G. Tinjauan Pustaka

Tinjauan pustaka ini terdiri dari beberapa jurnal sebagai referensi pelengkap guna terselesaikannya penelitian ini.

1. Penelitian Terkait

Adapun beberapa penelitian yang terkait dengan penelitian ini ditunjukkan pada Tabel 2.

Tabel 2. Penelitian Terkait

No	Peneliti	Judul Penelitian	Hasil
1	Didik Srianto, Edy Mulyanto (2016)	Perbandingan *K-Nearest Neighbor* dan *Naive Bayes* untuk Klasifikasi Tanah Layak Tanam Pohon Jati	Metode klasifikasi *K-Nearest Neighbor* di-anggap lebih baik dengan akurasi sebesar 96.66% di-bandingkan metode *Naive Bayes Classifier* dengan akurasi sebesar 82.63%.
2	Aida Indriani (2018)	Analisa Perbandingan Metode *Naive Bayes Classifier* dan *K-Nearest Neighbor* Terhadap Klasifikasi Data	Metode KNN lebih baik tingkat akurasinya daripada metode NBC. Hal ini dibuktikan dengan tingkat akurasi sebesar 80% untuk metode KNN dan sebesar 73% untuk NBC yang dihitung dengan menggunakan metode *confusion matrix*.

No	Peneliti	Judul Penelitian	Hasil
3	Yusra, Dhita Olivita, Yelfi Vitriani (2016)	Perbandingan Klasifikasi Tugas Akhir Mahasiswa Jurusan Teknik Informatika Menggunakan Metode *Naive Bayes Classifier* dan *K-Nearest Neighbor*	Pada seratus data tugas akhir dengan jumlah kelas acak, metode *Naive Bayes* menghasilkan nilai akurasi lebih baik, yaitu sebesar 87%. Pengujian pada metode *K-Nearest Neighbor* menghasilkan nilai akurasi 84% dengan nilai k=3, 85% dengan nilai k=5, 86% dengan nilai k=7 dan 84% dengan nilai k=9.
4	Paulus Dian Wicaksana (2015)	Perbandingan Algoritma *K-Nearest Neighbor* dan *Naive Bayes* untuk Studi Data "*Wisconsin Diagnosis Breast Cancer*"	Algoritma *Naive Bayes* mempunyai akurasi yang lebih akurat dengan hasil 97.7% dibandingkan algoritma *K-Nearest Neighbor* dengan hasil 95.8% dengan menggunakan evaluasi 10-*fold validation*.

H. Landasan Teori

Dalam penelitian ini juga terdapat beberapa landasan-landasan teori yang digunakan serta dijadikan sebagai acuan dalam penelitian ini antara lain:

1. ***Machine Learning***

Machine learning menyelidiki bagaimana komputer dapat belajar atau meningkatkan kinerjanya berdasarkan data. Area penelitian utama adalah agar program komputer belajar secara otomatis mengenali pola kompleks dan membuat keputusan cerdas berdasarkan data (Han dkk, 2011).

Machine learning memiliki dua teknik dasar belajar, yaitu *supervised learning* dan *unsupervised learning*. *Supervised learning* adalah metode klasifikasi di mana kumpulan data sepenuhnya diberikan label untuk mengklasifikasikan kelas yang tidak dikenal. *Supervised learning* dikelompokkan lebih lanjut dalam masalah klasifikasi dan regresi. Masalah klasifikasi adalah ketika variabel output berbentuk kategori, seperti merah atau biru atau penyakit dan tidak ada penyakit. Sedangkan masalah regresi adalah ketika variabel *output* adalah nilai riil, seperti *dollar* atau berat. Metode yang populer digunakan dalam *supervised learning,* yaitu *K-Nearest Neighbor* (KNN), *Naive Bayes Classifier* (NBC), *Support Vector Machine* (SVM), *Decision Tree*, dan lain-lain.

