BAB III

METODOLOGI PENELITIAN

3.1 Subyek dan Obyek Penelitian

Subyek penelitian merupakan data csv yang akan diamati. Subyek penelitian ini adalah dataset penyakit diabetes. Obyek penelitian ini merupakan permasalahan yang akan diamati. Obyek penelitian ini merupakan penerapan algoritma *K-Nearest Neighbors* dengan Teknik *cross validation.*

3.2 Alat dan Bahan Penelitian

Dalam melakukan penelitian algoritma *K-Nearest Neighbors* dengan Teknik *cross validation* pada prediksi penyakit diabetes, memerlukan alat dan bahan untuk kebutuhan penelitian.

3.2.1 Alat Penelitian

Dalam penelitian ini menggunakan rerangkat keras sebuah laptop dengan spesifikasi sebagai berikut:

a. Laptop ASUS
b. Processor: IntelI CoreI i3-6006U CPU @ 2.00GHz (4 CPUs), ~2.0GHz
c. RAM 4GB

Sedangkan perangkat lunak yang digunakan dalam pengembangan penelitian ini sebagai berikut:

a. Sistem Operasi Windows 10 Pro
b. Jupyter Notebook
c. Python
d. Google Chrome

3.2.2 Bahan Penelitian

Bahan penelitian ini adalah dataset penyakit diabetes dari berbagai usia tak terkecuali laki-laki maupun perempuan yang berasal dari *National Institute of Diabetes and Digestive and Kidney Diseases* pada tahun 2021.

3.3 Diagram Alir Penelitian

Gambar 3. 1 Tahapan-tahapan penelitian

3.3.1 Studi Pustaka

Penulis melakukan studi Pustaka. Pada tahap ini, penulis membaca dan memahami konsep dan permasalahan *machine learning* yang ada pada jurnal, buku maupun penelitian sebelumnya. Kemudian, hasil yang didapatkan menjadi landasan penulisan dan penelitian yang akan dilakukan.

Pada penelitian ini berfokus pada data yang berkaitan dengan penyakit diabetes dalam berbagai sumber penelitian terkait. Dengan melakukan studi pustaka ini berharap penulis akan lebih menguasai topik-topik yang berada didalamnya.

3.3.2 Perumusan Masalah dan Tujuan

Perumusan masalah dilakukan untuk mengetahui permasalahan yang ada sehingga diperlukan penelitian ini dan penyusunan tujuan penelitian dilakukan untuk mengetahui tujuan dari penelitian ini. Dalam hal ini, penulis berfokus kepada permasalahan di bidang kesehatan. Salah satunya penyakit diabetes dan jenis model algoritma machine learning yang digunakan untuk tujuan penelitian ini terhadap data penyakit tersebut.

3.3.3 Data Collection dan Preprocessing

Pada tahap ini, penulis melakukan *data collection dan preprocessing.* Langkah pertama dari tugas ilmu data adalah untuk mendapatkan, mengumpulkan, dan mengukur data yang diperlukan dan ditargetkan dari sumber data internal atau eksternal yang tersedia, dan kemudian dikompilasi ke dalam sistem yang mapan. Dalam hal ini, penulis mendapatkan dataset dari *National Institute of Diabetes and Digestive and Kidney Diseases* pada tahun 2021.

Kemudian melakukan *prepocessing. Preprocessing* adalah teknik penambangan data yang mengubah data mentah menjadi format yang dapat dipahami. Proses ini memiliki empat tahap utama yaitu *data cleaning, data integration, data transformation, and data reduction.* Dataset dan pengolahan datanya akan dijelaskan pada bagian perhitungan KNN.

3.3.3.1 Data Preprocessing

Pada tahap ini, akan menyaring, mendeteksi, dan menangani data kotor untuk memastikan kualitas data dan hasil analisis yang berkualitas. Dalam hal ini, mungkin ada *noise* dari nilai dan *outlier* yang tidak mungkin

dan ekstrim, dan nilai yang hilang. Kesalahan mungkin termasuk data yang tidak konsisten dan atribut dan data yang berlebihan. Dan empat tahap utama dalam *preprocessing* ikut terlibat. Contoh pengolahan data pada tahap *preprocessing* adalah filter attribute data yang berlebihan. Hal tersebut dapat mempengaruhi proses perhitungan akurasi dalam data.

