

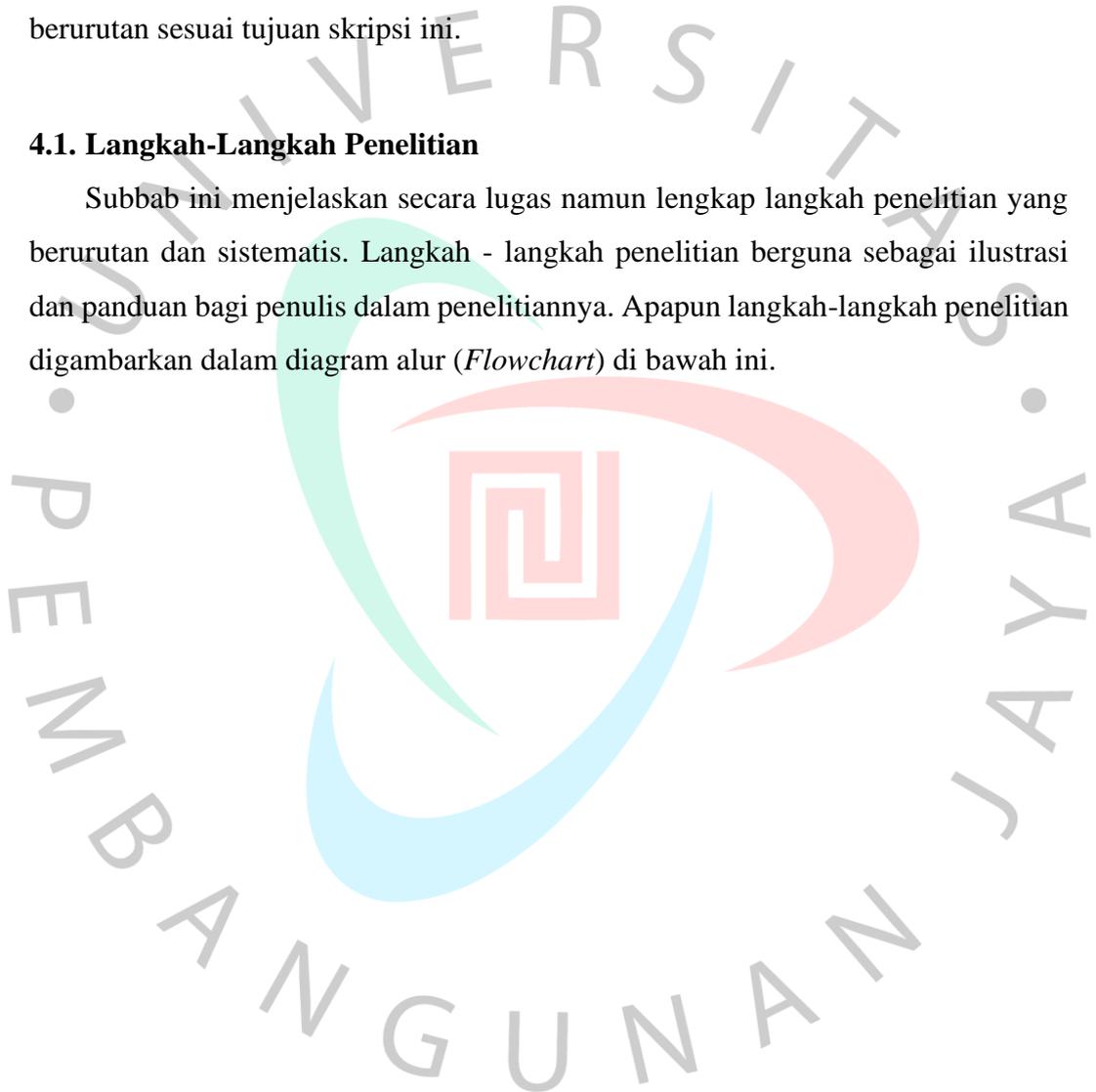
BAB IV

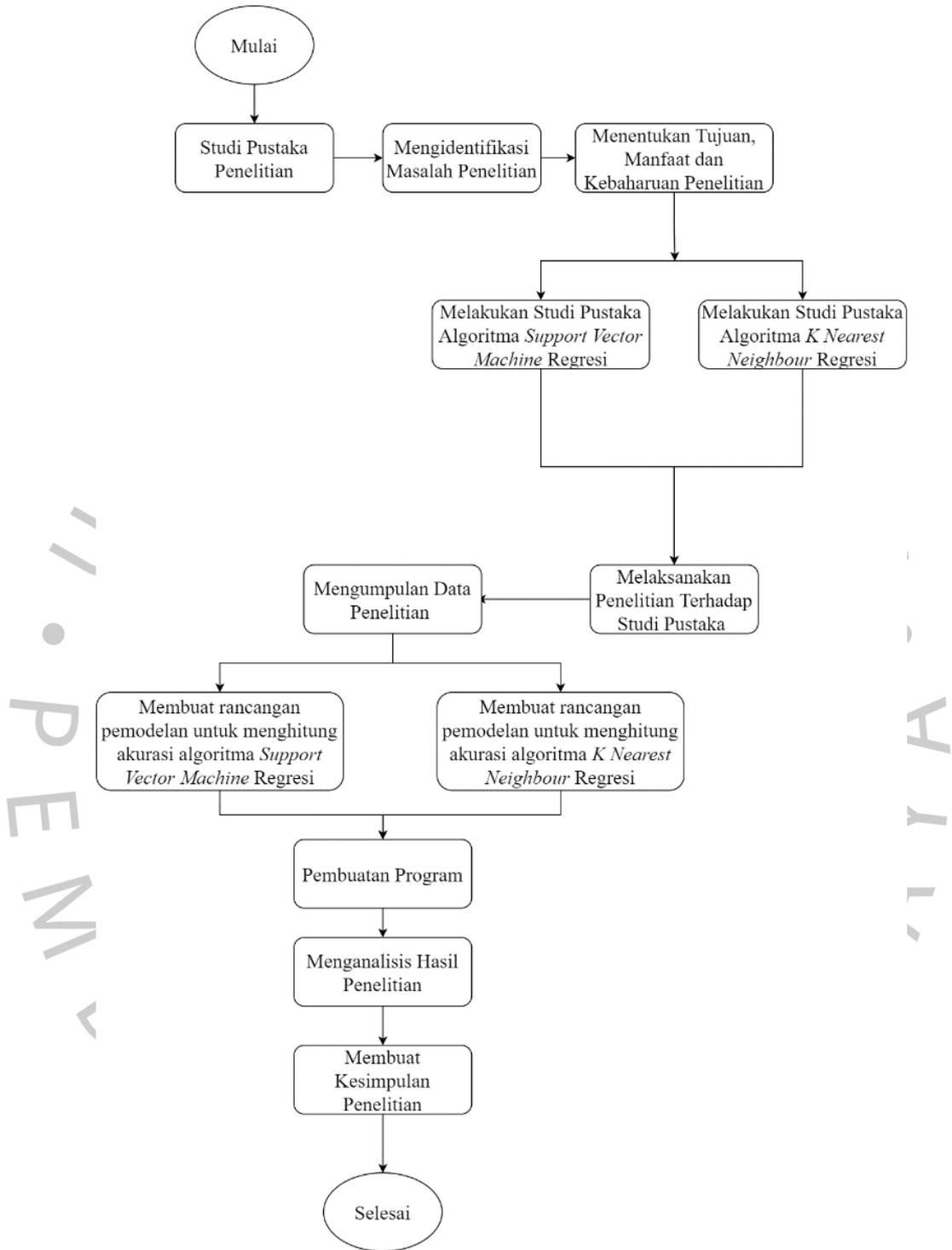
PERENCANAAN

Perencanaan pada penulisan skripsi dilakukan dengan membuat berbagai langkah maupun perencanaan penelitian. Setiap langkah dan rancangan yang sudah direncanakan bertujuan sebagai panduan untuk melaksanakan penelitian secara berurutan sesuai tujuan skripsi ini.