Sedangkan *unsupervised learning* sering disebut *cluster* atau pengelompokkan dikarenakan tidak ada kebutuhan untuk pemberian label dalam kumpulan data. *Unsupervised learning* dikelompokkan lebih lanjut dalam masalah *clustering* dan asosiasi. Masalah pengelompokan (*clustering*) adalah tempat untuk menemukan

pengelompokan yang melekat dalam data, seperti mengelompokkan pelanggan berdasarkan pada perilaku pembelian. Sedangkan masalah asosiasi adalah aturan yang menggambarkan sebagian besar data yang ada, seperti orang yang membeli A juga cenderung membeli B. Metode yang populer digunakan dalam *unsupervised learning* seperti *K-Means* dan *Apriori* (Roihan dkk, 2020).

2. ***Data Mining***

Data mining adalah penambangan atau penemuan informasi baru dengan mencari pola atau aturan tertentu dari sejumlah data yang sangat besar. *Data mining* juga disebut sebagai serangkaian proses untuk menggali nilai tambah berupa pengetahuan yang selama ini tidak diketahui secara manual dari suatu kumpulan data (Bustami, 2014).

3. ***K-Nearest Neighbor***

Klasifikasi adalah proses untuk menemukan model atau fungsi yang menjelaskan atau membedakan konsep atau kelas data, dengan tujuan untuk dapat mengklasifikasi kelas dari suatu objek yang labelnya tidak diketahui. Dalam mencapai tujuan tersebut, proses klasifikasi membentuk suatu model yang mampu membedakan data ke dalam kelas-kelas yang berbeda berdasarkan aturan atau fungsi tertentu (Bustami, 2014).

K-Nearest Neighbor merupakan sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Cara kerja dari *K-*

Nearest Neighbor perlu adanya penentuan inputan berupa data latih (*data training*), data uji (*data testing*), dan nilai k. Kemudian mengurutkan data latih berdasarkan kedekatan jaraknya, berdasarkan hitungan dari jarak data yang diuji dengan data latih. Dari k tetangga terdekat yang terpilih dilakukan voting dengan memilih kelas yang jumlahnya paling banyak sebagai label kelas hasil prediksi pada data uji (Sani dkk, 2016).

Ada banyak cara untuk mengukur jarak kedekatan antara data baru (*data testing*) dengan data lama (*data training*), diantaranya *euclidean distance* dan *manhattan distance* (Sumarlin, 2015), yang paling sering digunakan adalah *euclidean distance*. Persamaan *euclidean distance* ditunjukkan pada persamaan 1.

$$euc = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \cdots + (a_n - b_n)^2} \quad \text{.....(1)}$$

Dimana a = a_1,a_2, …, a_n, dan b = b_1, b_2, …, b_n mewakili n nilai atribut dari dua record.

Sedangkan persamaan *manhattan distance* ditunjukkan pada persamaan 2 (Halim, 2020).

$$D = \sum_{i=1}^{n} |\text{xi} - \text{yi}| \quad \text{.....(2)}$$

Keterangan:

D : jarak *manhattan* (*manhattan distance*)

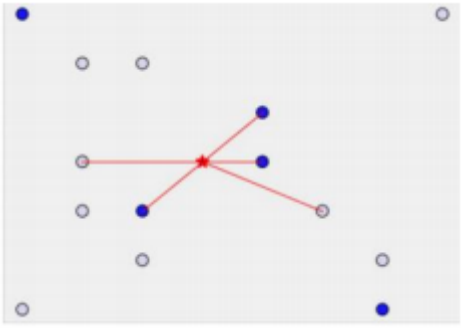
x : sampel data

y : data uji (*data testing*)

i : variabel data

n : dimensi data

Masalahnya, sampai saat ini k tidak dapat ditentukan secara matematik. Jadi proses pelatihan pada dasarnya adalah melakukan observasi terhadap sejumlah k sampai dihasilkan k yang paling optimum (Pamungkas dkk, 2020). Visualisasi dari *K-Nearest Neighbor* dapat dilihat pada Gambar 1.



