

BAB III

METODOLOGI PENELITIAN

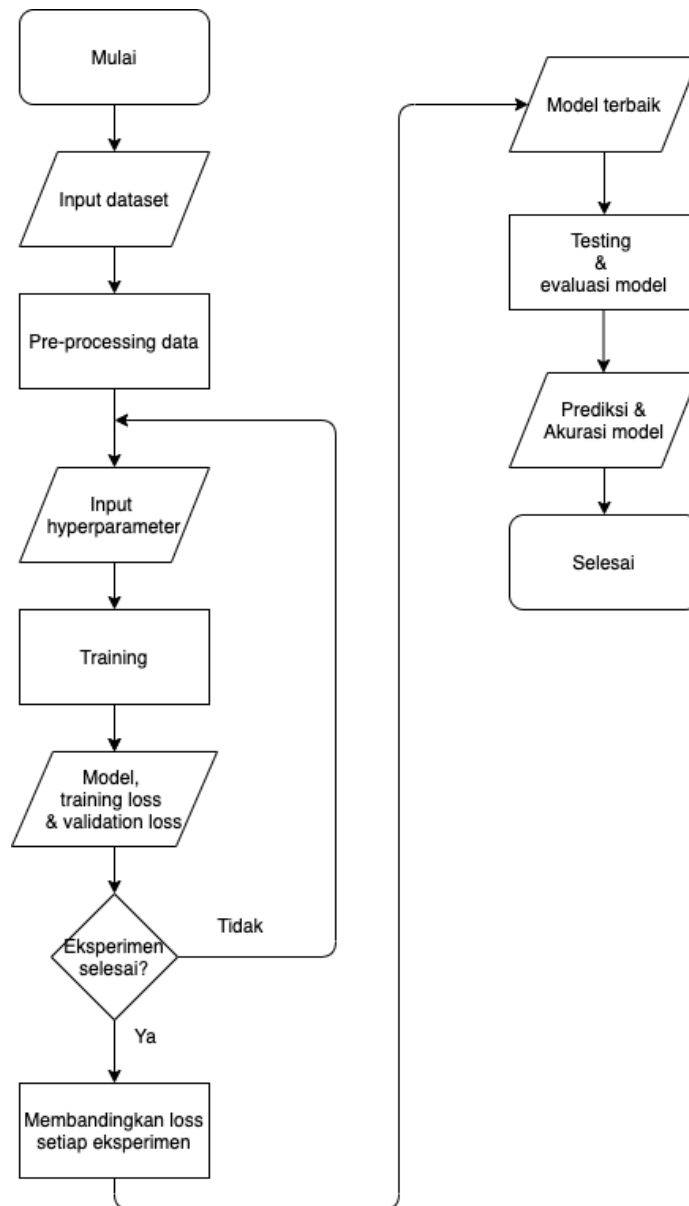
3.1 Analisis Persoalan

Pada penelitian ini digunakan metode *Gated Recurrent Unit* untuk mendapatkan sebuah model prediksi terbaik dari hasil membandingkan setiap model training yang memiliki *hyperparameter* berbeda-beda. Adapun tahap analisisnya sebagai berikut:

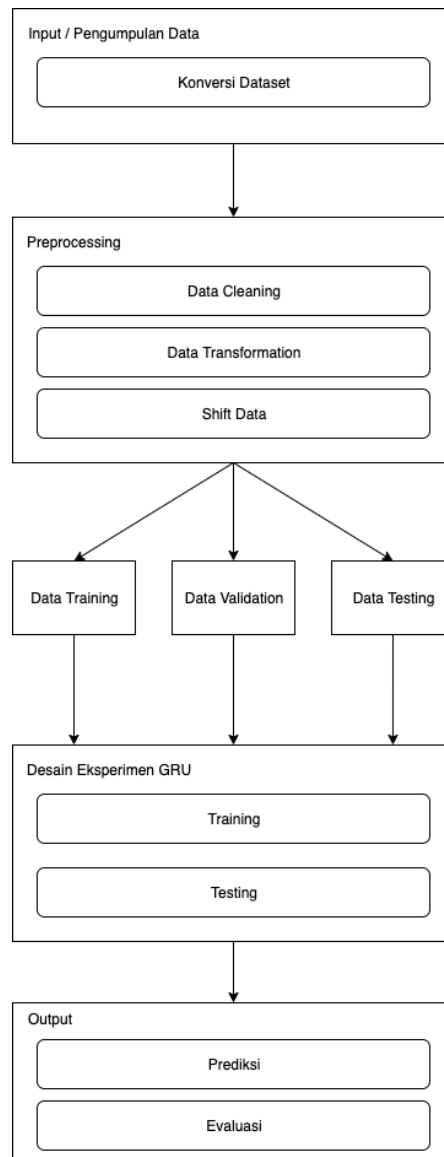
- a. *Dataset* ketinggian pasang surut yang didapat dari Badan Informasi Geospasial akan dikonversi dalam bentuk format CSV.
- b. Setiap data akan diperiksa ketinggian setiap jamnya, jika ada yang hilang maka diisi dengan interpolasi.
- c. Selanjutnya, data akan ditransformasi dalam bentuk skala umum dengan metode *Min-Max Scaller*.
- d. Data yang sudah ditransformasi selanjutnya akan dilakukan *shifting* untuk mempermudah proses *training*.
- e. Membagi data menjadi tiga bagian yaitu data *training*, data *validation* dan data *testing*.
- f. Melakukan beberapa proses *training* menggunakan GRU dengan data *training* dan *validation* serta *hyperparameter* yang berbeda-beda.
- g. Memilih model terbaik dari hasil *training* dengan membandingkan nilai RMSE setiap model.
- h. Melakukan proses *testing* menggunakan model terbaik yang telah dipilih untuk memprediksi data pasang surut untuk bulan November 2018 sampai Desember 2018.
- i. Evaluasi dengan menghitung nilai MAPE dan mendapatkan nilai akurasi dari hasil prediksi.

