BAB 2
LANDASAN TEORI

2.1 Landasan Teori

2.1.1 *Education Data Mining* (EDM)

Education Data Mining atau yang sering disingkat dengan EDM merupakan sebuah disiplin ilmu yang berfokus pada pengembangan, riset, dan penerapan metode-metode berbasis komputer untuk mengeksplorasi data unik yang berasal dari ekosistem pendidikan.

Definisi 1. *Education Data Mining* (EDM) merujuk pada sebuah proses penambangan data yang digunakan untuk mengekstrak informasi berharga serta pola-pola perilaku tersembunyi dari sekumpulan data besar yang dihasilkan oleh aktivitas pengajaran dan pembelajaran.

Mengacu pada penelitian dari Kord dkk.(2025), bidang ilmu ini memanfaatkan teknik analisis data komputasional untuk memahami karakteristik siswa serta lingkungan belajar secara menyeluruh [13]. Melalui penerapan EDM, data yang semula hanya tersimpan sebagai catatan niali administratif atau log aktivitas digital yang pasif dapat diubah menjadi sebuah pengetahuan strategis. Pengetahuan inilah yang kemudian dapat digunakan oleh para pendidik untuk mengevaluasi efektivitas metode pengajaran, mendeteksi hambatan belajar, serta memahami fenomena psikologis siswa di dalam kelas secara objektif.

Meskipun lahir dari akar keilmuan yang sama, terdapat perbedaan mendasar antara *Education Data Mining* dengan teknik *Data Mining* konvensional pada umumnya berfokus pada sektor komersial seperti retail, bisnis, perbankan, atau *e-commerce*, dengan tujuan utama untuk mengoptimalkan profit perusahaan, memprediksi perilaku konsumen dalam membeli barang, serta menganalisis efisiensi pasar. Di sisi lain, EDM memiliki karakteristik data yang jauh lebih kompleks dan sensitif karena objek yang di teliti adalah manusia yang sedang berada dalam proses perkembangan intelektual. Berdasarkan temuan Raj dan Renumol.(2021), data dalam dunia pendidikan memiliki sifat hierarkis dan multidimensional yang tidak hanya mencakup nilai angka saja, melainkan melibatkan aspek kognitif, tingkat motivasi, interaksi sosial, hingga pengaruh lingkungan tempat siswa belajar. Oleh karena itu, jika data mining biasa hanya

melihat hubungan sebab-akibat yang linear untuk transaksi ekonomi,maka EDM harus mampu menginterpretasikan dinamika psikologis manusia yang dinamis dan tidak selalu dapat diukur secara kaku dengan metrik bisnis standar.

Penerapan Educational Data Mining di dunia pendidikan saat ini telah mencakup spektrum yang sangat luas untuk menjawab berbagai tantangan operasional dan metode pengajaran di sekolah maupun perguruan tinggi. Salah satu aplikasi utamanya adalah pemodelan prediktif untuk memetakan tingkat kemampuan akademis siswa serta mendeteksi secara dini risiko kegagalan atau keterlambatan kelulusan. Sejalan dengan pendapat Ahmed (2024), pemanfaatan algoritma kecerdasan buatan dalam ranah EDM memberikan kemampuan bagi institusi pendidikan untuk mengambil tindakan evaluasi yang lebih proaktif sebelum siswa mengalami penurunan performa yang drastis [5]. Selain itu, merujuk pada temuan Pal dkk.(2025), EDM juga diaplikasikan secara masif dalam membangun sistem rekomendasi cerdas yang mampu mempersonalisasikan sumber daya, materi, atau jalur pembelajaran digital agar sesuai dengan preferensi, minat, dan kecepatan belajar masing-masing peserta didik tanpa harus menyamaratakan perlakuan kepada semua siswa [2].

