BAB II
TINJAUAN PUSTAKA DAN DASAR TEORI

2.1. Tinjauan Pustaka

Penelitian yang dilakukan oleh Putri (2023), dalam penelitiannya membandingkan algoritma *K-Nearest Neighbors* (KNN), *Naïve Bayes Classiffier* (NBC), dan *Support Vector Machine* (SVM) untuk memprediksi kelulusan mahasiswa tingkat akhir. Berdasarkan pengolahan data, pengujian, dan analisis yang dilakukan terhadap 379 data publik, disimpulkan bahwa algoritma KNN mencapai kinerja rata-rata yang lebih tinggi pada berbagai pemisahan data (90:10, 80:20, 70:30, 60:40, dan 50:50). Secara spesifik, KNN menunjukkan akurasi sebesar 87,8%, presisi sebesar 87,8%, dan *recall* sebesar 84%. Hasil ini menunjukkan bawha algoritma KNN lebih unggul dari algoritma *Naïve Bayes Clasifier* (NBC) dan *Support Vector Machine* (SVM) dalam memprediksi tingkat kelulusan mahasiswa tingkat akhir.

Penelitian yang dilakukan oleh Nuraeni (2024), dalam penelitiannya melakukan analisis akurasi algoritma *Naïve Bayes* dan KNN dilakukan untuk menentukan penerima PKH. Berdasarkan hasil penelitian dan tiga skenario pengujian, ditemukan bahwa algoritma *Naïve Bayes* secara umum menghasilkan akurasi tertinggi dibandingkan dengan KNN untuk merekomendasikan penerima PKH, dengan akurasi rata-rata 77% di ketiga skenario pengujian. Namun, dalam pengujian *recall,* KNN menunjukkan kinerja yang lebih baik daripada *Naïve Bayes*. Ditunjukkan dengan nilai *recall* KNN yang lebih tinggi dibandingkan dengan *Naïve Bayes*, yaitu mencapai 100% dalam skenario pengujian pertama dan 75% pada skenario pengujian kedua.

Penelitian yang dilakukan oleh Muryono (2021), dalam penelitiannya, dilakukan perbandingan antara algoritma *K-Nearest Neighbor* (KNN), *Decision Tree* (C4.5), dan *Naïve Bayes* untuk menilai kelayakan kredit. Tujuan dari penelitian ini adalah untuk membandingkan akurasi algoritma tersebut dalam menentukan kelayakan kredit. Sebelas atribut digunakan untuk perbandingan ini: status perkawinan, jumlah tanggungan, usia, tingkat pendidikan, pekerjaan,

pendapatan bulanan, kepemilikan rumah, agunan, jumlah pinjaman, durasi pinjaman, dan deskripsi hasil. Pengujian dilakukan dengan menggunakan alat *Rapid Miner*, dan hasilnya menunjukkan bahwa algoritma *K-Nearest Neighbor* (KNN) mencapai akurasi tertinggi yaitu 93,33% dalam pengujian kelima. Algoritma *Decision Tree* (C4.5) mencapai akurasi maksimum sebesar 98,00% pada pengujian ketiga, sedangkan algoritma *Naïve Bayes* mendapatkan akurasi tertingginya sebesar 86,67% pada pengujian ketiga.

Penelitian yang dilakukan oleh Hunafa (2023), dalam penelitiannya melakukan perbandingan antara algoritma *Naïve Bayes* dan *K-Nearest Neighbor* (KNN) pada *dataset* diabetes yang tidak seimbang. Hasil penelitiannya menunjukkan bahwa *Naïve Bayes* dengan teknik SMOTE mencapai akurasi sebesar 71,66%, sedangkan *Naïve Bayes* tanpa SMOTE memiliki akurasi yang sedikit lebih tinggi yaitu 76,03%. KNN dengan SMOTE juga berkinerja baik, dengan akurasi 80,47%, sedangkan KNN tanpa SMOTE mencapai akurasi tertinggi yaitu 83,02%. Penelitian ini menyoroti bahwa penggunaan teknik SMOTE merupakan faktor penting, khususnya untuk algoritma *Naïve Bayes*. Meskipun *Naïve Bayes* dengan SMOTE memiliki akurasi yang sedikit lebih rendah daripada tanpa SMOTE, penggunaan SMOTE secara signifikan meningkatkan presisi (dari 32,27% menjadi 44,7%) dan perolehan kembali (dari 57,58% menjadi 72,66%). Hal ini menunjukkan bahwa SMOTE dapat membantu mengurangi masalah ketidak seimbangan kelas dalam *dataset* diabetes.

Penelitian yang dilakukan oleh Sahar (2020), dalam penelitiannya melakukan analisis perbandingan antara algoritma *K-Nearest Neighbor* (KNN) dan *Naïve Bayes Classifier* (NBC) pada *dataset* penyakit jantung. Penerapan metode KNN pada *dataset* penyakit jantung mencapai kinerja terbaiknya dengan pengujian *split* 20%, menghasilkan akurasi 68%, presisi 66%, *recall* 74%, dan *F-measure* 70% pada *K*=236 dengan metrik jarak *Manhattan*. Demikian pula, metode KNN dengan pengujian *split* 20% dan metrik jarak *Euclidean* mencapai akurasi 66%, presisi 65%, *recall* 70%, dan *F-measure* 67%. Metode NBC, yang diterapkan pada *dataset* penyakit jantung, memperoleh kinerja terbaiknya dengan pengujian *split* 10%, menunjukkan akurasi 58%, presisi 55%, *recall* 90%, dan *F-measure* 68%.

