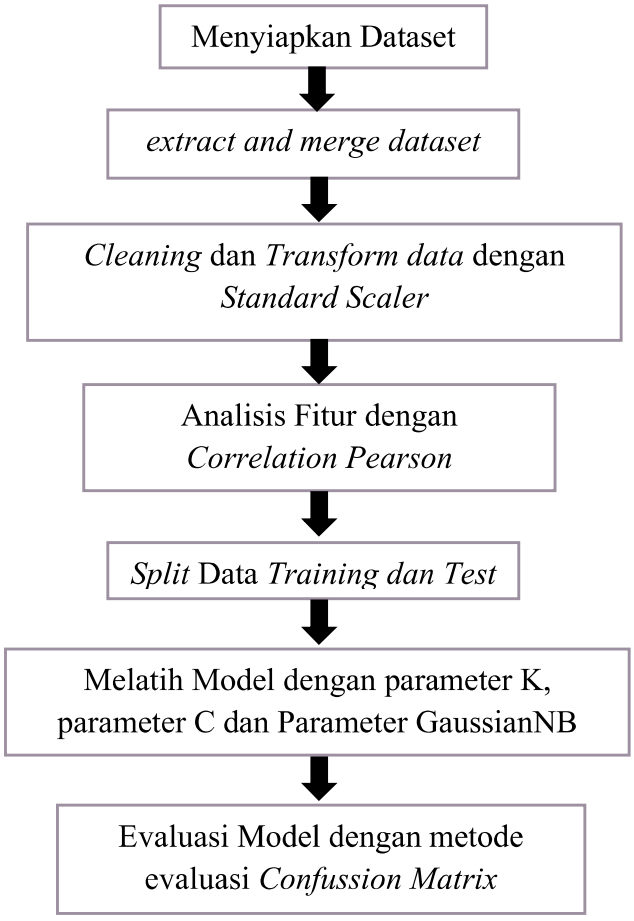
BAB III

METODE PENELITIAN

3.1 Desain Penelitian

Desain penelitian menyediakan kerangka dan alur kerja mencakup sepanjang proses penelitian. Dalam desain penelitian ini, penulis membagi penelitian menjadi beberapa tahap sebagai berikut :



Gambar 3. 1 Desain Penelitian

Dari gambar desain penelitian diatas, penjelasan setiap kotak adalah sebagai berikut :

1. Menyiapkan Dataset

Tahap pertama dalam penelitian ini adalah menyiapkan data, dalam rentang waktu pembuatan skripsi yaitu dari pertengahan September 2024 sampai Januari 2025, peneliti mampu mendapatkan informasi yang relevan dengan tujuan dari penelitian, dalam tahap ini peneliti menggunakan kumpulan data berupa *dataset* yang akan dipakai untuk melatih model *machine learning* dalam mendeteksi serangan.

2. *Extract and Merge dataset*

Tahap selanjutnya dalam penelitian ini mengekstrak *dataset* yang digunakan, data yang digunakan dalam penelitian ini terdiri dari 2 *dataset* berisi lalu lintas normal dan serangan, maka perlu untuk mengekstrak informasi penting dari *dataset* guna memahami isinya dan menggabungkan keduanya menjadi 1 dataset utuh.

3. *Cleaning* dan *Transform data* dengan *StandardScaler*

Tahap ketiga dalam penelitian adalah melakukan olah data terhadap *dataset*, dengan melakukan proses *cleaning* untuk melihat apakah terdapat nilai kosong atau duplikat pada data, *dataset* yang telah bersih kemudian akan di transformasi dengan metode *standardscaler.* Dikarenakan penggunaan algoritma yang sensitif terhadap jarak seperti KNN dan SVM, maka *dataset* perlu di normalisasikan ke rentang nilai yang tidak jauh atau besar untuk menghasilkan pelatihan model yang optimal.

4. Analisis Fitur dengan *Correlation Pearson*

Tahap keempat dari penelitian ini adalah menganalisis fitur, *dataset* yang telah dibersihkan akan dipilih fitur yang relevan dengan menggunakan teknik *Correlation*

Pearson dalam menentukan mana fitur yang berpengaruh terhadap penelitian dengan membandingkan fitur tersebut terhadap label, hasilnya kemudian disimpan kedalam *dataset* baru dengan fitur terbaik berdasarkan metode fitur seleksi yang dipilih.

