BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Tinjauan Pustaka bertujuan untuk memperluas pemahaman tentang konsep, metode, dan penelitian sebelumnya. Ini dilakukan dengan mengacu pada berbagai literatur sebagai landasan ilmiah yang kokoh untuk menganalisis dan menyelesaikan masalah utama dalam penelitian.

2.1.1 Definisi Data Mining

Data mining adalah proses eksplorasi dan penggalian informasi yang tersembunyi dalam data yang besar dan kompleks. Tujuannya adalah untuk menemukan pola, tren, dan hubungan yang berguna yang dapat membantu dalam pengambilan keputusan(Lasfeto, 2021).

Data mining merupakan suatu proses yang memanfaatkan teknik statistik, matematika, kecerdasan buatan, dan pembelajaran mesin untuk mengekstrak dan mengenali informasi yang berguna serta pengetahuan yang relevan dari berbagai basis data besar(Linawati et al., 2020).

Berdasarkan definisi-definisi di atas tentang *Data mining* dapat disimpulkan bahwa *data mining* adalah proses yang melibatkan serangkaian prosedur eksplorasi dan penggalian informasi tersembunyi dari data berukuran besar dan kompleks. Dengan memanfaatkan teknik-teknik seperti statistik, matematika, kecerdasan buatan, dan pembelajaran mesin, *data mining* memungkinkan identifikasi pola, tren, dan hubungan dalam data yang dapat memberikan wawasan penting untuk pengambilan keputusan. Melalui pendekatan ini, data mentah dapat diubah menjadi pengetahuan yang relevan dan bernilai bagi pengguna.

Dengan menerapkan *data mining* di SMK Hutama Bekasi, sekolah dapat mengolah dan menganalisis data siswa secara lebih efektif. Penerapan *data mining* di SMK Hutama Bekasi diharapkan dapat membantu sekolah dalam mengidentifikasi pola dan tren yang relevan dari data akademik, ekstrakurikuler, presensi, prestasi dan kepribadian siswa, yang sebelumnya mungkin tidak terdeteksi. Dengan demikian, penerapan *data mining* di SMK Hutama Bekasi memberikan dukungan dalam pengambilan keputusan yang lebih objektif, cepat, dan tepat sasaran, terutama terkait proses seleksi calon penerima beasiswa berprestasi.

2.1.1.1 Teknik Data Mining

Menurut (Ginantra, Ni Luh Wiwik Sri Rahayu et al., 2021) ada beberapa teknik dan sifat *data mining* adalah sebagai berikut:

1. *Clustering*. Teknik *clustering* atau pengelompokan adalah metode yang digunakan untuk membagi data menjadi beberapa kelompok berdasarkan kriteria tertentu. Tujuan dari teknik ini adalah untuk mengumpulkan data dengan kriteria yang serupa dalam satu kelompok yang sama, serta memisahkan data dengan kriteria yang berbeda ke kelompok yang lain.

   Berikut merupakan metode yang umum digunakan pada teknik *clustering* antara lain:

   A. *Partitioning method*, adalah teknik yang digunakan untuk mengelompokkan data berdasarkan jumlah cluster atau kelompok yang telah ditentukan.

   B. *Hierarchical method* adalah alternatif dari metode partitioning dan tidak memerlukan penentuan jumlah cluster atau kelompok di awal.

2. *Regression*. Teknik *regression* adalah metode dalam data mining yang digunakan untuk memahami hubungan antara dua atau lebih variabel dan memprediksi nilai dari variabel dependen dengan menggunakan nilai dari variabel-variabel prediktor atau independen.

   Terdapat dua buah metode teknik *regression* yaitu:

   A. Regresi Linear adalah teknik analisis yang digunakan untuk mengidentifikasi dan memahami hubungan antara dua variabel.
   B. Regresi Berganda adalah metode analisis yang digunakan untuk mengevaluasi hubungan antara satu variabel dependen dan lebih dari satu variabel independen.

3. *Classification.* Teknik *classification* adalah metode yang digunakan untuk mengklasifikasikan data. Teknik ini mirip dengan *clustering* yang juga membagi data ke dalam kelompok-kelompok. Namun, pada *classification,* kelompok-kelompok ini dikenal sebagai kelas data. Perbedaan utama dengan *clustering* terletak pada penggunaan *data training* yang telah memiliki label. Model *classification* dibangun menggunakan *data training* berlabel ini untuk mengklasifikasikan dan memprediksi kelas data baru yang belum memiliki label. Beberapa algoritma yang umum digunakan dalam metode *classification* antara lain adalah *Naive Bayes, Decision Tree, Logistic Regression, K-Nearest Neighbor, dan Support Vector Machine.*

4. *Association. Association rule* merupakan teknik yang biasanya berbentuk pernyataan jika/maka (if/then), yang berfungsi untuk mengidentifikasi hubungan antara data yang tampak tidak berhubungan dalam basis data

relasional atau tempat penyimpanan informasi lainnya. Beberapa algoritma yang sering digunakan dalam teknik *association rule* antara lain *Apriori, ECLAT, dan FP-Growth.*

2.1.2 Beasiswa

Beasiswa adalah bentuk pendanaan yang tidak berasal dari biaya pribadi atau orang tua, melainkan diberikan oleh pemerintah, perusahaan swasta, kedutaan besar, universitas, lembaga pendidikan atau penelitian, serta tempat kerja. Beasiswa ini dapat diberikan sebagai penghargaan atas prestasi, yang memungkinkan seseorang, seperti karyawan, untuk meningkatkan kualitas sumber daya manusia melalui Pendidikan(Apdian et al., 2024).

2.1.2.1 Beasiswa di SMK Hutama Bekasi

SMK Hutama Bekasi adalah sekolah menengah kejuruan yang berada di bawah naungan Yayasan Pendidikan 1988 (YP88), berlokasi di Kecamatan Pondok Melati, Kota Bekasi, Provinsi Jawa Barat. SMK Hutama Bekasi mengelola berbagai data siswa, termasuk data pribadi (profil) dan data hasil akademik selama proses pembelajaran. Di sekolah ini diterapkan program untuk siswa kelas 11 dan 12 yang berprestasi, di mana penilaian dilakukan berdasarkan nilai rapor sebelumnya.

Penilaian dalam program ini didasarkan pada beberapa kriteria sebagai berikut:

A. Nilai Akademik: Dipilih berdasarkan total nilai UTS dan UAS yang paling tinggi.
B. Presensi: Siswa tidak boleh memiliki alfa (ketidakhadiran tanpa keterangan sah).

C. Nilai Non-Akademik: Siswa harus memiliki nilai non-akademik minimal A dan B.

Siswa yang mendapatkan peringkat kelas dan memenuhi beberapa aspek penilaian lainnya berhak mendapatkan penghargaan berupa beasiswa atau pembebasan SPP selama 6 bulan. Selain aspek akademis (total nilai UTS dan UAS), penilaian juga mencakup aspek non-akademis seperti keaktifan dalam ekstrakurikuler (ekskul), presensi, prestasi dan kepribadian. Dengan demikian, siswa yang berprestasi dinilai tidak hanya berdasarkan nilai mata pelajaran saja, tetapi juga kemampuan lain yang dimiliki oleh siswa.

2.1.3 Prediksi

Prediksi merupakan proses sistematis dalam memperkirakan hal yang paling mungkin terjadi di masa depan, berdasarkan informasi yang tersedia dari masa lalu dan masa kini, dengan tujuan untuk meminimalkan kesalahan (selisih antara kejadian sebenarnya dan hasil perkiraan)(Adiguno et al., 2022).

