# BAB 2

# TINJAUAN PUSTAKA DAN DASAR TEORI

## 2.1 Tinjauan Pustaka

Penelitian ini membahas tentang bagaimana hasil prediksi kelulusan mahasiswa dengan metode klasifikasi *Naive Bayes Classifier*. Berikut penelitian terdahulu yang pernah dilakukan dengan menerapkan metode klasifikasi *Naive Bayes Classifier* :

Penelitian yang dilakukan oleh Kemal Refta Diska dan Khairi Budayawan (Diska & Budayawan, 2023), dengan judul "Sistem Informasi Prediksi Kelulusan Menggunakan Metode *Naive Bayes* (Studi Kasus: Prodi Pendidikan Teknik Informatika)". Data diuji menggunakan data mahasiswa PTI angkatan 2014 dan 2015 sebagai *data training* berjumlah 94 data, dan data mahasiswa angkatan 2016 sebagai *data testing* berjumlah 46 data. Atribut yang digunakan dalam penelitian ini adalah IPS semester 1 hingga 6, dan total sks. Hasil dari penelitian yang dilakukan menyatakan bahwa dari 46 *data testing* memperoleh hasil *accuracy* 82,61%, *precision* 91,66%, dan *recall* 61,11%.

Penelitian yang dilakukan oleh Firman Azhar Riyadi dan Donny Avianto (Riyadi & Avianto, 2020), dengan judul "Implementasi Metode *Naive Bayes* Untuk Prediksi Kelulusan Mahasiswa Tepat Waktu Prodi Informatika (Studi Kasus Universitas Teknologi Yogyakarta)". Penelitian ini menggunakan data mahasiswa angkatan 2014 sebanyak 200 data dengan atribut kelamin, sks1, sks2, sks3, sks4, ipk1, ipk2, ipk3, dan ipk4. Penelitian yang dilakukan yaitu dengan uji coba

perbandingan data uji sebanyak 60% dan data latih sebanyak 40% maka didapatkan akurasi 91,86%.

Penelitian yang dilakukan oleh Nurul Khasanah, dkk (Khasanah et al., 2022), dengan judul "Prediksi Kelulusan Mahasiswa Dengan Metode *Naive Bayes*". Penelitian ini menggunakan sebanyak 379 data, dengan metode *Naive bayes*, dengan rincian *data training* 303 data dan *data testing* 76 data. Atribut yang digunakan nama, status mahasiswa, status perkawinan, IPS, IPK, dan status kelulusan. Dengan tahapan identifikasi masalah, pengumpulan data, *data cleaning*, data *transformation* (dibagi menjadi *data training* dan *data tesing*), klasifikasi dengan NBC, validasi, evaluasi dan hasil. Hasil penelitian yang diperoleh yaitu akurasi = 88,16%, *precision* = 93,62% dan *recall* = 88%.

Penelitian yang dilakukan oleh Supardi Salmu dan Achmad Solichin (Salmu & Solichin, 2017), dengan judul "Prediksi Tingkat Kelulusan Mahasiswa Tepat Waktu Menggunakan *Naive Bayes* (Studi Kasus : UIN Syarif Hidayatullah Jakarta)". Atribut yang digunakan pada penelitian ini yaitu meliputi jenis kelamin, jenis seleksi, pendapatan ayah, pendidikan ibu, IP semester 1 sampai 4, dan SKS semester 1 sampai 4. Penelitian ini menunjukkan hasil akurasi *Naive Bayes* sebesar 80,72%, presisi 28,13%, *recall* 28,57%, dan *F1-Score* 32,53%. Dengan rincian *data training* sebanyak 1662 data dan *data testing* sebanyak 587 data.

