pengumpulan Data

3.3.1. Data Real

Data *real* yang digunakan dalam penelitian ini bersumber dari data *event* gelombang gravitasi yang dihimpun oleh *Gravitational Wave Open Science Center* (GWOSC) yang merupakan bentuk kolaborasi dari beberapa detektor gelombang gravitasi seperti LIGO, Virgo, KAGRA, dan GEO600 untuk memberikan akses data kepada publik, yang dapat diakses melalui laman <https://gwosc.org/>. Pada laman tersebut setiap *event* dapat diakses melalui katalog gelombang gravitasi seperti GWTC-1-confident, GWTC-2.1-confident, dan GWTC-3-confident. Katalog GWTC-1-confident merupakan katalog *event* gelombang gravitasi yang berhasil diobservasi oleh LIGO selama pengoperasian pertama dan kedua (O1 dan O2) yakni dari tahun 2015 hingga 2017 (B. P. Abbott dkk., 2019). Sedangkan, katalog GWTC-2.1-confident menghimpun data-data *event* gelombang gravitasi selama pengoperasian O3a yakni selama tahun 2019 (T. Abbott dkk., 2021). Sementara itu, GWTC-3-confident merupakan katalog gelombang gravitasi yang berhasil terdeteksi pada pengoperasian O3b selama akhir pertengahan 2019 hingga awal pertengahan 2020 (R. Abbott, Abbott, dkk., 2023).

Untuk setiap *event*-nya GWOSC menyediakan dua durasi waktu data yakni 4096 detik dan 32 detik. Selain itu, juga menyediakan dua *sample rate* yakni 4kHz dan 16 kHz. Durasi waktu ini dapat dianalogikan sebagai *field of view* (FOV) pada data gambar, sedangkan *sample rate* dapat dianalogikan sebagai resolusinya. Dalam penelitian ini akan digunakan data berdurasi waktu 32 detik agar lebih terfokus hanya pada satu *event* dengan *sample rate* 4 kHz. Hal ini dilakukan agar selama proses pelatihan data pada CNN dapat dilakukan dengan lebih cepat dan kompleksitas komputasi tidak terlalu tinggi.

3.3.2. Data Simulasi

Dalam penelitian ini, data latih (*training data*) yang digunakan merupakan data simulasi numerik yang telah dilakukan oleh penelitian terdahulu. Data ini diperoleh menggunakan fungsi Python *get_td_waveform* yang terdapat pada pustaka Python PyCBC (*Python Compact Binary Coalescence*). Parameter-parameter yang akan digunakan sebagai masukan terhadap fungsi tersebut di antaranya ialah m_1 dan m_2 dengan rentang antara $5M_{\odot}$ hingga $45M_{\odot}$ dengan bin

$1M_{\odot}$, serta jarak dengan rentang antara 600Mpc hingga 1945Mpc dengan bin ± 96 Mpc.

Data simulasi tersebut merupakan *waveform* dalam domain waktu yang berdurasi kurang dari 2 detik. Mengingat bahwa untuk keperluan penelitian ini data simulasi haruslah dibuat sedemikian sehingga dapat semirip mungkin dengan data yang diperoleh detektor LIGO melalui GWOSC yang berdurasi 32 detik dengan *sample rate* 4kHz, maka *waveform* yang telah diperoleh terlebih dahulu dilakukan pelebaran/ekstensi dengan cara dimasukkan ke dalam *numpy array* yang berisi deretan angka nol sedemikian sehingga *array* tersebut memiliki durasi 32 detik ($32 \times \text{sample rate}$). Kemudian untuk mendapatkan efek gangguan, digunakan fungsi *colored_noise* (dari PyCBC) terhadap data berdasarkan PSD yang disesuaikan dengan karakteristik PSD pada detektor LIGO. Penyesuaian PSD dilakukan dengan mempertimbangkan beberapa fungsi PSD yang tersedia pada pustaka Python PyCBC seperti *aLIGOZeroDetHighPower*, *aLIGOaLIGO175MpcT1800545*, *aLIGOThermal*, dll. Untuk dapat memastikan bahwa data simulasi ini memiliki karakteristik yang sama dengan detektor LIGO, dilakukan uji statistik terhadap distribusi spektrum tersebut.