3.3.4 Training dan Klasifikasi

Pada tahap ini, dimulai dari eksplorasi analisis data, *data modelling*, dan evaluasi model yang diuji. Kemudian akan tampil hasil akurasi dari model yang digunakan untuk proses implementasi akurasi model ke input manual dari pengguna berbasis streamlit.

3.3.4.1 Eksplorasi analisis data (EDA)

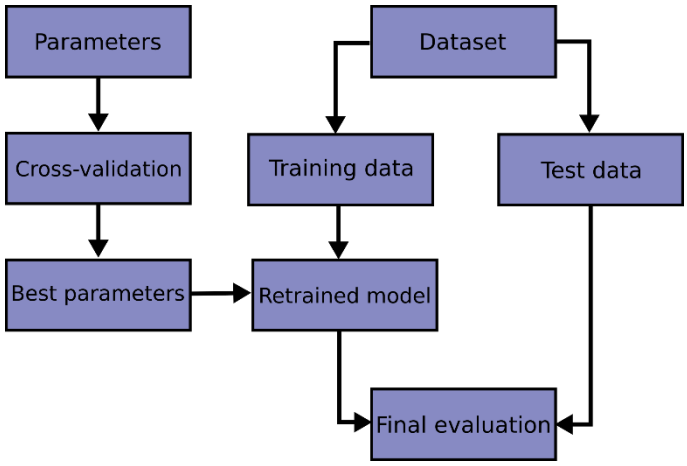
EDA bertujuan untuk melakukan penyelidikan awal pada data sebelum pemodelan formal dan representasi grafis dan visualisasi, untuk menemukan pola, melihat asumsi, dan menguji hipotesis. Ringkasan informasi tentang karakteristik utama dan tren tersembunyi dalam data dapat membantu dokter mengidentifikasi area dan masalah yang menjadi perhatian, dan penyelesaiannya dapat meningkatkan akurasi dalam mendiagnosis diabetes.

3.3.4.2 Data Modelling

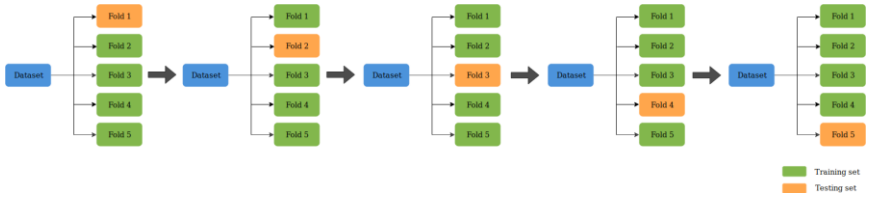
Dataset dibagi menjadi dua set terpisah yaitu set pelatihan dan set tes. Keduanya terdiri dari atribut yang sama, tetapi nilai atributnya tidak sama. *Training set* digunakan untuk melatih dan membangun model klasifikasi. *Test set* digunakan untuk memprediksi klasifikasi data baru yang tidak bias yang tidak digunakan untuk melatih model, sebelum mengevaluasi kinerja model berdasarkan metrik kinerja akurasi, *presisi*, *recall*, dan skor *F1* dari klasifikasi tersebut.

3.3.4.3 Cross Validation (CV)

Cross Validation adalah metode yang digunakan untuk mengevaluasi dan memvalidasi kinerja sistem kami dalam konteks klasifikasi. Ide utama di balik validasi silang adalah bahwa setiap sampel dalam kumpulan data kami memiliki peluang untuk diuji. Kemudian pemilihan CV dapat didasarkan pada ukuran dataset yang digunakan. Biasanya *CV K-fold* digunakan karena dapat mengurangi waktu komputasi dengan tetap menjaga keakuratan estimasi. Berikut alur *cross validation* yang ditunjukkan pada gambar 3.2 dan 3.3 ini:



Gambar 3. 2 Alur *cross validation*



Gambar 3. 3 Alur *cross validation*

Merujuk pada Gambar 3.2 dan 3.3, terlihat alur *cross validation* yang bermula dataset awal yang menggunakan *train-test split* untuk membagi data menjadi dua yaitu data pelatihan dan data uji. *K-Fold Cross Validation* ini digunakan untuk menyelesaikan permasalahan *train-test split*. Karena *K-Fold Cross Validation* akan memilih nilai k sebagai pembagian data. Misalnya nilai k=5 maka jumlah data dibagi 5 dan

melakukan putaran *training* dan *testing* sebanyak 5 kali. Kemudian hasilnya pembagian sebagai data uji dan sisanya sebagai data pelatihan serta dilakukan sebanyak 5 kali. Tiap putaran akan menghasilkan akurasi dan banyaknya akurasi tersebut dilakukan mean atau rata-rata. Hasil *mean* tersebut adalah akurasi sebenarnya dari model algoritma yang digunakan menggunakan teknik cross validation.

3.3.4.4 Model Evaluation

Pada tahap ini, akan dilakukannya evaluasi model yang akan uji hasil akurasi, *presisi, recall,* dan skor *F1*. Kemudian akan digunakan untuk implementasi ke tahap selanjutnya. Sehingga hasil tiap model bisa dilakukan perbandingan model yang lebih baik dari model lainnya. *Confusion matrix* adalah suatu metode yang digunakan sebagai pengevalusi kinerja metode klasifikasi. *Confusion matrix* menghasilkan sebuah perbandingan antara hasil klasifikasi yang dibuat oleh sistem (Prediksi) dan hasil klasifikasi yang sebenarnya (Actual). Secara teknis, ada empat istilah yang digunakan di *confusion matrix* sebagai representasi hasil akhir. Yaitu *True Positive* (TP) yang menunjukkan hasil positif yang dideteksi dengan benar. *True Negative* (TN) menunjukkan hasil negatif yang dideteksi dengan benar. *False Positive* (FP) adalah data negatif, namun terdeteksi positif dari sistem. *False Negative* (FN) adalah data positif yang terdeteksi sebagai data bernilai *negative*.[25] *Confusion matrix* juga bisa menghasilkan sebuah perhitungan nilai untuk hasil *accuracy, precision*, dan *recall.*

a. Akurasi

Dari model arsitektur dan *confusion matrix* maka dalam mendapatkan hasil akurasi dapat menggunakan persamaan berikut ini:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{3.1}$$

b. *Presisi*

Selanjutnya dalam mendapatkan hasil precition dapat menggunakan persamaan berikut ini:

$$Precision = \frac{TP}{TP + FP} \tag{3.2}$$

c. *Recall*

Terakhir dalam mendapatkan hasil recall dapat menggunakan persamaan yang ada di bawah ini:

$$Recall = \frac{TP}{TP + FN} \tag{3.3}$$

d. *F1 Score*

Nilai presisi dan recall berguna untuk memahami performa algoritme tertentu dan membantu dalam menghasilkan hasil berdasarkan persyaratan. Tetapi saat harus membandingkan beberapa algoritme yang di latih pada data yang sama, menjadi sulit untuk memahami algoritme mana yang lebih cocok dengan data hanya berdasarkan kedua nilai tersebut. Sehingga dibutuhkannya *F1 Score*. *F1 Score* dapat digambarkan sebagai rata-rata harmonis atau tertimbang dari presisi dan perolehan. Berikut persamaanya.

$$\text{F1 Score} = 2 \text{ x } \frac{\text{Precision x Recall}}{\text{Precision + Recall}} \tag{3.4}$$