Berdasarkan kinerja komparatif ini, metode KNN menunjukkan hasil keseluruhan yang lebih baik dibandingkan dengan metode *Naïve Bayes Classiffier* (NBC).

Penelitian yang dilakukan oleh Wulandari (2024), penelitian dilakukan dengan menggunakan *Tool Rapid Miner* pada kumpulan data Penyakit Ginjal Kronis yang diperoleh dari *Kaggle.com*. Algoritma *Naïve Bayes* diuji menggunakan *K-Fold Cross Validation*, menghasilkan akurasi 94,25%, presisi 98,40%, *recall* 94,23%, dan AUC 0,961. Selain itu, penelitian yang dilakukan dengan menggunakan algoritma K-NN dengan *K-Fold Cross Validation*, menghasilkan akurasi 77,79%, presisi 80,20%, *recall* 95,06%, dan AUC 0,627. Hasil ini menunjukkan bahwa algoritma *Naïve Bayes* memberikan kinerja klasifikasi yang lebih baik dibandingkan dengan algoritma K-NN, yang menunjukkan potensinya untuk memprediksi pasien CKD dan non-CKD, yang dapat menjadi referensi untuk penelitian mendatang.

2.2. Dasar-Dasar Teori

2.2.1. *Data Mining*

Data mining adalah langkah-langkah untuk mengumpulkan dan mengolah data guna mendapatkan informasi berharga dari dalam data tersebut. Proses ini dapat dilakukan dengan memanfaatkan perangkat lunak yang memanfaatkan kalkulasi sistematis, matematika, atau teknologi *Artificial Intelligence* (AI) yang merupakan bidang dari berbagai disiplin ilmu (Syariful dkk., 2021). Dalam *data mining*, pemilihan fitur dan ekstraksi adalah dua aspek penting yang harus dipertimbangkan (Dol & Jawandhiya, 2023). Pada *data mining* terdapat beberapa metode yang dapat digunakan dan juga diterapkan untuk menganalisis pengetahuan secara mudah. Dalam sistem yang bertugas mengklasifikasikan seluruh *dataset* yang tersedia, kinerja sistem tidak dapat mencapai tingkat akurasi 100%. Oleh karena itu, kinerja sistem juga perlu untuk diukur. Biasanya, pengukuran kinerja klasifikasi dilakukan menggunakan *Confusion Matrix* (Shabrilianti dkk., 2023).

Data mining merupakan proses yang bersifat berulang dan melibatkan interaksi, yang bertujuan untuk mendapatkan aturan, pola, atau model baru yang dapat dipertanggungjawabkan kebenarannya serta memberikan manfaat signifikan

untuk pengambilan keputusan. Proses ini melibatkan analisis data yang mendalam untuk menemukan korelasi tersembunyi, tren, dan anomali yang tidak terlihat secara langsung. Dengan memanfaatkan teknik-teknik statistik, *machine learning*, dan algoritma kompleks, *data mining* memungkinkan meningkatkan kualitas keputusan yang lebih baik (Budiarti & Cendana, 2022). Berikut adalah beberapa karakteristik utama dari *data mining* dalam konteks analisis kelayakan pemberian bantuan untuk nelayan:

1. Penemuan Pola Tersembunyi

   Data mining berkaitan dengan mengidentifikasi pola yang sebelumnya tidak terdeteksi dalam kumpulan data. Proses ini melibatkan analisis data untuk menemukan korelasi, tren, dan anomali yang tersembunyi dalam *dataset*. Misalnya, dalam analisis kelayakan pemberian bantuan untuk nelayan, *data mining* dapat mengungkap faktor-faktor yang mempengaruhi nelayan untuk mendapatkan bantuan.

2. Penggunaan *Dataset* Besar

   Data mining biasanya melibatkan penggunaan kumpulan data yang sangat besar. Kumpulan data yang berjumlah banyak dan beragam sangat penting untuk memperoleh hasil yang lebih tepat dan dapat dipercaya. Semakin banyak data yang digunakan, semakin besar peluang untuk menemukan pola yang signifikan dan membuat prediksi yang lebih tepat.

3. Pembuatan Keputusan Kritis

   Pemanfaatan *data mining* dapat membantu dalam pengambilan keputusan-keputusan yang bersifat krusial, khususnya dalam hal pengembangan strategi dan operasional. Pengetahuan yang dihasilkan dari proses *data mining* dapat dimanfaatkan untuk mendukung keputusan manajerial yang penting, seperti menentukan kelayakan penerima bantuan, mengidentifikasi nelayan yang paling membutuhkan, atau mengembangkan program pemberdayaan yang lebih efektif. Dalam sektor pemberian bantuan nelayan, hasil *data mining* dapat membantu lembaga pemerintah dalam menilai resiko dan kelayakan masyarakat nelayan sebagai penerima bantuan.