3.2 Metodologi dan Arsitektur Umum

Pada sub bab ini, akan membahas langkah-langkah metodologi penelitian yang akan dilakukan berdasarkan tujuan pada penelitian ini. Gambar 3.1 merupakan diagram alir yang digunakan untuk mencari model GRU terbaik.



Gambar 3.1 Diagram alir metodologi penelitian.



Gambar 3.2 Diagram alir arsitektur umum.

Pada Gambar 3.2 merupakan arsitektur umum dari langkah-langkah metodologi yang antara lain pengumpulan data, *preprocessing*, desain eksperimen GRU, serta prediksi dan evaluasi.

3.2.1 Pengumpulan data

Pada penelitian ini, disiapkan sebuah kumpulan data dengan mencari data pasang surut air laut di wilayah provinsi Lampung. Data pasang surut air laut yang

diperoleh dari Badan Informasi Geospasial (BIG) milik pemerintah Indonesia yang menyediakan data-data terkait geospasial di seluruh Indonesia. Data yang didapat berupa data pasang surut air laut setiap jam yang berada di pantai Krui, Kabupaten Pesisir Barat, yang direkam dari tanggal 1 Oktober 2017 sampai dengan 31 Desember 2018 dengan jumlah *record* sebanyak 10801. Data tersebut dengan format *file text* (TXT) dan data terdiri dari tiga kolom yaitu tanggal, pukul dan ketinggian dengan skala sentimeter. Format data tersebut akan dikonversi dalam bentuk *Comma Separated Value* (CSV) untuk mempermudah pemrosesan sistem pada tahap selanjutnya. Tabel 3.1 merupakan rincian dari dataset pasang surut yang didapatkan dan digunakan pada penelitian ini.

Tabel 3.1 Deskripsi *dataset* pasang surut air laut.

Jumlah	10.801
Rata-rata	103,81
Standar deviasi	34,93
Nilai terendah	13
Q1	78
Q2	102
Q3	128
Nilai tertinggi	235

Pada Tabel 3.1 terdapat jumlah data ketinggian air laut sebanyak 10.801, dengan nilai rata-rata ketinggian 103,81 cm, dan nilai standar deviasi dari *dataset* adalah 34,93. Nilai ketinggian air laut terendah dalam *dataset* adalah 13 cm dan paling tertinggi adalah 235 cm. Dalam *dataset* terdapat kuartil 1, 2 dan 3 yang masing-masing memiliki nilai yaitu 78, 102 dan 128.

3.2.2 Preprocessing Data

Pada tahap *preprocessing* data ini terdapat beberapa langkah sebagai berikut:

- a. *Data cleaning* merupakan salah satu metode untuk menghapus data yang tidak dibutuhkan dan mengisi data yang hilang. Pada penelitian ini, data yang didapat memiliki nilai yang konsisten dari alat pencatat pasang surut air laut. Data yang digunakan terlebih dahulu dicari kolom waktu yang hilang, selanjutnya dilakukan penambahan baris data, lalu mengisi kolom waktu serta mengatur ulang indeks data. Untuk mengisi data baru untuk ketinggian menggunakan Persamaan II-25.
- b. *Data transformation* merupakan cara menormalisasi data yang berfungsi untuk penyamaan format dalam bentuk skala umum. Dalam penelitian ini, digunakan Persamaan II-26.
- c. Pada penelitian ini dilakukan *Shift Data* untuk membuat sampel *input* yang digunakan pada tahap *training* dan *testing*. Berikut adalah contoh langkah *shift data* untuk data *training*:

1. Membuat kolom target untuk jam selanjutnya.

Tabel 3.2 merupakan contoh *dataset* yang digunakan proses *training* pada penambahan kolom target, dan hasilnya seperti pada tabel 3.3.

Tabel 3.2 Contoh *dataset* pasang surut air laut.

