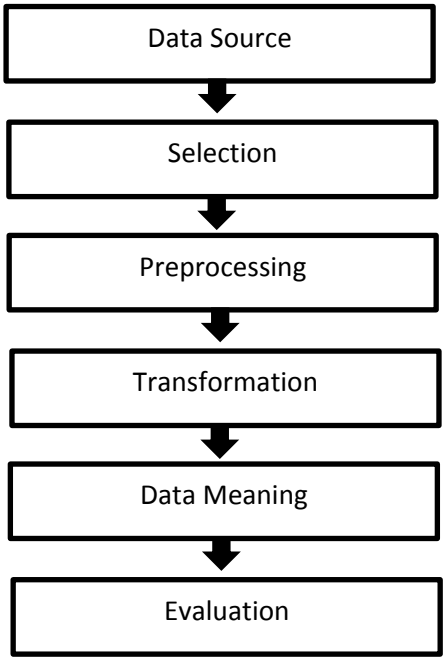
BAB II
LANDASAN TEORI

2.1 *Knowledge Discovery in Database*

Knowledge Discovery in Databases (KDD) adalah suatu proses sistematis untuk menemukan pengetahuan yang berguna dari kumpulan data yang besar. KDD mencakup serangkaian tahapan yang saling terkait, yamı: pemilihan data (*selection*), pra-pemrosesan (*preprocessing*), transformasi data, data mining, dan evaluasi atau interpretasi hasil.



Gambar 2. 1 Knowledge Discovery in Databases

Proses ini bertujuan untuk mengekstrak informası tersembunyi yang sebelumnya tidak diketahui namun memiliki nilai strategis bagi pengambilan keputusan.

Pada tahap data mining, algoritma akan diterapkan untuk mengelompokkan pelanggan berdasarkan pola pembelian mereka. Hasil dari pengelompokan ini kemudian dianalisis dan diinterpretasikan untuk membantu pemilik toko dalam memahami karakteristik setiap klaster pelanggan, sehingga dapat digunakan sebagai dasar dalam pongambilan keputusan bisnis yang lebik efektif dan efisien.

2.2 Data Mining

Data mining merupakan proses ekstraksi atas penggalian informasi yang bermanfaat dan tersembunyi dari sejumlah besar data. Data mining sering digunakan dalam berbagai bidang seperti bisnis, kesehatan, pendidikan, dan keuangan untuk mendukung pengambilan keputusan yang lebih cerdas. Beberapa metode populer dalam data mining meliputi klasifikasi, asosiasi, regresi, dan clustering. Clustering sendiri merupakan teknik untuk mengelompokkan data ke dalam kelompok-kelompok (klaster) berdasarkan kemiripan karakteristik, tanpa adanya label atau kategori yang sudah ditentukan sebelumnya (*unsupervised learning*)

Dalam konteks penelitian ini, data mining diterapkan untuk menganalisis data penelitian ini dengan menggunakan metode naïve bayes, salah satu teknik pengelompokan data yang paling umam digunakan. Metode ini memungkinkan peneliti untuk membagi data pelanggan berdasarkan pola pembelian mereka ke dalam beberapa klaster, sehingga dapat diperoleh gambaran yang lebih jelas mengenai karakteristik masing-masing kelompok pelanggan. Dengan pendekatan ini, dapat mengoptimalkan strategi bisnisnya melalui pemahaman yang lebih baik terhadap preferensi pelanggan, pengelolaan stok barang yang lebih efisien, serta

penetapan strategi promosi yang pat sasaran. Penerapan data mining diharapkan menjadi solusi praktis dan efesien dalam memanfaatkan.

2.2 Algoritma Naive Bayes

Algoritma *Naive Bayes* adalah metode klasifikasi berbasis probabilitas yang menggunakan Teorema Bayes dengan asumsi independensi antar variabel[5]. Dalam konteks prediksi kelulusan mahasiswa, algoritma ini dapat mengklasifikasikan data akademik, seperti nilai mata kuliah, Indeks Prestasi Kumulatif (IPK), dan faktor demografis, untuk menentukan probabilitas seorang mahasiswa lulus tepat waktu.

Naive Bayes mengacu pada pendekatan yang "*naive*" (sederhana), karena mengasumsikan bahwa semua fitur dalam dataset saling independen atau tidak saling bergantung, meskipun dalam kenyataannya fitur-fitur tersebut mungkin saja saling berhubungan.