5. *Split Data Training and Test*

Dataset ini kemudian dibagi menjadi data latih dan data uji dengan rasio 80:20, menggunakan *randomizer* agar data teracak dan tidak bias dalam mendeteksi 1 kelas data saja.

6. Melatih Model dengan parameter K, parameter C dan Parameter *GaussianNB*

Tahap Selanjutnya adalah melatih model, masing masing algoritma dilatih dengan parameter yang disesuaikan terlebih dahulu, Algoritma KNN menggunakan parameter K, algoritma SVM dengan parameter C dan kernel *linear*, algoritma NB dengan parameter *GaussianNB*, hasil dari ketiga algoritma akan ditentukan dengan menggunakan teknik *CrossValidation* dan *GridSearchCV* untuk menemukan nilai terbaik dari masing masing parameter berdasarkan metrik *Accuracy, Precision, Recall,* dan *F1-Score*.

7. Evaluasi Model dengan metode evaluasi *Confussion Matrix*

Setelah berhasil melatih algoritma, model algoritma akan digunakan untuk menguji data uji. Untuk mengevaluasi kinerja pengujian, hasil akan dievaluasi dengan metrik *Confussion Matrix* untuk melihat nilai TP,TN,FP dan FN. Hasil *Confussion matrix* kemudian digunakan untuk menghitung metrik lainnya yaitu *Accuracy, Precision, Recall,* dan *F1-Score*.

Untuk menghitung nilai *Accuracy :*

$$Accuracy = \frac{TN+TP}{TP+TN+FP+FN},$$

untuk menghitung nilai *Precision* :

$$Precision = \frac{TP}{TP+FP},$$

untuk menghitung nilai *Recall :*

$$Recall = \frac{TP}{TP+FN}$$

untuk menghitung nilai *F1* :

$$F1 - score = 2 \text{ x } ((Precision \text{ x } Recall) / (Precision + Recall)).$$

Hasil evaluasi kemudian akan dibandingkan untuk menganalisis performa model algoritma dalam mendeteksi serangan *botnet.*

3.2 Metode Pengumpulan Data

Pada penelitian ini peneliti menggunakan *dataset* berformat CSV yang diambil dari github, ini merupakan hasil olahan dari *CTU-13 Dataset,* sebuah *dataset* jaringan *botnet* yang ditangkap di CTU University, Czech Republic pada tahun 2011. *Dataset* ini dihasilkan dengan memproses file *Pcap* dari *Dataset CTU-13* menggunakan *CICFlowMeter Tool*.Untuk membuat kumpulan data CSV ini, semua file *Pcap* dari *Dataset CTU-13* diimpor menggunakan Alat *CICFlowMeter*. Alat ini mengubah File *Pcap* yang diberikan menjadi File CSV berdasarkan Aliran Paket, kemudian semua sampel lalu lintas serangan digabung dalam *Attack_Traffic.csv* berdasarkan ip

Serangan. Demikian pula, semua sampel lalu lintas normal digabung ke dalam *Normal_Traffic.csv* dengan mempertimbangkan semua ip selain ip Serangan.

3.3 Eksplorasi Data

Eksplorasi Data pada dataset adalah proses awal dalam analisis data yang bertujuan untuk memahami karakteristik, pola, dan struktur data sebelum dilakukan pemrosesan lebih lanjut. Langkah pertama sebelum menggabungkan kedua dataset adalah memastikan kedua file memiliki struktur kolom, jenis data,, korelasi antara file dan fitur yang sama. Pada penelitian ini peneliti menggunakan software *JupyterLab* untuk mengimport kedua file csv normal dan serangan ke dalam *cell,* peneliti menggunakan library *pandas, numpy, matplotlib.pyplot* dan *seaborn* untuk membantu menganalisis kedua file ini. Pertama peneliti mengecek distribusi data dengan mengelompokkan data sesuai labelnya hasilnya bisa dilihat pada tabel dibawah :

No.	Data	Jenis	Label
1	53.314	Normal	0
2	38898	Attack	1

Tabel 3. 1 Tabel distribusi data

Pada tabel 3.1 menunjukan jika distribusi label yang ada pada dataset yaitu 0 dan 1, dengan jenis data normal sebagai kelas mayoritas dan data *botnet* sebagai kelas minoritas.