Waktu	Tinggi
2017-10-01 00:00:00	0.41
2017-10-01 01:00:00	0.37
2017-10-01 02:00:00	0.33
2017-10-01 03:00:00	0.34
2017-10-01 04:00:00	0.38
2017-10-01 05:00:00	0.42
2017-10-01 06:00:00	0.45
2017-10-01 08:00:00	0.46
2017-10-01 09:00:00	0.46

Tabel 3.3 Hasil *shift data* penambahan kolom target.

Waktu	Tinggi	T+1
2017-10-01 00:00:00	0.41	0.37
2017-10-01 01:00:00	0.37	0.33
2017-10-01 02:00:00	0.33	0.34
2017-10-01 03:00:00	0.34	0.38
2017-10-01 04:00:00	0.38	0.42
2017-10-01 05:00:00	0.42	0.45
2017-10-01 06:00:00	0.45	0.46
2017-10-01 08:00:00	0.46	0.46
2017-10-01 09:00:00	0.46	0.45

2. Menambahkan kolom untuk waktu sebelumnya.

Tabel 3.4 Hasil *shift data* penambahan kolom waktu sebelumnya.

Waktu	Tinggi	T+1	T-5	T-4	T-3	T-2	T-1	T
2017-10-01 00:00:00	0.41	0.37	NaN	NaN	NaN	NaN	NaN	0.41
2017-10-01 01:00:00	0.37	0.33	NaN	NaN	NaN	NaN	0.41	0.37
2017-10-01 02:00:00	0.33	0.34	NaN	NaN	NaN	0.41	0.37	0.33
2017-10-01 03:00:00	0.34	0.38	NaN	NaN	0.41	0.37	0.33	0.34
2017-10-01 04:00:00	0.38	0.42	NaN	0.41	0.37	0.33	0.34	0.38

Waktu	Tinggi	T+1	T-5	T-4	T-3	T-2	T-1	T
2017-10-01 05:00:00	0.42	0.45	0.41	0.37	0.33	0.34	0.38	0.42
2017-10-01 06:00:00	0.45	0.46	0.37	0.33	0.34	0.38	0.42	0.45
2017-10-01 08:00:00	0.46	0.46	0.33	0.34	0.38	0.42	0.45	0.46
2017-10-01 09:00:00	0.46	0.45	0.34	0.38	0.42	0.45	0.46	0.46

Tabel 3.4 merupakan hasil penambahan kolom untuk waktu sebelumnya dengan 5 waktu sebelumnya. Pada tabel tersebut terlihat NaN (*Not a Number*). Tabel 3.5 merupakan hasil penghapusan data NaN.

3. Menghapus nilai NaN pada kolom waktu sebelumnya dan kolom tinggi.

Tabel 3.5 Hasil penghapusan data NaN.

Waktu	T+1	T-5	T-4	T-3	T-2	T-1	T
2017-10-01 05:00:00	0.45	0.41	0.37	0.33	0.34	0.38	0.42
2017-10-01 06:00:00	0.46	0.37	0.33	0.34	0.38	0.42	0.45
2017-10-01 08:00:00	0.46	0.33	0.34	0.38	0.42	0.45	0.46
2017-10-01 09:00:00	0.45	0.34	0.38	0.42	0.45	0.46	0.46

3.2.3 Desain Eksperimen Gated Recurrent Unit

Data yang sudah melewati tahap *preprocessing*, selanjutnya akan digunakan untuk pembuatan sebuah model. Data tersebut akan dibagi menjadi tiga yaitu data

training, *validation* dan *testing* yang digunakan untuk menguji model *hyperparameter* yang berbeda. Data *validation* digunakan untuk mengevaluasi model setiap periode *training* (*epoch*) dan menentukan kategori model yang dihasilkan *overfitting* atau *underfitting*.

Dalam pembuatan sebuah model GRU, digunakan beberapa parameter diantaranya adalah jumlah *hidden layer*, *epoch* menggambarkan jumlah iterasi training, *batch size* adalah jumlah data training yang harus dipertimbangkan sebelum memperbarui bobot dalam satu *epoch* dan *optimizer* untuk mencari bobot terbaik untuk arsitektur jaringan akhir. Berikut adalah cara untuk menentukan *hidden layer*, *epoch*, *batch size* dan *optimizer*:

a. Penentuan *hidden layer*

Ada beberapa aturan untuk menentukan jumlah neuron yang dapat diterima untuk digunakan dalam *hidden layer*, seperti berikut [31]:

1. Jumlah *hidden neuron* harus antara ukuran *input layer* dan *output layer*.
2. Jumlah *hidden neuron* harus $2/3$ ukuran *input layer*, ditambah ukuran *output layer*.
3. Jumlah *hidden neuron* harus kurang dari dua kali jumlah *input layer*.