2.1.4 Naïve Bayes Classifier

Naive Bayes Classifier adalah metode klasifikasi probabilistik yang sederhana yang menghitung probabilitas dengan menjumlahkan frekuensi dan kombinasi nilai dari dataset yang tersedia. Algoritma ini didasarkan pada teorema Bayes, yang memperhitungkan hubungan antara probabilitas bersyarat dan marginal. Salah satu asumsi utamanya adalah bahwa semua atribut dianggap independen atau tidak saling bergantung satu sama lain, meskipun kenyataannya dalam banyak kasus mungkin terdapat keterkaitan antara atribut-atribut tersebut. Karena kesederhanaan dan efisiensinya, *Naive Bayes* sering digunakan dalam

berbagai aplikasi seperti klasifikasi teks, diagnosis penyakit, dan analisis sentimen, meskipun asumsi independensi sempurna jarang terpenuhi dalam situasi nyata. Algoritma ini tetap efektif dalam memberikan hasil yang cukup akurat bahkan ketika asumsi independensinya tidak sepenuhnya benar(Mustika et al., 2021).

Rumus Teorema Bayes adalah:

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

Keterangan:

X : Data dengan kelas yang belum diketahui.

H : Hipotesis data X merupakan suatu label kelas tertentu.

$P(H|X)$: Probabilistik hipotesis H berdasarkan kondisi X (posteriori probability).

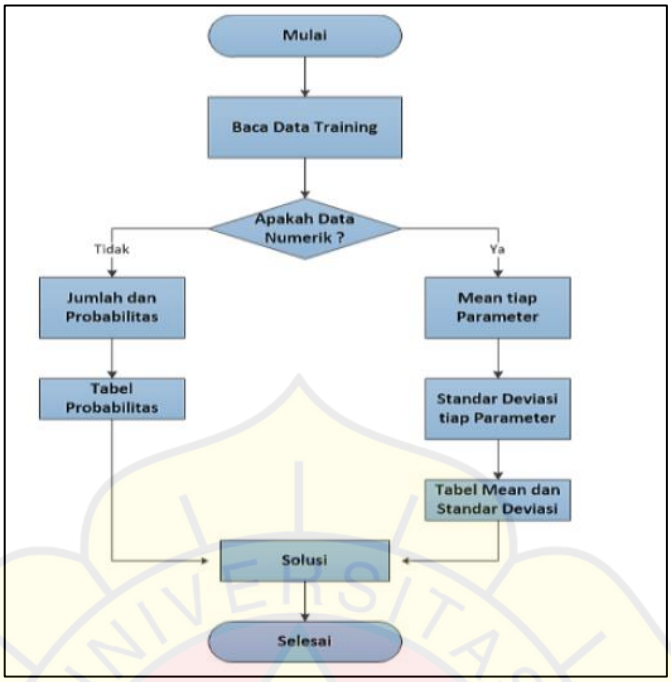
$P(H)$: Probabilistik hipotesis H (prior probability).

$P(X|H)$: Probabilitias X berdasarkan kondisi pada hipotesis H.

$P(X)$: Probabilistik X

Berikut alur dari metode *Naïve Bayes* pada dibawah ini:

1. Membaca *data training*.
2. Menghitung jumlah dan probabilitas. Namun, jika data berupa numerik, maka:
    a. Hitung nilai rata-rata (mean) dan simpangan baku (standar deviasi) dari setiap parameter yang berupa data numerik. Seperti Gambar 2.1 dibawah ini:



Gambar 2. 1 Alur Metode Naive Bayes (sumber:(Rayuwati et al., 2022))

$$\mu = \sum_{i=1}^{n} x_i \quad (2)$$

atau

$$\mu = \frac{X1 + X2 + X3 + \ldots + Xn}{n} \quad (3)$$

Keterangan:

μ : rata – rata hitung (mean)

x_i : nilai sampel ke-i

n : jumlah sampel

Dan persamaan untuk menghitung nilai simpangan baku (standar deviasi dapat dilihat sebagai berikut:

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

(4)

Keterangan:

σ : Standar deviasi

x_i : nilai x ke-i

μ : rata-rata hitung

n : jumlah sampel

b. Hitung nilai probabilitas dengan membagi jumlah data yang memenuhi kategori tertentu dengan total data dalam kategori tersebut.

3. Menentukan nilai dalam tabel yang mencakup rata-rata (mean), simpangan baku (standar deviasi), dan probabilitas.
4. Solusi kemudian dihasilkan(Rayuwati et al., 2022).

Kelebihan dan Kekurangan pada Naïve Bayes Classifier

1. Kelebihan *Naïve Bayes Classifier* meliputi:
   a. Algoritma *Naïve Bayes* sederhana dan mudah diimplementasikan, bahkan pada dataset yang besar.
   b. *Naïve Bayes* sering memberikan hasil yang memuaskan dalam berbagai jenis masalah klasifikasi, terutama pada data teks dan klasifikasi biner. Efisiensi algoritma ini membuatnya cocok digunakan pada aplikasi real-time.
2. Kekurangan *Naïve Bayes Classifier* meliputi:
   a. *Naïve Bayes* mengasumsikan bahwa setiap fitur tidak saling bergantung, yang sering kali tidak mencerminkan kondisi sebenarnya. Fitur-fitur dalam data biasanya memiliki korelasi atau hubungan, dan asumsi independensi

ini tidak dapat ditangani secara efektif oleh *Naïve Bayes*, sehingga dapat mempengaruhi akurasi hasil klasifikasi.

b. *Naïve Bayes* kurang optimal jika diterapkan pada data yang memerlukan model yang menangkap keterkaitan kompleks antar variabel, karena model ini tidak dapat menangani interaksi variabel dengan baik tanpa praproses tambahan(Ginantra, Ni Luh Wiwik Sri Rahayu et al., 2021).

2.1.5 Support Vector Machine (SVM)

Support Vector Machine (SVM) adalah sistem pembelajaran yang memanfaatkan ruang hipotesis berupa fungsi-fungsi linier dalam ruang fitur berdimensi tinggi. Konsep SVM secara sederhana dapat dijelaskan sebagai usaha menemukan *hyperplane* terbaik yang memisahkan dua kelas dalam ruang input. *Hyperplane* terbaik ini adalah yang memaksimalkan margin, yaitu jarak antara pemisah dan titik-titik data terdekat dari masing-masing kelas. Pemisahan yang optimal tercapai ketika *hyperplane* tersebut berada di tengah-tengah, memisahkan secara jelas antara kelas negatif dan kelas positif(Muttaqin et al., 2023).

Rumus untuk perhitungan Support Vector Machine (SVM):

Konsep dasar dari SVM adalah menemukan garis pemisah optimal yang dapat memisahkan dua kelas data dengan jelas. Garis pemisah ini atau *hyperplane*, ditentukan oleh persamaan (1):

$$(w.x_i) + b = 0 \quad (1)$$

Persamaan $(w.x_i) + b = 0$, menggambarkan *hyperplane* atau garis batas pemisah optimal dalam SVM. *Hyperplane* ini adalah garis (untuk data 2D) atau

bidang (untuk data 3D) yang memisahkan dua kelas data yang berbeda. Parameter w (vektor bobot) menentukan orientasi *hyperplane*, sedangkan b adalah bias atau offset dari *hyperplane* terhadap titik asal. Dengan kata lain, *hyperplane* ini merupakan batas yang memaksimalkan margin antara dua kelas data.