Penjelasan mengenai ruang lingkup dan fungsi dari EDM tersebut memiliki hubungan relevansi yang sangat kuat dengan penelitian skripsi yang sedang dijalankan saat ini. Penelitian ini berfokus pada klasifikasi tingkat kemampuan siswa ke dalam tiga kelompok utama, yaitu Lemah, Sedang, dan Pintar, dengan memanfaatkan fitur gabungan yang mengintegrasikan aspek nilai pelajaran dan aspek perilaku sehari-hari siswa. Sejalan dengan argumen Muhaimin dkk.(2024), penggunaan metode penambangan data pendidikan menggunakan algoritma klasifikasi terbukti mampu memberikan hasil pengelompokan profil siswa yang akurat untuk mendukung sistem evaluasi sekolah [14]. Melalui koridor ilmu EDM, keputusan untuk memasukkan variabel non-kognitif seperti durasi tidur, waktu penggunaan gawai (*screen time*), dan kecemasan ujian ke dalam model klasifikasi *K-Nearest Neighbor* mendapatkan validasi ilmiah yang kuat. Hal ini membuktikan bahwa faktor gaya hidup dan kondisi psikologis merupakan komponen data pendidikan yang sah serta esensial dalam menentukan arah pemberian rekomendasi pembelajaran yang bersifat personal bagi siswa.

2.1.2 Klasifikasi

Dalam ranah *machine learning* dan *Data Mining*, klasifikasi merupakan salah satu teknik fundamental yang termasuk ke dalam kategori *supervised learning* atau pembelajaran terawasi. Definisi dari klasifikasi itu sendiri merujuk pada sebuah proses menempatkan atau memprediksi kategori target untuk setiap *instans* atau baris data berdasarkan pola-pola yang telah dipelajari dari data historis sebelumnya. Secara sederhana, komputer diajarkan menggunakan sekumpulan data latih yang sudah memiliki label atau jawaban yang benar, sehingga komputer dapat memahami karakteristik unik dari setiap kelompok. Mengacu pada penjelasan Ahmed (2024), model klasifikasi bekerja dengan cara memetakan hubungan antara variabel input atau fitur-fitur yang ada dengan variabel output yang bersifat diskret atau kategorikal [5]. Ketika model tersebut berhadapan dengan data baru yang belum pernah dilihat sebelumnya, model akan menganalisis fitur-fitur data tersebut dan secara otomatis memasukkannya ke dalam salah satu kategori atau kelas yang paling sesuai.

A Perbedaan klasifikasi vs *clustering* vs regresi

Untuk memahami klasifikasi secara utuh, penting untuk melihat perbedaannya secara mendasar dengan teknik *machine learning* populer lainnya, seperti *clustering* (penggugusan) dan regresi. Perbedaan utama antara klasifikasi dengan *clustering* terletak pada keberadaan label atau target pada data yang digunakan. Jika klasifikasi merupakan pembelajaran terawasi karena jalannya pemodelan dipandu oleh label yang sudah jelas, maka *clustering* dikategorikan sebagai *unsupervised learning* atau pembelajaran tidak terawasi. Dalam *clustering*, data sama sekali tidak memiliki label kategori sejak awal, sehingga tugas algoritma adalah mencari kesamaan karakteristik di antara data tersebut secara mandiri dan mengelompokkannya ke dalam beberapa kelompok atau gugusan baru tanpa arahan target yang pasti. Sementara itu, perbedaan antara klasifikasi dengan regresi terletak pada tipe data dari output atau label yang diprediksi. Meskipun klasifikasi dan regresi sama-sama merupakan bagian dari pembelajaran terawasi, regresi digunakan untuk memprediksi nilai numerik yang bersifat kontinu atau kontinu-berkelanjutan, seperti memprediksi perkiraan harga rumah, suhu ruangan, atau estimasi pendapatan di masa depan. Sebaliknya, klasifikasi murni digunakan untuk menentukan label yang berbentuk kategori tegas atau diskret yang tidak memiliki nilai antara, seperti memprediksi status kelulusan, tingkat prestasi, atau mendeteksi

apakah sebuah email termasuk spam atau bukan.

B Jenis-jenis klasifikasi (*binary*, *multi-class*)