b. Penentuan *epoch*

Penentuan jumlah *epoch* tidak signifikan, karena yang lebih penting adalah *error training* dan *validation*. Jika *error validation* mulai meningkat yang mungkin sebagai indikasi *overfitting*, maka harus menetapkan jumlah *epoch* setinggi mungkin dan mengakhiri training berdasarkan tingkat *error* [32].

c. Penentuan *batch size*

Menentukan *batch size* tergantung dari spesifikasi atau kemampuan komputer yang digunakan. Semakin kecil *batch size* maka lebih mudah untuk memuat data *training* pada satu *batch* di dalam memori, akibatnya membutuhkan memori yang lebih besar jika menggunakan *batch size* yang lebih kecil.

d. Penentuan *optimizer*

Pada penelitian ini, *optimizer* yang digunakan adalah *Adaptive Learning Rate* (Adadelta), *Adaptive Moment Estimation* (Adam), Adamax, RMSprop dan *Stochastic Gradient Descent* (SGD).

Untuk mendapatkan hasil yang terbaik dari model GRU, proses *training* akan digunakan parameter yang berbeda-beda agar mendapatkan model yang terbaik. Dari sejumlah model yang dihasilkan akan dibandingkan serta dianalisis dari nilai *loss training* dan *loss validation*. Adapun ketentuan nilai parameter yang digunakan untuk pembuatan model GRU sebagai berikut:

Tabel 3.6 Parameter pengujian model GRU.

Parameter	Jumlah	Keterangan
<i>Input layer</i>	6	Data ketinggian pasang surut setiap jam
<i>Hidden layer</i>	<i>Trial error</i>	6 – 11 <i>neuron</i>
<i>Output</i>	1	Jumlah ketinggian jam berikutnya
<i>Epoch</i>	<i>Trial error</i>	50 – 500
<i>Batch Size</i>	<i>Trial error</i>	4 – 40
<i>Optimizer</i>	<i>Trial error</i>	Adadelata, Adam, Adamax, RMSProp, SGD

Pada Tabel 3.6 terdapat beberapa parameter yang akan digunakan pada penelitian ini. Pada input layer digunakan 6 layer, 6 layer diperoleh dari referensi tutorial GRU yang dilakukan oleh Chen, dkk. [33]. Eksperimen akan dilakukan setiap parameter, dimana hasil pengujian parameter yang telah didapat akan digunakan untuk pengujian berikutnya sehingga mendapatkan model yang terbaik. Tabel 3.7 adalah tahapan pengujian parameter.

Tabel 3.7 Tahapan pengujian parameter.

Tahap ke	Parameter
1	<i>Hidden Layer</i>
2	<i>Optimizer</i>
3	<i>Epoch</i>
4	<i>Batch Size</i>