Selanjutnya peneliti mengecek informasi dasar pada dataset, adapun detail setiap tipe data pada fitur bisa dilihat pada tabel dibawah ini.

No.	*Feature*	*Type*
1	Unnamed: 0	Int64
2	Flow Duration	Int64
3	Tot Fwd Pkts	int64
4	Tot Bwd Pkts	int64
5	TotLen Fwd Pkts	int64
6	TotLen Bwd Pkts	int64
7	Fwd Pkt Len Max	int64
8	Fwd Pkt Len Min	int64
9	Fwd Pkt Len Mean	float64
10	Fwd Pkt Len Std	float64
11	Bwd Pkt Len Max	int64
12	Bwd Pkt Len Min	int64
13	Bwd Pkt Len Mean	float64
14	Bwd Pkt Len Std	float64
15	Flow Byts/s	float64
16	Flow Pkts/s	float64
17	Flow IAT Mean	float64
18	Flow IAT Std	float64
19	Flow IAT Max	int64
20	Flow IAT Min	int64

No.	*Feature*	*Type*
21	Fwd IAT Tot	int64
22	Fwd IAT Mean	float64
23	Fwd IAT Std	float64
24	Fwd IAT Max	int64
25	Fwd IAT Min	int64
26	Bwd IAT Tot	int64
27	Bwd IAT Mean	float64
28	Bwd IAT Std	float64
29	Bwd IAT Max	int64
30	Bwd IAT Min	int64
31	Bwd PSH Flags	int64
32	Fwd Header Len	int64
33	Bwd Header Len	int64
34	Fwd Pkts/s	float64
35	Bwd Pkts/s	float64
36	Pkt Len Min	int64
37	Pkt Len Max	int64
38	Pkt Len Mean	float64
39	Pkt Len Std	float64
40	Pkt Len Var	float64

No.	*Feature*	*Type*
41	FIN Flag Cnt	int64
42	SYN Flag Cnt	int64
43	RST Flag Cnt	int64
44	ACK Flag Cnt	int64
45	Down/Up Ratio	int64
46	Pkt Size Avg	float64
47	Fwd Seg Size Avg	float64
48	Bwd Seg Size Avg	float64
49	Init Bwd Win Byts	int64
52	Fwd Act Data Pkts	int64
51	Active Mean	float64
52	Active Std	float64
53	Active Max	int64
54	Active Min	int64
55	Idle Mean	float64
56	Idle Std	float64
57	Idle Max	int64
58	Idle Min	int64
59	Label	int64

Tabel 3. 2 Tabel Tipe Data *Dataset*

Pada tabel diatas menunjukkan jika *dataset* ini memiliki 59 fitur yang terdiri dari 2 jenis tipe data, *float64* dengan 24 fitur dan *int64* dengan 35 fitur.

Selanjutnya untuk mengetahui karakteristik dari data, penggunaan statistik deskriptif akan sangat berguna dalam membantu memahami distribusi data dan rentang nilai, jika datanya mengandung *outlier* yang perlu ditangani maka teknik ini bisa menampilkannya. Adapun rumus statisktik deskriptif bisa dilihat dibawah ini :

1. Mean, untuk mencari nilai rata-rata dan melihat distribusi data normal dan serangan

$$\bar{X} = \frac{\sum X_i}{n}$$

Rumus 3. 1 Rumus Mean

Penjelasan :

X *i* = nilai data ke-i

n = jumlah data

cara menghitung mean dengan menjumlahkan semua nilai pada fitur dan bagi dengan jumlah data.

2. Standar Deviasi, untuk mengukur variasi data pada fitur.

$$s_j = \sqrt{\frac{\sum (X_{ij} - \bar{X}_j)^2}{n-1}}$$

Rumus 3. 2 Rumus Standar Deviasi

Keterangan :

s j = standar deviasi fitur ke-*j*

Xij = nilai fitur ke-j pada data ke-i

Xj = rata-rata fitur ke- j

n = jumlah sampel dalam dataset

Cara menghitung standar deviasi dengan pertama menghitung mean fitur, kemudian kurangkan setiap nilai pada fitur dengan mean lalu kuadratkan. Jumlahkan hasilnya dan bagi dengan n-1 dan terakhir kuadratkan hasilnya.