4. Integrasi Dengan Teknik Lain

Data mining sering kali diintegrasikan dengan teknik dan teknologi lain seperti *machine learning*, statistik, dan *database management*. Kombinasi ini memperkaya *analisis* dan meningkatkan kemampuan untuk menemukan wawasan yang lebih mendalam. Misalnya, penggunaan algoritma pembelajaran mesin dalam *data mining* dapat meningkatkan akurasi prediksi dan efisiensi proses analisis, sehingga membantu dalam menentukan alokasi bantuan yang optimal.

Di dalam *data mining*, teknik pelatihan diterapkan untuk mengembangkan model yang mampu memprediksi dan mengklasifikasikan data baru dengan mengacu pada data historis yang tersedia. Umummya, teknik pelatihan dalam *data mining* dibedakan menjadi dua kategori:

1. *Supervised Learning*

Supervised learning adalah jenis *machine learning* yang model atau algoritma belajar dari data berlabel. Ini berarti bahwa setiap contoh data dalam set pelatihan sudah dilabeli atau *output* yang sesuai yang menjadi acuan bagi model untuk mempelajari hubungan antara fitur (*input*) dan label (*output*). Metode ini bertujuan untuk mengkategorikan data baru ke dalam kelompok data yang sudah ada. Tujuan lain dari metode ini adalah untuk mempelajari fungsi pemetaan dari *input* ke *output*, sehingga memungkinkan model untuk melakukan prediksi atau klasifikasi untuk data baru yang belum ditemui. Contoh algoritma *supervised learning* yaitu, *Decision Tree, Random Forest, Naïve Bayes, Support Vector Machine,* dan *K-Nearest Neighbor.*

2. *Unsupervised Learning*

Unsupervised learning adalah jenis *machine learning* yang modelnya atau algoritma belajar dari data yang tidak berlabel atau *output* yang sudah ditentukan. Tujuannya yakni untuk mengenali pola atau struktur yang tidak terlihat dalam data tersebut. *Unsupervised learning*, merupakan model yang bekerja dengan menemukan hubungan atau struktur dalam data tanpa panduan dari label atau *output* yang sudah ada. Contoh algoritma *unsupervised learning*

yaitu, *K-Means, Association Rule Mining, Fuzzy C-Means* dan *Hierarchial Clustering.*

Data mining dan *Knowledge Discovery in Database* (KDD) adalah istilah yang menggambarkan proses penemuan informasi yang tidak terlihat dalam kumpulan data besar. Walaupun kedua konsep itu mempunyai perbedaan, keduanya saling berkaitan antara satu dengan yang lainnya, dan *data mining* merupakan satu tahap dalam KDD. *Knowledge Discovery in Database* (KDD) adalah proses yang tujuannya untuk mengeksplorasi, menganalisis, dan mengambil informasi atau wawasan yang berguna dari sejumlah data besar.

Knowledge Discovery in Database (KDD) merupakan salah satu metode ilmiah yang diterapkan didalam *data mining*. *Data mining* merupakan salah satu tahap pada proses KDD, yaitu mengidentifikasi pola atau tren yang diharapkan dalam kumpulan data besar. Keuntungan dari *data mining* adalah terletak pada kemampuannya untuk membantu pengambilan keputusan di masa yang akan datang. Dalam tahap ini, pola yang dihasilkan dapat diidentifikasi oleh perangkat tertentu, yang kemudian dapat memberikan solusi untuk meningkatkan proses pengambilan keputusan (Pahlevi dkk., 2023).

Berikut ini adalah langkah-langkah penting dalam proses *Knowledge Discovery in Database* (KDD):

1. *Data Cleaning*

   Data cleaning atau pembersihan data adalah proses menghapus duplikat data, menghapus data yang tidak diperlukan, memeriksa ketidak konsistenan dan mengoreksi kesalahan, seperti kesalahan ketik dan nilai yang hilang, tujuannya untuk meningkatkan kualitas data.

2. *Data Integration*

   Data integration atau integrasi data adalah tahap melakukan proses menggabungkan data yang sudah ada dengan informasi relevan lainnya, sering kali dengan menggabungkan data dari beberapa kumpulan data menjadi satu data terpadu yang diperlukan untuk proses KDD.

3. *Data Selection*

   Data selection atau seleksi data di mana pada tahap ini, data yang relevan dipilih untuk melakukan analisis dari data operasional. Data yang sudah dipilih kemudian disimpan dalam kumpulan data terpisah. Dengan melakukan data seleksi, proses selanjutnya menjadi lebih efisien dan fokus pada data yang signifikan, mengurangi beban komputasi dan memastikan hasil analisis yang lebih akurat dan tepat sasaran.

4. *Data Transformation*

   Pada tahap ini, data diubah kedalam format tertentu agar kompetibel dengan proses *data mining.* Transformasi data diperlukan dalam penelitian ini untuk mengonversi tipe data teks menjadi data numerik. Proses ini bertujuan untuk memperbaiki konsistensi, akurasi, dan efisiensi dalam proses analisis, sehingga algoritma dapat lebih efektif dalam mengidentifikasi pola atau informasi tersembunyi yang ada di dalam data.