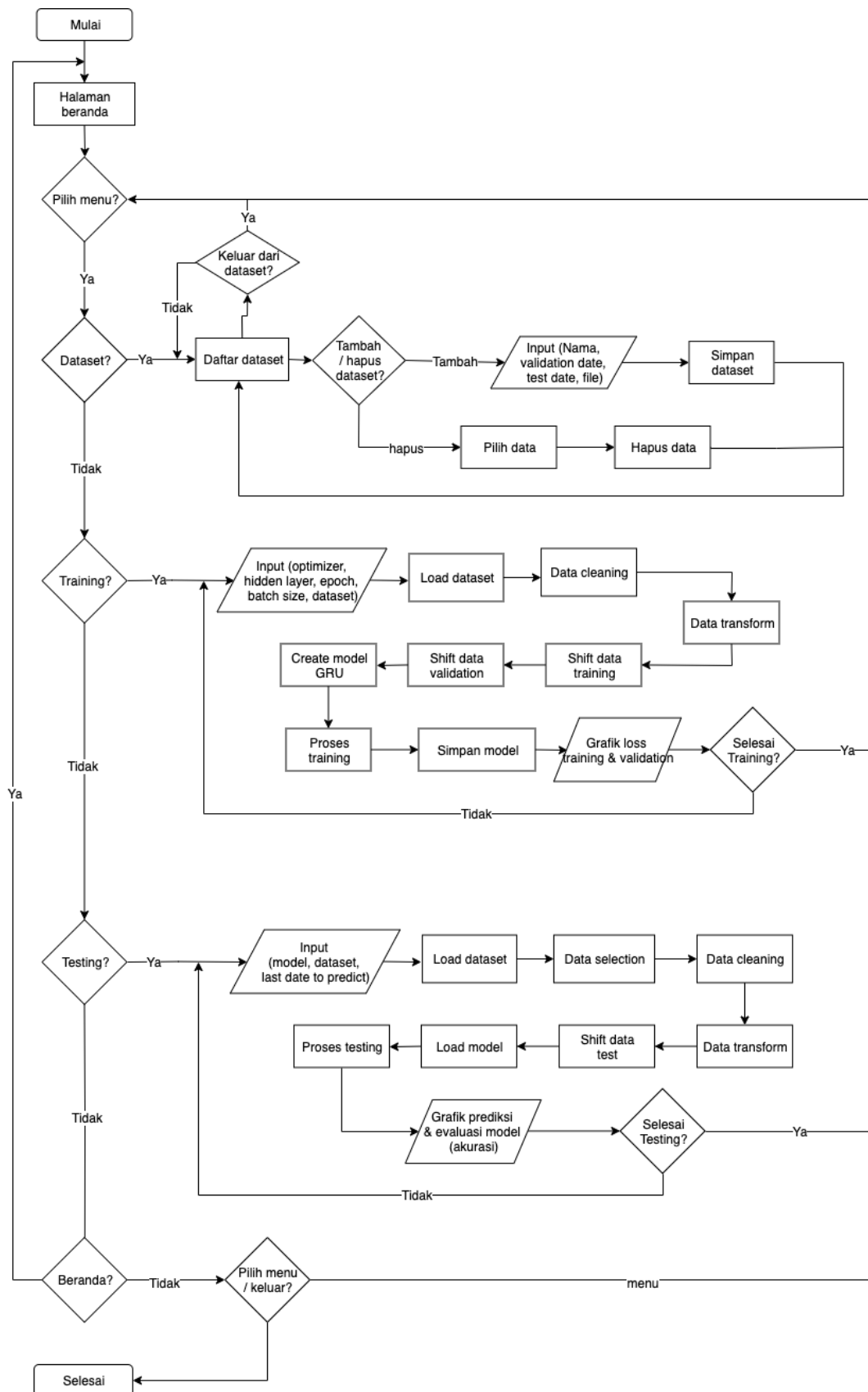
3.2.4 Evaluasi dan Hasil Prediksi

Tahap *testing* akan dilakukan proses pengujian untuk dapat menentukan keberhasilan model GRU terbaik yang telah didapat dari tahap *training*. Proses *testing* dilakukan dengan mengambil nilai bobot dan bias dari model terbaik, lalu diprediksi dengan perhitungan GRU yang menghasilkan prediksi ketinggian pasang surut, selanjutnya hasil prediksi akan dicari nilai ketepatan prediksi pada tahap evaluasi.

Pada tahap evaluasi merupakan tahap akhir dari sistem untuk mengetahui nilai ketepatan prediksi. Hasil prediksi dan data aktual akan dikalkulasi untuk mencari nilai ketepatan prediksi dengan Persamaan II-27, selanjutnya hasil nilai ketepatan prediksi digunakan untuk mendapatkan nilai akurasi. Penggunaan MAPE untuk evaluasi dikarenakan dapat menyatakan persentase kesalahan hasil peramalan terhadap permintaan aktual selama periode tertentu yang akan memberikan informasi persentase kesalahan terlalu tinggi atau terlalu rendah. Hasil prediksi akan menampilkan informasi hasil prediksi yang divisualisasikan berupa grafik garis. Hasil prediksi ketinggian pasang surut air laut akan ditampilkan dalam bentuk *timeseries* setiap jamnya.