Contoh : data fitur 1 = 10,20,30,40 dengan Mean = 25

$$s_1 = \sqrt{\frac{(10-25)^2 + (20-25)^2 + (30-25)^2 + (40-25)^2}{4-1}}$$

$$= \sqrt{\frac{(-15)^2 + (-5)^2 + (5)^2 + (15)^2}{3}}$$

$$= \sqrt{\frac{225 + 25 + 25 + 225}{3}}$$

$$= \sqrt{\frac{500}{3}} = \sqrt{166.67} \approx 12.91$$

Gambar 3. 2 Hasil perhitungan Standar Deviasi

3. Min, nilai terkecil pada data.

$$X_{\text{min}} = \text{nilai terkecil dalam data}$$

Rumus 3. 3 Rumus Min

Nilai terkecil yang ada pada fitur, misal fitur 1 = 1,2,3,4,5 maka nilai Min = 1.

4. Max

$$X_{\text{max}} = \text{nilai terbesar dalam data}$$

Rumus 3. 4 Rumus Max

Nilai terbesar yang ada pada fitur, misal fitur 1 = 1,2,3,4,5 maka nilai Max = 5.

Hal terakhir yang bisa dilakukan adalah mengecek korelasi fitur *dataset*. Tujuan dari mengecek korelasi antara fitur *dataset* normal dan serangan untuk mencari fitur yang punya perbedaan signifikan antara serangan dan normal, semakin berbeda polanya maka semakin baik fitur untuk klasikasi. Untuk perhitungan korelasi pilih dua fitur yang akan dibandingkan, misalnya X dan Y. Langkah pertama adalah menghitung rata-rata dari masing-masing fitur. Setelah itu, hitung selisih setiap nilai dengan rata-ratanya. Selanjutnya kalikan selisih tersebut untuk setiap pasangan data dan jumlahkan hasilnya. Kemudian hitung standar deviasi masing masing fitur dan terakhir menerapkan rumus korelasi pearson untuk mendapatkan nilai korelasi antara kedua fitur. pertama-tama pilih dua fitur yang akan dibandingkan, misalnya Jika hasil korelasi mendekati 1, maka kedua fitur memiliki hubungan positif yang kuat, sedangkan jika mendekati -1, hubungan antar fitur bersifat negatif kuat. Jika nilai korelasi mendekati 0, berarti tidak ada hubungan yang signifikan antara kedua fitur tersebut.

3.4 Tahap Pre Processing

Pre processing yang akan peneliti lakukan terdiri dari *data cleaning* dan normalisasi data. Berikut tahap – tahap dalam pre processing yang peneliti lakukan pada penelitian ini.

3.4.1 Data Cleaning

Data cleaning adalah tahap pertama yang peneliti lakukan, proses ini mencari adanya *missing value* dan *data duplicate*, kedua hal ini akan mengganggu proses pelatihan model *machine learning* jika tidak ditanggulangi. Untuk mengecek apakah dataset memiliki data duplikat atau memiliki data yang kosong, peneliti menggunakan metode *data.duplicate* dan *data.isnull* dari *library pandas* di *python*, *data.duplicate* digunakan untuk mengecek data duplikat pada dataset sedangkan *data.isnull* digunakan untuk mengecek *missing values* pada dataset.

3.4.2 Normalisasi Data

Dataset yang telah bersih akan dipisah fitur dan labelnya untuk melalui proses normalisasi, pada penelitian ini metode normalisasi yang peneliti gunakan adalah *standardscaler* pada *library scikit-learn* di *python* pada data yang nantinya akan menjadi data latih. Adapun rumus dari *standardcaler* bisa dilihat dibawah ini :

$$X' = \frac{X - \mu}{\sigma}$$

Rumus 3. 5 Rumus *StandardScaler*

Keterangan :

X’ = nilai fitur setelah di-standardisasi

X = nilai asli dari fitur

μ = rata-rata dari fitur

σ = standar deviasi dari fitur

Perhitungan *standardscaler* digunakan untuk menstandarkan data sehingga memiliki rata-rata 0 dan standar deviasi 1. Proses perhitungannya dimulai dengan menghitung rata-rata (μ), yang diperoleh dengan menjumlahkan semua nilai dalam dataset kemudian dibagi dengan jumlah data, Setelah itu, standar deviasi (σ) dihitung dengan mengurangkan setiap nilai data dengan rata-rata, menguadratkan hasilnya, menjumlahkan seluruhnya, lalu membaginya dengan n−1 dan diakhiri dengan operasi akar kuadrat. Setelah kedua parameter didapatkan, maka data diubah menjadi rumus diatas di mana setiap nilai data dikurangi dengan rata-rata lalu dibagi dengan standar deviasi. Hasil akhirnya adalah data yang terdistribusi dengan rata-rata 0 dan standar deviasi 1, meskipun nilainya masih bisa lebih dari 1 atau kurang dari -1 tergantung pada penyebaran data aslinya.