Tabel 3. 1 Contoh Confusion Matrix

		Nilai Sebenarnya	
		TRUE	FALSE
Nilai Prediksi	TRUE	90	20
	FALSE	10	880

Presisi = $\frac{90}{90+20} = \frac{90}{110} = 0.82 = 82\%$

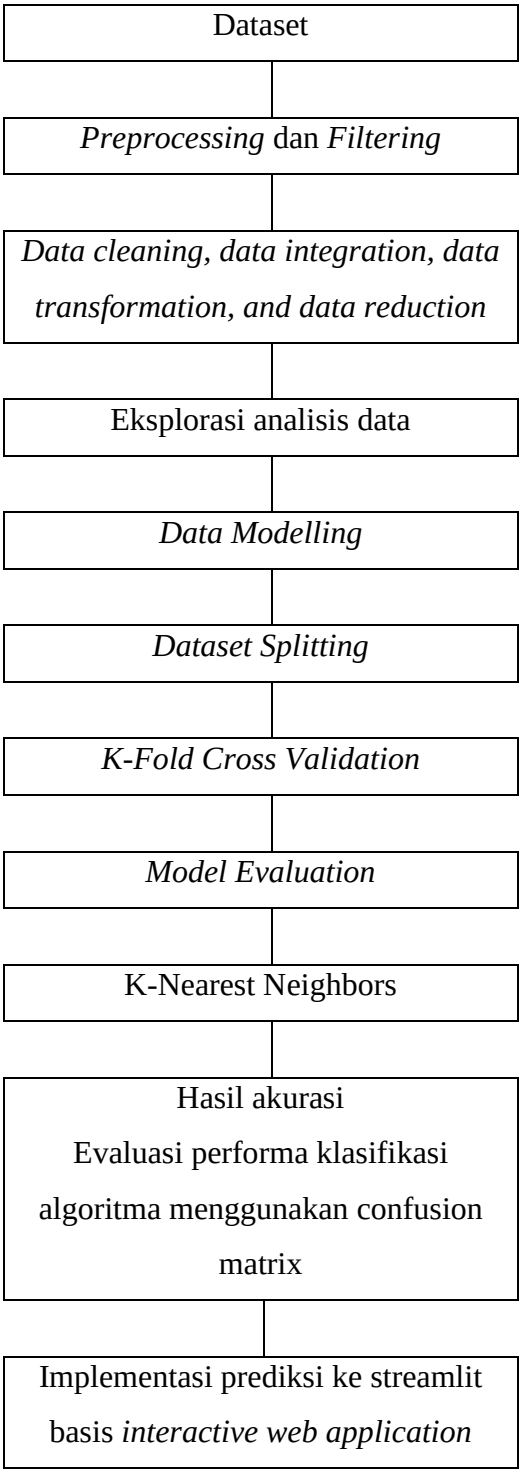
Recall = $\frac{90}{90+10} = \frac{90}{100} = 0.9 = 90\%$

Akurasi = $\frac{90+880}{90+880+20+10} = \frac{970}{1000} = 0.97 = 97\%$

F1 Score = 2 x $\frac{0.82\ X\ 0.9}{0.82 + 0.9} = 2\ x\ \frac{0.738}{1.72} = 2\ X\ 0.43 = 0.86 =\ 86\%$

Merujuk pada table 3.1 terlihat contoh confusion matrix dan dilanjut contoh perhitungan akurasi, presisi, akurasi dan f1 score dari data diatas.

3.3.5 Proses Pengujian



Gambar 3. 4 Alir Pengujian KNN

Pada gambar 3.4 menjelaskan mengenai proses penelitian menggunakan algoritma *K-Nearest Neighbors.* Dataset dilakukan

preprocessing dan *filtering* seperti *data cleaning, data integration, data transformation, and data reduction.* Kemudian eksplorasi analisis data dan membuat model data. Setelah itu, pembagian data untuk dilakukan *k-fold cross validation* dan evaluasi algoritma *k-nearest neighbors.* Sehingga hasil akurasi terlihat dan bisa di implementasikan kedalam streamlit basis *interactive web application* untuk prediksi penyakit diabetes dengan attributes dataset yang telah difilter sebelumnya.

3.3.6 Proses Perhitungan KNN dengan K-Fold Cross Validation

Dalam penelitian ini, dataset yang digunakan akan dibagi menggunakan metode validasi silang. Data dibagi menjadi 2 jenis data yaitu data pelatihan dan data uji yang digunakan untuk membandingkan dengan jarak data uji dan proses klasifikasi KNN dengan metode validasi silang. Tujuannya untuk memvalidasi algoritma KNN menjadi lebih teruji dan kinerja yang dihasilkan valid. Dataset yang digunakan memiliki 390 *rows* dan 14 *attributes.*

Tabel 3. 2 Dataset yang digunakan

Attributes	cholesterol	glucose	hdl_chol	chol_hdl_ratio	age	weight	height
1	193	77	49	3.9	19	61	119
2	146	79	41	3.6	19	60	135
…	…	…	…	…	…	…	…
390	165	94	69	2.4	92	62	217

Atrributes	Bmi	systolic_bp	diastolic_bp	waist	hip	waist_hip_ratio	diabetes
1	22.5	118	70	32	38	0.84	0
2	26.4	108	58	33	40	0.83	0
…	…	…	…	…	…	…	…
390	39.7	160	82	51	51	1	0

Dalam dataset yang digunakan, misalnya kita akan bagi menjadi data latih dan uji dengan perbandingan 80:20. Artinya, penulis menyimpan 80% dari total data untuk melatih model dan sisanya 20% untuk mengujinya.