4.1. Langkah-Langkah Penelitian

Subbab ini menjelaskan secara lugas namun lengkap langkah penelitian yang berurutan dan sistematis. Langkah - langkah penelitian berguna sebagai ilustrasi dan panduan bagi penulis dalam penelitiannya. Apapun langkah-langkah penelitian digambarkan dalam diagram alur (*Flowchart*) di bawah ini.





Gambar 4. 1 Diagram *Flowchart* Langkah-Langkah Penelitian

Berdasarkan diagram *flowchart* di atas, langkah-langkah penelitian dapat diuraikan sebagai berikut.

1. Melakukan studi pustaka penelitian dengan mengidentifikasi isu dan masalah yang akan diteliti. Penelitian dibuat dengan melakukan riset pada jurnal, buku maupun skripsi yang telah terbit 5 tahun sebelum penelitian ini ditulis dan memiliki kredibilitas yang baik. Berdasarkan topik tersebut dapat menggambarkan masalah yang akan diteliti. Lalu dipersingkat dengan menentukan algoritma yang akan digunakan dan studi kasus penelitian.
2. Mengidentifikasi masalah penelitian dengan menentukan Rumusan dan membuat batasan Masalah yang akan diteliti.
3. Menentukan tujuan dari melaksanakan penelitian, manfaat untuk pemerintah, ilmu pengetahuan, dan peneliti serta kebaharuan berdasarkan penelitian terdahulu yang sesuai dengan topik penelitian.
4. Melakukan studi pustaka terkait algoritma *Support Vector Machine* dan *K Nearest Neighbor* seperti definisi, alur kerja, metode yang digunakan dan hasil akurasi pada algoritma dan lain-lain. Studi pustaka dilakukan sebagai bentuk pembelajaran untuk menemukan hubungan antara algoritma *Machine Learning* dan topik penelitian.
5. Melaksanakan studi pustaka terhadap kasus-kasus yang digunakan dalam penelitian, diperlukan sebagai bentuk pengetahuan bagi penulis tentang gambaran dan detail mengenai kasus COVID-19 di Indonesia.
6. Mengumpulkan data penelitian yang berupa *dataset* prediksi Kasus COVID-19 di Indonesia yang diperoleh dari halaman *web OurWorldInData.org* sesuai dengan batasan penelitian.
7. Membuat rancangan pemodelan dalam bentuk diagram *flowchart* berupa alur kerja algoritma *Support Vector Machine* dan *K Nearest Neighbor* untuk menjabarkan bagaimana algoritma itu berkerja untuk menyelesaikan kasus COVID-19 di Indonesia.
8. Membuat program dari algoritma *Support Vector Machine* dan *K Nearest Neighbor* dengan menggunakan bahasa pemrograman *Python* melalui aplikasi *Google Colab* dan dilakukan evaluasi model berdasarkan data akurasi yang diperoleh.