Penelitian yang dilakukan oleh M. Riski Qisthiano, dkk (Qisthiano et al., 2021), dengan judul "Pengembangan Model Untuk Prediksi Tingkat Kelulusan Mahasiswa Tepat Waktu Dengan Metode *Naive Bayes*". Atribut yang digunakan dalam penelitian ini adalah Jurusan, Perguruan Tinggi, Jenis Kelas, Nilai IP

Sementara dari semester 1 hingga 4, tahun lulus, dan angkatan kuliah. Selanjutnya pembagian dataset dibagi 70% untuk *data training* dan 30% sebagai *data testing*. Penelitian ini menguji proses algoritma *Naive Bayes* menggunakan K-Fold Validation. Hasil penelitian ini merupakan akurasi dari model prediksi yang dilakukan, dimana hasil akurasi yang didapatkan adalah 0.8103 atau sebesar 81,03%.

Penelitian yang dilakukan oleh Neni Purwati dan Agnes Dwi Januanti (Purwati & Januanti, 2021), dengan judul "Prediksi Tingkat Kelulusan Mahasiswa Dengan Algoritma *Naive Bayes*". Atribut yang digunakan dalam penelitian ini yaitu meliputi jurusan, biaya, jenis kelamin, beasiswa, jumlah mata kuliah yang disetujui semester pertama, nilai rata-rata semester pertama, jumlah mata kuliah yang disetujui semester 2, dan nilai rata-rata semester 2. Menggunakan dataset sebanyak 500 data, kemudian dibagi menjadi 80% *data training* dan 20% *data testing*. Penelitian yang dilakukan mendapatkan hasil akurasi sebesar 95%, presisi 95,16%, *recall* 95%, dan *F1-Score* 95%.

Penelitian yang dilakukan oleh Muhammad Gunawan, Muhammad Zarlis, dan Roslina (Gunawan et al., 2021), dengan judul "Analisis Komparasi Algoritma *Naive Bayes* dan *k-Nearest Neighbor* Untuk Memprediksi Kelulusan Mahasiswa Tepat Waktu". Data yang digunakan dalam penelitian ini adalah Data Alumni Mahasiswa Fakultas Kedokteran Universitas Muhammadiyah Sumatera Utara Tahun Angkatan 2015. Data mahasiswa yang digunakan adalah 100 data mahasiswa. Data penelitian yang diambil memiliki atribut *input* indeks prestasi semester (IPS) 1 hingga 5 dan atribut output ialah Kelulusan. Penelitian ini

menghasilkan kesimpulan bahwa algoritma *Naive Bayes* memiliki tingkat akurasi yang sama dengan algoritma KNN dalam memprediksi kelulusan mahasiswa program studi Pendidikan Kedokteran yaitu sebesar 90 %. Hasil ini diperoleh dari analisis yang dilakukan dengan aplikasi Weka terhadap 30 *data testing* dan 100 *data training*.

Penelitian yang dilakukan oleh Sidik Rahmatullah dan Ema Utami (Rahmatullah, 2019), dengan judul "Prediksi Tingkat Kelulusan Tepat Waktu Dengan Metode *Naive Bayes* Dan *K-Nearest Neighbor*". Atribut-atribut data yang digunakan adalah NPM, jenis kelamin, IPS 1-5, konsentrasi, asal tinggal, jumlah sks, status pekerjaan, dan status kelulusan. Uji coba dilakukan dengan menggunakan data lulusan mahasiswa S1 Sistem informasi STMIK Dian Cipta Cendikia Kotabumi sebanyak 600 data untuk *training* dan 180 data untuk *testing*. Hasil uji coba menunjukkan bahwa dengan menggunakan *Naive Bayes* menghasilkan akurasi sebesar 85%, sedangkan menggunakan algoritma *K-nearest neighbor* menghasilkan akurasi sebesar 68,89 %.

## 2.2 Dasar Teori

### 2.2.1 *Data Mining*

*Data mining* merupakan proses untuk mendapatkan informasi yang berguna dari gudang basis data yang besar (Qisthiano et al., 2021). *Data mining* sering disebut dengan *Knowledge Discovery in Database* atau disingkat dengan KDD, yaitu merupakan proses pengumpulan, analisis, dan penggunaan data historis untuk menelusuri data yang ada guna membangun model yang dapat mengenali pola data lain yang berukuran besar. *Data mining* adalah proses yang menggunakan teknik

statistik, matematika, kecerdasan buatan dan *machine learning* untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai database besar (Widaningsih, 2019).