Selanjutnya, penulis melatih model dengan nilai “K” yang berbeda dan menangkap akurasinya pada data pengujian penulis. Asumsikan penulis mendapat table seperti dibawah ini:

Tabel 3. 3 Sample akurasi dari berbagai nilai K

Value dari “K”	Akurasi di data uji
K=1	0.83
K=2	0.81
K=3	0.78
K=4	0.90
K=5	0.88

Terlihat pada table diatas, jika penulis amati maka pada K=4 akan mendapatkan akurasi tertinggi 90% dan setelah itu, penulis melihat tren penurunan akurasi. Jadi, dasar itu penulis sampai pada kesimpulan bahwa nilai yang sesuai untuk K=4. Jika penulis mempertimbangkan contoh di atas dan berdasarkan pemahaman penulis, untuk titik data yang tidak terlihat di masa depan seberapa akurat model akan memprediksi label kelas. Ketika suatu algoritma berkinerja baik pada titik data yang tidak terlihat, itu disebut generalisasi. Seluruh tujuan pembelajaran mesin adalah generalisasi.

Jika penulis memikirkan KNN, penulis menggunakan data uji untuk pada dasarnya menentukan nilai K yang tepat dan data latih untuk menemukan nearest neighbors. Penulis mendapatkan akurasi 90% pada data uji yang juga penulis gunakan untuk menentukan nilai “K” yang tepat. Tapi penulis tidak dapat mempertahankan tingkat akurasi perkiraan ini pada data yang tidak terlihat di masa depan. Untuk mengatakan dengan yakin bahwa penulis dapat mencapai akurasi sekitar 90% pada data yang tidak terlihat di masa mendatang, penulis harus terlebih dahulu menguji model ini pada data yang tidak terlihat. Untuk itu digunakan metode validasi silang untuk mengatasi hal diatas atau meminimalkan *overfitting.*

Langkah-langkah proses perhitungan KNN dengan metode validasi silang dalam kinerja dari *K-fold cross validation* yaitu:

1. Total *Instance* dibagi menjadi N bagian.
2. Fold ke-1 adalah saat bagian ke-1 menjadi data uji dan sisanya menjadi data latih. Kemudian, hitung akurasi atau kedekatan suatu hasil pengukuran dengan angka atau data yang sebenarnya berdasarkan porsi data tersebut. Perhitungan akurasi tersebut menggunakan persamaan sebagai berikut.

$$\text{Akurasi} = \frac{\sum data\ uji\ benar\ klasifikasi}{\sum total\ data\ uji} 100X \qquad (3.5)$$

3. Demikian seterusnya dan sesuaikan nilai fold yang akan digunakan untuk proses perhitungan akurasi. Hitung rata-rata akurasi dari k-buah akurasi yang ada. Rata-rata akurasi menjadi final
4. Kemudian diimplementasikan kedalam algoritma KNN yang memiliki berbagai tahapan diantaranya seperti menentukan nilai k, hitung jarak Euclidean dari data uji dan data latih dari dataset, menampilkan jarak Euclidean secara *ascending*, ambil jarak terkecil sesuai jumlah k, dan mendapatkan hasil klasifikasi data menggunakan KNN.

Misalnya bahwa untuk setiap nilai K penulis harus menghitung akurasi sebanyak 4 kali. Ini karena penulis secara acak membagi kumpulan data pelatihan menjadi 4 bagian yang sama. Kemudian penulis secara acak membagi kumpulan data menjadi 5 bagian yang sama maka penulis harus menghitung 5 akurasi berbeda untuk setiap nilai K dan mengambil rata-ratanya. Walaupun memiliki proses yang tidak sedikit tapi prosesnya sangat berharga karena meningkatkan generalisasi model algoritma.