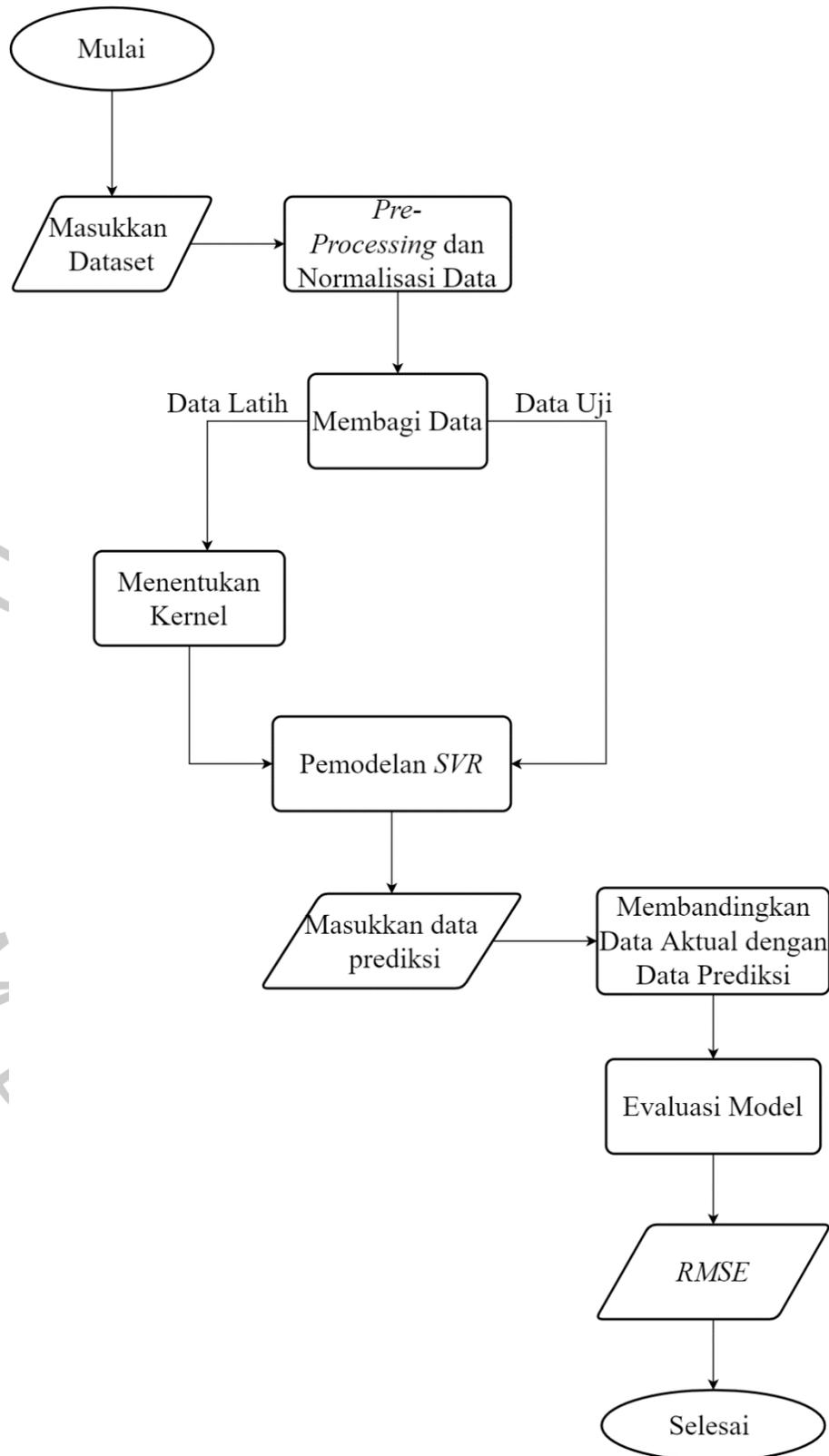
9. Menganalisis hasil penelitian diantara kedua algoritma berdasarkan evaluasi model yang dilakukan menggunakan *RMSE* untuk dilihat tingkat akurasi dan selisih *error* dari data tersebut.
10. Membuat kesimpulan berdasarkan hasil penelitian yang sudah didapatkan. Kesimpulan ini dapat mengetahui algoritma mana yang memiliki akurasi yang lebih tinggi dalam membuat pemodelan prediksi Kasus COVID-19 di Indonesia. Selain itu, pada bagian ini memberikan saran penelitian untuk peneliti selanjutnya untuk dijadikan sumber referensi agar dapat melakukan penelitian dengan topik yang sama.