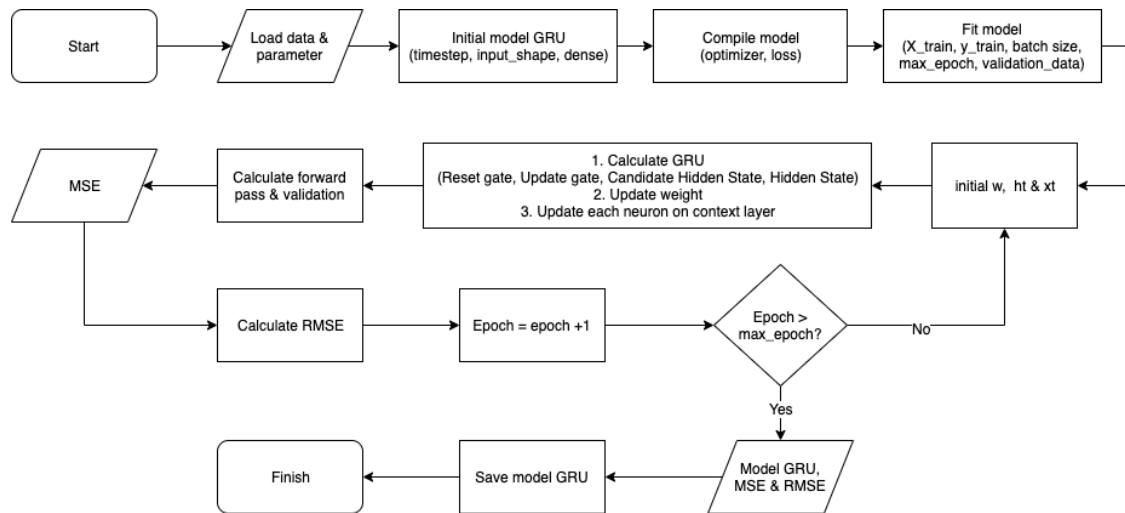
3.3 Rancangan Sistem

Pada penelitian ini, akan dibangun sebuah sistem berbasis *web* untuk mendukung kegiatan penelitian ini. Terdapat empat menu dalam rancangan sistem yaitu beranda, *dataset*, *training*, dan prediksi. Menu beranda merupakan halaman awal ketika pengguna membuka sistem, pada halaman ini hanya menampilkan judul penelitian. Menu *dataset* merupakan halaman daftar *dataset* yang akan digunakan untuk *training* dan prediksi yang pengguna unggah ke dalam sistem. Menu *training* merupakan halaman untuk melakukan proses *training* dan menampilkan hasilnya. Sedangkan, menu *testing* atau prediksi merupakan halaman untuk melakukan proses *testing* dan menampilkan hasil prediksi. Adapun diagram alir dari rancangan struktur sistem pada penelitian ini yang dapat dilihat pada Gambar 3.3.



Gambar 3.3 Diagram alir rancangan struktur sistem.

Pada Gambar 3.3 terdapat proses *training* yang secara detail digambarkan pada Gambar 3.4.



Gambar 3.4 Diagram alir proses *training*.

Gambar 3.4 menunjukkan proses *training*, *input* yang digunakan pada proses *training* yaitu hasil *preprocessing* data *training* dan *validation* serta parameter model digunakan pada tahap ini. Model akan disesuaikan dengan parameter yang akan di-*input* untuk membangun model. Proses *training* GRU akan menghasilkan jumlah *error* dengan kalkulasi MSE dan RMSE setiap *epoch*. Ketika proses *training* selesai maka model akan disimpan untuk digunakan pada tahap *testing*.

3.4 Kebutuhan Perangkat

3.4.1 Perangkat Keras

Tabel 3.8 merupakan spesifikasi perangkat keras yang akan digunakan pada implementasi penelitian ini.

Tabel 3.8 Spesifikasi Perangkat Keras.

No.	Perangkat Keras	Spesifikasi
1	<i>Processor</i>	Intel Core i5 2,3 GHz Quad-Core
2	RAM	8 GB 2133 MHz LPDDR3
3	<i>Storage</i>	256 GB SSD
4	<i>Graphics</i>	Intel Iris Plus Graphics 655 1536 MB

3.4.2 Perangkat Lunak

Tabel 3.9 merupakan spesifikasi perangkat lunak yang akan digunakan pada implementasi penelitian ini.

Tabel 3.9 Spesifikasi perangkat lunak.

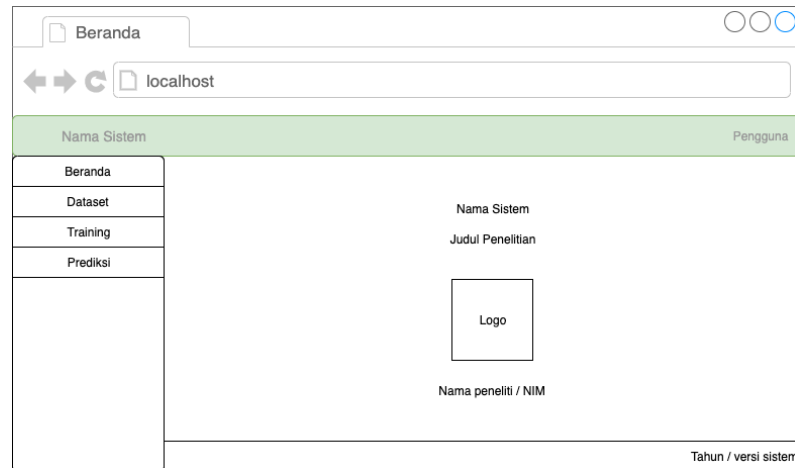
No.	Perangkat Lunak	Spesifikasi
1	Sistem Operasi	MacOS Catalina (10.15)
2	Bahasa Pemograman	Python 3.6.8
3.	<i>Framework</i>	Flask v1.1.1, Tensorflow v1.15.0, Keras v2.2.5, Pandas v0.25.2, Matplotlib v3.1.1, Numpy v1.17.3

3.5 Rancangan Antarmuka Prototipe

Rancangan antarmuka prototipe yang dibangun berbasis *website*. Terdapat beberapa halaman pada sistem yaitu beranda, *dataset*, *training*, dan prediksi.