5. *Data Mining*

   Pada tahap ini, dilakukan upaya untuk menemukan pola atau informasi dengan memanfaatkan algoritma tertentu. Dalam penelitian ini, algoritma klasifikasi *Naïve Bayes* dan *K-Nearest Neighbor* diimplementasikan dan digabungkan dengan teknik penyeimbang data yaitu SMOTE.

6. *Pattern Evaluation*

   Pada tahap ini, proses untuk mengenali pola yang penting dari hasil *data mining*. Selama fase ini, hasil teknik penambangan data, yang terdiri dari pola-pola spesifik atau model prediksi akan dievaluasi untuk menentukan apakah sesuai dengan hipotesis yang telah ditetapkan.

7. *Knowledge Presentation*

   Pada tahap ini, pola informasi yang diperoleh dari proses *data mining* ditampilkan. Visualisasi ini berfungsi untuk menyampaikan hasil *data mining* dengan jelas dan mudah dipahami (Rahman dkk., 2022).

2.2.1.1. Fungsi *Data Mining*

Data mining memiliki beberapa fungsi penting yang memungkinkan peneliti dan praktisi untuk mengekstrak informasi berharga dari kumpulan data besar,

mengidentifikasi tren tersembunyi dan membuat keputusan berdasarkan data yang diperoleh. Berikut adalah fungsi dari *data mining*:

1. *Classification*

   Klasifikasi merupakan proses mengidentifikasi kategori atau kelas dari data tertentu untuk mendeskripsikan serta membedakan berbagai kelas atau kategori dalam data. Proses ini menggunakan algoritma untuk menganalisis data yang telah diklasifikasi sebelumnya, sehingga model yang terbentuk dapat memprediksi kelas data baru dengan tingkat akurasi yang tinggi.

2. *Clustering*

   Clustering berfungsi untuk mengelompokkan atribut ke dalam segmen berdasarkan kesamaannya. Proses ini bertujuan untuk menemukan struktur tersembunyi dalam data tanpa perlu label atau klasifikasi awal, sehingga pola-pola baru dapat dikenali.

3. *Associasion*

   Berfungsi untuk mengidentifikasi hubungan antara atribut atau set item berdasarkan frekuensi kemunculan item secara bersamaan dan *rule associasion* yang telah ditentukan. Proses ini membantu dalam mengenali pola atau hubungan yang mungkin tersembunyi di dalam data.

4. *Regression*

   Regresi merupakan teknik statistik yang berfungsi untuk menganalisis keterkaitan antara satu atau lebih variabel independen (umumnya disebut variabel prediktor) dan satu variabel dependen (sering disebut variabel respons). Fungsi utama dari regresi adalah untuk membuat model dan meramalkan nilai variabel independen dan variabel dependen.

5. *Forecasting*

   Berfungsi untuk memperkirakan hasil di masa mendatang berdasarkan tren yang diamati pada waktu yang sebelumnya.

6. *Sequence Analysis*

   Bertujuan untuk mengidentifikasi pola dalam runtututan kejadian yang terjadi secara berurutan.

7. *Deviation Analysis*

   Fungsi untuk mengidentifikasi kejadian tidak biasa yang menyimpang secara signifikan dari kondisi pada umumnya (juga dikenal sebagai anomali).

2.2.1.2. Teknik *Data mining*

Berbagai teknik dalam *data mining* yang paling umum adalah sebagai berikut:

1. Teknik statistik klasik, seperti *quadratic, linear*, dan *logistic discriminate analyses.*
2. Teknik statistik modern, seperti *density estimation, K-Nearest Neighbor,* dan *bayesian networks* merupakan metode yang lebih maju dalam analisis data.
3. *Artificial Neural Network* (ANN) atau Jaringan Syaraf Tiruan merujuk pada model terstruktur yang dirancang untuk meniru atau mensimulasikan karakteristik struktural dan fungsional jaringan saraf yang ada pada makhluk hidup.
4. *Support Vektor Machine* (SVM), adalah serangkaian metode *supervised learning* yang difungsikan melakukan regresi dan klasifikasi.
5. *Associayion Rules* (AR), adalah metode penelitian yang digunakan untuk menemukan keterkaitan antara variabel dalam basis data.
6. *Decision Tree* (DT), merupakan sebuah model prediktif dalam *data mining* dan pemberlajaran mesin digunakan untuk memodelkan hubungan antara sekumpulan variabel *input* dan variabel target. Model ini berbentuk pohon keputusan (*Decision Tree*).
7. *Case Base Reasoning* (CBS), adalah pendekatan penyelesaian masalah baru dengan memanfaatkan solusi dari kasus sebelumnya yang serupa.
8. *Fuzzy Logic System* (FLS), adalah jenis logika yang memiliki nilai ganda dan berkaitan dengan penarikan kesimpulan melalui pendekatan yang bersifat aproksimasi. Dalam logika *fuzzy*, nilai kebenaran berada di antara 0 dan 1. *Genetic Algorithms* (GA), adalah metode komputasi yang terinspirasi dari proses evolusi biologis, khususnya mekanisme seleksi alam dan genetika (Setiyani dkk., 2020).