Secara garis besar, *data mining* adalah proses pencarian dan analisis pada suatu kumpulan data (*database*) untuk menemukan pola yang menarik dengan tujuan mengekstrak informasi dan pengetahuan yang akurat serta potensial, sehingga dapat dipahami dan berguna dalam pengambilan keputusan (Setiyani et al., 2020). Menurut (Mustafa et al., 2018) tahap – tahap *data mining* adalah sebagai berikut :

1. Pembersihan data (*data cleaning*)

   Pembersihan data merupakan proses menghilangkan *noise* dan data yang tidak konsisten atau data yang tidak relevan.

2. Integrasi data (*data integration*)

   Integrasi data merupakan penggabungan data dari berbagai *database* ke dalam satu *database* baru.

3. Seleksi data (*data selection*)

   Seleksi data merupakan proses penyeleksian data. Data yang ada pada *database* sering kali tidak semuanya dipakai, oleh karena itu hanya data yang sesuai untuk dianalisis yang akan diambil dari *database*.

4. Transformasi data (*data transformation*)

   Data diubah atau digabung ke dalam format yang sesuai untuk diproses dalam *data mining*.

5. Proses *mining*

Merupakan suatu proses utama saat metode diterapkan untuk menemukan pengetahuan berharga dan tersembunyi dari data.

6. Evaluasi pola (*pattren evaluation*)

   Untuk mengidentifikasi pola – pola menarik ke dalam *knowledge based* yang ditemukan.

7. Presentasi pengetahuan (*knowledge presentation*)

   Merupakan visualisasi dan penyajian pengetahuan mengenai metode yang digunakan untuk memperoleh pengetahuan yang diperoleh pengguna.

### 2.2.2 Praproses Data

Praproses data melibatkan beberapa langkah penting untuk memastikan data yang digunakan dalam pengembangan model prediksi adalah data yang bersih, lengkap, dan sesuai. Praproses merupakan sebuah tahap awal yang harus dilakukan pada *data mining* untuk mempersiapkan data mentah sebelum dilakukan proses lain. Praproses data dilakukan dengan cara mengeliminasi data yang tidak sesuai atau mengubah data menjadi bentuk yang lebih mudah diproses oleh sistem. Tujuan praproses adalah untuk mendapatkan hasil yang lebih akurat, pengurangan waktu perhitungan untuk *large scale problem*, dan membuat nilai data menjadi lebih kecil tanpa merubah informasi didalamnya (Nasution et al., 2019).

### 2.2.3 Pembersihan Data

Pembersihan data (*Data Cleaning*) adalah tahap awal dalam proses *Knowledge Discovery in Databases* (KDD) yang melibatkan penghapusan atau perbaikan data yang tidak lengkap, salah, atau tidak konsisten. *Data cleaning* adalah

proses mendeteksi, memperbaiki atau bahkan menghapus catatan, tabel, dan database yang salah atau tidak akurat (Baiq Nurul Azmi et al., 2023). Pada tahap ini, akan dilakukan proses penghapusan data yang tidak relevan, duplikat, atau yang memiliki banyak nilai kosong. Atribut-atribut yang nilainya kosong seluruhnya atau sebagian besar kosong akan dihilangkan untuk mendapatkan atribut yang relevan dan valid. Tujuannya adalah untuk memastikan bahwa data yang digunakan adalah data yang valid, dapat diandalkan, dan bebas dari *noise* seperti *missing value*, inkonsistensi, dan redundansi.

Proses penggantian *missing value* dilakukan dengan mengganti data yang hilang atau tidak sesuai dengan atribut yang ada. Menurut Amien dkk, *missing value* dapat diganti dengan nilai *median* atau *mean* dari masing-masing atribut, dengan tujuan untuk mengoptimalkan hasil yang didapatkan (Amien et al., 2023). Sedangkan inkonsistensi data terjadi ketika ada data yang redundan. Data redundan adalah menumpuknya data-data yang sama yang tidak dibutuhkan di dalam database. Salah satu cara untuk menyelesaikan masalah inkonsistensi data adalah dengan melakukan eliminasi atau menghapus beberapa data yang tidak konsisten pada database (Sanjaya & Sulistyo, 2015).