3.5.1 Rancangan Halaman Beranda

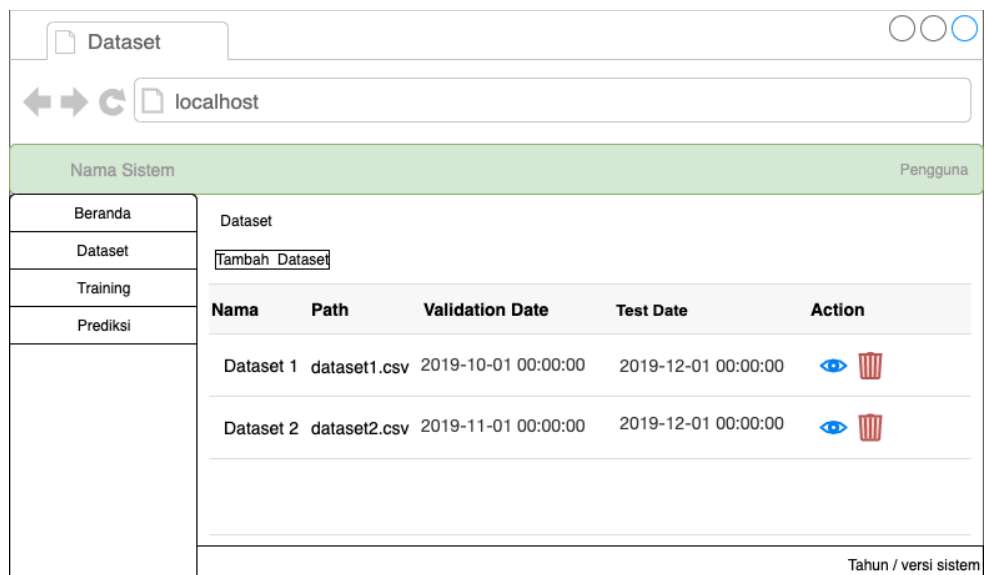
Halaman beranda merupakan halaman awal ketika pengguna membuka sistem pada *browser*. Halaman ini tidak mempunyai fungsi khusus, hanya menampilkan judul dari penelitian. Gambar 3.5 merupakan rancangan dari halaman beranda.



Gambar 3.5 Rancangan halaman beranda.

3.5.2 Rancangan Halaman Dataset

Halaman *dataset* menampilkan tabel *dataset* yang sudah diunggah dan sebelumnya telah dikonversi dalam bentuk CSV. Tabel *dataset* akan menampilkan nama, *path*, *validation date* dan *test date*. Pengguna dapat menambahkan dataset melalui tombol tambah dataset yang akan menampilkan form, pengguna akan mengunggah dataset serta mengisi informasi *validation date* dan *test date*. Pengguna dapat melihat detail informasi dataset dengan menekan tombol ikon mata pada kolom *Action*, serta menghapus dataset dengan menekan tombol ikon tong sampah. Gambar 3.6 merupakan rancangan halaman *dataset*.

Gambar 3.6 Rancangan halaman *dataset*.

3.5.3 Rancangan Halaman *Training*

Halaman *training* menampilkan *form* untuk melakukan *training* model. Untuk melakukan *training*, pengguna diminta memasukkan nilai parameter jumlah *hidden layer*, *epoch* dan *batch size* serta memilih *optimizer* dan *dataset* dalam bentuk *drop-down*, selanjutnya menekan tombol proses untuk *training*. Setelah proses *training* selesai, maka halaman *training* akan langsung *redirect* ke halaman hasil *training* untuk menampilkan hasil. Gambar 3.7 merupakan rancangan dari halaman *training*.

The image shows a web application interface for training a model. It features a sidebar menu on the left with the following items: Beranda, Dataset, Training (which is highlighted), and Prediksi. The main content area is titled 'Form Training' and contains the following fields:

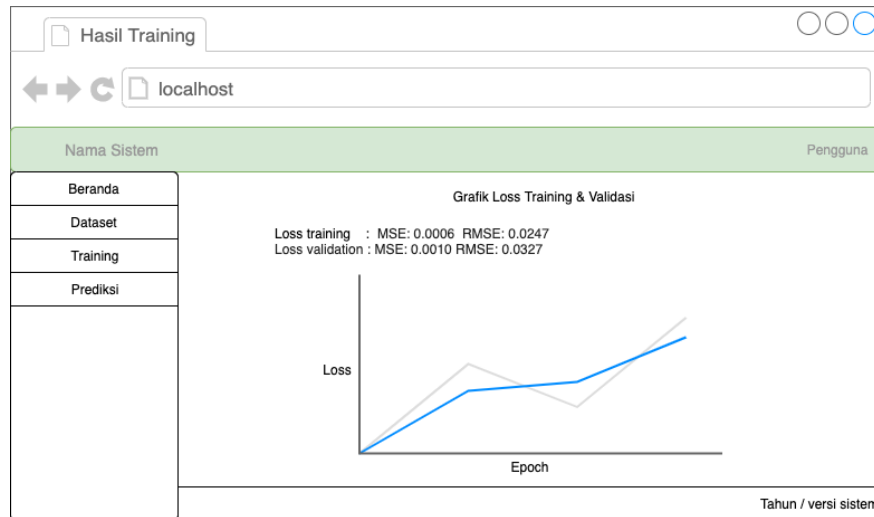
- Optimizer :** A dropdown menu with 'RMSProp' selected.
- Hidden Layer :** A text input field with 'Jumlah' as a placeholder.
- Epoch :** A text input field with 'Jumlah' as a placeholder.
- Batch Size :** A text input field with 'Jumlah' as a placeholder.
- Dataset :** A dropdown menu with 'Dataset 1' selected.