2.2.2. Klasifikasi

Klasifikas yaitu proses yang bertujuan untuk membangun serangkain model atau fungsi yang mampu menguraikan serta memisahkan data kedalam kategori tertentu, sehingga model tersebut dapat digunakan untuk mengidentifikasi kelas objek yang kelasnya masih belum diketahui. Proses ini sangat penting dalam berbagai aplikasi, seperti pengenalan pola, diagnosis medis, dan pengolahan citra, di mana pengelompokan data yang akurat dapat menghasilkan keputusan yang lebih baik. Dalam klasifikasi terdapat dua proses utama, yaitu proses pembelajaran atau pelatihan, yang membangun model menggunakan data pelatihan, dan proses pengujian, yang menguji data uji menggunakan model yang didapatkan dari langkah pelatihan (Raysyah dkk., 2021). Dalam pembagian *dataset* untuk klasifikasi, data dipisahkan dengan dua bagian, yaitu data pelatihan dan data uji. Semua *dataset* akan dipisahkan dengan dua klaster yang digunakan untuk melatih dan menguji model. Pemisahan data ini nantinya akan mempengaruhi tingkat akurasi dari pelaksanaan suatu metode dalam klasifikasi. Dengan cara ini, model dapat dioptimalkan untuk mengenali pola dan melakukan perkiraan yang lebih akurat mengacu pada data yang telah dilatih (Bianto dkk., 2019).

Pemilihan model klasifikasi seperti *Naïve Bayes* dan *K-Nearest Neighbor* dalam penelitian ini dipilih melalui pertimbangan fokus dari masalah yang akan diselesaikan untuk menganalisis kelayakan pemberian bantuan nelayan, yang merupakan tugas klasifikasi. Model klasifikasi, seperti *Naïve Bayes* dan KNN, memiliki kemampuan untuk memperkirakan kelas atau label dari data yang mengacu pada fitur-fitur yang diamati (Ariyanti & Iswardani, 2020). Model ini dapat digunakan untuk memprediksi kelayakan nelayan dalam menerima bantuan, dengan mempertimbangkan karakteristik dan atribut tertentu, seperti jumlah kapal, sampan motor, sampan biasa, jumlah pancing, jumlah jaring, serta keterangan lainnya.

Model *clustering*, tidak cocok dalam kasus ini karena tujuan utama dari *clustering* adalah untuk mengelompokkan data menjadi beberapa kelompok berdasarkan kemiripan fitur, tanpa memperhatikan label kelas atau prediksi (Sholeh

& Aeni, 2023). Dalam kasus analisis kelayakan pemberian bantuan nelayan sudah memiliki label kelas yang ditentukan (yaitu, layak dan tidak layak).

Dengan memilih model klasifikasi *Naïve Bayes* dan KNN, penelitian ini dapat lebih efektif dalam memenuhi tujuannya untuk melakukan analisis kelayakan pemberian bantuan nelayan untuk mendukung pemberdayaan nelayan, dengan mengklasifikasi label kelas berdasarkan data dan fitur yang tersedia.

2.2.3. *Synthetic Minority Over-sampling Technique* (SMOTE)

Synthetic Minority Over-sampling Technique (SMOTE) merupakan metode untuk menyeimbangkan berbagai kelas melalui penerapkan teknik *oversampling*. Metode SMOTE menggandakan data dalam kelas minoritas supaya data kelas minoritas dapat seimbang dengan data dalam kelas mayoritas (Sholihah & Hermawan, 2023). *Dataset* yang tidak seimbang dapat menyebabkan kesalahan dalam klasifikasi, karena data dalam kelas minoritas ini selalu salah diklasifikasikan sebagai kelas mayoritas. Ketidakseimbangan kelas ini terjadi ketika kelas mempunyai total *instance* yang jauh lebih banyak dibandingkan dengan kelas lainnya, yang dapat menyebabkan model pembelajaran mesin menjadi bias terhadap kelas mayoritas (Prianata, 2024). Dengan menggunakan teknik SMOTE ini dapat membantu model untuk lebih memahami pola dalam kelas minoritas tanpa *overfitting* yang dapat terjadi jika hanya menduplikasi data. Dengan menyeimbangkan jumlah sampel pada kelas minoritas dan mayoritas, SMOTE mengurangi bias model terhadap kelas mayoritas, meningkatkan kapasitas model dalam megidentifikasi dan memprediksi kelas minoritas (Nugroho & Harini, 2024).

Teknik ini berfungsi dengan membangun data sintetis dari kelas minoritas sehingga seimbang dengan kelas mayoritas. *Synthetic Minority Over-sampling Technique* (SMOTE) adalah metode lanjutan dari *oversampling.* Metode ini membuat beberapa salinan data, yang disebut dengan data sintetis. Tujuan penerapan SMOTE adalah untuk mengurangi ketidak seimbangan kelas sehingga model yang dihasilkan lebih akurat. Teknik SMOTE ini menciptakan data sintetis atau data tiruan dengan mengukur kedekatan antara data numerik menggunakan jarak *Euclidean*, sedangkan untuk data kategoris menggunakan nilai modus (Iskandar & Nataliani, 2021). Dengan menerapkan SMOTE, model pembelajaran

mesin dapat belajar lebih baik dari data minoritas, meningkatkan akurasi, dan mengurangi bias yang mungkin ada terhadap kelas mayoritas. Metode ini sangat bermanfaat dalam situasi di mana kelas minoritas memiliki pengaruh yang besar. Berikut adalah persamaan yang digunakan ditunjukkan pada persamaan 2.1:

$$X_{syn} = X_i + (X_{knn} - X_i)\ x\ \delta \tag{2.1}$$

Dimana, X_{syn} mewakili data sintesis yang akan dibuat, X_{knn} adalah jarak terdekat dari data yang akan disintetis, X_i adalah titik data asli yang akan di duplikasi dan δ nilai acak antara 0 dan 1 (Azizah dkk., 2022).

