BAB II
LANDASAN TEORI

2.1 Landasan Teori

2.1.1. Penyakit Jantung Koroner

Penyakit jantung koroner adalah keadaan ketika pembuluh darah koroner, yang bertugas menyuplai darah ke otot jantung, menyempit atau tersumbat karena penumpukan plak aterosklerosis. Kondisi ini berpotensi memicu angina, serangan jantung, gagal jantung, bahkan kematian mendadak. Proses *aterosklerosis* berkembang secara perlahan dan sering kali tanpa gejala signifikan hingga muncul komplikasi serius [10].

Faktor risiko penyakit jantung koroner terbagi menjadi dua kategori:

1. Faktor yang tidak bisa diubah, seperti usia, jenis kelamin, dan riwayat keluarga.
2. Faktor yang dapat dikelola, meliputi hipertensi, kolesterol tinggi, diabetes, obesitas, kurangnya aktivitas fisik, serta kebiasaan merokok.

Deteksi dini terhadap faktor-faktor risiko ini sangat penting dalam mencegah perkembangan penyakit lebih lanjut dan menurunkan angka mortalitas. Oleh karena itu, upaya sistematis dan teknologi berbasis data sangat dibutuhkan untuk mengidentifikasi potensi risiko secara lebih akurat.

2.1.2. Machine Learning

Machine Learning adalah bagian dari kecerdasan buatan yang memungkinkan komputer belajar dan membuat keputusan sendiri, menggunakan data masa lalu tanpa memerlukan arahan langkah demi langkah. Di bidang medis, pembelajaran mesin telah menjadi alat penting untuk membantu diagnosis, memprediksi perkembangan penyakit, dan membuat pilihan pengobatan. Algoritma ini diajarkan menggunakan informasi seperti tekanan darah, kadar kolesterol, berat badan, riwayat kesehatan keluarga, dan hasil tes. Karena dapat menemukan pola tersembunyi dalam data, sistem dapat menganalisis informasi ini dengan cepat dan

akurat untuk memberikan prediksi yang lebih akurat tentang masalah kesehatan [11].

Dalam kasus penyakit jantung koroner, *machine learning* menjadi solusi potensial untuk deteksi dini dan penilaian risiko pasien secara individual. Model klasifikasi yang dibangun mampu memberikan *output real-time* berupa rekomendasi medis berdasarkan data pasien, sehingga dapat mempercepat proses diagnosis. Pendekatan ini diyakini dapat meminimalisir kesalahan manusia dan subjektivitas dalam interpretasi klinis. Dengan demikian, penerapan machine learning tidak hanya meningkatkan efisiensi pelayanan kesehatan, tetapi juga mendukung intervensi dini yang lebih tepat bagi pasien dengan risiko tinggi [12].

2.1.3. K-Nearest Neighbors (KNN)

K-Nearest Neighbor (KNN) merupakan metode klasifikasi non-parametrik yang bekerja dengan memprediksi kelas suatu objek berdasarkan seberapa dekatnya dengan objek lain dalam data dengan data latih yang telah diketahui sebelumnya. Dalam prosesnya, KNN memanfaatkan prinsip kemiripan jarak antar data, di mana objek baru akan diklasifikasikan ke dalam kelas yang paling umum di antara tetangga terdekatnya. Metode ini terbagi menjadi dua fase utama, yaitu fase pembelajaran (*training*) yang menyimpan data historis dan fase pengujian (*testing*) yang menentukan kelas berdasarkan tetangga terdekat. Algoritma ini bersifat instan karena tidak membentuk model *eksplisit*, melainkan menyimpan data latih sebagai *referensi* klasifikasi.