Dalam implementasinya, model klasifikasi secara garis besar dapat dibagi menjadi beberapa jenis berdasarkan jumlah kelas atau label target yang terlibat dalam proses pengujian, di mana jenis yang paling umum adalah klasifikasi biner (*binary classification*) dan klasifikasi multi-kelas (*multi-class classification*). Klasifikasi biner terjadi apabila label target yang ingin diprediksi hanya memiliki dua kemungkinan pilihan kategori yang saling bertolak belakang. Sejalan dengan analisis dari Fauzan dkk.(2024), contoh klasik dari klasifikasi biner dalam dunia pendidikan adalah prediksi status kelulusan siswa yang hanya dibagi menjadi dua opsi tegas, yaitu lulus atau tidak lulus, atau prediksi performa yang hanya dibatasi pada hasil berhasil atau gagal [15]. Di sisi lain, klasifikasi multi-kelas digunakan apabila label target dalam dataset memiliki lebih dari dua jenis pilihan kategori yang berbeda. Pada jenis klasifikasi multi-kelas ini, algoritma dituntut untuk memiliki kemampuan memisahkan batasan data yang lebih kompleks karena *decision boundary* (ruang keputusan) yang dibentuk harus memisahkan beberapa area kelas sekaligus secara bersamaan dengan tetap menjaga tingkat presisi yang optimal.

Penjelasan mengenai konsep klasifikasi multi-kelas tersebut merepresentasikan karakteristik dan struktur data yang digunakan secara riil dalam penelitian skripsi ini. Fokus utama dari penelitian ini tidak menggunakan pendekatan biner yang mengotak-atik status kelulusan secara sederhana, melainkan menerapkan model klasifikasi multi-kelas untuk memetakan tingkat kemampuan siswa ke dalam tiga kelompok spesifik, yaitu Lemah, Sedang, dan Pintar. Berdasarkan temuan dari Hidayatullah dkk.(2022), pengelompokan tingkat performa akademik ke dalam beberapa jenjang kelas yang lebih spesifik sangat efektif untuk memberikan gambaran peta kemampuan siswa secara utuh bagi institusi sekolah [9]. Pelabelan tiga kelas dalam penelitian ini diderivasi langsung secara objektif dari data nilai akhir (*grade*) siswa yang ada di dataset, di mana kategori Lemah mencakup representasi siswa dengan capaian *grade D* dan *F* kategori Sedang untuk siswa dengan capaian *grade C*, serta kategori Pintar untuk siswa dengan capaian *grade A* dan *B*. Mengacu pada pendapat Kotjek dkk.(2025), pembagian kemampuan ke dalam tiga spektrum kelas yang komprehensif ini menjadi fondasi yang sangat kuat dan relevan untuk menghasilkan sistem

penunjang keputusan pengajaran, karena sistem dapat mendeteksi kelompok siswa yang membutuhkan perhatian khusus secara otomatis sekaligus membedakan strategi penanganan tindakan belajar yang tepat untuk tiap tingkatan profil siswa tersebut [4].

2.1.3 Algoritma *K-Nearest Neighbor* (KNN)

A Definisi dan Konsep Dasar *K-Nearest Neighbor* (KNN)

Algoritma *K-Nearest Neighbor* atau yang lebih dikenal dengan singkatan KNN merupakan salah satu metode klasifikasi yang sangat populer dan fundamental dalam dunia kecerdasan buatan serta penambangan data pendidikan. Definisi dari *K-Nearest Neighbor* itu sendiri merujuk pada sebuah algoritma *supervised learning* (pembelajaran terawasi) yang bekerja dengan prinsip dasar mencari kedekatan jarak antara objek data baru yang belum diketahui kelasnya dengan objek-objek data historis yang sudah ada di dalam kumpulan *training data* (data latih). Mengacu pada penelitian yang dilakukan oleh Wahyuni dan Kurniawan (2025), konsep dasar dari algoritma ini bertumpu pada sebuah premis sederhana bahwa objek-objek data yang memiliki karakteristik atau fitur yang serupa akan cenderung berada di dalam ruang geometris atau ruang keputusan yang berdekatan satu sama lain [1]. Di dalam struktur kerja *machine learning*, KNN dikategorikan sebagai salah satu algoritma *memory-based* (berbasis memori) atau *instance-based learning* (Pembelajaran berbasis contoh). Karakteristik ini muncul karena algoritma KNN tidak melakukan proses pembelajaran untuk membangun sebuah model matematis yang eksplisit atau fungsi keputusan global selama fase pelatihan data berlangsung, melainkan menyimpan seluruh data latih di dalam memori sistem dan baru akan melakukan kalkulasi komputasi yang intensif ketika ada data pengujian baru yang masuk dan membutuhkan prediksi label kelas.