Gambar 1. Visualisasi *K-Nearest Neighbor*

4. ***Naive Bayes Classifier***

Metode *Naive Bayes* merupakan salah satu metode yang terdapat pada teknik klasifikasi. *Naive Bayes* dikemukan oleh ilmuwan Inggris Thomas Bayes, yaitu memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya sehingga dikenal sebagai *Teorema Bayes* (Bustami, 2014).

Naive Bayes Classifier merupakan sebuah metode pengklasifikasian berdasarkan probabilitas sederhana dan dirancang untuk dipergunakan dengan asumsi bahwa antar satu kelas dengan kelas yang lain tidak saling tergantung (independen). Pada klasifikasi *Naive Bayes*, proses pembelajaran lebih ditekankan pada mengestimasi

probabilitas (Putri dkk, 2014). Dalam ilmu statistik, probabilitas bersyarat dinyatakan pada persamaan 3.

$$P(C|\mathrm{X}) = \frac{P(x|\mathrm{c})P(c)}{P(x)} \text{.....(3)}$$

Keterangan:

x : data dengan *class* yang belum diketahui

c : hipotesis data merupakan suatu *class* spesifik

P(c|x) : probabilitas hipotesis berdasar kondisi (*posteriori probability*)

P(c) : probabilitas hipotesis (*prior probability*)

P(x|c) : probabilitas berdasarkan kondisi pada hipotesis

P(x) : probabilitas c

Selain persamaan seperti di atas, metode *Naive Bayes Classifier* juga dapat menangani data berupa numerik. Untuk menangani data numerik, metode *Naive Bayes Classifier* menggunakan asumsi distribusi normal. Persamaan distribusi normal ditunjukkan pada persamaan 4:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \text{.....(4)}$$

Untuk mendapatkan *mean* menggunakan persamaan 5:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i \text{.....(5)}$$

Untuk mendapatkan *standard deviation* menggunakan persamaan 6:

$$\sigma = \left[\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2\right]^{0.5} \text{.....(6)}$$

Keterangan:

x_i : sampel data

n : jumlah sampel

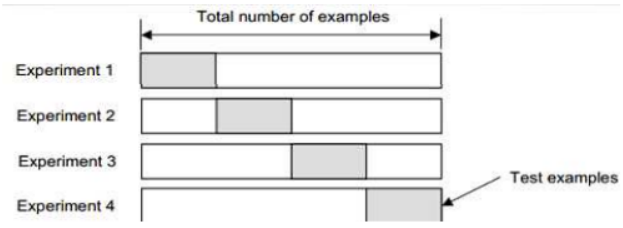
μ : nilai rata-rata (*mean*)

σ : standar deviasi

5. ***Cross Validation***

Cross validation adalah metode statistik untuk mengevaluasi dan membandingkan algoritma pembelajaran (*learning algorithms*) dengan membagi data menjadi dua segmen, yaitu data latih (*data training*) untuk belajar dan data uji (*data testing*) digunakan untuk memvalidasi model. Dalam *cross validation* kumpulan pelatihan dan validasi harus *crossover* berturut-turut sehingga setiap data memiliki kesempatan tervalidasi (Saifudin, 2018).

K-fold cross validation adalah teknik untuk mengestimasi performansi dari model pelatihan yang telah dibangun. Metode ini membagi *data training* dan *data testing* sebanyak k bagian data. Fungsi dari *k-fold cross validation* adalah agar tidak ada *overlapping* pada *data testing* (Sasongko, 2016). Berikut merupakan ilustrasi dari *k-fold cross validation* yang ditunjukkan pada Gambar 2.



Gambar 2. Ilustrasi *K-fold Cross Validation*

K-*fold cross validation* dilakukan dengan menggunakan kembali dataset yang sama, sehingga menghasilkan k perpecahan dari kumpulan data menjadi *non-overlapping* dengan proporsi pelatihan (k-1)/k dan 1/k untuk pengujian (Saifudin, 2018).

6. **Deteksi Tepi Metode *Sobel***

Metode *sobel* adalah salah satu algoritma yang digunakan dalam mendeteksi tepi pada saat proses pengolahan citra. Tujuan pendeteksian tepi ini yaitu untuk meningkatkan penampakan garis batas suatu daerah atau objek di dalam citra (Halim, 2020).