3.3.3. Ekstraksi *Waveform* dan Prapemrosesan Data

Data yang digunakan sebagai masukan pada model CNN merupakan data berupa serial waktu hanya pada bagian sinyal gelombang gravitasi. Untuk itu, proses selanjutnya adalah dengan mengekstrak sinyal gelombang gravitasi dari data *real* dan simulasi yang sebelumnya berdurasi 32 detik menjadi hanya 1 detik. Hal ini dilakukan dengan terlebih dahulu dilakukan *bandpass* sedemikian sehingga frekuensinya pada hanya pada rentang 30Hz hingga 300Hz. kemudian dilakukan suatu proses sedemikian sehingga amplitudo di setiap frekuensi secara rata-rata bernilai sama dan ternormalisasi, proses ini disebut dengan *whitening*. Kemudian, proses ekstraksi dilakukan dengan mengambil data dalam rentang 0,8 detik sebelum *merge time* hingga 0,2 detik setelah *merge time*. Pra-pemrosesan data dilakukan dengan mengubah skala sedemikian sehingga data berada pada rentang -1 hingga 1. Hal ini dilakukan dengan terlebih dahulu menormalisasi data sehingga tidak lagi dalam orde $\sim 10^{-21}$. Kemudian, perubahan skala dilakukan dengan menggunakan fungsi *StandardScaler* pada pustaka Python Scikit-learn.

3.3.4. Konversi Menjadi Spektogram

Data yang sebelumnya berupa *timeseries*, selanjutnya dikonversi menjadi spektogram. Hal ini dilakukan karena tidak seperti *timeseries* biasa yang hanya merepresentasikan data dalam domain waktu, spektogram merepresentasikan data dalam domain waktu dan frekuensi dalam satu grafik sehingga lebih banyak menyimpan informasi terkait karakteristik sinyal (Chen dkk., 2010). Pada spektogram sumbu-x menyatakan waktu sedangkan sumbu-y menyatakan frekuensinya, sementara itu amplitudo dinyatakan dengan distribusi warna.

Konversi dari *timeseries* menjadi spektogram ini dilakukan dengan memanfaatkan modul Signal dalam pustaka Python Scipy. Luaran data setelah dilakukan konversi ini berupa matriks yang berisi nilai pixel dengan ukuran 65 x 15 yang dapat divisualisasikan menjadi gambar. Kemudian agar proses pelatihan data dengan CNN dapat dilakukan dengan efisien, ukuran matriks diubah menjadi 65 x 65 dengan menginterpolasi data pada domain waktu.

3.4. Melatih Data dengan *Convolutional Neural Network*

3.4.1. Arsitektur Model

Model deep learning dibuat menggunakan bahasa pemrograman python dengan bantuan pustaka Tensorflow dan Scikit-Learn. Tensorflow digunakan dalam merancang model dan proses pelatihan data. Sedangkan, Scikit-Learn digunakan dalam pra-pemrosesan data, pembagian data menjadi data latih dan data tes, serta dalam evaluasi hasil pelatihan. Arsitektur model sebagian besar terdiri dari lapisan konvolusi dengan tambahan beberapa lapisan *Dense*. Lapisan konvolusi digunakan untuk mentransformasi gambar sedemikian sehingga selama proses pelatihan data model dapat melakukan ekstraksi fitur. Sedangkan lapisan *Dense* digunakan agar model mengenali pola-pola non-linear yang lebih rumit. Pada beberapa blok digunakan lapisan *MaxPooling* yang digunakan untuk mengecilkan ukuran pixel sehingga proses pelatihan lebih cepat. Dalam model ini juga digunakan Batch normalization yang digunakan untuk meningkatkan efisiensi proses pelatihan data dengan cara menormalisasi data luaran dari setiap lapisan neural network (Muhammad dkk., 2023). Penggunaan batch normalization tidak hanya mempercepat konvergensi pada nilai loss juga dapat meningkatkan performansi dan

generalisasi model (Peerthum & Stamp, 2023). Untuk melakukan kompilasi model, digunakan metode Adam (Kingma & Ba, 2014) sebagai optimizer dalam proses back propagation, MSE (mean squared error) sebagai fungsi loss, dan nilai R^2 sebagai metric. Proses pelatihan data menggunakan callback berupa early stopping dan reduce LR on plateau untuk menghentikan proses secara otomatis dan mereduksi learning rate jika nilai loss tidak lagi mengalami penurunan yang signifikan.