4.2. Rancangan Pengujian

Subbab ini menyampaikan rancangan pemodelan untuk pengujian., sub bab ini akan berisi penjelasan tentang rancangan pemodelan untuk menguji akurasi algoritma *Support Vector Machine* dan algoritma *K Nearest Neighbor* pada kasus prediksi COVID-19 di Indonesia. Ini akan berupa diagram *flowchart* disertai dengan penjelasan secara deskriptif dan rinci atas setiap bagiannya.

4.2.1. Rancangan Algoritma *Support Vector Regression*

Support Vector Regression merupakan salah satu model algoritma *Support Vector Machine* yang umumnya digunakan untuk memprediksi suatu data dengan menggunakan suatu bidang dua dimensi (*Hyperplane*) untuk menguji antara data aktual dengan data prediksi. Algoritma tersebut memiliki rancangan pemodelan yang digambarkan menggunakan *flowchart* berikut ini.



Gambar 4. 2 Diagram Flowchart Support Vector Regression

Berdasarkan diagram *flowchart* pada algoritma *Support Vector Regression* pada Gambar 4.2, maka dapat diuraikan pemodelan sebagai berikut.

1. Masukkan *Dataset* kasus harian COVID-19 di Indonesia yang telah diunduh melalui *OurWorldInData.org* dalam satuan tanggal yang sesuai batasan penelitian dan disimpan pada folder di *Google Drive* berekstensi *comma sparated values (.csv)*.
2. *Dataset* yang telah dimasukkan akan dilakukan pembersihan data terlebih dahulu setelah itu lakukan normalisasi dengan menggunakan metode *Standard Scaler*. Penggunaan normalisasi digunakan untuk menormalkan seluruh angka kasus harian agar memiliki status nilai yang sama.
3. Dilakukan pembagian data menjadi data latih dan data uji (variabel x dan y). Pada penelitian ini data latih yang diinisialisasikan sebagai x merupakan kolom data $d(n-7)$, $d(n-6)$, $d(n-5)$, $d(n-4)$, $d(n-3)$, $d(n-2)$, $d(n-1)$ dan $d(n)$ yaitu data historis kasus dari 7 hari sebelumnya hingga data saat ini. Selain itu, data uji yang diinisialisasikan sebagai y berisikan kolom $d(n+1)$ yaitu data historis kasus yang akan diprediksi 1 hari selanjutnya. Data latih dan data uji yang sudah diinisialisasikan sebagai variabel x dan y Akan dibagi menjadi 3 buah rasio yaitu 90 : 10 , 80 : 20 dan 70 : 30.
4. Setelah dilakukan pembagian data, perlu melatih model SVR pada data latih. Fungsi SVM diimpor dan ditetapkan ke variabel *SVR_Model* menggunakan *kernel Linear* dengan nilai C parameter yaitu 1000 . Kernel Linear digunakan ketika data dapat dipisahkan secara Linier, yaitu dapat dipisahkan menggunakan satu Garis. Hal ini dilakukan karena data yang digunakan yaitu linear. Setelah itu, Fungsi *SVR_model.fit* digunakan untuk menyesuaikan variabel X_{train} dan y_{train} dengan membentuk kembali data yang sesuai.
5. Pemodelan SVR dilakukan untuk mengukur akurasi skor dari nilai X_{test} dan y_{test} untuk algoritma *Support Vector Regression*.
6. Membuat variabel baru yang berisikan Nilai yang akan diprediksi pada periode sehari selanjutnya.
7. Proses prediksi dilakukan untuk membandingkan nilai y_{test} sebagai Nilai Aktual dan nilai y_{pred} sebagai Nilai yang Prediksi untuk setiap X_{test} satu