Pada data xi, yang termasuk dalam kelas -1, kondisi ini dirumuskan dalam persamaan (2):

$$(w.x_i + b) \leq 1, y_i = -1 \quad (2)$$

Persamaan $(w.x_i + b) \leq 1, y_i = -1$, berlaku untuk data xi yang berada di kelas -1, di mana $y_i = -1$. Artinya, semua data yang termasuk dalam kelas ini berada di sisi *hyperplane* yang sesuai dengan batas minimum jarak. Persamaan ini memastikan bahwa data dalam kelas -1 berada di luar margin *hyperplane*, sehingga membantu memaksimalkan margin antara kelas -1 dan kelas +1.

Sedangkan untuk data xi yang berada dalam kelas +1, kondisinya dirumuskan dengan persamaan (3):

$$(w.x_i + b) \geq 1, y_i = 1 \quad (3)$$

Persamaan $(w.x_i + b) \geq 1, y_i = 1$, berlaku untuk data xi yang berada di kelas +1, di mana $y_i = 1$. Sama seperti pada kelas -1, persamaan ini memastikan bahwa data dalam kelas +1 berada di sisi *hyperplane* yang tepat, di luar margin *hyperplane* untuk kelas ini. Dengan kata lain, setiap data dalam kelas +1 harus berada pada jarak yang ditentukan oleh margin dari *hyperplane*.

Keterangan:

w : vector bobot

x_i : vector data ke-i

b : bias (*offset*)

y_i : label kelas(Rahayu et al., 2022).

Kelebihan dan Kekurangan pada Support Vector Machine (SVM)

1. Kelebihan *Support Vector Machine* (SVM)
   A. SVM bekerja dengan memaksimalkan margin antara kelas, yang menghasilkan pemisahan yang jelas dan optimal.
   B. SVM dapat bekerja dengan baik pada data berdimensi tinggi (banyak fitur), bahkan ketika jumlah fitur lebih banyak dari jumlah sampel.

2. Kekurangan *Support Vector Machine* (SVM)
   A. SVM membutuhkan waktu komputasi yang besar saat diterapkan pada dataset besar, terutama dalam proses optimasi untuk menentukan margin maksimum.
   B. SVM sangat sensitif terhadap *outlier*, terutama pada data yang overlap antar kelas. Adanya *outlier* dapat mengubah posisi *hyperplane*, sehingga mengurangi akurasi model.

2.1.6 Confusion Matrix

Confusion Matrix merupakan alat evaluasi yang digunakan untuk menilai kinerja model klasifikasi dengan menggambarkan performa model dalam bentuk

matriks. Matriks ini memberikan informasi mengenai hasil klasifikasi yang benar maupun salah, serta memberikan rincian mengenai bagaimana model memprediksi setiap kelas(Dan et al., 2024). Konteks penelitian ini, *confusion matrix* digunakan untuk menganalisis hasil model dengan mengukur metrik evaluasi seperti akurasi, recall, precision, dan f1-score, yang masing-masing memberikan gambaran lebih mendalam tentang kinerja model dalam memprediksi kelas yang tepat.

2.1.7 Basis Data

Basis data adalah kumpulan data yang diorganisir sedemikian rupa agar mudah diakses, disimpan, dan dimanipulasi menggunakan konsep hubungan yang menggambarkan suatu domain tertentu. Dengan ini, pengguna dapat memperoleh informasi yang diperlukan dari basis data tersebut dengan lebih mudah(Yulianingsih et al., 2022).

2.1.8 CRISP DM

CRISP-DM (Cross Industry Standard for Data Mining) adalah metode yang umum digunakan oleh para ahli dalam pemodelan data. Metode ini bertujuan untuk mengidentifikasi pola yang bermakna dan relevan dalam data yang dianalisis. Dengan kerangka kerja yang terstruktur, CRISP-DM membantu pengguna dalam mengikuti langkah-langkah yang jelas selama proses penelitian(Dhewayani et al., 2022). Metode ini terdiri dari enam tahap, yaitu:

1. *Business Understanding*

   Tahap awal ini bertujuan memahami tujuan bisnis, merumuskan *masalah data mining* dan menyusun strategi untuk mencapainya. Fokusnya adalah memastikan tujuan bisnis selaras dengan model yang akan dibangun.

2. *Data Understanding*

   Tahap ini melibatkan eksplorasi data untuk mengidentifikasi masalah, memahami distribusi data, dan mendapatkan insight awal. Masalah seperti data hilang atau outlier diidentifikasi untuk ditangani pada tahap berikutnya.

3. *Data Preparation*

   Pada tahap ini, data diperbaiki, diubah, dan disiapkan agar sesuai dengan algoritma yang akan digunakan. Proses ini mencakup pembersihan data, transformasi, dan pembagian data menjadi *training* dan *testing*, serta sering diulang untuk menyesuaikan kebutuhan model.

4. *Modelling*

   Tahap ini melibatkan penerapan algoritma *machine learning* atau teknik statistik untuk membangun model prediktif atau deskriptif. Jika model membutuhkan penyesuaian, proses kembali ke *data preparation*.

5. *Evaluation*

   Hasil model dianalisis dan dievaluasi untuk memastikan model tersebut sesuai dengan tujuan yang telah ditetapkan pada tahap awal.

6. *Deployment*

   Model diterapkan dalam sistem operasional. Tahap ini mencakup perencanaan penggunaan model, integrasi dalam proses bisnis, serta pemantauan dan pembaruan model sesuai perubahan data(Muttaqin et al., 2023).

2.1.9 Pemodelan Sistem UML

Permodelan sistem menggunakan *Unified Modeling Language* sebagai perancangan sistem perangkat lunak yang membantu dalam memvisualisasikan struktur dan alur kerja sistem. Pada sub bab ini akan menjelaskan mengenai pengertian dari UML dan diagram-diagram seperti *use case* dan *activity*.

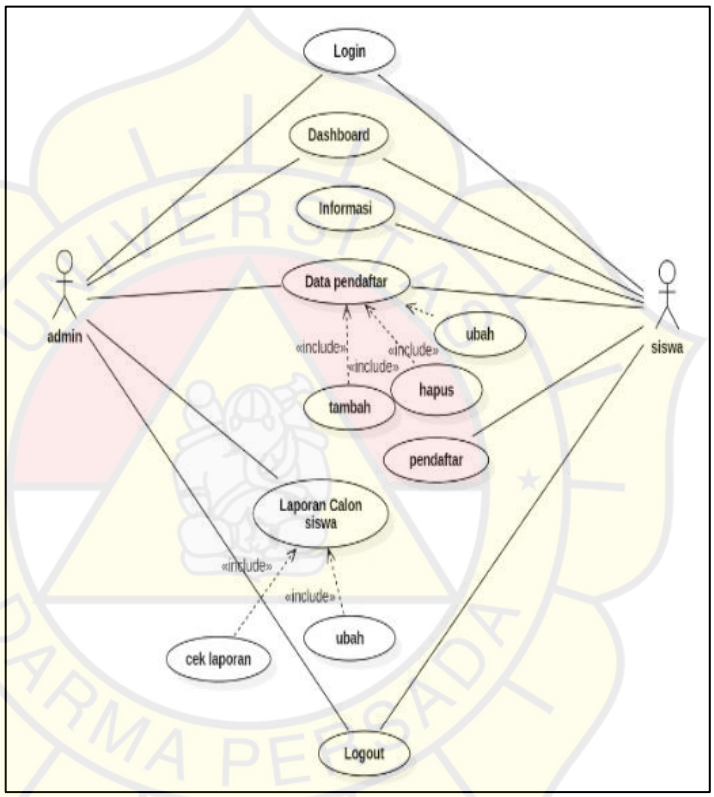
2.1.9.1 Unified Modelling Language (UML)

Unified Modeling Language (UML) adalah bahasa pemodelan standar yang digunakan untuk mendokumentasikan, merancang, dan membangun sistem perangkat lunak. UML membantu dalam mendefiniskan spesifikasi, konstruksi, serta dokumentasi komponen-komponen sistem, sehingga memudahkan pengembang dalam proses pengembangan(Abdillah, 2021). Dalam studi kasus, UML terbukti efektif untuk memvisualisasikan sistem secara grafis, menghasilkan kode program siap implementasi(Nistrina & Sahidah, 2022). UML juga memperlancar komunikasi antara pengguna dan pengembang, membuat perancangan sistem lebih efisien dengan berbagai diagram baik statis maupun dinamis(Primadasa & Juliansa, 2020).