2.2.4. *Naïve Bayes*

Naïve Bayes merupakan algoritma klasifikasi yang memiliki struktur algoritma praktis dan efisiensi komputasi yang tinggi (Chen dkk., 2021). Model ini efektif untuk melakukan klasifikasi pada teks. Model ini adalah bentuk dasar dari *Bayesian Network*, di mana setiap karakteristik independen diberikan nilai variable kelas (Farhana, 2021). *Naïve Bayes* mempunyai sejumlah kelebihan diantaranya adalah kesederhanaannya, kecepatan dalam proses perhitungan dan tingkat akurasi yang tinggi (Indrayuni, 2019). *Naïve Bayes* ini menggunakan teori probabilitas untuk mengklasifikasikan data (Prabha dkk., 2019). Metode *Naïve Bayes* cukup membutuhkan sedikit data pelatihan dalam memperkirakan parameter yang dibutuhkan pada proses klasifikasi. Metode ini juga cukup efisien untuk diterapkan pada berbagai jenis *dataset*. Algoritma *Naïve Bayes* lebih mudah diimplementasikan karena proses perhitungannya yang singkat (Putro dkk., 2020). Hal ini menjadikannya pilihan yang efektif, terutama dalam situasi di mana data pelatihan terbatas.

Secara garis besar, algoritma *Naïve Bayes* merupakan pengklasifikasi himpunan data statistik yang memperkirakan semua probabilitas untuk setiap anggota kelas. Algoritma *Naïve Bayes* telah terbukti menunjukkan kecepatan dan akurasi yang tinggi saat digunakan terhadap kumpulan data besar, kecil maupun sedang (Ridwan, 2020). Adapun struktur umum dari *Teorema Bayes* ditunjukkan dalam persamaan 2.2:

$$P(H \mid X) = \frac{P(H \mid X)P(H)}{PX} \quad (2.2)$$

Keterangan rumus *Teorema Naïve Bayes*:

X = Data yang kelasnya masih tidak teridentifikasi.

H = Hipotesis untuk data X yaitu suatu kelas tertentu

P(H|X) = Probabilitas hipotesis H ditentukan kondisi x (*posteriori* prob.)

P(H) = Probabilitas hipotesis H (*prior* prob.)

P(X|H) = Probabilitas X

P(X) = Probabilitas dari X

Berikut adalah proses penyelesaian dari algoritma *Naïve Bayes*, yang melibatkan serangkaian langkah sistematis untuk mengklasifikasikan data berdasarkan prinsip-prinsip probabilistik.

1. Baca dan perhatikan data *training*, dengan cara memperhatikan struktur data dan mengidentifikasi fitur-fitur yang relevan.
2. Menghitung total dan probabilitas.
   a. Apabila terdapat data numerik dalam *dataset,* langkah yang dilakukan adalah mencari nilai rata-rata dan standar deviasi untuk masing-masing parameter yang mengilustrasikan data numerik tersebut. Untuk menghitung nilai rata-rata dapat diselesaikan dengan menerapkan persamaan 2.3:

$$\mu \sum_{i=1}^{n} x_i \text{ Atau } \mu = \frac{x_1+x_2+x_3+\cdots+x_n}{n} \quad (2.3)$$

Keterangan :

μ : rata – rata hitung (*mean*)

xi : nilai sampel ke –i

n : jumlah sampel

Rumus yang digunakan untuk mengkalkulasi nilai standar deviasi dilakukan dengan persamaan 2.4 dibawah ini:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i-\mu)^2}{n-1}} \quad (2.4)$$

Keterangan:

σ : simpangan baku (standar deviasi)

x_i : nilai x ke –i

μ : rata-rata hitung

n : jumlah sampel

b. Apabila tidak terdapat data numerik, maka hitunglah probabilitas di setiap fitur dengan cara membagi total data dalam fitur tersebut dengan total data dalam kategori itu.

3. Nilai probabilitas dari setiap fitur dalam masing-masing kelas.

Dalam menentukan nilai probabilitas masing-masing fitur dalam suatu kelas, perlu menghitung total data yang sesuai dalam fitur tersebut, lalu membaginya dengan total data pada kategori tersebut.