### 2.2.4 Pembagian Data

Pembagian data adalah proses membagi dataset menjadi dua bagian, yaitu data latih (*training data*) dan data uji (*testing data*). Data latih digunakan untuk melatih model, sedangkan data uji digunakan untuk mengevaluasi performa model. Pada penelitian ini akan dilakukan pembagian data dengan proporsi 80% untuk data latih dan 20% untuk data uji. Pembagian data ini bertujuan untuk memastikan

bahwa model dapat memprediksi dengan akurat pada data baru atau data yang belum pernah dilihat sebelumnya. Pembagian jumlah data latih dan data uji adalah salah satu faktor yang menentukan akurasi, sehingga kesalahan dalam menentukan komposisi kedua tipe data tersebut akan mempengaruhi nilai akurasi dan presisi yang diperoleh (Baiq Nurul Azmi et al., 2023).

### 2.2.5 Klasifikasi

Klasifikasi merupakan proses penemuan model atau fungsi yang menggambarkan dan membedakan kelas data atau konsep yang bertujuan agar bisa digunakan untuk memprediksi kelas dari objek yang label kelasnya tidak diketahui (Haditsah, 2018). Klasifikasi banyak digunakan untuk memprediksi kelas pada label tertentu, yaitu dengan mengklasifikasikan data atau membangun model berdasarkan *training set* dan nilai – nilai atau label kelas yang digunakan dalam mengklasifikasikan atribut tertentu (Tangkelayuk, 2022).

Dalam proses klasifikasi, terjadi identifikasi kelompok dari suatu obyek berdasarkan kesamaan fitur tertentu, dimana setiap kelompok telah terbentuk melalui suatu proses tertentu. Proses klasifikasi biasanya dibagi menjadi dua fase yaitu fase *learning* dan fase *test*. Pada fase *learning* sebagian data yang telah diketahui kelas datanya diumpankan untuk membentuk model perkiraan. Kemudian pada fase *test* model yang sudah dibentuk diuji dengan sebagian data lainnya untuk mengetahui akurasi model. Teknik ini dapat memberikan klasifikasi pada data baru dengan memanipulasi data yang ada yang telah diklasifikasi.

### 2.2.6 *Naive Bayes*

*Naive Bayes Classifier* yaitu salah satu metode *machine learning* yang menggunakan perhitungan probabilitas dan statistik yang dikemukakan oleh ilmuan Inggris Thomas Bayes, yaitu memperkirakan probabilitas di masa depan berdasarkan pengalaman di masa sebelumnya sehingga dikenal sebagai Teorema Bayes (Gunawan et al., 2021). Teorema tersebut dikombinasikan dengan *naive* yang mana diasumsikan keadaan antar atribut saling bebas. Jadi didalam penerapannya, algoritma *Naive Bayes Classifier* tidak ada hubungan antara satu atribut dengan atribut yang lain, atau dengan kata lain satu atribut tidak berpengaruh dengan atribut yang lain, sekalipun mungkin atribut tersebut saling berhubungan (Sigid Widodo et al., 2023).

Teorema bayes merupakan dasar dari *naive bayes classifier*. *Naive Bayes Classifier* berfungsi menghitung dan mencari nilai probabilitas paling tinggi untuk mengklasifikasikan sebuah data uji dengan kategori yang tepat. Teknik prediksi probabilitas yang sederhana didasarkan pada penerapan teorema bayes yang secara umum dinyatakan sebagai berikut :

$$P(H|X) = \frac{P(X|H).P(H)}{P(X)} \quad \textbf{(2. 1)}$$

Keterangan :

| | | |
|---|---|---|
| $X$ | = | Data dengan kelas yang belum diketahui |
| $H$ | = | Hipotesis data X merupakan suatu label kelas tertentu |
| $P(H\|X)$ | = | Probabilitas hipotesis H berdasarkan kondisi X (Probabilitas Posterior) |