2.1.9.1.1 Use Case Diagram

Use Case Diagram digunakan untuk menggambarkan fungsi utama sistem serta hubungan antara aktor (pengguna) dan fungsi tersebut. Aktor di sini mencakup peran seperti pelanggan, admin, atau pemilik toko. Diagram ini membantu mengidentifikasi kebutuhan pengguna dan alur kerja yang diperlukan untuk mencapai tujuan sistem(Priyambodo et al., 2024).

Secara teknis, *use case* mencakup skenario yang menjelaskan urutan langkah-langkah interaksi antara pengguna dan sistem, serta sebaliknya. Dengan demikian, *use case* menggambarkan fungsionalitas yang disediakan sistem, interaksi pengguna dengan sistem, serta koneksi antara pengguna dan setiap fungsi dalam system. Dapat dilihat seperti Gambar 2.2 dibawah ini:

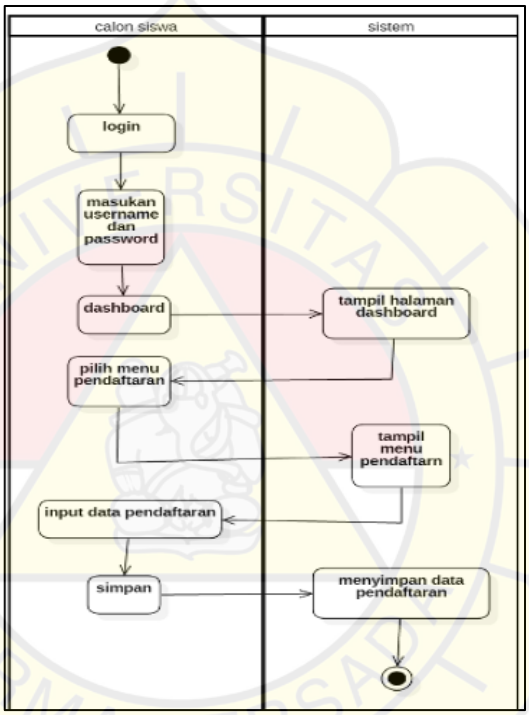


Gambar 2. 2 Contoh Use Case Diagram (sumber: (Nistrina & Sahidah, 2022))

2.1.9.1.2 Activity Diagram

Activity Diagram adalah diagram yang memodelkan aliran kerja atau proses bisnis dalam bentuk aktivitas dan keputusan. Diagram ini menggambarkan alur kerja atau aktivitas sistem pada perangkat lunak, yang mencakup berbagai tahap seperti pemilihan produk, pembayaran, dan pengiriman.

Berbeda dengan *Use Case* Diagram, yang berfokus pada interaksi antara aktor dan sistem untuk menunjukkan apa yang dilakukan aktor saat menggunakan sistem, *Activity* Diagram berfokus pada aktivitas internal sistem itu sendiri, bukan pada apa yang dilakukan oleh aktor(Priyambodo et al., 2024). Berikut ini adalah contoh penggunaan *activity* diagram seperti Gambar 2.3 dibawah ini:



Gambar 2. 3 Contoh Activity Diagram (sumber:(Nistrina & Sahidah, 2022))

2.1.10 Software dan Bahasa Pemrograman Terkait

Software dan pemrograman yang terkait dapat memengaruhi efisiensi, fungsi, serta penerapan sistem yang akan dikembangkan. Dalam konteks ini, *software* dan pemrograman yang digunakan meliputi editor pengolah data serta bahasa pemrograman yang sesuai dengan penelitian.

2.1.10.1 Editor Jupyter Notebook

Jupyter adalah aplikasi web gratis yang digunakan untuk membuat dan membagikan dokumen yang memuat kode, hasil perhitungan, visualisasi, dan teks. Nama *Jupyter* berasal dari tiga bahasa pemrograman populer, yaitu *Julia* (Ju), *Python* (Py), dan *R*, yang sering digunakan oleh para data scientist. Jupyter berperan penting dalam mendukung pembuatan narasi komputasi untuk memberikan makna serta wawasan dari data yang dianalisis.

Selain itu, *Jupyter* memudahkan kolaborasi antara insinyur dan *data scientist* karena memungkinkan penulisan dan berbagi teks serta kode dengan mudah. Fitur-fitur ini membantu *data scientist* untuk bekerja sama dengan peneliti data atau data engineer lainnya(Asyrofi & Asyrofi, 2023).

2.1.10.2 Streamlit

Streamlit adalah framework open-source yang memudahkan pengembang Data Science dan Machine Learning untuk membuat web deployment interaktif dengan cepat dan sederhana menggunakan bahasa pemrograman *Python.* Dengan *Streamlit*, pengembang dapat membangun aplikasi web yang menampilkan data secara interaktif, termasuk grafik, tabel, dan berbagai fitur interaktif lainnya dengan mudah. Streamlit menawarkan beragam pustaka dan fitur yang mendukung pengembangan aplikasi web secara efisien, seperti alat untuk visualisasi data, elemen interaktif seperti dropdown dan slider, serta integrasi dengan pustaka *Python* populer seperti *Pandas*, *Numpy*, dan *Matplotlib*. Fitur seperti autoreload memungkinkan aplikasi untuk memuat ulang secara otomatis saat kode diperbarui, mempercepat proses pengembangan.

Streamlit juga mendukung deployment ke berbagai platform cloud seperti *AWS* dan *Google Cloud*. Dalam pengembangan aplikasi web, pengembang dapat menyesuaikan tema dan antarmuka pengguna (UI) sesuai kebutuhan. *Streamlit* menyediakan dokumentasi yang lengkap serta komunitas yang aktif, yang membantu pengembang mendapatkan dukungan dan informasi yang diperlukan(Junaidi Surya, 2024).

2.1.10.3 CSS

CSS (Cascading Style Sheets) adalah bahasa yang digunakan untuk mengatur tampilan dan pemformatan dokumen berbasis HTML. CSS memungkinkan pengaturan elemen visual seperti warna, font, dan tata letak, serta memberikan fleksibilitas dalam desain web. Konsep "Cascading" mengatur prioritas aturan gaya, memungkinkan perubahan tampilan tanpa mengubah struktur konten. Dengan CSS, pengembang dapat lebih efisien dalam menciptakan dan memelihara situs web, menjadikannya elemen penting dalam pengembangan web modern.(Arisantoso et al., 2023).