Karakteristik unik lain dari *K-Nearest Neighbor* yang menjadikannya sangat fleksibel untuk berbagai macam kasus klasifikasi adalah sifatnya yang dikenal sebagai *lazy learner* atau pembelajar malas. Istilah ini tidak berarti algoritma tersebut tidak efisien, melainkan merujuk pada strategi penundaan proses generalisasi data di mana algoritma ini sengaja menunda tahap pembentukan pola hubungan fitur hingga waktu pengujian (testing data) benar-benar dieksekusi terhadap data baru. Sejalan dengan pendapat dari Hidayutullah dkk.(2022), karena algoritma KNN tidak membuat asumsi fungsional atau perkiraan awal mengenai

bentuk distribusi dan hubungan antara fitur input dengan label target, maka algoritma ini mampu menangani struktur data yang kompleks, tidak linear, serta multidimensional secara alami tanpa memerlukan manipulasi data yang rumit [9]. Konsep dasar kemiripan dalam KNN diwujudkan melalui interpretasi posisi data sebagai titik-titik koordinat di dalam sebuah ruang fitur berdimensi banyak, di mana setiap dimensi grafiknya mewakili satu fitur atau variabel dari instans data tersebut. Berdasarkan temuan dari Jasri dkk.(2024), tingkat kemiripan atau kesamaan di antara instans data tersebut dihitung secara kuantitatif dengan menggunakan fungsi metrik jarak tertentu [16]. Melalui pendekatan matematis ini, konsep ”tetangga” didefinisikan sebagai sekumpulan data historis yang memiliki nilai jarak paling kecil atau paling dekat dengan posisi koordinat data baru yang sedang diuji, sehingga data-data tetangga terdekat tersebut dianggap memiliki perilaku atau kategori yang sejenis.

Dalam konteks penelitian klasifikasi kemampuan siswa ini, konsep dasar kesamaan geometris yang dimiliki oleh algoritma KNN menjadi sangat relevan dan kuat ketika diterapkan pada fitur gabungan yang melibatkan aspek akademik dan non-kognitif. Ketika data seorang siswa baru dimasukkan ke dalam sistem, model KNN tidak akan melihat siswa tersebut secara terisolasi, melainkan memproyeksikan seluruh fiturnya, mulai dari nilai ujian mata pelajaran, persentase kehadiran, durasi tidur, tingkat kecemasan, hingga waktu penggunaan gawai ke dalam ruang fitur multidimensi yang telah diisi oleh data 1.000 siswa historis dari dataset. Merujuk pada pemikiran dari Muhaimin dkk. (2024), efektivitas KNN dalam pengelompokan performa siswa terletak pada kemampuannya mengenali pola kemiripan perilaku keseharian dan performa belajar yang saling berinteraksi secara simultan di dunia nyata [14]. Sebagai contoh, jika seorang siswa baru memiliki pola perilaku berupa jam tidur yang minim dan kecemasan ujian yang tinggi, maka dalam ruang koordinat, titik data siswa tersebut akan secara otomatis mendekat dan berkumpul dengan titik-titik data siswa historis lain yang memiliki kebiasaan serupa. Jika mayoritas siswa historis yang berada di lingkungan koordinat terdekat tersebut memiliki label kelas ”Lemah”, maka berdasarkan konsep voting mayoritas dari sejumlah ’K’ tetangga terdekat, siswa baru tersebut akan diklasifikasikan ke dalam kategori yang sama. Hal ini menunjukkan bahwa konsep dasar KNN tidak hanya bertumpu pada kalkulasi angka matematika murni, melainkan mampu merefleksikan kesamaan sosiologis dan psikologis siswa ke dalam sebuah sistem keputusan personal yang terukur secara ilmiah.