Penentuan tetangga terdekat dilakukan dengan mengukur seberapa jauh data uji dari setiap data latih. Biasanya, *Euclidean Distance* digunakan untuk tujuan ini, yang secara spesifik mengukur jarak lurus di antara titik-titik dalam ruang fitur. Setelah jarak dihitung, sistem akan mengurutkan data latih berdasarkan jarak terpendek lalu memilih k tetangga terdekat sebagai bahan klasifikasi. Kelas dari objek baru akan ditentukan berdasarkan mayoritas label dari k tetangga tersebut. Keunggulan KNN terletak pada kesederhanaannya dan kemampuannya menangani berbagai tipe data, meskipun performanya dapat dipengaruhi oleh skala fitur dan distribusi data [13].

$$d = \sqrt{\Sigma_{i=1}^{\Pi}}(x_i - y_i)^z$$

Tahapan perhitungan dalam algoritma KNN adalah sebagai berikut:

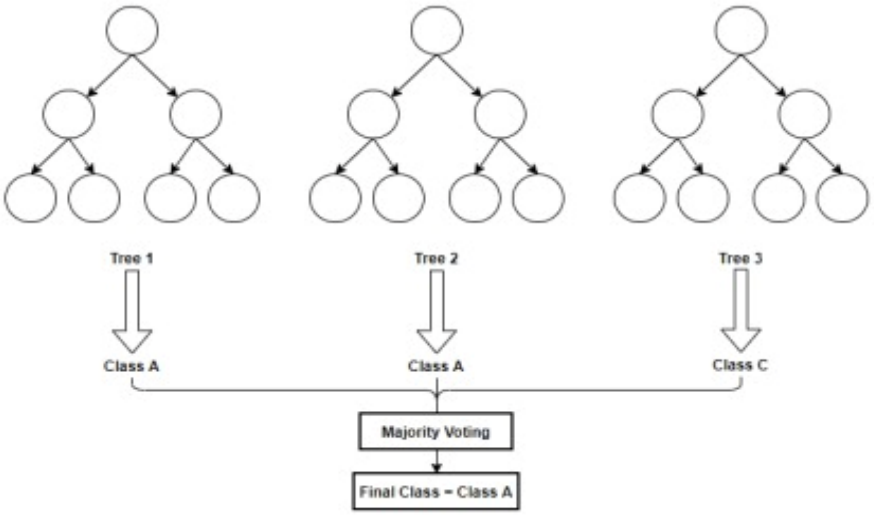
1. Tentukan nilai parameter k.
2. Hitung jarak antara data pelatihan dan data uji menggunakan rumus Jarak Euclidean.
3. Urutkan hasil jarak Euclidean dari yang terkecil.
4. Identifikasi k jarak terdekat.
5. Kumpulkan label kelas (Y) dari tetangga terdekat tersebut.
6. Tentukan kelas data yang dievaluasi berdasarkan mayoritas kelas dari tetangga terdekat.

2.1.4. Random Forest

Algoritma *Random* Forest merupakan bagian dari keluarga *CART (Classification and Regression Tree)* yang telah banyak digunakan dalam pengolahan data dan pengembangan model *machine learning*. Keunggulan utama dari metode ini terletak pada fleksibilitasnya karena tidak membutuhkan asumsi statistik seperti normalitas distribusi atau linearitas hubungan antar variabel. Hal ini menjadikan *Random Forest* sangat ideal untuk diterapkan pada dataset medis yang kompleks dan *heterogen*. Dalam penerapannya, *Random Forest* membangun banyak pohon keputusan independen yang masing-masing dilatih pada subset data acak, sehingga memntuk sebuah *ensemble* yang menyerupai hutan keputusan .

Setiap pohon dalam *Random Forest* melakukan prediksi berdasarkan karakteristik data pelatihan yang dibagi secara acak melalui teknik *bootstrap aggregating (bagging)*. Metode ini menghasilkan variasi pohon yang kaya dan membantu mengurangi *overfitting*. Selain itu, pemilihan fitur pada setiap node pemisahan dilakukan secara acak, yang meningkatkan keberagaman model dan kestabilan klasifikasi. Dengan demikian, *Random Forest* dikenal sebagai algoritma yang mampu menangani data besar, bersifat robust terhadap noise, dan menghasilkan akurasi tinggi dalam berbagai kasus klasifikasi, termasuk diagnosis

penyakit jantung koroner [14]. Gambar Ilustrasi *random forest* ditunjukkan pada Gambar 2.1 sebagai berikut:



Gambar 2.1. Ilustrasi *Random Forest*

Kelebihan *random forest* adalah sebagai berikut:

1. Keakuratannya sangat unggul bila dibandingkan dengan teknik klasifikasi lainnya.
2. Mampu menangani data imbalanced
3. Dapat menangani data dengan sampel besar
4. Dapat mengatasi data dengan noise dan missing data
5. Error yang dihasilkan relatif rendah

Kekurangan *random forest* adalah sebagai berikut:

1. Akurasi yang dihasilkan seringkali tidak konsisten.
2. Proses pengolahan data membutuhkan waktu yang cukup panjang.
3. Diperlukan penyesuaian model yang presisi agar sesuai dengan karakteristik data.

2.1.5. Jupyter Notebook

Jupyter Notebook merupakan aplikasi berbasis web yang dirancang untuk menulis, menjalankan, dan mendokumentasikan kode secara interaktif dalam satu antarmuka terpadu. *Platform* ini sangat populer di kalangan peneliti dan praktisi

data science karena kemampuannya dalam menggabungkan *skrip Python*, visualisasi data, serta penjelasan naratif dalam satu dokumen dinamis. Pengguna dapat memanfaatkan pustaka seperti *pandas, scikit-learn*, dan *matplotlib* untuk analisis dan model *machine learning* langsung dalam notebook. Hal ini menjadikan Jupyter sebagai alat yang sangat fleksibel dan efisien dalam mendukung alur kerja analitik berbasis data [15].

Dalam konteks diagnosis penyakit jantung koroner, *Jupyter Notebook* menjadi solusi ideal karena memungkinkan eksplorasi data medis secara transparan dan sistematis. *Notebook* ini mendukung praktik *reproducible research*, yaitu penelitian yang dapat diulang dengan hasil yang konsisten, yang sangat penting dalam validasi klinis. Dokumen notebook dapat mencakup semua tahapan mulai dari *preprocessing, training* model, evaluasi hingga visualisasi, sehingga memudahkan kolaborasi antar peneliti maupun integrasi dengan sistem informasi rumah sakit. Oleh karena itu, penerapan *Jupyter Notebook* dalam penelitian di RSUD Dr. Soeselo Slawi memiliki nilai strategis dalam mendorong inovasi teknologi medis yang akurat dan terdokumentasi dengan baik.

2.1.6. Python

Python hadir dengan berbagai *tools* dan fungsi bawaan yang lengkap, membantu pengembang dalam berbagai tugas pemrograman. Komunitasnya yang besar juga aktif berbagi pengetahuan dan saling mendukung. Meskipun sering dipakai untuk membuat *script, Python* juga dimanfaatkan secara luas di berbagai bidang lain dan mampu mengembangkan berbagai jenis *software*.

Python dapat beroperasi di berbagai sistem operasi, termasuk *Linux* dan *Unix, Windows, Mac OS X, Java Virtual Machine, Amiga, Palm, Symbian* (untuk produk-produk Nokia). *Python* didistribusikan dengan beberapa *lisensi* yang berbeda dari beberapa versi. Namun pada prinsipnya p*ython* dapat diperoleh dan dipergunakan *Python* dapat digunakan secara bebas, termasuk untuk kepentingan komersial, karena lisensinya sesuai dengan prinsip *Open Source* dan *General Public License* [16].

2.1.7. K-Fold Cross Validation

K-Fold Cross Validation adalah teknik evaluasi dalam pembelajaran mesin yang bertujuan untuk memperoleh pengukuran performa model yang lebih stabil dan tidak bias. Metode ini bekerja dengan membagi dataset ke dalam sejumlah lipatan (*folds*) yang proporsional, biasanya sebanyak 5 atau 10 bagian. Setiap lipatan akan bergiliran menjadi data uji, sementara sisa lipatan berperan sebagai data latih. Dengan melakukan pelatihan dan pengujian sebanyak K kali, evaluasi model menjadi lebih menyeluruh terhadap berbagai variasi distribusi data [17].