2.1.10.4 Python

Python adalah bahasa pemrograman tingkat tinggi yang sangat terkenal dan multifungsi. Dikembangkan pertama kali oleh Guido van Rossum pada awal 1990-an, *Python* telah menjadi salah satu bahasa pemrograman yang paling banyak digunakan di kalangan pengembang perangkat lunak di seluruh dunia. Salah satu keuntungan utama dari *Python* adalah kemudahan penggunaannya serta sintaksisnya yang sederhana dan mudah dipahami, menjadikannya pilihan ideal bagi pemula maupun pengembang berpengalaman.

Karakteristik khas dari *Python* adalah filosofi desainnya yang menekankan pada keterbacaan kode. Ini tercermin dalam sintaks yang bersih, yang memungkinkan pengembang untuk menulis kode dengan cepat dan efisien. *Python* juga dilengkapi dengan perpustakaan standar yang luas dan komprehensif, yang menyediakan berbagai modul dan fungsi bawaan untuk mempermudah pengembangan aplikasi. Pustaka standar ini membantu pengembang menghemat waktu dan usaha saat menerapkan fungsionalitas umum yang sering digunakan dalam pengembangan perangkat lunak(Santos et al., 2024). *Python* adalah bahasa pemrograman open-source yang efisien dan mudah digunakan. Keunggulan utama *Python* terletak pada kemudahan penggunaannya, di mana untuk menjalankan operasi, pengguna tidak perlu menulis banyak baris kode. Bahkan, masalah yang kompleks sekalipun dapat diselesaikan hanya dengan beberapa baris kode saja. Bahasa pemrograman ini dapat digunakan pada bidang ilmu komputer dan ilmu data. *Data science* adalah sebuah disiplin ilmu yang fokus mempelajari data, terutama data kuantitatif (numerik), baik yang terstruktur maupun tidak terstruktur. Bidang ini mencakup berbagai aspek terkait data, mulai dari pengumpulan, analisis, dan pengolahan data, hingga manajemen, penyimpanan, pengelompokan, presentasi, distribusi, serta cara mengubah data menjadi informasi yang bermakna(Adhisyanda Aditya et al., 2020).

Python memiliki banyak library yang membantu berbagai kebutuhan pengembangan perangkat lunak, seperti:

2.1.10.4.1 Library Pandas

Pandas adalah pustaka open-source pada Python yang dirancang khusus untuk analisis data, menyediakan struktur data yang mudah digunakan namun berkinerja tinggi. Dengan *pandas*, data atau file CSV dapat diproses dan diolah untuk menghasilkan dataset yang terstruktur dan siap dianalisis lebih lanjut. Selain itu, *pandas* memungkinkan manipulasi data dengan berbagai cara, seperti menggabungkan, menyaring, atau merapikan data, sehingga sangat berguna dalam tahap pra-pemrosesan dan eksplorasi data dalam proyek data science(Setiawan et al., 2020).

```
import pandas as pd

df = pd.read_csv('data.csv')

print(df.to_string())
```

Gambar 2. 4 Library Pandas (sumber: w3schools)

2.1.10.4.2 Library Matplothlib

Matplotlib adalah pustaka Python yang berfokus pada visualisasi data, memungkinkan pembuatan berbagai jenis grafik dan plot. Dengan *Matplotlib*, pengguna dapat membuat visualisasi seperti grafik garis, batang, histogram, dan scatter plot untuk membantu menganalisis serta memahami pola dalam data. Selain itu, *Matplotlib* dapat dikustomisasi dengan berbagai opsi seperti warna, label, dan gaya, sehingga mendukung pembuatan grafik yang informatif dan menarik dalam analisis data maupun laporan hasil penelitian(Nur & Cahyani, 2024).

```
import matplotlib.pyplot as plt
import numpy as np

xpoints = np.array([0, 6])
ypoints = np.array([0, 250])

plt.plot(xpoints, ypoints)
plt.show()
```

Gambar 2. 5 Library Matplotlib (sumber: w3schools)

2.1.10.4.3 Library Seaborn

Seaborn adalah pustaka Python yang digunakan untuk membuat grafik statistik, dirancang untuk meningkatkan pemahaman data melalui visualisasi yang menarik dan informatif. Dikembangkan di atas pustaka *Matplotlib*, *Seaborn* memanfaatkan struktur data dari *Pandas*, yang memudahkan integrasi data secara terorganisir. Tujuan utama *Seaborn* adalah menyediakan alat visualisasi yang intuitif untuk membantu analisis data dengan menghasilkan grafik yang berfokus pada interpretasi semantik dan agregasi statistik. Dengan kemampuan plotting berdasarkan dataset yang terstruktur dalam DataFrame atau array, Seaborn mempermudah pembuatan grafik yang informatif dan membantu pengguna dalam menemukan pola dan wawasan dari data yang diolah(Rozzika et al., 2023).

```
import matplotlib.pyplot as plt
import seaborn as sns

sns.distplot([0, 1, 2, 3, 4, 5])

plt.show()
```

Gambar 2. 6 Library Seaborn (sumber: w3schools)

2.1.10.4.4 Library Sklearn

Scikit-learn, sering disebut sebagai *sklearn*, adalah perangkat lunak open-source untuk machine learning yang dikembangkan dalam bahasa pemrograman Python. Pustaka ini mendukung berbagai algoritma pembelajaran mesin, seperti klasifikasi, regresi, dan clustering, termasuk algoritma populer seperti k-means. *Scikit-learn* sangat efektif digunakan untuk tugas-tugas analisis data dan pemodelan karena menyediakan antarmuka yang mudah dipahami serta dokumentasi yang lengkap. Dengan popularitasnya yang tinggi di GitHub, *scikit-learn* menjadi salah satu pustaka paling banyak digunakan oleh komunitas data scientist dan machine learning untuk membangun model prediktif dan mengolah data secara efisien(Rakhmawati et al., 2020).

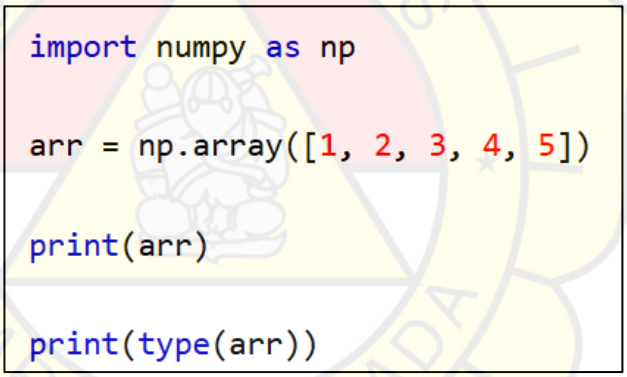
```
from sklearn import datasets

X, y = datasets.load_iris(return_X_y=True)
```

Gambar 2. 7 Library Sklearn (sumber: w3schools)

2.1.10.4.5 Library Numpy

NumPy (Numerical Python) adalah pustaka *Python* yang dirancang khusus untuk komputasi ilmiah. Pustaka ini menyediakan dukungan untuk operasi matematika tingkat lanjut dan manipulasi array atau matriks multidimensi yang sangat efisien, menjadikannya komponen penting dalam analisis data dan pemrosesan numerik. Dengan fungsionalitas seperti aljabar linier, transformasi Fourier, dan operasi acak, *NumPy* menjadi fondasi bagi banyak pustaka lain dalam ekosistem *Python*, seperti *Pandas* dan *SciPy*, yang juga berfokus pada analisis dan pemrosesan data ilmiah(Nur & Cahyani, 2024).