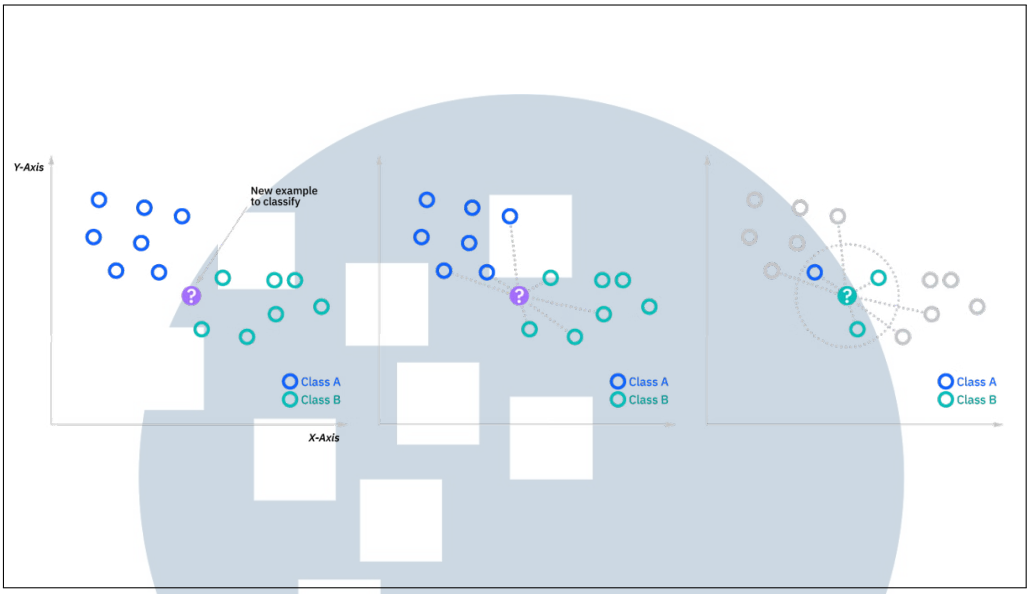
B Cara Kerja *Algoritma K-Nearest Neighbor* (KNN)

Proses operasional dari algoritma *K-Nearest Neighbor* dalam menentukan kategori kemampuan siswa berjalan secara bertahap melalui empat langkah utama yang terstruktur secara logis dan matematis. Tahap paling awal dalam alur kerja ini dimulai dengan proses input data baru, yaitu memasukkan profil siswa yang belum memiliki label kelas ke dalam sistem. Data baru ini bertindak sebagai sebuah titik koordinat baru yang membawa nilai untuk seluruh variabel pendukung yang telah ditentukan oleh peneliti. Mengacu pada struktur penelitian ini, data input tersebut terdiri dari kombinasi fitur akademik, seperti nilai mata pelajaran sebelumnya serta persentase kehadiran, dan fitur non-kognitif yang meliputi durasi tidur, tingkat kecemasan ujian, hingga waktu penggunaan gawai. Sejalan dengan pendapat Muhaimin dkk. (2024), akurasi model pada tahap input awal ini sangat bergantung pada kelengkapan dan kebersihan nilai fitur yang dimasukkan, karena setiap variabel perilaku dan nilai tersebut akan langsung memengaruhi posisi geometris data siswa di dalam ruang komputasi [14].

Setelah data baru berhasil dimasukkan dan dipetakan posisinya, langkah kedua yang dieksekusi oleh sistem adalah melakukan perhitungan jarak antara titik data baru tersebut dengan seluruh data latih yang tersimpan di dalam memori. Algoritma KNN akan memindai total 1.000 baris data siswa historis yang ada dalam *dataset training* dan mengukur tingkat kemiripan di antara mereka secara individual menggunakan rumus metrik jarak tertentu. Berdasarkan temuan dari Wahyuni dan Kurniawan (2025), proses kalkulasi ini melibatkan perhitungan selisih nilai untuk setiap dimensi fitur yang ada, di mana perbedaan nilai numerik dari jam belajar harian, skor motivasi, maupun nilai rapor akan diakumulasikan untuk menghasilkan satu nilai jarak tunggal yang mutlak [1]. Melalui tahapan ini, komputer secara matematis mengonversi kemiripan perilaku nyata dan capaian nilai siswa menjadi sebuah representasi angka jarak, di mana semakin kecil nilai jarak yang dihasilkan, maka tingkat kemiripan antara siswa baru tersebut dengan siswa historis tertentu akan dinilai semakin tinggi. Langkah berikutnya setelah seluruh nilai jarak komparatif berhasil dihitung adalah melakukan pemilihan sejumlah K tetangga terdekat. Pada tahap ketiga ini, sistem akan mengurutkan seluruh hasil perhitungan jarak dari nilai yang paling kecil hingga nilai yang paling besar, lalu menyaring data untuk mengambil sejumlah instans data historis teratas yang posisinya paling dekat dengan data baru sesuai dengan nilai konstanta K yang telah ditentukan sebelumnya. Mengacu pada pengujian yang dilakukan oleh Hidayatullah

dkk.(2022), penentuan jumlah *K* ini memegang peranan yang sangat penting karena *K* menentukan berapa banyak sampel tetangga yang akan dijadikan acuan untuk mengambil keputusan akhir [9]. Jika nilai *K* yang ditetapkan adalah lima, maka sistem hanya akan mengambil lima data siswa historis yang memiliki karakteristik paling mirip dengan profil siswa baru tersebut, lalu mengabaikan sisa data lainnya yang memiliki nilai jarak lebih jauh dalam ruang koordinat.