Algoritma *Naïve Bayes* juga sebagai metode klasifikasi berbasis probabilitas yang menggunakan Teorema Bayes dengan asumsi bahwa setiap fitur dalam dataset bersifat independen satu sama lain (*independent features assumption*). Algoritma ini sering digunakan dalam berbagai bidang, termasuk klasifikasi teks, deteksi spam, analisis sentimen, dan dalam penelitian ini, untuk prediksi kelulusan mahasiswa. *Naïve Bayes* bekerja dengan menghitung probabilitas suatu kelas tertentu berdasarkan distribusi fitur dalam dataset pelatihan.

2.3.1 Teorema Bayes

Dasar dari algoritma ini adalah *Teorema Bayes*, yang didefinisikan sebagai berikut:

$$P(X \mid Y) = \frac{P(X \mid C)P(C)}{P(X)}$$

Dimana:

P(C/ X) merupakan Probabilitas suatu kelas *C* diberikan data fitur *X* (*Posterior Probability*).

P(C/X) merupakan Probabilitas fitur *X* muncul dalam kelas *C* (*Likelihood*).

P(C) merupakan Probabilitas awal dari kelas *C* (*Prior Probability*).

P(X) merupakan Probabilitas keseluruhan dari fitur *X* dalam dataset (*Evidence*).

Karena *P(X)* adalah konstanta dalam semua kelas, maka persamaan ini sering disederhanakan menjadi ;

$$P(C/X) \propto P(X/C) \cdot P(C)$$

Dimana kita hanya perlu menghitung *Likelihood* dan *Prior* untuk menentukan kelas dari suatu data baru.

Naive Bayes sering digunakan karena kemudahannya dalam implementasi dan performanya yang baik meskipun dataset memiliki jumlah data pelatihan yang terbatas[6].

2.3.2 Asumsi "Naïve" (Independensi Fitur)

Algoritma ini disebut "*Naive*" karena membuat asumsi yang sangat kuat bahwa semua fitur (atribut) dalam data adalah independen satu sama lain[7]. Artinya, keberadaan suatu fitur tidak memengaruhi keberadaan fitur lainnya. Meskipun asumsi ini seringkali tidak sepenuhnya benar dalam dunia nyata, algoritma ini tetap memberikan hasil yang baik dalam banyak kasus[8].

Algoritma *Naive Bayes* disebut "*naive*" (sederhana atau naif) karena didasarkan pada asumsi independensi fitur, yaitu menganggap bahwa semua fitur atau variabel dalam dataset tidak saling bergantung satu sama lain, meskipun dalam kenyataannya sering kali ada hubungan atau keterkaitan antar fitur.

Naïve Bayes disebut "*Naïve*" karena membuat asumsi bahwa semua fitur dalam dataset tidak saling bergantung atau independen. Sebagai contoh, dalam prediksi kelulusan mahasiswa, kita memiliki fitur: IPK, Jumlah SKS yang diambil, Tingkat Kehadiran, dan Status Sosial Ekonomi

Algoritma ini mengasumsikan bahwa IPK tidak bergantung pada jumlah SKS atau kehadiran, padahal dalam kenyataannya bisa ada korelasi. Meskipun asumsi ini sederhana, algoritma ini tetap sering memberikan hasil yang cukup akurat.

2.3.3 Jenis-Jenis Naïve Bayes

Naïve Bayes memiliki beberapa varian tergantung pada cara menghitung *Likelihood* dari fitur:

1) *Gaussian* Digunakan ketika fitur bersifat *Naïve Bayes* (GNB)
    a. Digunakan ketika fitur bersifat numerik dan kontinu.
    b. Mengasumsikan bahwa fitur mengikuti distribusi *Gaussian* (Normal).
    c. *Likelihood* dihitung dengan distribusi normal.

Dimana:

xi adalah nilai fitur.

μC adalah rata-rata fitur dalam kelas *C*.

$\sigma C2$ adalah variansi fitur dalam kelas *C*.

2) *Multinomial Naïve Bayes* (MNB)

a. Cocok untuk klasifikasi teks dan kategori diskrit.
b. Menggunakan frekuensi fitur dalam setiap kelas untuk menentukan probabilitas.

3) *Bernoulli Naïve Bayes* (BNB)

a. Digunakan untuk data biner (0/1).
b. Cocok untuk klasifikasi teks dalam bentuk ada/tidaknya kata tertentu dalam dokumen.