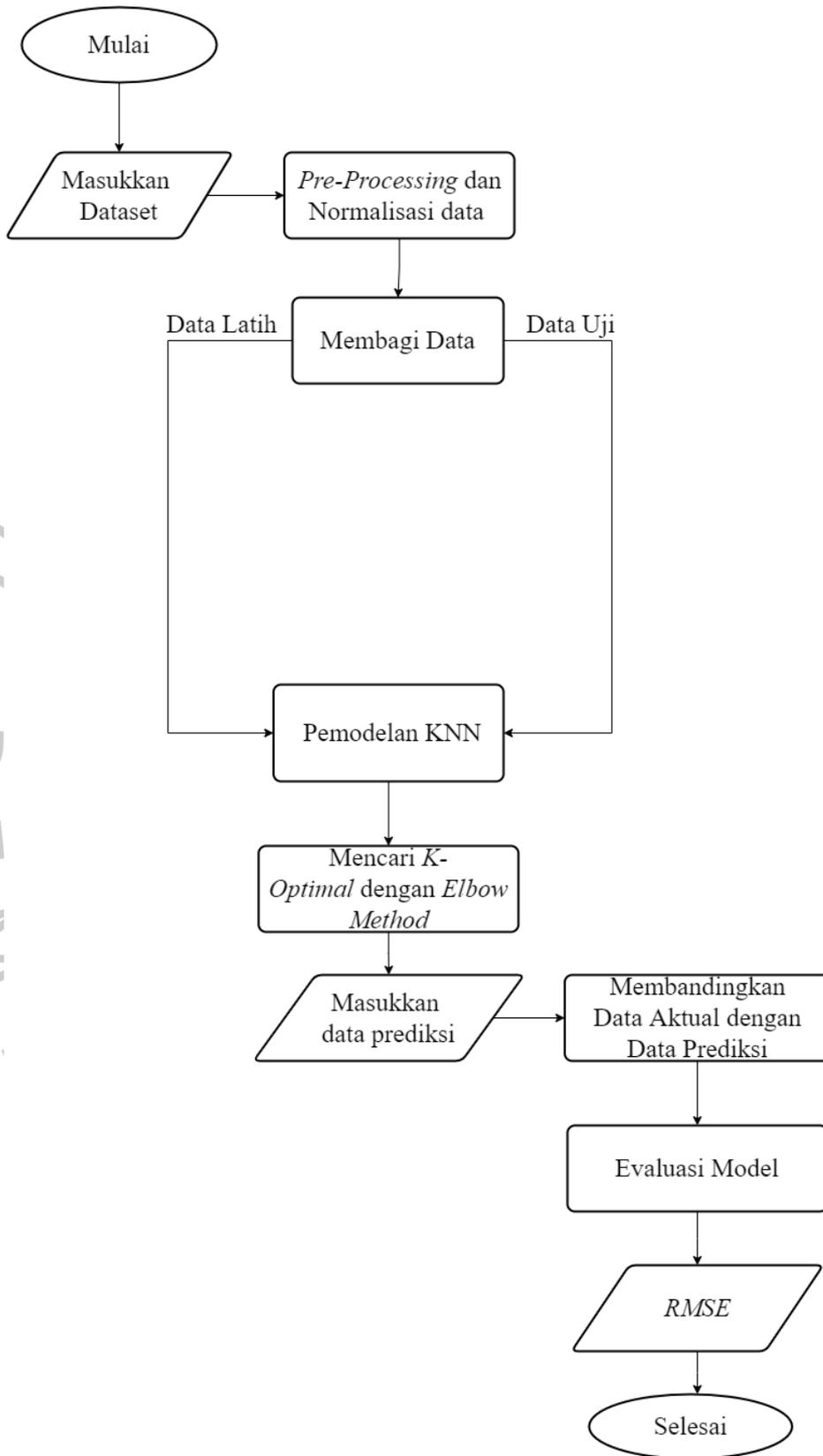
sama lain dalam *Pandas DataFrame* untuk melihat seberapa akurat prediksi menggunakan algoritma *Support Vector Regression*.