Untuk mendeteksi tepi dengan metode *sobel*, menggunakan gradien G(x,y), yang merupakan sebuah vektor yang terdiri dari dua unsur yaitu Gx dan Gy. Deteksi tepi dilakukan dengan cara membaca setiap *pixel* pada citra dengan cara membaca dari *pixel* paling kiri atas (timur utara) dan bergerak ke *pixel* paling kanan bawah (barat selatan). Oleh karena itu, untuk membantu penelusuran tepi, gradien Gx dan Gy masing-masing dihitung dengan *matrix* metode *sobel mask* 3x3 (Zalukhu, 2016).

7. **Ekstraksi Fitur *Moment Invariant***

Moment invariant merupakan proses untuk menghasilkan nilai-nilai fitur berupa vektor dari citra biner. Fitur yang digunakan yaitu *seven moment invariant* yang akan menghasilkan tujuh nilai pada vektor fitur (Liantoni, 2016). Vektor fitur tersebut kemudian digunakan untuk tahap klasifikasi.

Proses perhitungan nilai *HuMoment* adalah sebagai berikut (Halim, 2020):

a. Menghitung momen orde 0 (m_{00}) dan momen orde 1 (m_{10} dan m_{01}) dengan persamaan berikut:

$$m_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} x^p y^q \, f(x,y) \quad \text{.....(7)}$$

Keterangan:

M : jumlah baris/resolusi panjang citra

N : jumlah kolom/resolusi lebar citra

p,q : orde momen

f (x,y): intensitas pixel dititik x,y

b. Menghitung pusat kordinat dari area atau massa ($\bar{x}$, $\bar{y}$) dengan persamaan sebagai berikut:

$$\bar{x} = \frac{m_{10}}{m_{00}} \quad , \quad \bar{y} = \frac{m_{10}}{m_{00}} \text{.....(8)}$$

c. Menghitung momen pusat (μ_{pq}) orde 2 (μ_{11}, μ_{20}, μ_{02}) dan orde 3 (μ_{21}, μ_{12}, μ_{30}, μ_{03}) dengan persamaan berikut:

$$\mu_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (x - \bar{x})^p (y - \bar{y})^q \, f(x,y) \text{.....(9)}$$

d. Menghitung normalisasi momen pusat (μ_{pq}) orde 2 (μ_{11}, μ_{20}, μ_{02}) dan orde 3 (μ_{21}, μ_{12}, μ_{30}, μ_{03}) dengan persamaan berikut:

$$\mu_{pq} = \frac{\mu_{pq}}{\mu_{00\gamma}} \quad \gamma = \frac{p+q}{2} + 1 \text{.....(10)}$$

e. Setelah menghitung normalisasi momen pusat (μ_{pq}), nilai *HuMoment* dihitung dengan persamaan berikut:

$$Hu_1 = \eta_{20} + \eta_{02}$$

$$Hu_2 = (\eta_{20} + \eta_{02})^2 + 4{\eta_{11}}^2$$

$$Hu_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$$

$$Hu_4 = (\eta_{30} + \eta_{12})^2 + (3\eta_{21} + \eta_{03})^2$$

$$Hu_5 = (\eta_{30} + 3\eta_{12})(\eta_{30} + \eta_{12})\,[(\eta_{30} + \eta_{12})^2 - 3(\eta_{12} + \eta_{03})^2] + (3\eta_{21} + \eta_{03})(\eta_{21} + \eta_{03})\,[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

$$Hu_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]\,4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$

$$Hu_7 = (3\eta_{21} - \eta_{30})(\eta_{30} + \eta_{12})\,[(\eta_{30} + \eta_{12})^2 - 3(\eta_{12} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})\;[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \quad \text{.....(11)}$$

8. **Pemrograman Python**

Bahasa pemrograman Python ini pertama kali dibuat oleh Guido van Rossum tahun 1990 di negeri Belanda. Bahasa pemrograman Python merupakan bahasa pemrograman yang dapat dikembangkan oleh siapa saja karena besifat *open source* atau dengan kata lain bahasa pemrograman ini gratis, dapat digunakan tanpa lisensi, dan dapat dikembangkan semampu yang dapat dilakukan (Saptono dkk, 2013).