3.4.2. Proses Pelatihan Data

Data simulasi dan data yang diperoleh dari LIGO yang sebelumnya ada dalam file berekstensi .txt masing-masing dimuat ke dalam numpy ndarray sebagai masukan data, di mana setiap baris yang berbeda merupakan *waveform* yang telah dikonversi menjadi spektrogram dengan nilai parameter fisis yang berbeda. Tiap nilai parameter fisis yang terdapat pada nama *file* juga dimuat ke dalam numpy 2darray sebagai data luaran, dengan baris yang bersesuaian terhadap data masukan. Pada data luaran ini kolom pertama menyatakan nilai massa sekunder, kolom kedua menyatakan massa primer, dan kolom ketiga menyatakan nilai jarak. Data masukan dan luaran dilakukan pemisahan masing-masing menjadi data latih sebanyak 80% dan data tes sebanyak 20% sehingga didapatkan data latih masukan, data latih luaran, data tes masukan, dan data tes luaran. Kemudian, data latih dilakukan pemisahan lanjutan untuk mendapatkan data validasi sebanyak 20%.

3.4.3. Evaluasi Model

Proses evaluasi dilakukan dengan melihat perkembangan nilai *loss* dan metrik selama proses pelatihan data berdasarkan hasil plot terhadap jumlah pengulangan yang dilakukan. Diharapkan bahwa nilai *loss* mengalami penurunan secara signifikan mengingat nilai ini menyatakan *error* atau penyimpangan hasil prediksi model terhadap nilai yang sesungguhnya, sedangkan nilai metrik berupa R^2 -score diharapkan mengalami peningkatan mengingat nilai menyatakan akurasi regresi. Kemudian, dilakukan proses prediksi terhadap data tes masukan yang diharapkan bersesuaian dengan nilai sesungguhnya yaitu data tes luaran. Pada akhirnya, model ini diaplikasikan terhadap data yang diperoleh dari LIGO untuk mengetahui keakuratan model untuk data yang nyata (*real*). Beberapa metrik yang

digunakan dalam menilai performansi model terhadap data nyata di antaranya ialah, *mean absolute error* (MAE), *root mean squared error* (RMSE), R^2 -score. Pada proses evaluasi ini dilakukan percobaan variasi pada beberapa parameter model yang dilakukan secara bertahap di mulai dari *kernel size*, *kernel initializer*, *dropout* dan *activation functions*. Hal ini dilakukan untuk mendapatkan model dengan performansi optimal.

3.5. Analisis Data

Analisis data dilakukan pada data latih simulasi, dalam hal ini terhadap domain frekuensi data *strain* simulasi sebelum dilakukan konversi. Uji statistik dilakukan untuk memastikan data tersebut sudah sesuai dan memiliki karakteristik yang sama dengan data *strain* nyata, sehingga data simulasi ini dapat merepresentasikan data nyata dengan baik.

Selain itu, analisis data dilakukan pada hasil implementasi model terhadap data *real*. Distribusi posterior yang diperoleh LIGO dari setiap parameter fisis untuk setiap *event* digunakan untuk mengevaluasi hasil estimasi model. Dalam hal ini digunakan untuk mengetahui seberapa jauh penyimpangan hasil estimasi tersebut terhadap nilai sebenarnya. Di samping itu, juga untuk mengetahui seberapa signifikan hasil estimasi tersebut. Pengujian statistik dilakukan untuk mengetahui seberapa mirip karakteristik data hasil estimasi terhadap data aktualnya dan apakah kemiripan tersebut dapat diterima secara statistik.