Gambar 2. 8 Library Numpy (sumber: w3schools)

2.1.11 Waterfall

Waterfall adalah metode pengembangan perangkat lunak yang menerapkan proses secara berurutan atau linear. Setiap langkah dalam pengembangan harus diselesaikan secara berurutan, sehingga tahap berikutnya hanya dapat dimulai setelah tahap sebelumnya selesai. Sebagai contoh, tahap ketiga dapat dilaksanakan hanya setelah tahap pertama dan kedua selesai dilakukan(Fitri Khoiry Tamami Salam & Septanto, 2024). Metode ini terdiri dari lima tahapan yaitu:

1. Analisis Kebutuhan (*Requirement Analysis*)

   Tahap awal dalam pengembangan perangkat lunak dimulai dengan pengumpulan kebutuhan secara mendalam dan terfokus untuk memahami jenis perangkat lunak yang dibutuhkan oleh pengguna. Spesifikasi kebutuhan perangkat lunak harus didokumentasikan dengan baik(Voutama & Novalia, 2022).

2. Desain Sistem (*System Design*)

   Setelah analisis kebutuhan selesai, tahap selanjutnya adalah perancangan sistem. Pada tahap ini, arsitektur sistem dirancang, termasuk struktur database serta logika bisnis aplikasi. Hasil dari proses ini berupa dokumen perancangan yang digunakan sebagai pedoman dalam tahap implementasi. Perancangan sistem menggunakan UML (*Unified Modeling Language*), yang mencakup pembuatan Use Case Diagram, Activity Diagram, dan diagram lainnya.(Fitri Khoiry Tamami Salam & Septanto, 2024).

3. Implementasi (*Implementation/Coding*)

   Setelah perancangan selesai, langkah berikutnya adalah menerjemahkan rancangan tersebut ke dalam program perangkat lunak. Tahap ini menghasilkan program komputer yang sesuai dengan desain yang telah dibuat sebelumnya. Pada tahap ini, penulis mengembangkan program menggunakan bahasa pemrograman Python dan database MySQL.(Badrul et al., 2021)

4. Pengujian (*Testing*)

   Pada tahap ini, dilakukan pengujian terhadap program yang telah dikembangkan untuk mengidentifikasi kekurangan yang mungkin ada.

Pengujian mencakup validasi halaman login serta memastikan setiap fungsi dalam menu dapat berjalan dengan baik(Voutama & Novalia, 2022).

5. Pemeliharaan (*Maintenance*)

   Setiap perangkat lunak memerlukan pemeliharaan, salah satunya dalam bentuk pengembangan. Seiring waktu, kebutuhan perangkat lunak terus berubah, sehingga diperlukan penambahan fitur baru yang sebelumnya belum tersedia(Badrul et al., 2021).

2.2 Tinjauan Literatur/Kajian Penelitian Terdahulu

Tinjauan literatur atau kajian penelitian sebelumnya dilakukan untuk membangun dasar teori yang kuat dan memastikan bahwa penelitian ini memberikan kontribusi baru dalam bidang prediksi beasiswa. Kajian ini mencakup berbagai penelitian terkait penerapan algoritma *Naïve Bayes* dan *Support Vector Machine* (SVM) dalam klasifikasi data, serta penggunaan data mining untuk mendukung proses seleksi dan prediksi kelayakan penerima beasiswa.

2.2.1 Paper 1

Judul: Implementasi Algoritma *Naive Bayes* Untuk Menentukan Calon Penerima Beasiswa Di SMK YPM 14 Sumobito Jombang

Author: Wahyuningsih, Budiman, dan Izzatul Umami

Publikasi: Jurnal Teknologi Dan Sistem Informasi Bisnis (JTEKSIS)

Tahun: 2022

Klasifikasi Journal: Sinta 4

2.2.1.1 Tujuan Penelitian

Tujuan penelitian ini adalah untuk mengetahui kelayakan siswa dalam menerima bantuan beasiswa dan meningkatkan akurasi dari perhitungan algoritma klasifikasi yang digunakan, yaitu algoritma *Naïve Bayes*. Penelitian ini juga bertujuan untuk mengatasi tantangan dalam proses seleksi beasiswa yang sebelumnya bersifat subjektif dan tanpa metode prediktif.

2.2.1.2 Metodelogi Yang Digunakan

Metodologi yang digunakan dalam penelitian ini adalah model *Cross-Industry Standard Process for Data Mining* (CRISP-DM), yang terdiri dari enam tahap proses: *Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, dan Deployment.* Penelitian ini juga melibatkan pengumpulan dan analisis data siswa untuk menentukan kelayakan penerima beasiswa menggunakan algoritma *Naïve Bayes.*

2.2.1.3 Temuan Utama

Temuan utama dari penelitian ini menunjukkan bahwa penggunaan algoritma *Naïve Bayes* dalam menentukan calon penerima beasiswa menghasilkan akurasi yang tinggi, yaitu sebesar 90.48%, dengan precision 96.88% dan recall 83.33%. Kriteria yang dianalisis, seperti pekerjaan orang tua, penghasilan, tanggungan, jarak rumah, nilai, dan sikap siswa, terbukti berpengaruh signifikan terhadap kelayakan penerima beasiswa. Penelitian ini juga menekankan pentingnya pendekatan sistematis dalam pengambilan keputusan untuk seleksi beasiswa, yang sebelumnya bersifat subjektif. Dengan penerapan teknik data mining, proses seleksi menjadi lebih efisien dan akurat. Selain itu, evaluasi

berkelanjutan dalam proses seleksi beasiswa sangat penting, mengingat jumlah peserta didik dapat berubah seiring waktu. Metodologi CRISP-DM yang digunakan dalam penelitian ini membantu dalam struktur dan proses penelitian, mulai dari pemahaman data hingga penerapan model, sehingga dapat disimpulkan bahwa penerapan algoritma *Naïve Bayes* dapat meningkatkan proses seleksi beasiswa dengan cara yang lebih objektif dan berbasis data.

2.2.1.4 Kesimpulan Penelitian

Kesimpulan dari penelitian ini adalah bahwa penerapan algoritma *Naïve Bayes* dalam menentukan calon penerima beasiswa di SMK YPM 14 Sumobito, Jombang, terbukti efektif dengan menghasilkan akurasi sebesar 90.48%. Kriteria yang dianalisis, seperti pekerjaan orang tua, penghasilan, tanggungan, jarak rumah, nilai, dan sikap siswa, memiliki pengaruh signifikan terhadap kelayakan penerima beasiswa. Penelitian ini menekankan pentingnya pendekatan sistematis dalam pengambilan keputusan untuk seleksi beasiswa, serta perlunya evaluasi berkelanjutan dalam proses tersebut. Dengan demikian, algoritma *Naïve Bayes* dapat meningkatkan proses seleksi beasiswa menjadi lebih objektif dan berbasis data.

2.2.2 Paper 2

Judul: Perbandingan algoritma klasifikasi *naive bayes* dan svm pada studi kasus pemberian penerima beasiswa PPA

Author: Safitri Linawati, Rizky Ade Safitri, Ahmad Rifqy Alfiyan, dan Witriana Endah Pangesti, serta Monikka Nur Winarto

Publikasi: Jurnal Swabumi

Tahun: 2020

Klasifikasi Journal: Sinta 4

2.2.2.1 Tujuan Penelitian

Tujuan penelitian ini adalah untuk membandingkan dua algoritma klasifikasi, yaitu *Naïve Bayes* dan *Support Vector Machine* (SVM), guna mengetahui algoritma mana yang memiliki keakuratan lebih tinggi dalam pengambilan keputusan penerima beasiswa PPA.