8. Evaluasi model menggunakan Akurasi dan *Root Mean Squared Error* (RMSE) Evaluasi dilakukan untuk melihat tingkat akurasi dan selisih *error* RMSE yang akan dibandingkan dengan algoritma *K Nearest Neighbor*.

4.2.2. Rancangan Algoritma *K Nearest Neighbor*

K-Nearest Neighbor adalah prosedur pemecahan *Machine Learning* yang digunakan buat pengklasifikasian dan Regresi Numerik terhadap data yang jaraknya paling dekat menggunakan objek (K-Optimal). Algoritma tersebut memiliki rancangan pemodelan untuk pengujian pada *flowchart* berikut ini.





Gambar 4. 3 Diagram *Flowchart K Nearest Neighbor*

Berdasarkan diagram *flowchart* pada algoritma *K Nearest Neighbor* pada Gambar 4.3, maka dapat diuraikan pemodelan sebagai berikut.

1. Masukkan *Dataset* kasus harian COVID-19 di Indonesia yang telah diunduh melalui *OurWorldInData.org* dalam satuan tanggal yang sesuai batasan penelitian dan disimpan pada folder di *Google Drive* berekstensi *comma sparated values (.csv)*.
2. *Dataset* yang telah dimasukkan akan dilakukan normalisasi dengan menggunakan metode *Standard Scaler*. Penggunaan normalisasi digunakan untuk menormalkan seluruh angka kasus harian agar memiliki status nilai yang sama.
3. Dilakukan pembagian data menjadi data latih dan data uji (variabel x dan y). Pada penelitian ini data latih yang diinisialisasikan sebagai x merupakan kolom data $d(n-7)$, $d(n-6)$, $d(n-5)$, $d(n-4)$, $d(n-3)$, $d(n-2)$, $d(n-1)$ dan $d(n)$ yaitu data historis kasus dari 7 hari sebelumnya hingga data saat ini. Selain itu, data uji yang diinisialisasikan sebagai y berisikan kolom $d(n+1)$ yaitu data historis kasus yang akan diprediksi 1 hari selanjutnya. Data latih dan data uji yang sudah diinisialisasikan sebagai variabel x dan y Akan dibagi menjadi 3 buah rasio yaitu $90 : 10$, $80 : 20$ dan $70 : 30$.
4. Pemodelan KNN dilakukan untuk mengukur akurasi skor dari nilai X_{test} dan y_{test} untuk algoritma *K Nearest Neighbor*.
5. Mencari Nilai K Optimal menggunakan metode *Elbow* untuk dilihat nilai K mana saja yang memiliki tingkat rata – rata *error* (MSE) yang rendah sehingga bisa digunakan sebagai acuan untuk menghitung jarak *Euclidean*.
6. Setelah dicari nilai K Optimal maka dicari nilai prediksi dari algoritma KNN. Fungsi `KNN_model.predict` ini digunakan untuk memprediksi nilai untuk X_{test} setelah itu nilai fungsi diinisialisasikan dengan nama y_{pred} . Sekarang kita memiliki dua data yaitu y_{test} (Nilai Aktual) dan y_{pred} (Nilai yang diprediksi).
7. Pemodelan Data KNN dilakukan untuk mengukur akurasi skor dari nilai X_{test} dan y_{test} untuk algoritma *K Nearest Neighbor*.
8. Buat variabel yang berisikan Nilai yang akan diprediksi untuk melakukan proses prediksi.

9. Proses prediksi dilakukan untuk membandingkan nilai y_{test} sebagai Nilai Aktual dan nilai y_{pred} sebagai Nilai yang Prediksi untuk setiap X_{test} satu sama lain dalam *Pandas DataFrame* untuk melihat seberapa akurat prediksi menggunakan algoritma *K Nearest Neighbor*.
10. Evaluasi model menggunakan Akurasi *Root Mean Squared Error* (RMSE) Evaluasi dilakukan untuk melihat tingkat akurasi dan selisih *error* RMSE yang akan dibandingkan dengan algoritma *Support Vector Regression*.