9. ***Confusion Matrix***

Confusion matrix adalah *tool* yang digunakan untuk evaluasi model klasifikasi untuk memperkirakan objek yang benar atau salah. Sebuah *matrix* dari prediksi yang akan dibandingkan dengan kelas yang asli dari inputan atau dengan kata lain berisi informasi nilai aktual dan prediksi pada klasifikasi (Mustakim, 2016).

Pada Tabel 3, nilai TP (*true positive*) dan TN (*true negative*) menunjukkan tingkat ketepatan klasifikasi. Jika label prediksi keluaran bernilai benar (*true*) dan nilai sebenarnya bernilai salah (*false*) disebut sebagai FP (*false positive*). Sedangkan jika prediksi label keluaran bernilai salah (*false*) dan nilai sebenarnya bernilai benar (*true*) maka hal ini disebut sebagai FN (*false negative*) (Sasongko, 2016).

Tabel 3. Tabel *Confusion Matrix*

		Nilai Sebenarnya	
		True	***False***
Nilai Prediksi	***True***	TP (*True Positive*)	FP (*False Positive*)
	False	FN (*False Negative*)	TN (*True Negatif*)

10. **Performa**

Confusion matrix akan menguji hasil performa sebuah metode klasifikasi berupa akurasi, presisi, *recall*, dan *f-measure*.

a. Akurasi (*Accuracy*)

Akurasi didefinisikan sebagai tingkat kedekatan antara nilai prediksi dengan nilai aktual (Amelia dkk, 2017). Persamaan akurasi ditunjukkan pada persamaan 12:

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad \text{.....(12)}$$

b. Presisi (*Precision*)

Presisi menunjukan tingkat ketepatan atau ketelitian dalam pengklasifikasian. Persamaan presisi ditunjukkan pada persamaan 13:

$$Precision = \frac{TP}{TP+FP} \quad \text{.....(13)}$$

c. *Recall*

Recall berfungsi untuk mengukur proporsi positif aktual yang benar diidentifikasi (Sasongko, 2016). Persamaan *recall* ditunjukkan pada persamaan 14:

$$Recall = \frac{TP}{TP+FN} \quad \text{.....(14)}$$

d. *F-measure*

F-measure adalah *harmonic mean* antara nilai presisi dan *recall*, *f-measure* juga kadang disebut dengan nama *F1-Score* (Baharuddin dkk, 2019). Persamaan *f-measure* ditunjukkan pada persamaan 15:

$$F - measure = 2\frac{presisi\ x\ recall}{presisi+recall} \quad \text{.....(15)}$$

Keterangan:

TP = *True Positive*

TN = *True Negative*

FP = *False Positive*

FN = *False Negative*

11. ***Receiver Operating Characteristic-Area Under the Curve of* (ROC-AUC)**

ROC menunjukkan akurasi dan membandingkan klasifikasi secara visual. ROC mengekspresikan *confusion matrix*. ROC adalah grafik dua dimensi dengan *false positive* sebagai garis horizontal dan *true positive* sebagai garis vertikal. AUC dihitung untuk mengukur perbedaan performansi metode yang digunakan (Rosandy, 2016).