2.2.2.2 Metodologi Yang Digunakan

Metodologi yang digunakan dalam penelitian ini terdiri dari beberapa langkah penting. Pertama, dilakukan pengumpulan data mengenai pemberian beasiswa PPA, yang mencakup 122 sampel dengan 5 variabel, yaitu semester, pekerjaan, orang tua, penghasilan, dan IPK. Data tersebut kemudian diolah dan diimpor ke dalam format CSV untuk analisis lebih lanjut. Selanjutnya, proses *data mining* diterapkan, yang memanfaatkan teknik statistik, matematika, kecerdasan buatan, dan machine learning untuk mengekstraksi informasi yang bermanfaat dari database besar. Setelah itu, data dalam format CSV diolah menggunakan tools WEKA, di mana pengujian dilakukan dengan memilih algoritma yang akan digunakan, yaitu *Naïve Bayes* dan *Support Vector Machine* (SVM). Terakhir, evaluasi dilakukan terhadap akurasi, presisi, dan recall dari kedua algoritma, serta analisis *confusion matrix* untuk membandingkan hasil dari masing-masing algoritma. Metodologi ini bertujuan untuk membandingkan keakuratan kedua algoritma dalam pengambilan keputusan penerima beasiswa PPA.

2.2.2.3 Temuan Utama

Temuan utama dari penelitian ini menunjukkan bahwa algoritma *Naïve Bayes* memiliki akurasi tertinggi sebesar 90,90% dalam klasifikasi data penerima beasiswa PPA, dibandingkan dengan algoritma *Support Vector Machine* (SVM) yang mencapai akurasi 89,25%. Meskipun terdapat perbedaan dalam tingkat akurasi, nilai presisi dan recall untuk kedua algoritma tidak menunjukkan perbedaan yang signifikan. Hasil ini menegaskan bahwa dengan jumlah fitur yang sama, akurasi yang dihasilkan oleh kedua algoritma cenderung serupa. Penelitian ini juga menyoroti pentingnya penerapan data mining dalam proses pengambilan keputusan untuk seleksi penerima beasiswa.

2.2.2.4 Kesimpulan Penelitian

Kesimpulan dari penelitian ini menunjukkan bahwa algoritma *Naïve Bayes* lebih unggul dibandingkan dengan *Support Vector Machine* (SVM) dalam hal akurasi, dengan nilai akurasi tertinggi mencapai 90,90% untuk *Naïve Bayes*, sedangkan SVM mencapai 89,25%. Meskipun terdapat perbedaan dalam tingkat akurasi, nilai presisi dan recall untuk kedua algoritma tidak menunjukkan perbedaan yang signifikan. Penelitian ini menegaskan pentingnya penerapan data mining dalam pengambilan keputusan, khususnya dalam seleksi penerima beasiswa PPA, untuk meningkatkan keakuratan dan efektivitas proses seleksi.

2.2.3 Paper 3

Judul: Klasifikasi Pemberian Beasiswa Berprestasi Menggunakan Perbandingan Tiga Algoritma

Author: Nanda Tri Haryati, Edi Surya Negara, dan Tri Basuki Kurniawan

Publikasi: Jurnal Teknologi Komputerisasi Akuntansi (TEKNO KOMPAK)

Tahun: 2023

Klasifikasi Journal: Sinta 4

2.2.3.1 Tujuan Penelitian

Tujuan penelitian ini adalah untuk membandingkan tiga algoritma klasifikasi, yaitu *Naïve Bayes, Random Forest, dan Support Vector Machine*, dalam memprediksi siswa yang kemungkinan akan menerima beasiswa. Dengan demikian, pihak sekolah dapat dengan mudah menentukan data siswa mana yang akan diajukan untuk menerima beasiswa.

2.2.3.2 Metodelogi Yang Digunakan

Penelitian ini menggunakan metode eksperimen untuk menganalisis data siswa yang mengajukan beasiswa PIP di SMK Nurul Iman Palembang dari tahun 2018 hingga 2022, dengan mengumpulkan 2434 entri yang mencakup 21 fitur, termasuk informasi pribadi siswa, nilai mata pelajaran, dan data orang tua. Proses penelitian meliputi pengumpulan data primer, analisis, pembersihan, dan transformasi data sebelum membagi data menjadi set pelatihan dan pengujian. Tiga algoritma klasifikasi, yaitu *Naïve Bayes, Random Forest,* dan *Support Vector Machine* (SVM), digunakan untuk membangun model dan mengukur akurasi. Hasil penelitian menunjukkan bahwa algoritma *Random Forest* memiliki akurasi tertinggi sebesar 75,52%, diikuti oleh SVM dengan 59,18% dan *Naïve Bayes* dengan 55,07%. Pengujian menggunakan *cross-validation* menunjukkan konsistensi hasil, dengan *Random Forest* tetap unggul (akurasi rata-rata 73,55%). Selain itu, pemilihan fitur menunjukkan bahwa penghasilan merupakan faktor

paling berpengaruh dalam menentukan kelayakan beasiswa, sementara fitur matematika dan bahasa Inggris kurang signifikan. Penelitian ini menekankan pentingnya penggunaan metode berbasis data untuk meningkatkan efisiensi alokasi beasiswa.

2.2.3.3 Temuan Utama

Penelitian ini menggunakan metode eksperimen untuk menganalisis data siswa yang mengajukan beasiswa PIP di SMK Nurul Iman Palembang dari tahun 2018 hingga 2022, dengan mengumpulkan 2434 entri yang mencakup 21 fitur, termasuk informasi pribadi siswa, nilai mata pelajaran, dan data orang tua. Tiga algoritma klasifikasi, yaitu *Naïve Bayes*, *Random Forest*, dan *Support Vector Machine* (SVM), digunakan untuk membangun model dan mengukur akurasi. Hasil penelitian menunjukkan bahwa algoritma *Random Forest* memiliki akurasi tertinggi sebesar 75,52%, diikuti oleh SVM dengan 59,18% dan *Naïve Bayes* dengan 55,07%. Pengujian menggunakan *cross-validation* menunjukkan konsistensi hasil, dengan *Random Forest* tetap unggul (akurasi rata-rata 73,55%). Selain itu, analisis pemilihan fitur mengungkapkan bahwa penghasilan merupakan faktor paling berpengaruh dalam menentukan kelayakan beasiswa, sementara fitur terkait nilai mata pelajaran, seperti matematika dan bahasa Inggris, kurang signifikan. Penelitian ini menekankan pentingnya penggunaan metode berbasis data untuk meningkatkan efisiensi alokasi beasiswa.

2.2.3.4 Kesimpulan Penelitian

Kesimpulan dari penelitian ini menunjukkan bahwa algoritma *Random Forest* adalah yang paling efektif dalam memprediksi siswa yang berpotensi

menerima beasiswa, dengan akurasi tertinggi sebesar 75,52%. Selain itu, hasil penelitian juga mengindikasikan bahwa penghasilan orang tua merupakan faktor paling berpengaruh dalam menentukan kelayakan beasiswa, sedangkan nilai mata pelajaran seperti matematika dan bahasa Inggris kurang signifikan. Penelitian ini menekankan pentingnya penggunaan metode berbasis data untuk meningkatkan efisiensi alokasi beasiswa dan menyarankan pemilihan fitur yang tepat untuk meningkatkan akurasi model prediksi.