A green 'Proses' button is located at the bottom right of the form. The interface also includes a top navigation bar with 'Training' and 'Pengguna' tabs, and a browser address bar showing 'localhost'.

Gambar 3.7 Rancangan halaman *training*.

3.5.4 Rancangan Halaman Hasil *Training*

Halaman hasil *training* merupakan halaman setelah proses *training* model selesai, dalam halaman ini pengguna dapat melihat grafik *loss training* dan *loss validation* terhadap setiap *epoch*. Gambar 3.8 merupakan rancangan halaman hasil *training*.



Gambar 3.8 Rancangan halaman hasil *training*.

3.5.5 Rancangan Halaman Prediksi

Halaman prediksi menampilkan *form* untuk melakukan proses *testing* untuk mendapatkan prediksi pasang surut. Untuk melakukan testing, pengguna akan memilih model, *dataset* dan waktu prediksi terakhir dalam bentuk *drop-down*, selanjutnya menekan tombol proses untuk *testing* model. Setelah proses testing selesai, maka halaman prediksi akan langsung *redirect* ke halaman hasil prediksi untuk menampilkan hasil prediksi berupa grafik dan evaluasi model. Gambar 3.9 merupakan rancangan dari halaman prediksi.

The screenshot shows a web browser window titled 'Prediksi'. The address bar shows 'localhost'. The page has a green header with 'Nama Sistem' and 'Pengguna'. On the left is a sidebar menu with 'Beranda', 'Dataset', 'Training', and 'Prediksi'. The main content area is titled 'Form Prediksi'. It contains the following fields:

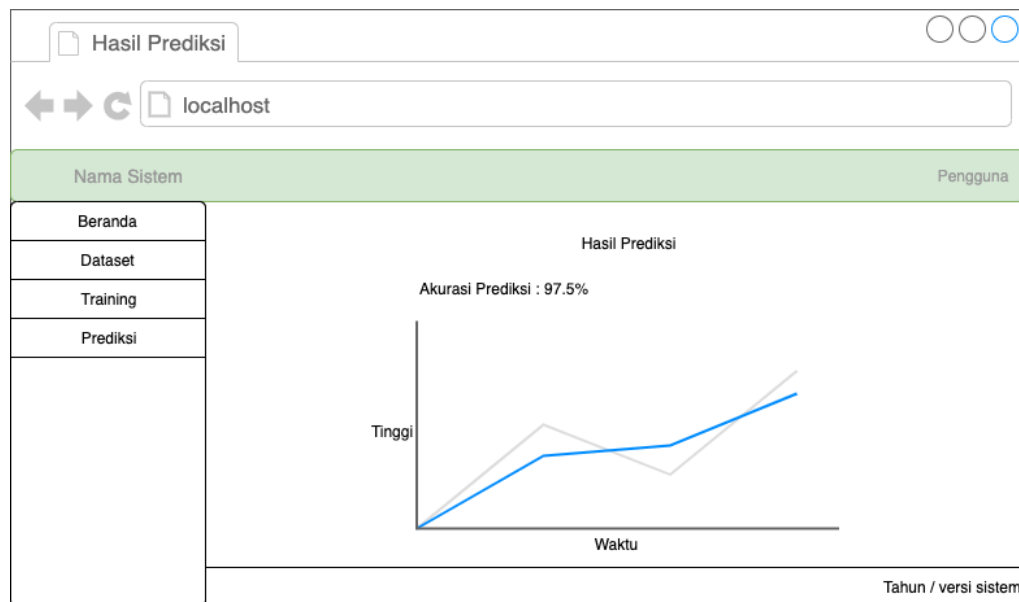
- Model**: A dropdown menu with the selected value 'KRUI-6-5-1-50-32-RMSProp.h5'.
- Dataset**: A dropdown menu with the selected value 'Dataset 1'.
- Waktu Tampil Prediksi Terakhir**: A text input field with the value '2019-12-10 00:00:00'.

At the bottom right of the form is a green button labeled 'Proses'.

Gambar 3.9 Rancangan halaman prediksi.

3.5.6 Rancangan Halaman Hasil Prediksi

Halaman hasil prediksi merupakan halaman setelah melakukan proses *testing* pada halaman prediksi untuk evaluasi model, dalam halaman ini akan menampilkan grafik data ketinggian air laut secara aktual dan prediksi terhadap setiap jam serta menampilkan nilai akurasi dari proses prediksi. Gambar 3.10 merupakan rancangan halaman hasil prediksi.



Gambar 3.10 Rancangan halaman hasil prediksi.