$P(X|H)$ = Probabilitas X berdasarkan kondisi pada hipotesis H (Probabilitas *Likelihood*)

$P(H)$ = Probabilitas hipotesis H (Probabilitas Prior)

$P(X)$ = Probabilitas X (Probabilitas *Evidence*)

Berikut tahapan dari algoritma *Naive Bayes Classifier* :

1. Mengitung probabilitas prior (P(H)), yaitu menentukan probabilitas prior dari setiap kelas H.
2. Menghitung probabilitas *likelihood* (P(X|H)), yaitu menghitung probabilitas bahwa data X muncul dalam setiap kelas H.
3. Menghitung probabilitas *evidence* (P(X)), yaitu menghitung probabilitas dari data yang diamati secara keseluruhan, tanpa memperhatikan kelasnya.
4. Menghitung probabilitas posterior (P(H|X)), yaitu dengan cara menggunakan rumus diatas untuk menghitung probabilitas bahwa hipotesis H terjadi berdasarkan data X yang diamati.
5. Prediksi kelas, yaitu menentukan kelas dengan probabilitas posterior tertinggi.

### 2.2.7 *Confusion Matrix*

*Confusion matrix* merupakan alat yang sangat penting dalam evaluasi kinerja model klasifikasi dalam *machine learning*. Matriks ini menyajikan jumlah prediksi yang benar dan salah yang dibuat oleh model dibandingkan dengan hasil sebenarnya. *Confusion matrix* memberikan gambaran yang lebih detail tentang kesalahan klasifikasi yang dilakukan oleh model. *Confusion matrix* merupakan matriks yang menampilkan prediksi klasifikasi dan klasifikasi yang aktual, serta

digunakan untuk memperoleh nilai *precision*, *recall*, dan *accuracy* (Rahayu et al., 2021). Berikut *confusion matrix* dengan dua label kelas berbentuk tabel 2x2 yang memberikan informasi tentang *true positive*, *false positive*, *true negative*, dan *false negative*.

**Tabel 2. 1.** Struktur *Confusion Matrix* 2x2

| **Prediksi** / **Aktual** | **Positif** | **Negatif** |
|---|---|---|
| **Positif** | TP | FN |
| **Negatif** | FP | TN |

Dimana :

- *True Positive* (TP) : Jumlah prediksi positif yang benar (model memprediksi positif dan sebenarnya positif).
- *True Negative* (TN) : Jumlah prediksi negatif yang benar (model memprediksi negatif dan sebenarnya negatif).
- *False Positive* (FP) : Jumlah prediksi positif yang salah (model memprediksi positif tetapi sebenarnya negatif).
- *False Negative* (FN) : Jumlah prediksi negatif yang salah (model memprediksi negatif tetapi sebenarnya positif).

Berikut adalah rumus-rumus untuk mengukur performa model klasifikasi menggunakan metrik seperti akurasi, presisi, *recall*, dan *F1-score* :

1. Akurasi (*accuracy*), yaitu untuk mengukur seberapa sering model klasifikasi membuat prediksi yang benar dari semua prediksi yang dibuat.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad \textbf{(2. 2)}$$

2. Presisi (*precision*), yaitu untuk mengukur seberapa banyak prediksi positif yang benar dari semua prediksi positif yang dibuat oleh model.

$$Precision = \frac{TP}{TP + FP} \quad \textbf{(2. 3)}$$

3. *Recall*, yaitu untuk mengukur seberapa banyak kasus yang sebenarnya positif yang benar diprediksi oleh model.

$$Recall = \frac{TP}{TP + FN} \quad \textbf{(2. 4)}$$

4. *F1-Score*, yaitu kombinasi dari presisi dan *recall* untuk memberikan keseimbangan antara kedua metrik.