2.2.4 Paper 4

Judul: Implementasi Algoritma *Naive Bayes* Untuk Klasifikasi Penerima Beasiswa (Studi Kasus Universitas Hamzanwadi)

Author: Nurhidayati, Yahya, Fathurrahman, L.M Samsu, dan Wajizatul Amnia

Publikasi: Jurnal Informatika dan Teknologi (INFOTEK)

Tahun: 2023

Klasifikasi Journal: Sinta 4

2.2.4.1 Tujuan Penelitian

Tujuan penelitian ini adalah untuk memahami dan mengidentifikasi permasalahan dalam proses seleksi penerima beasiswa bidikmisi di Universitas Hamzanwadi, yang masih menggunakan metode manual. Penelitian ini bertujuan untuk mengimplementasikan algoritma *Naive Bayes* dalam klasifikasi penerima beasiswa, sehingga dapat meningkatkan efisiensi dan akurasi dalam proses seleksi.

2.2.4.2 Metodelogi Yang Digunakan

Metodologi yang digunakan dalam penelitian ini mengikuti model CRISP-DM (*Cross Industry Standard Process for Data Mining*), yang terdiri dari enam

tahapan. Pertama, pada tahap Pemahaman Bisnis, peneliti mengidentifikasi permasalahan dalam proses seleksi penerima beasiswa yang masih menggunakan metode manual. Selanjutnya, pada tahap Pemahaman Data, data dikumpulkan melalui observasi dan wawancara untuk mendapatkan informasi yang diperlukan. Pada tahap Persiapan Data, peneliti menentukan variabel yang akan dianalisis dan mempersiapkan data untuk pemodelan. Kemudian, pada tahap Pemodelan, algoritma *Naive Bayes* digunakan untuk klasifikasi penerima beasiswa. Setelah itu, pada tahap Evaluasi, model diuji menggunakan confusion matrix untuk mendapatkan akurasi dan nilai AUC. Terakhir, pada tahap Penyebaran, proses data mining diterapkan sebagai solusi untuk permasalahan yang ada.

2.2.4.3 Temuan Utama

Temuan utama dari penelitian ini menunjukkan bahwa algoritma *Naive Bayes* efektif dalam mengklasifikasikan mahasiswa yang berhak menerima beasiswa bidikmisi di Universitas Hamzanwadi. Dengan menggunakan *metode k-fold cross-validation*, penelitian ini mencapai akurasi tertinggi sebesar 91,43%, presisi 90,53%, dan recall 99,67%. Nilai AUC yang mencapai 0,996 menandakan bahwa klasifikasi yang dilakukan sangat baik. Analisis melalui confusion matrix menunjukkan adanya 306 true positif, 47 true negatif, 32 false negatif, dan 1 false positif. Temuan ini mengindikasikan bahwa algoritma *Naive Bayes* dapat menjadi alat evaluasi yang efektif dalam proses seleksi beasiswa, serta dapat meningkatkan efisiensi dan akurasi dibandingkan dengan metode manual yang saat ini digunakan.

2.2.4.4 Kesimpulan Penelitian

Kesimpulan dari penelitian ini menunjukkan bahwa algoritma *Naive Bayes* terbukti efektif dalam mengklasifikasikan mahasiswa yang berhak menerima beasiswa bidikmisi di Universitas Hamzanwadi. Dengan penerapan metode *k-fold cross-validation*, penelitian ini berhasil mencapai akurasi tertinggi sebesar 91,43%, presisi 90,53%, dan recall 99,67%. Nilai AUC yang mencapai 0,996 menandakan bahwa klasifikasi yang dilakukan sangat baik. Hasil analisis melalui *confusion matrix* menunjukkan adanya 306 true positif, 47 true negatif, 32 false negatif, dan 1 false positif. Temuan ini mengindikasikan bahwa algoritma *Naive Bayes* dapat menjadi alat evaluasi yang efektif dalam proses seleksi beasiswa, serta dapat meningkatkan efisiensi dan akurasi dibandingkan dengan metode manual yang saat ini digunakan.

2.2.5 Paper 5

Judul: Implementasi Algoritma SVM dan C4.5 dalam Klasifikasi Calon Penerimaan Beasiswa

Author: Rahmaddeni, Syarfi Aziz, Zairi Saputra, Hafid Azis Supahri, Ryan Ismanizan

Publikasi: Jurnal Komunikasi, Teknologi, dan Informasi (KOMTEKINFO)

Tahun: 2024

Klasifikasi Journal: Sinta 4

2.2.5.1 Tujuan Penelitian

Tujuan penelitian ini adalah untuk mengevaluasi algoritma *Support Vector Machine* (SVM) dan C4.5 dalam memprediksi kandidat penerima beasiswa

dengan menggunakan data historis yang tersedia. Penelitian ini bertujuan untuk meningkatkan akurasi dalam penyaluran beasiswa, sehingga lebih banyak siswa yang dapat terbantu dalam pencapaian cita-cita mereka.

2.2.5.2 Metodelogi Yang Digunakan

Metodelogi yang digunakan dalam penelitian ini melibatkan beberapa langkah sistematis. Pertama, data siswa yang layak mendapatkan beasiswa dikumpulkan dan disiapkan untuk analisis. Selanjutnya, dilakukan preprocessing yang mencakup pembersihan data, transformasi data, dan pembagian data menjadi set pelatihan dan pengujian dengan rasio 70:30. Setelah data diproses, algoritma *Support Vector Machine* (SVM) dan C4.5 diterapkan untuk menghasilkan hasil klasifikasi, di mana model dilatih menggunakan set pelatihan dan kemudian digunakan untuk mengklasifikasi data baru. Terakhir, hasil klasifikasi dievaluasi untuk mengukur kinerja model dalam menkategorikan data, termasuk pengukuran akurasi dan metrik lainnya. Metodologi ini dirancang untuk memastikan setiap tahap penelitian berjalan sesuai dengan tujuan yang ditetapkan serta untuk meningkatkan efisiensi dan efektivitas dalam penyaluran beasiswa.

2.2.5.3 Temuan Utama

Temuan utama dari penelitian ini adalah bahwa algoritma C4.5 menunjukkan performa yang lebih baik dibandingkan dengan algoritma *Support Vector Machine* (SVM) dalam klasifikasi calon penerima beasiswa. Model C4.5 mencapai akurasi 78.27%, presisi 60.27%, recall 53.01%, dan F1 Score 56.41%, sementara model SVM memiliki akurasi 75.72%, presisi 57.45%, recall 32.53%, dan F1 Score 41.54%. Meskipun SVM menunjukkan akurasi yang baik, C4.5 lebih

efektif dalam menangkap kasus positif. Penelitian ini merekomendasikan penggunaan model C4.5 dalam sistem klasifikasi penerima beasiswa dan menekankan pentingnya preprocessing data, serta potensi pengembangan lebih lanjut melalui tuning parameter dan eksplorasi algoritma ensemble untuk meningkatkan kinerja klasifikasi.

2.2.5.4 Kesimpulan Penelitian

Kesimpulan dari penelitian ini menunjukkan bahwa algoritma C4.5 lebih efektif dibandingkan dengan algoritma *Support Vector Machine* (SVM) dalam klasifikasi calon penerima beasiswa. Model C4.5 mencapai akurasi 78.27%, sementara SVM hanya mencapai 75.72%. Meskipun SVM menunjukkan akurasi yang baik, C4.5 lebih unggul dalam menangkap kasus positif, dengan presisi dan recall yang lebih baik. Penelitian ini merekomendasikan penggunaan model C4.5 dalam sistem klasifikasi penerima beasiswa dan menekankan pentingnya preprocessing data untuk meningkatkan kinerja klasifikasi. Selain itu, ada potensi untuk pengembangan lebih lanjut melalui tuning parameter dan eksplorasi algoritma ensemble.