I. Metodologi Penelitian

1. Tahapan Penelitian

a. Pengumpulan Data

Pengumpulan data dan informasi sebagai acuan dalam melakukan penelitian, diantaranya adalah studi pustaka dan dataset dari *repository* Kaggle, yang mana data tersebut diunggah pada tanggal 20 April 2020 dan diperbarui pada tanggal 24 September 2020. Data diolah oleh Shanggong Medical Technology Co., Ltd. dari berbagai rumah sakit dan pusat medis di China.

b. Analisis Data

Pada tahap analisis data, kegiatan yang dilakukan adalah menganalisis seluruh data yang telah diperoleh, yaitu menganalisis *missing value* sebagai acuan pada tahap *preprocessing* citra yang akan dilakukan selanjutnya.

c. Preprocessing Data

Pada tahap ini menggunakan model kombinasi *sobel* sebagai segmentasi citra dan *moment invariant* sebagai ekstraksi fitur.

d. Implementasi Metode

Pada penelitian ini, metode klasifikasi yang digunakan adalah *K-Nearest Neighbor* dan *Naive Bayes Classifier*.

e. Perhitungan Performa

Pada tahap ini akan menguji hasil performa sebuah metode klasifikasi berupa akurasi, presisi, *recall*, *f-measure*, dan ROC-AUC.

f. Pengambilan Kesimpulan

Kesimpulan diambil berdasarkan hasil perbandingan performa (akurasi, presisi, *recall*, *f-measure*, ROC-AUC) metode *K-Nearest Neighbor* dan *Naive Bayes Classifier* dalam mengklasifikasi dataset *multiclass* penyakit.

2. **Instrumen Penelitian**

Instrumen penelitian adalah alat bantu yang dipilih dalam kegiatannya agar sistematis dan mempermudah peneliti selama melakukan penelitian. Instrumen ini terbagi menjadi dua yaitu:

a. Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan adalah :

1. *Processor* Intel® Core ™ i7-2630QM CPU @ 2.00GHz.
2. RAM 6,00 GB.

b. Perangkat Lunak (*Software*)

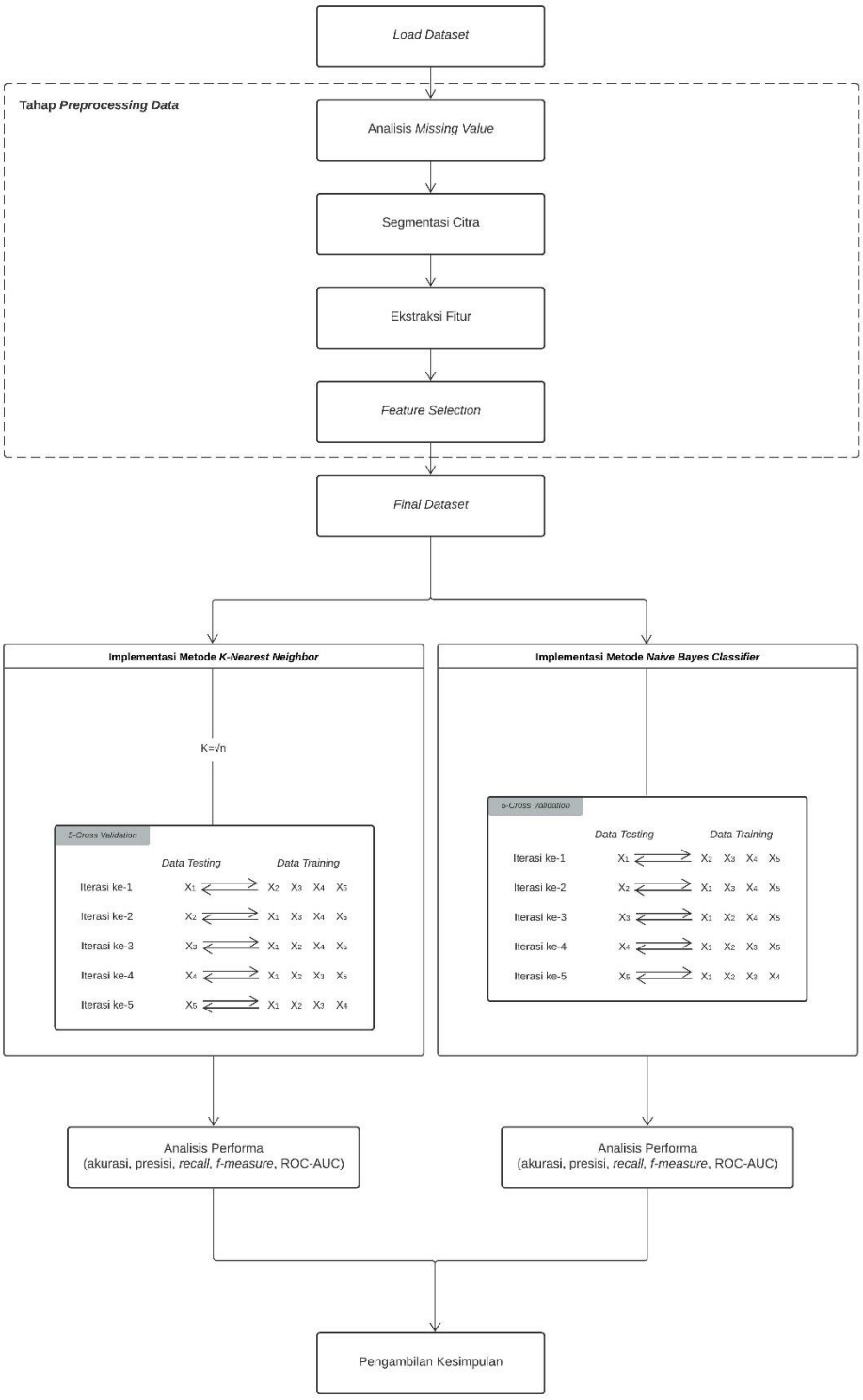
Perangkat lunak atau *software* yang digunakan adalah :