Contoh Pehitungan KNN dan Jarak menggunakan Euclidean

Tabel 3.4 Contoh Data Training

Cholesterol	Glucose	Diabetes
193	77	0
220	60	1
165	94	0
123	99	1
99	101	0
65	194	?

Dari tabel diatas, ada 3 data yang sudah berlabel dan 1 satu data yang harus penulis tentukan kelasnya dengan detail informasi sebagai berikut:

- Ada 2 kelas yatu 0 dan 1
- Cholesterol dan Glucose adalah independent variables atau variable yang nilainya tidak dipengaruhi oleh variable lain dan akan digunakan untuk menghitung jarak.
- Diabetes merupakan dependent variable, variable yang nilainya dipengaruhi oleh variable lain (Cholesterol dan Glucose)

Misalnya penulis ambil nilai k=3 dan hitung jarak antara data baru dan masing-masing data lainnya.

Data 1 : dis = $\sqrt{(65-193)^2+(194-77)^2}$ = 173,4157

Data 2 : dis = $\sqrt{(65-220)^2+(194-60)^2}$ = 204,8927

Data 3 : dis = $\sqrt{(65-165)^2+(194-94)^2}$ = 141,4214

Data 4 : dis = $\sqrt{(65-123)^2+(194-99)^2}$ = 111,3059

Data 5 : dis = $\sqrt{(65-99)^2+(194-101)^2}$ = 99,9599

Setelah memperoleh hasil jarak dari data baru dengan data training, penulis urutkan dari jarak terdekat secara ascending dan menentukan tetangga terdekat berdasarkan jarak minimum K. Berikut urutan hasil jarak terdekat data baru dengan data training secara ascending.

Tabel 3.5 Urutan Jarak Terdekat data baru dengan data training

Cholesterol	Glucose	Hasil Jarak	Urutan	Apakah termasuk 3-NN
99	101	99,9599	1	Ya (K<3)
123	99	111,3059	2	Ya (K<3)
165	94	141,4214	3	Ya (K=3)
193	77	173,4157	4	Tidak (K>3)
220	60	204,8927	5	Tidak (K>3)

Pada kolom kelima berisi keterangan apakah termasuk 3-NN, itu pun karena nilai K sudah ditentukan sama dengan 3. Kemudian penulis tentukan kategori dari tetangga terdekat. Penulis perhatikan baris 1,2, dan 3 termasuk kategori Ya dan sisanya Tidak. Penulis memberikan kategori berdasarkan table awal. Berikut datanya.

Tabel 3.6 Penentuan Kategori Data

Cholesterol	Glucose	Apakah termasuk 3-NN	Kategori 0 atau 1
99	101	Ya (K<3)	0
123	99	Ya (K<3)	1
165	94	Ya (K=3)	0

Selanjutnya penulis gunakan kategori mayoritas yang sederhana dari tetangga yang terdekat sebagai nilai prediksi data yang baru. Merujuk pada tabel 3.6, terlihat penulis memiliki 2 kategori 0 dan 1 kategori 1. Sehingga penulis simpulkan bahwa data baru termasuk dalam kategori 0.

3.3.7 Analisis Hasil

Pada tahap analisis hasil dilihat dari optimasi model k-nearest neighbors dengan Teknik *cross validation*. Percobaan menguji model k-nearest neighbors tanpa Teknik *cross validation* memiliki perbedaan hasil akurasi. Sehingga ada perbandingan hasil akurasi model k-nearest neighbors dengan dan tanpa menggunakan *K-Fold Cross Validation*. Hasil optimasi model k-nearest neighbors seperti penggunaan dan tanpa Teknik *cross validation* yang kemudian dianalisis melalui beberapa parameter nilai seperti

akurasi, presisi, recall dan f1 score serta merujuk pada manfaat penggunaan Teknik *cross validation* itu sendiri pada suatu data.

3.3.8 Kesimpulan

Setelah melakukan proses pengolahan data sampai tahap evaluasi hasil, dilakukan tahap pengambilan kesimpulan. Kesimpulan didapatkan berdasarkan hasil dari proses pengolahan data, hasil, evaluasi data sampai analisis akurasi yang akurat yang melibatkan Teknik *cross validation* dalam menguji model algoritma k-nearest neighbors.