3.5 Split Data

Pada penelitian ini, peneliti membagi *dataset* yang telah dinormalisasikan ke dalam rasio 80:20, seperti yang dijelaskan di tahap eksplorasi data jika dataset ini terdiri dari 53.314 data normal dan 38.899 data *botnet*, walaupun ketidak seimbangan antara data kelas normal dan kelas serangan tidak besar, namun hal ini mungkin masih bisa mengakibatkan bias pada pelatihan model dan pendeteksian data lebih condong mendeteksi data normal. untuk mengatasinya peneliti akan menggunakan *randomize* dalam membagi dataset kedalam data latih dan data uji, data yang masuk kedalam data

latih dan uji akan diacak untuk menjaga keberagaman isi data kemudian disimpan menjadi 2 file baru file latih dan file uji.

3.6 Analisis Fitur

Untuk memilih fitur yang berpengaruh terhadap deteksi serangan *botnet* dan membuang fitur yang menjadi redundansi terhadap proses deteksi pada penelitian ini peneliti menggunakan metode Korelasi Pearson untuk membandingkan fitur X terhadap Label (Y). Adapun rumus korelasi pearson bisa dilihat pada gambar dibawah :

$$r = \frac{\sum(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum(X_i - \bar{X})^2} \times \sqrt{\sum(Y_i - \bar{Y})^2}}$$

Rumus 3. 6 Rumus Korelasi Pearson

Keterangan:

Xi = Nilai dari fitur X

Yi = Nilai dari fitur Label

X = Rata-rata dari fitur X

Y = Rata-rata dari fitur Label

Untuk langkah kerja dari korelasi pearson ini pertama-tama hitung rata-rata dari fitur X dan Label, selanjutnya hitung deviasi dari rata-rata untuk setiap data, kemudian kalikan hasil deviasi untuk setiap pasang data, selanjutnya hitung penyebut dengan menjumlahkan kuadrat deviasi dari masing-masing variabel, masukkan semua nilai ke

dalam rumus untuk mendapatkan r atau hasil korelasi. Sebagai contoh, perhitungan korelasi pearson untuk sampel dari fitur *Flow Pkts/s* dengan 10 sampel acak adalah sebagai berikut :

Mean *Flow Pkts/s* (X) = 16678.8074

Mean Label (Y) = 0.7

Contoh untuk salah satu data pertama, deviasi dari mean :

X1 - X = 31.916184 – 16678.8074 = -16646.8912

Y1 – Y = 1 – 0.7 = 0.3

Perkalian deviasi = (Xi – X) x (Yi – Y) = (-16646.8912) x (0.3) = −4994.067373

Kemudian menghitung Kuadrat deviasi kedua fitur =

$(Xi – X)^2$, $(Yi – Y)^2$ = 277,119,000.73 dan 0.09

Setelah mendapatkan semua nilai maka dimasukkan ke rumus, hasil yang didapatkan adalah :

$$r = \frac{\sum(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum(X_i - \bar{X})^2} \times \sqrt{\sum(Y_i - \bar{Y})^2}}$$

$$r = \frac{6,742.3676}{\sqrt{698,882,435.6} \times \sqrt{2.1}}$$

$$r = \frac{6,742.3676}{26,438.216 \times 1.4491}$$

$$r = \frac{6,742.3676}{38,304.13}$$

$$r \approx 0.377$$

Gambar 3. 3 Hasil Perhitungan Korelasi Pearson

Pada gambar diatas didapatkan nilai korelasi dari fitur *flow pkts/s* dengan 10 sampel adalah 0.377. Sedangkan Untuk mendapatkan nilai keseluruhan dari suatu fitur memerlukan kekuatan komputasi, ini dikarenakan banyaknya data pada tiap fitur sehingga mustahil untuk dihitung secara manual.