1. Microsoft Windows 10 Pro 64-bit, sebagai sistem operasi.
2. Python sebagai bahasa pemrograman.
3. *Scikit learn* sebagai *library machine learning*.
4. *Open CV* sebagai *library* pengolah data citra
5. *Browser*
6. Kaggle *Repository* sebagai tempat penyimpanan data.
7. Kaggle *Kernel* sebagai *tools* mengolah data.

3. **Lokasi Penelitian**

Lokasi penelitian ini dilakukan di Fakultas Ilmu Komputer Universitas Muslim Indonesia.

4. Metode Penelitian



Gambar 3. Alur Perancangan Proses

Metode yang akan digunakan dalam penelitian yaitu metode kuantitatif yang dipaparkan sebagai berikut:

a. Analisis Data

Data yang dikumpulkan akan dilakukan analisis *missing value*, yaitu pengecekan hilangnya suatu informasi dari data karena alasan-alasan tertentu.

b. *Preprocessing Data*

Setelah data dianalisis, maka data tersebut akan masuk ke tahap *preprocessing data*. Tahap pertama adalah melakukan segmentasi citra dengan menggunakan metode *sobel*.

Setelah melakukan segmentasi citra, tahap kedua adalah melakukan ekstraksi fitur dengan menggunakan *moment invariant* untuk mengkonversi data citra menjadi data numerik, dimana hasilnya berupa 8 nilai array, yaitu fitur yang diberi label H1 sampai H7 dan target.

Setelah melakukan ekstraksi fitur, data tersebut dilakukan *feature selection*, dimana fitur-fitur yang tidak digunakan dalam memproses data akan dihapus. *Final dataset* tersebut akan digunakan untuk tahap klasifikasi.

c. Implementasi *K-Nearest Neighbor* dan *Naive Bayes Classifier*

Pada metode *K-Nearest Neighbor*, untuk menghitung jarak kedekatan antara data lama (*data training*) dan data baru (*data*

testing) menggunakan *euclidean distance* pada persamaan 1 dan *manhattan distance* pada persamaan 2.

Sedangkan pada metode *Naive Bayes Classifier* menggunakan persamaan data numerik pada persamaan 4, 5, dan 6.

d. Perhitungan Performa

Pada tahap ini akan menguji hasil performa sebuah metode klasifikasi menggunakan *confusion matrix*, performa tersebut berupa akurasi, presisi, *recall*, *f-measure*, dan ROC-AUC. Akurasi menggunakan persamaan 12, presisi menggunakan persamaan 13, *recall* menggunakan persamaan 14, dan *f-measure* menggunakan persamaan 15.