Pada tahap akhir dari operasional algoritma ini, sistem akan menjalankan mekanisme voting mayoritas untuk menentukan keputusan final mengenai label kelas yang akan diberikan kepada data siswa baru tersebut. Komputer akan memeriksa label kelas yang melekat pada sejumlah *K* tetangga terdekat yang sudah terpilih pada tahap sebelumnya, di mana dalam konteks penelitian ini label tersebut terdiri dari tiga kategori, yaitu Lemah, Sedang, atau Pintar. Berdasarkan temuan eksperimen dari Febrian dkk.(2025), proses penentuan keputusan ini bekerja layaknya pemungutan suara secara demokratis, di mana label kelas yang paling banyak muncul atau dominan di antara tetangga terdekat tersebut secara otomatis akan diadopsi sebagai label kelas bagi siswa baru. Sebagai contoh, apabila dari lima tetangga terdekat yang terpilih terdapat tiga siswa berlabel Pintar, satu siswa berlabel Sedang, dan satu siswa berlabel Lemah, maka suara mayoritas dimenangkan oleh kelas Pintar, sehingga sistem secara cerdas dan objektif menyimpulkan bahwa siswa baru tersebut diklasifikasikan ke dalam kategori Pintar yang kemudian menjadi dasar kuat untuk merumuskan rekomendasi pembelajaran personal yang sesuai.



Gambar 2.1. Prosedur Tahapan Klasifikasi Spasial pada Algoritma K-Nearest Neighbor (Sumber: https://www.ibm.com/id-id/think/topics/knn)

Visualisasi mekanistik dari proses pengelompokan spasial ini disajikan secara berurutan melalui tiga panel utama pada Gambar 2.1. Pada panel pertama, diperlihatkan masuknya sampel data uji baru yang belum memiliki label kelas (direpresentasikan oleh simpul berwarna ungu dengan simbol tanda tanya) ke dalam ruang fitur dua dimensi yang dibentuk oleh sumbu X dan sumbu Y. Selanjutnya, panel kedua mengilustrasikan proses komputasi pada langkah kedua dan ketiga, yaitu ketika model KNN secara simultan menghitung metrik jarak garis lurus (*Euclidean distance*) dari koordinat data baru tersebut ke setiap titik sampel pada data latih historis di sekitarnya, yang ditunjukkan oleh sebaran garis putus-putus.

Pada panel ketiga, ditunjukkan penentuan radius lingkaran imajiner berdasarkan hiperparameter K yang telah ditetapkan untuk memilih sejumlah tetangga terdekat. Di dalam radius tersebut, model melakukan proses *majority voting* untuk menentukan label kelas yang dominan (pada ilustrasi ini didominasi oleh Class B berwarna hijau). Dengan demikian, simbol tanda tanya pada data baru secara otomatis berubah warna dan mengadopsi keputusan kelas Class B sebagai keluaran prediksi akhir model secara objektif.

C Rumus Jarak *Euclidean Distance*

Pengukuran kemiripan atau kedekatan antar-instans data di dalam algoritma *K-Nearest Neighbor* dilakukan dengan menghitung jarak geometris antara dua titik

koordinat di dalam ruang multidimensi. Salah satu metrik perhitungan jarak yang paling mendasar, populer, dan digunakan secara luas dalam aplikasi penambangan data adalah *Euclidean Distance*. Secara definisi, *Euclidean Distance* merupakan rumus matematika yang digunakan untuk mengukur jarak garis lurus terpendek antara dua buah titik pada ruang berdimensi banyak (n-dimensi). Konsep ini sebenarnya diturunkan secara alami dari Teorema Pythagoras yang biasa digunakan untuk mencari sisi miring segitiga siku-siku, namun diperluas skalanya agar mampu menghitung selisih koordinat pada banyak variabel sekaligus. Mengacu pada penelitian dari Hidayatullah dkk.(2022), pemilihan *Euclidean Distance* sebagai metrik kedekatan dalam KNN dinilai sangat efektif karena mampu menghasilkan visualisasi jarak fisik yang konsisten, objektif, dan bernilai mutlak di dalam ruang fitur komputasi [9].