Misalnya, kita ingin membuat model klasifikasi untuk memprediksi apakah suatu email adalah spam atau bukan[9]. Fitur-fitur yang kita pertimbangkan adalah adanya kata "gratis", adanya tanda seru, dan panjang email.

2.3.4 Langkah-Langkah Implementasi Algoritma Naïve Bayes

1) Menghitung *Prior Probability (P(C))*

a. Probabilitas awal dari masing-masing kelas dalam dataset dihitung sebagai:

$$P(C) = \frac{Jumlah\ data\ kelas\ C}{Total\ jumlah\ data}$$

b. Misalnya, dalam data mahasiswa:

Lulus adalah 70 mahasiswa dari total 100.

Tidak Lulus adalah 30 mahasiswa dari total 100.

Maka:

$$P(Lulus) = \frac{70}{100} = 0{,}7 \quad P(Tidak\ Lulus) = \frac{30}{100} = 0{,}3$$

2) Menghitung *Likelihood Probability (P(X/C))*

a. Probabilitas setiap fitur berdasarkan kelas dihitung dari dataset pelatihan.

b. Contoh: Jika rata-rata IPK mahasiswa yang lulus adalah 3.5 dan memiliki distribusi normal, maka likelihood dihitung dengan *Gaussian Naïve Bayes* formula.

3) Menghitung *Posterior Probability (P(C/X))*

Untuk setiap kelas, hitung probabilitas gabungan fitur:

$$P(C/X) \propto P(C) \times P(X1/C) \times P(X2/C) \times \dots$$

Kelas dengan probabilitas tertinggi menjadi hasil klasifikasi.

4) Memprediksi Kelas

a. Bandingkan hasil perhitungan untuk semua kelas.

b. Pilih kelas dengan probabilitas tertinggi sebagai prediksi[10].

2.3.5 Penerapan Naïve Bayes dalam Prediksi Kelulusan Mahasiswa

Misalkan kita memiliki data mahasiswa dengan fitur berikut:

Tabel 2. 1 Prediksi Kelulusan Mahasiswa

Mahasiswa	IPK	Kehadiran (%)	Lulus (Y/N)
A	3.8	90%	Ya
B	2.7	60%	Tidak
C	3.5	85%	Ya
D	2.0	40%	Tidak
E	3.2	75%	Ya

Jika ada mahasiswa baru dengan IPK = 3.0 dan Kehadiran = 80%**,** kita dapat menghitung probabilitas lulus/tidak lulus menggunakan *Naïve Bayes* dan memilih hasil dengan probabilitas lebih tinggi. Penggunaan Algoritma Naive Bayes banyak digunakan dalam berbagai aplikasi, seperti[11]:

a. Klasifikasi teks seperti *Spam filtering, sentiment analysis, topik modeling*.

b. Klasifikasi multi-kelas seperti Klasifikasi dokumen, pengenalan gambar.

c. Sistem rekomendasi seperti Menyaring item yang relevan untuk pengguna.

Naive Bayes sangat populer dalam pemrosesan teks karena dapat menangani data berdimensi tinggi dengan baik. Mengklasifikasikan email sebagai spam atau tidak berdasarkan kata-kata yang terkandung dalam email. Menentukan apakah suatu teks (misalnya ulasan produk) memiliki sentimen positif, negatif, atau netral. Mengelompokkan artikel berita ke dalam kategori seperti olahraga, politik, teknologi, dll. Algoritma ini digunakan dalam sistem rekomendasi berbasis konten untuk memprediksi preferensi pengguna. Rekomendasi buku, film, atau produk berdasarkan ulasan dan preferensi sebelumnya.

Naïve Bayes adalah algoritma yang kuat untuk klasifikasi berbasis probabilitas, cocok digunakan dalam prediksi kelulusan mahasiswa karena kemampuannya dalam mengolah data kategori dan numerik secara efisien. Namun, asumsi independensi fitur menjadi keterbatasan yang perlu diperhitungkan. Dalam penelitian ini, *Naïve Bayes* akan digunakan bersama untuk meningkatkan akurasi prediksi kelulusan mahasiswa berbasis kompetensi akademik.