Metode menyeleksi atau memilih fitur yang terbaik atau relevan dalam mendeteksi serangan adalah dengan menghitung korelasi setiap fitur X yang ada pada *dataset* terhadap label, hasilnya fitur - fitur yang paling berkorelasi dengan label akan digunakan untuk melatih model algoritma. Berdasarkan ini dapatlah 15 fitur terbaik dari hasil output korelasi pearson, Ke-15 fitur yang dipilih bisa dilihat pada tabel dibawah ini :

Korelasi Pearson			
No.	Fitur	Korelasi	Penjelasan fitur
1	*SYN Flag Cnt*	0.399891	Menghitung jumlah paket dalam aliran yang memiliki bendera *SYN* di *header* TCP diatur ke 1. Menunjukkan upaya memulai koneksi TCP, biasanya tingginya nilai ini menunjukan serangan *SYN flood*.
2	*Fwd Pkts/s*	0.322526	*Forward packet/second*. Jumlah rata-rata paket yang dikirim dari sumber ke tujuan per detik

3	*Flow Pkts/s*	0.322064	*Flow packet/second*. Jumlah total paket dalam satu aliran per detik (baik dari sumber ke tujuan maupun sebaliknya).
4	*Bwd Pkts/s*	0.315983	*Backward packet/second*. Jumlah rata-rata paket yang dikirim dari tujuan ke sumber per detik.
5	*Init Bwd Win Byts*	0.270555	*Initial backward window bytes*. Jumlah *byte* dalam jendela TCP awal yang dikirim dari tujuan ke sumber.
6	*Active Min*	0.232310	*Active minimum.* Waktu minimum koneksi tetap aktif antara pengiriman paket berurutan selama sesi.
7	*Active Mean*	0.212623	*Active mean*. Rata-rata waktu koneksi tetap aktif antara pengiriman paket berurutan.
8	*Fwd Pkt Len Min*	0.178448	*Forward packet length minimum*. Panjang paket terkecil yang dikirim dari sumber ke tujuan.
9	*FIN Flag Cnt*	0.161627	Menghitung jumlah paket dengan bendera *FIN* diatur ke 1. Untuk menunjukkan upaya untuk mengakhiri koneksi. Nilai tinggi bisa mengindikasikan lalu lintas mencurigakan.

10	*Active Max*	0.134268	*Active maximum*. Waktu maksimum koneksi tetap aktif antara pengiriman paket berurutan.
11	*Flow IAT Mean*	0.099705	*Flow IAT (Inter - arrival Time) mean.* Menunjukkan rata-rata waktu antara kedatangan dua paket berturut-turut dalam aliran yang sama. Nilai tinggi atau rendah bisa memberikan petunjuk tentang pola lalu lintas yang tidak biasa.
12	Down/Up Ratio	0.072822	Rasio ini memberi gambaran tentang perbandingan antara trafik yang diterima oleh pengguna dan yang dikirim, yang sering kali digunakan untuk mendeteksi pola komunikasi yang tidak wajar atau anomali.
13	*Fwd Seg Size Avg*	0.072722	*Forward Segment Size Average*. menunjukkan seberapa besar ukuran data yang ditransmisikan dalam setiap segmen dari pengirim. Serangan atau anomali dapat tercermin pada ukuran segmen yang lebih besar atau lebih kecil dari yang diharapkan.

14	*Fwd Pkt Len Mean*	0.072722	*Forward Packet Length Mean*. memberikan informasi tentang ukuran paket rata-rata yang dikirim dalam aliran tersebut. Dalam serangan, seperti DDoS, panjang paket dapat berfluktuasi secara tidak wajar dibandingkan dengan lalu lintas normal.
15	*Flow Byts/s*	0.072661	*Flow Byte per second*. mengukur kecepatan transfer data dalam satu aliran, yang dapat mencerminkan volume lalu lintas.

Tabel 3. 3 Fitur terbaik *Korelasi Pearson*

Setelah mendapatkan 15 fitur terbaik, dataset baru dibuat untuk menyimpan 15 fitur terbaik yang akan digunakan untuk melatih dan menguji model algoritma *machine learning*. Data uji juga akan disimpan dengan format fitur yang sama dengan data latih.