Secara matematis, persamaan kalkulasi *Euclidean Distance* untuk menghitung jarak antara titik data pengujian (p) dan titik data pelatihan (q) dinyatakan melalui persamaan berikut:

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} \tag{2.1}$$

Di dalam persamaan tersebut, simbol $d(p,q)$ mewakili nilai total jarak skalar yang dihasilkan antara objek p dan objek q. Simbol $p_1, p_2, \ldots, p_n$ menunjukkan nilai fitur atau variabel ke-1 hingga ke-n yang dimiliki oleh data siswa baru yang sedang diuji. Sementara itu, simbol $q_1, q_2, \ldots, q_n$ merupakan nilai fitur ke-1 hingga ke-n dari baris data siswa historis yang ada di dalam kumpulan data latih (*training data*). Alur operasional dari rumus ini bekerja dengan cara mencari selisih angka antara fitur data baru dengan fitur data lama untuk setiap variabel yang sama (misalnya nilai p_1 dikurangi q_1). Hasil pengurangan tersebut kemudian dipangkatkan dua untuk menghilangkan nilai negatif, sehingga semua selisih berubah menjadi angka positif. Setelah seluruh variabel selesai dikurangkan dan dipangkatkan, komputer akan menjumlahkan total keseluruhan nilai tersebut dan mengeksekusi operasi akar kuadrat ($\sqrt{\ldots}$) sebagai tahap akhir untuk mendapatkan satu angka penanda jarak yang mutlak.

Di dalam konteks penelitian klasifikasi kemampuan siswa ini, rumus *Euclidean Distance* memegang peranan krusial sebagai mesin penggerak utama yang mengonversi data gabungan akademik dan non-kognitif menjadi angka kemiripan profil yang konkret. Ketika sistem menghitung jarak antara seorang

siswa baru dengan salah satu dari 1.000 data siswa di dataset, setiap dimensi atau suku di dalam tanda akar (n-dimensi) akan mewakili satu kolom fitur yang ada. Sebagai contoh nyata, suku pertama melambangkan selisih fitur akademik berupa nilai matematika sebelumnya (*math_prev_score*), suku kedua melambangkan selisih tingkat kehadiran (*attendance_percentage*), suku ketiga melambangkan selisih fitur non-kognitif seperti durasi tidur (*sleep_hours*), suku keempat melambangkan selisih skor kecemasan ujian (*exam_anxiety_score*), dan begitu seterusnya hingga seluruh fitur gabungan selesai dihitung. Sejalan dengan pendapat dari Kotjek dkk.(2025), karena rumus ini menggabungkan variabel performa nilai dan pola perilaku psikologis sekaligus, perbedaan tipis pada gaya hidup siswa akan langsung memengaruhi hasil penjumlahan di dalam akar [4]. Siswa baru yang memiliki kombinasi nilai akademik rendah serta tingkat kecemasan tinggi secara otomatis akan menghasilkan nilai jarak $d(p,q)$ yang sangat kecil (sangat dekat) ketika diselisihkan dengan kelompok siswa historis yang berlabel "Lemah". Hal ini membuktikan bahwa *Euclidean Distance* mampu menyatukan berbagai jenis variabel yang berbeda ke dalam satu ruang keputusan geometris yang adil dan terukur secara empiris.

D Pemilihan Nilai *K*

Nilai *K* (jumlah tetangga terdekat) adalah *hyperparameter* krusial dalam algoritma KNN yang harus ditentukan secara manual sebelum pemodelan dimulai. Penentuan nilai *K* ini sangat memengaruhi performa akhir model karena menentukan tingkat akurasi dan kemampuan generalisasi sistem.