2.3 Evaluasi Model Prediksi

Untuk mengukur seberapa baik model yang dihasilkan, digunakan beberapa metrik evaluasi:

2.6.1 *Confusion Matrix*

Membandingkan hasil prediksi dengan data aktual untuk menghitung akurasi, *precision, recall, dan F1-score.*

Tabel 2. 2 Perbandingan Hasil Prediksi

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

$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN}$$

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

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

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

2.6.2 Evaluasi K-Means dengan Rapidminer

*RapidMine*r adalah aplikasi berbasis GUI yang dirancang untuk analisis data, mulai dari pra-pemrosesan hingga penerapan algoritma *machine learning*. Dengan fitur *drag-and-drop*, pengguna dapat membuat model analitik tanpa pemrograman. *RapidMiner* mendukung berbagai teknik, termasuk klasifikasi, *clustering*, dan prediksi, sehingga cocok untuk aplikasi seperti analisis kelulusan mahasiswa. Kelebihan utama *RapidMine*r adalah kemampuannya dalam mengintegrasikan visualisasi hasil analisis, mempermudah interpretasi, dan memungkinkan eksplorasi pola data secara langsung.

2.4 Penelitian Terdahulu

Berikut adalah beberapa penelitian terdahulu dalam 5 tahun terakhir yang relevan dengan implementasi algoritma *Naive Bayes* dan *K-Means* untuk prediksi kelulusan mahasiswa berbasis kompetensi akademik:

Tabel 2. 4 Penelitian Terdahulu

Referensi Penelitian	1
Judul	Implementasi Algoritma Naïve Bayes untuk Memprediksi Kelulusan Mahasiswa Berdasarkan Data Akademik dan Status Sosial Ekonomi dengan Optimalisasi Model
Nama	Abdullah
Tahun	2022
Hasil	Menggunakan algoritma Naive Bayes untuk memprediksi kelulusan mahasiswa berdasarkan data akademik seperti indeks prestasi (IP), nilai ujian nasional (UN), dan status sosial ekonomi, dengan

	akurasi mencapai 89% setelah optimalisasi model
Referensi Penelitian	2
Judul	Kombinasi Algoritma Naïve Bayes dan C4.5 untuk Prediksi Kelulusan Mahasiswa di STIMK Bina Nusantara Jaya
Nama	Etriyanti
Tahun	2020
Hasil	Mengombinasikan algoritma Naive Bayes dan C4.5 untuk prediksi kelulusan mahasiswa STIMK Bina Nusantara Jaya. Pendekatan ini bertujuan mengidentifikasi faktor utama yang memengaruhi kelulusan tepat waktu
Referensi Penelitian	3
Judul	Penerapan Algoritma K-Means dan C4.5 untuk Memprediksi Kesiapan Kerja Mahasiswa Berdasarkan Data Akademik dan Non-Akademi
Nama	Noviyanto
Tahun	2020
Hasil	Menggunakan algoritma K-Means dan C4.5 untuk memprediksi kesiapan kerja mahasiswa berdasarkan data akademik dan non-akademik. Pendekatan ini memberikan insight penting untuk meningkatkan kualitas lulusan
Referensi Penelitian	4
Judul	Pengembangan Model Prediksi Kelulusan Menggunakan Naïve Bayes Classifier dengan Integrasi Data Demografis dan Akademik Mahasiswa
Nama	Yustira
Tahun	2020
Hasil	Mengembangkan model prediksi kelulusan menggunakan Naive Bayes Classifier, dengan fokus pada integrasi data demografis dan akademik mahasiswa
Referensi Penelitian	5
Judul	Analisis Kompetensi Alumni Berdasarkan Masa Tunggu Kerja Menggunakan Algoritma C4.5 yang Relevan dengan Prediksi Kelulusan Berbasis Faktor Akademik
Nama	Cahyaningtyas
Tahun	2020
Hasil	Meneliti kompetensi alumni berdasarkan masa tunggu kerja menggunakan algoritma C4.5, yang relevan dengan prediksi kelulusan berbasis faktor

	akademik
Referensi Penelitian	10
Judul	Prediksi Performa Pascasarjana Menggunakan Jaringan Saraf Tiruan dengan Variabel Demografi dan Akademik yang Dapat Diintegrasikan dengan Algoritma Naïve Bayes atau K-Mean
Nama	Pal & Bhatt
Tahun	2019
Hasil	Menggunakan jaringan saraf tiruan (Artificial Neural Network) yang melibatkan variabel demografi dan akademik untuk memprediksi performa pascasarjana, yang dapat diintegrasikan dengan algoritma Naive Bayes atau K-Means