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad \textbf{(2. 5)}$$

Jika penelitian ini menggunakan 4 label kelas, maka *confusion matrix* akan menjadi matriks 4x4. Setiap baris mewakili jumlah *instance* kelas sebenarnya, dan setiap kolom mewakili jumlah *instance* kelas yang diprediksi. Bentuk tabel *confusion matrix* 4x4 adalah sebagai berikut:

**Tabel 2. 2.** Struktur *Confusion Matrix* 4x4

| **Prediksi / Aktual** | **A** | **B** | **C** | **D** |
|---|---|---|---|---|
| **A** | TP | FN | FN | FN |
| **B** | FP | TP | FN | FN |
| **C** | FP | FP | TP | FN |
| **D** | FP | FP | FP | TP |

### 2.2.8 *Use Case Diagram*

*Use case diagram* atau diagram *use case* adalah diagram untuk memodelkan perilaku suatu sistem yang akan dirancang dengan menggambarkan interaksi antara satu atau lebih aktor yang akan menggunakan sistem. *Use case diagram* terdiri dari sebuah aktor dan interaksi yang dilakukannya, aktor tersebut dapat berupa manusia, perangkat keras, sistem lain, ataupun yang berinteraksi dengan sistem (Kurniawan & Syarifuddin, 2020). Antara *use case* dengan aktor atau dengan *use case* terdapat beberapa *links* hubungan *include*, *extend*, *generalization*, dan lain-lain (Setiyani, 2021). *Use Case Diagram* adalah salah satu diagram yang digunakan dalam perancangan sistem untuk menggambarkan hubungan interaksi antara pengguna (aktor) dengan aplikasi yang sedang dirancang (sistem). Diagram ini dapat mendeskripsikan jenis-jenis interaksi yang terjadi antara pengguna dan sistem, serta mendokumentasikan kebutuhan fungsional yang harus dipenuhi oleh sistem. Dengan *use case diagram*, maka dapat memvisualisasikan dan memahami alur kerja aplikasi yang dibuat dari sudut pandang pengguna.

### 2.2.9 *Sequence Diagram*

Diagram sekuen (*Sequence Diagram*) mendeskripsikan perilaku objek pada *use case* dengan menjelaskan alur waktu hidup dari objek dan pesan atau *message* yang diterima dan dikirim antar objek. *Sequence Diagram* adalah diagram yang digunakan untuk menggambarkan urutan interaksi antar objek dalam sistem berdasarkan urutan waktu. Diagram ini menunjukkan bagaimana objek-objek saling berinteraksi melalui pertukaran pesan (*message*) dalam suatu skenario. *Sequence diagram* memiliki dimensi vertikal yang mewakili urutan waktu, dan dimensi

horizontal yang mewakili objek-objek yang terlibat dalam interaksi tersebut. Dengan *sequence diagram*, maka dapat memvisualisasikan dan memahami alur proses bisnis serta kolaborasi antar komponen sistem secara lebih rinci.

### 2.2.10 *Website*

*Website* adalah kumpulan halaman web yang saling terkait dan dapat diakses melalui Internet. Halaman-halaman tersebut dapat berisi teks, gambar, video, dan elemen interaktif lainnya yang disajikan menggunakan bahasa markup seperti HTML (*HyperText Markup Language*). Struktur dasar sebuah *website* terdiri dari berbagai elemen, seperti halaman beranda (*homepage*), halaman-halaman konten, *navigasi* (menu), *footer*, dan lain-lain. Halaman-halaman ini saling terhubung dengan tautan atau *hyperlink*.

Menurut Tangkudung dkk, *website* merupakan kumpulan halaman web yang berhubungan antara satu dengan lainnya, halaman pertama sebuah *website* adalah *home page*, sedangkan halaman demi halamannya secara mandiri disebut *web page*, dengan kata lain *website* adalah situs yang dapat diakses dan dilihat oleh para pengguna internet di seluruh dunia (Tangkudung et al., 2019). Aplikasi *website* dibagi menjadi dua jenis, yaitu *website* statis dan *website* dinamis. *Website* statis merupakan *website* yang isinya jarang berubah atau tetap, serta memiliki sifat satu arah dan tidak interaktif. Sedangkan *website* dinamis merupakan *website* yang isinya sering berubah, serta memiliki sifat dua arah dan interaktif.