Pendekatan ini sangat berguna untuk menghindari hasil evaluasi yang terlalu bergantung pada satu pembagian data tertentu. Selain meningkatkan keandalan pengukuran akurasi, teknik ini membantu meminimalkan risiko *overfitting* karena model diuji dalam berbagai skenario. *K-Fold Cross Validation* juga sangat cocok digunakan dalam studi klasifikasi medis seperti diagnosis penyakit jantung koroner, karena data pasien umumnya memiliki distribusi kelas yang tidak seimbang. Dengan demikian, metode ini memberikan estimasi performa model yang lebih representatif terhadap data aktual yang akan dihadapi dalam implementasi nyata.

2.1.8. Confusion Matrix

Confusion matrix adalah alat bantu perhitungan yang dipakai untuk menunjukkan tingkat akurasi suatu klasifikasi. Cara kerjanya membandingkan jumlah total data yang berhasil diklasifikasikan dengan benar dengan seluruh data yang ada dalam *matrix* tersebut [18]. Tabel *Confusion Matrix* ditunjukkan pada Tabel 2.1 sebagai berikut:

Tabel 2.1. *Confusion Matrix*

		Prediction Class	
		Positive	*Negative*
Actual Class	*Positive*	*True Positive (TP)*	*False Negative (FN)*
	Negative	*False Positive (FP)*	*True Negative (TN)*

Penjelasan Komponen *Confusion Matrix* adalah sebagai berikut:

a. TN (*True Negative*): Model secara akurat memprediksi bahwa data tersebut termasuk dalam kelas negatif, dan pada kenyataannya, data tersebut memang negatif.
b. TP (*True Positive*): Model dengan benar mengklasifikasikan data sebagai positif, dan data tersebut memang benar-benar positif.
c. FN (*False Negative*): Model keliru memprediksi data sebagai negatif, padahal sebenarnya data tersebut adalah positif.
d. FP (*False Positive*): Model salah mengklasifikasikan data sebagai positif, meskipun data tersebut sebenarnya negatif.

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \times 100$$

Untuk memahami seberapa baik sebuah model klasifikasi bekerja, kita bisa melihat nilai *precision*, *recall*, dan F1-*score*. *Precision* mengindikasikan seberapa akurat model dalam mengidentifikasi kelas positif di antara semua yang dikatakannya positif. Sebaliknya, *recall* menilai seberapa lengkap model menemukan semua kasus positif yang sebenarnya.

$$precision = \frac{TP}{TP + FP}$$

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

F1-Score merupakan *harmonic mean* dari *precision* dan *recall*. Nilai terbaik *F1-Score* adalah 1.0 sedangkan nilai terburuknya adalah 0

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

2.2 Penelitian Terkait

Beberapa penelitian terdahulu yang relevan dengan topik ini telah dilakukan dan menunjukkan potensi machine learning dalam diagnosis penyakit. Penelitian-

penelitian ini menggunakan berbagai algoritma dan dataset dengan hasil yang bervariasi, namun secara umum menunjukkan akurasi yang *promising* untuk implementasi klinis.

Penelitian yang dilakukan oleh Ainul Yaqin, Defri Kurniawan, Junta Zeniarja yang memanfaatkan dataset Pima Indians Diabetes menunjukkan efektivitas algoritma *K-Nearest Neighbors* (KNN) dalam membangun model prediksi penyakit diabetes. Dengan menerapkan teknik *Random Over-Sampling* untuk mengatasi ketidakseimbangan kelas, serta optimasi parameter melalui *GridSearchCV*, model KNN yang dikembangkan berhasil mencapai akurasi sebesar 88%, *precisio*n 75%, *recall* 89%, dan *F1-score* 82%. Studi ini memperkuat pemahaman bahwa *machine learning*, khususnya KNN, dapat memberikan kontribusi nyata dalam deteksi dini penyakit kronis seperti diabetes serta mendukung peningkatan efisiensi diagnosis berbasis data [19].