4. Nilai distribusi *gaussian*.

Tahap berikutnya adalah mengkalkulasikan nilai probabilitas untuk fitur dari data pengujian yang memiliki data numerik. Untuk mencari nilai distribusi *gaussian* ditunjukkan pada persamaan 2.5:$A = \pi r^2$

$$P = \left(X_i = x_i \middle| Y = y_j\right) = \frac{1}{\sqrt{2\pi\sigma_{ij}}} \times e^{-\frac{(x_i-\mu_{ij})^2}{2\sigma^2 ij}} \quad (2.5)$$

Keterangan:

P : Probabilitas

X_i : Fitur ke-i dari data yang diukur

x_i : Nilai aktual dari fitur X_i yang diamati pada sampel

$Y = y_j$: Kelas y_j yang diambil oleh variabel target Y

μ_{ij} : Rata-rata (mean) dari fitur X_i untuk kelas y_j

σ_{ij} : Simpangan baku (*standard deviation*) dari fitur X_i untuk kelas y_j

$\frac{1}{\sqrt{2\pi\sigma_{ij}}}$: Faktor normalisasi untuk distribusi *gaussian*

e : Bilangan eksponensial

5. Probabilitas akhir dari setiap kelas

Langkah berikutnya adalah mengkalkulasikan probabilitas akhir dari masing-masing kelas, dengan memasukkan seluruh nilai data distribusi *gaussian* ke dalam kelas yang sama sesuai persamaan 2.6 berikut:

$$P(X|Kelas)=P(V1|Kelas)\times P(V2|Kelas)\times P(V3|Kelas)\times P(V4|Kelas)\times P(V5|Kelas) \quad (2.6)$$

Keterangan :

P(X|Kelas) : Probabilitas bersyarat dari *instance* XXX (gabungan fitur V1,V2,V3,V4,V5) berada dalam suatu kelas tertentu

P(V1|Kelas): Probabilitas bersyarat dari setiap fitur V1,V2,V3,V4,V5 muncul, dengan syarat *instance* tersebut berada dalam kelas yang ditentukan

6. Probabilitas keseluruhan

Probabilitas keseluruhan ini didapatkan dengan memasukkan nilai probabilitas akhir dari setiap kelas pada persamaan *Naïve Bayes*. Di bawah ini adalah perhitungan probabilitas dengan persamaan 2.7 di bawah ini:

$$P(Kelas|X) = P(Kelas) * P(X) \quad (2.7)$$

Keterangan :

P(Kelas|X) : Probabilitas posterior, yaitu probabilitas suatu *instance* X termasuk dalam kelas tertentu setelah memperhitungkan bukti dari data X

P(Kelas) : Probabilitas prior dari kelas, yaitu probabilitas awal suatu kelas tanpa memperhitungkan data X (berdasarkan distribusi kelas dalam *dataset*)

P(X) : Probabilitas *likelihood* dari data X, yaitu probabilitas data X terjadi tanpa memperhitungkan kelasnya (sering diabaikan dalam perbandingan antar kelas)

Setelah memperoleh probabilitas akhir, tahap berikutnya adalah melakukan normalisasi dengan membagi nilai probabilitas dari kategori tertentu dengan total

nilai dari semua kategori, dapat dilakukan dengan persamaan 2.8 (Imandasari dkk., 2019).

$$P(Kelas)= P(Kelas|X) / (P(X|Kelas)+P(X|Kelas)) \quad (2.8)$$

Keterangan :

P(Kelas) : Probabilitas akhir dari sebuah kelas setelah normalisasi

P(Kelas|X) : Probabilitas posterior dari kelas, yaitu probabilitas kelas berdasarkan data X

P(X|Kelas) : Probabilitas data X muncul dalam suatu kelas tertentu (*likelihood* dari data dalam kelas)

2.2.5. *K-Nearest Neighbor* (KNN)

Algoritma *K-Nearest Neighbor* (KNN) merupakan metode untuk mengklasifikasikan target dengan merujuk pada titik data pelatihan yang lebih dekat dengan target tersebut. Algoritma klasifikasi KNN adalah metode yang sederhana dan non-parametrik. *K-Nearest Neighbor* didasarkan pada konsep "belajar dengan analogi". Data pembelajaran digambarkan melalui atribut numerik berdimensi-n. Pada data pembelajaran mewakili titik yang diberi label c dalam ruang berdimensi-n (Raysyah dkk., 2021).

K-Nearest Neighbor merupakan metode umum yang sering diterapkan dalam berbagai tugas klasifikasi teks. Yaitu dengan melakukan proses pembelajaran dari data pelatihan guna menentukan kelompok objek (Sahu dkk., 2024). Untuk menentukan hasil klasifikasi *K-Nearest Neighbor* yaitu dilakukan melalui melihat jarak terdekat dari setiap objek dalam kelompok yang berbeda. Jarak ini dihitung dengan mengukur kedekatan antara data masukan dan data kelompok berdasarkan nilai dari berbagai fitur yang ada (Argina, 2020). Algoritma ini berfungsi dengan mengidetifikasi jarak terdekat dari sampel uji ke sampel pelatihan untuk menetapkan tetangga terdekatnya. Setelah mendapatkan KNN, mayoritas dari KNN tersebut diambil untuk digunakan sebagai prediksi dari sampel uji. Jarak atau ukuran perbedaan antara sampel dapat dihitung dengan mengukur jarak menggunakan metode *Euclidean* (Putra dkk., 2022).