3.7 Model Machine Learning

Pada penelitian ini, peneliti menggunakan 3 algoritma machine learning yaitu *K-Nearest Neighbours, Support Vector Machine,* dan *Naïve Bayes*, ketiga algoritma ini adalah algoritma yang cukup umum digunakan dalam *machine learning* untuk masalah klasifikasi dan pendeteksian.

3.7.1 KNN

Algoritma KNN pada umumnya menggunakan rumus *Ecludien Distance,* adapun rumusnya bisa dilihat pada gambar dibawah :

$$D = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Rumus 3. 7 Rumus KNN

D: Jarak antara titik uji (x) dengan titik pelatihan (y).

xi,yi: nilai fitur ke-*i* dari masing-masing titik

n: Jumlah fitur pada *dataset*

untuk cara kerjanya pertama – tama tentukan jumlah K, kemudian gunakan rumus *Euclidean Distance* untuk menghitung jarak antara K tetangga, ambil K tetangga yang paling dekat per perhitungan ini untuk dihitung angkanya dari data point pada setiap kategori antara K tetangga ini, data point baru akan ditetapkan pada kategori yang mana nilai K tetangga tertinggi.

3.7.2 SVM

Algoritma SVM bekerja dengan mencari *hyperplane* terbaik yang memisahkan data dalam ruang fitur dengan margin maksimum. Adapun rumus dasar SVM bisa dilihat pada gambar dibawah :

$$w \cdot x + b = 0$$

Rumus 3. 8 Rumus SVM

Keterangan :

w = vektor bobot

x = vektor fitur data

b = bias

· = operasi dot product

Persamaan *hyperplane* dinyatakan dalam bentuk dot product antara vektor bobot w dan vektor fitur x, ditambah bias. w adalah vektor bobot yang menentukan arah dan kemiringan *hyperplane*, x adalah vektor fitur dari suatu data, dan b adalah bias yang menggeser *hyperplane* agar tidak selalu melalui titik asal. Algoritma SVM berusaha mencari hyperplane terbaik yang memaksimalkan margin, yaitu jarak antara hyperplane dengan data terdekat dari kedua kelas.

3.7.2 Naïve Bayes

Algoritma *Naïve Bayes* umumnya digunakan ketika fitur bersifat numerik dan diasumsikan mengikuti distribusi normal atau *gaussian, GaussianNB* pada *Naive Bayes* adalah rumus yang umum digunakan, adapun rumusnya bisa dilihat pada gambar dibawah :

$$P(X_i|Y) = \frac{1}{\sqrt{2\pi\sigma_Y^2}} \exp\left(-\frac{(X_i - \mu_Y)^2}{2\sigma_Y^2}\right)$$

Rumus 3. 9 Rumus NB

Keterangan :

μY = rata-rata (mean) dari fitur Xi untuk kelas

$\sigma Y2$ = varians dari fitur Xi untuk kelas

π = konstanta 3.14159...

exp = fungsi eksponensial

Cara kerja *GaussianNB* dengan mengasumsikan bahwa setiap fitur dalam dataset mengikuti distribusi normal (*Gaussian*) dalam setiap kelas. Pertama, model menghitung mean (μ) dan varians ($\sigma 2$) untuk setiap fitur berdasarkan kelasnya. Saat menerima data baru, model menghitung *probabilitas likelihood* setiap fitur menggunakan rumus distribusi *Gaussian*, yaitu dengan memasukkan nilai fitur ke dalam fungsi eksponensial berdasarkan mean dan varians yang telah dihitung. Selanjutnya, probabilitas posterior dihitung menggunakan *Teorema Bayes*, yang mengalikan *prior probability* kelas dengan *likelihood* dari semua fitur. Akhirnya, model memilih kelas dengan probabilitas posterior terbesar sebagai hasil prediksi. Karena setiap fitur dianggap independen satu sama lain, perhitungan dilakukan secara terpisah untuk masing-masing fitur, lalu dikalikan untuk mendapatkan probabilitas akhir.

3.8 Jadwal Penelitian

Jadwal penelitian adalah rangkaian kegiatan yang peneliti lakukan dalam jangka waktu pembuatan skripsi.

No	Kegiatan	Jadwal Penelitian					
		2024					2025
		Agus	Sept	Okt	Nov	Des	Jan
1.	Bab 1						
2.	Bab 2						
3	Bab 3						
4.	Bab 4						
5.	Bab 5						

Tabel 3. 4 Jadwal Penelitian