Penelitian berikutnya yang menggunakan metode Random Forest dilakukan olah Ade Christian, Hariyanto, Ahmad Yani, Sumanto untuk prediksi penyakit paru-paru menunjukkan kinerja model yang sangat baik, dengan akurasi mencapai 94,7% dan *F1-score* sebesar 0,946. Pengujian dilakukan menggunakan parameter tertentu melalui *tools Orange* Data *Mining*, dan evaluasi berdasarkan *confusion matrix* serta kurva *klasifikasi* membuktikan stabilitas model dalam menggeneralisasi data. Studi ini menegaskan bahwa algoritma *Random Forest* merupakan pendekatan yang efektif dan andal dalam pengembangan model diagnosis penyakit [20].

Berbeda dengan penelitian yang dilakukan oleh Mahdiawan Nurkholifah, Jasmarizal, Yusran Umar, Rahmaddeni pada klasifikasi penyakit liver menggunakan rekam medis pasien, studi tersebut membandingkan performa algoritma *Support Vector Machine*, *Naïve Bayes*, dan *K-Nearest Neighbor* dalam mendeteksi peradangan hati. Penelitian ini menyoroti pentingnya pengolahan data medis secara otomatis untuk mendukung proses diagnosis yang lebih akurat. Hasil analisis menunjukkan bahwa *Naïve Bayes* memberikan akurasi tertinggi saat menggunakan teknik *SMOTE* untuk mengatasi ketidakseimbangan data, yaitu

sebesar 65,51%, sedangkan tanpa *SMOTE*, algoritma *Support Vector Machine* mencapai akurasi 72,41% [21].

Penerapan algoritma *machine learning* dalam klasifikasi jenis kanker payudara seperti yang dilakukan oleh Novita Ranti Muntiari dan Kharis Hudaiby Hanif menunjukkan hasil yang sangat menjanjikan. Penelitian ini menggunakan tujuh algoritma, termasuk *neural network, decision tree, naïve bayes, k-nearest neighbor, logistic regression, random forest, dan support vector machines*. Berdasarkan pengolahan data menggunakan aplikasi *RapidMiner,* algoritma *logistic regression, decision tree, naïve bayes, dan k-nearest neighbor* memiliki tingkat akurasi tertinggi sebesar 95,00%. Tingginya akurasi tersebut mempercepat proses pengambilan keputusan medis dalam mengidentifikasi jenis kanker payudara. Dengan demikian, *machine learning* dapat menjadi solusi efektif dan efisien dalam mendukung diagnosis kanker payudara [22].

Penelitian lain yang dilakukan oleh Wiji Lestari dan Sri Sumarlinda yang membandingkan algoritma machine learning dan deep learning untuk prediksi penyakit kardiovaskular berbasis data HER, studi ini melakukan analisis komparatif terhadap tujuh algoritma machine learning dalam klasifikasi kerentanan penyakit jantung menggunakan dataset dari UCI Machine Learning Repository. Dengan 300 data latih dan 100 data uji, fitur yang dianalisis meliputi usia, jenis kelamin, tekanan darah sistolik, kolesterol, thalach, oldpeak, dan slope, serta label cardio sebagai indikator status kardiovaskular. Model klasifikasi dibangun menggunakan algoritma Naïve Bayes, K-Nearest Neighbor (KNN), Decision Tree, Random Forest, Backpropagation, Logistic Regression, dan Support Vector Machine (SVM), yang kemudian dievaluasi menggunakan metrik akurasi, precision, recall, dan F1-score. Hasil menunjukkan bahwa Logistic Regression dan Backpropagation memberikan performa terbaik dengan akurasi masing-masing sebesar 81,00% dan 80,00%, serta F1-score 87,59% dan 87,02%, menunjukkan bahwa kedua algoritma tersebut unggul dalam menangani data outlier dan memiliki potensi tinggi dalam sistem klasifikasi penyakit jantung berbasis data mining [23].