Persamaan berikut menggunakan matrik satuan rak atau biasanya dikenal dengan istilah *Euclidean* ditunjukkan dalam persamaan 2.9.

$$Euclidean = \sqrt{\sum_{i=1}^{n}(pi - pi)^2} \quad (2.9)$$

Keterangan :

Euclidean : jarak terdekat

pi : data latih atau sampel data

qi : Data uji

n : Total atribut untuk setiap kelas

i : Atribut masing-masing dari 1 hingga n

Langkah-langkah berikut dapat diikuti untuk melakukan perhitungan klasifikasi dengan metode *K-Nearest Neighbor* :

1. Tentukan parameter *K* yang sesuai dengan data yang digunakan. Nilai K minimal bernilai 1 dan maksimalnya jumlah dari data latih.
2. Menghitung jarak antara data uji dan data latih. Untuk menghitung jarak tersebut di dalam perhitungan algoritma KNN biasanya menggunakan perhitungan jarak *Euclidean* dengan rumus seperti persamaan 2.9.
3. Kemudian jarak tersebut diurutkan dari yang terbesar ke yang terkecil.
4. Menentukan jarak terdekat sampai pada parameter *K*.
5. Pasangkan kelas yang sesuai.
6. Mencari jumlah kelas dari tetangga terdekat dan menetapkan kelas tersebut sebagai kelas data yang akan diklasifikasikan atau dievaluasi (Kurniawan & Barokah, 2020).

2.2.6. *K-Fold Cross Validation*

K-Fold Cross Validation merupakan metode yang diterapkan dalam *machine learning* untuk menilai efektifitas model. Teknik *K-Fold Cross Validation* ini dapat dimanfaatkan untuk melatih dan menguji *dataset,* di mana pembagian dilakukan secara efektif dengan ukuran *K* yang sama untuk memperoleh nilai akurasi (Sandi dkk., 2023). Teknik ini memisahkan data menjadi dua segmen yaitu data pelatihan dan data pengujian. Data pelatihan digunakan untuk pembelajaran, adapun data pengujian digunakan untuk menilai kinerja model. Dalam *Cross Validation* ini data

di silangkan berturut sehingga setiap titik data berkesempatan untuk divalidasi. Pada penelitian ini menggunakan *dataset* sebanyak 302 data nelayan yang kemudian akan dibagi menjadi 10 *folds*. Masing-masing *fold* akan berisi sekitar 30 atau 31 data. Model dilatih sepuluh kali, setiap kali pelatihan menggunakan sembilan *fold* dan satu *fold* sebagai data pengujian.

2.2.7. *Confusion Matrix*

Confusion Matrix merupakan sebuah tabel yang berfungsi untuk menilai kinerja model klasifikasi pada *dataset* yang sudah melalui proses klasifikasi. *Confusion Matrix* terdiri dari empat gabungan yang berbeda antara hasil prediksi dan nilai sebenarnya (Normawati & Prayogi, 2021). Funggsi dari *Confusion Matrix* ini adalah untuk membuat ilustrasi yang jelas tentang sejauh mana model mampu membedakan antara kelas target, dengan melihat jumlah perkiraan yang benar dan keliru yang dihasilkan oleh model.

Dalam evaluasi performa dengan *Confusion Matrix*, mempunyai empat elemen utama yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). *True Negative* (TN) menunjukkan total data negatif yang berhasil diidentifikasi dengan benar, sedangkan *False Positive* (FP) adalah data negatif yang justru teridentifikasi sebagai data positif. Di sisi lain, *True Positive* (TP) adalah data positif yang teridentifikasi dengan benar, sementara *False Negative* (FN) adalah lawan dari *True Positive*, yakni data positif yang teridentifikasi sebagai data negatif (Khairi dkk., 2021). Bentuk tabel *Confusion Matrix* ditunjukkan pada Tabel 2.1.

Tabel 2. 1. *Confusion Matrix*

Aktual	**Prediksi**	
	True	*False*
True	TP (*True Positive*)	FP (*False Positive*)
False	FN (*False Negative*)	TN (*True Negative*)

Berdasarkan niali (TP), (TN), (FP) dan (FN) dapat dihitung akurasinya. Akurasi yaitu rasio antara total data yang terklasifikasi dengan benar terhadap total data keseluruhan. Perhitungan nilai akurasi dapat dilakukan dengan menggunakan persamaan 2.10:

$$Akurasi = \frac{TP+TN}{TP+TN+FP+F} x\ 100\% \quad (2.10)$$

Nilai presisi mencerminkan proporsi antara jumlah data positif yang berhasil diklasifikasikan dengan benar dibandingkan dengan total seluruh data yang diklasifikasikan sebagai positif. Perhitungan nilai presisi dapat dilakukan dengan menggunakan persamaan 2.11:

$$Presisi = \frac{TP}{TP+F} x\ 100\% \quad (2.11)$$

Nilai *recall* (sensitifitas) yaitu rasio antara data yang diprediksi positif benar terhadap jumlah total data positif benar (Pratiwi dkk., 2020). Perhitungan nilai *recall* dapat dilakukan dengan menggunakan persamaan 2.12:

$$Recall = \frac{TP}{TP+FN} x\ 100\% \quad (2.12)$$

Nilai $F_{measure}$ atau di sebut juga *F1-score* merupakan nilai harmonis, ini didapatkan dari perhitungan *recall* dan *precision*. Berikut persamaan yang dapat digunakan untuk menghitung nilai $F_{measure}$:

$$F_{measure} = 2\frac{presisi \times recall}{presisi+r} x\ 100\% \quad (2.13)$$