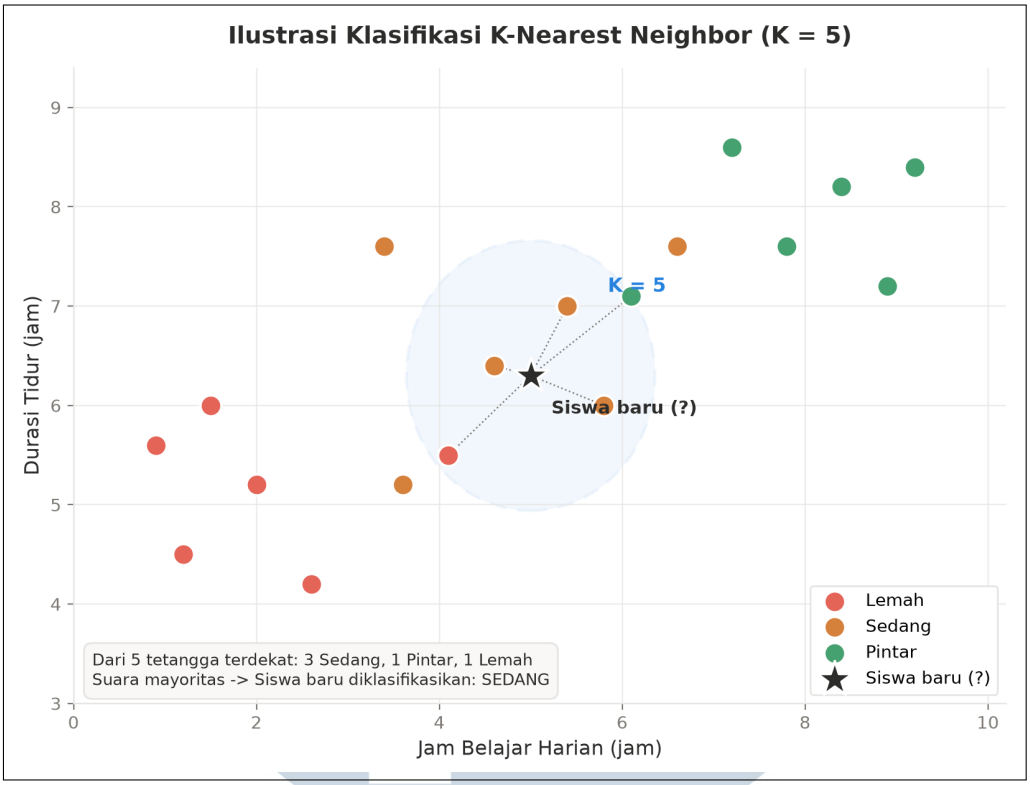
Format dampak dan penentu nilai *K* dijabarkan sebagai berikut:

1. Jika nilai *K* Terlalu Kecil (memicu *Overfitting*):
   Ketika nilai *K* yang dipilih terlalu kecil, model klasifikasi akan sangat menjadi sensitif terhadap titik data pencilan (*outliers*) maupun *noise* yang terdapat di dalam dataset. Kondisi ini menyebabkan garis keputusan (*decision boundary*) yang terbentuk menjadi terlalu kompleks dan kaku demi mengikuti variasi acak dari data latih. Berdasarkan temuan dari Hidayatullah dkk. (2022), nilai parameter yang terlalu rendah membuat akurasi model tampak sempurna pada data latihan (training data), namun kinerjanya akan anjlok drastis saat diuji dengan data baru (testing data) yang belum pernah dikenali sebelumnya [9]. Dalam kasus nyata, nilai *K* yang terlalu kecil membuat model rawan salah mengklasifikasikan kategori kemampuan siswa

hanya karena ada satu data historis terdekat yang perilakunya kebetulan menyimpang dari tren kelompok aslinya.

2. Jika Nilai *K* Terlalu Besar (Memicu *Underfitting*):
   Sebaliknya, apabila peneliti menetapkan nilai *K* yang terlalu besar, garis keputusan model akan menjadi terlalu mulus (*oversmoothed*) sehingga kehilangan kemampuan untuk mengenali pola-pola lokal yang spesifik di dalam data. Jika hal ini terjadi, keputusan akhir model sepenuhnya akan didominasi oleh kelompok atau kelas mayoritas yang memiliki populasi paling banyak di dalam *dataset*. Mengacu pada penelitian dari Novianto dkk. (2023), kondisi ini menyebabkan algoritma gagal menangkap perbedaan esensial antara fitur akademik dan non-kognitif [10]. Akibatnya, kelompok kecil siswa yang sebenarnya berada dalam kategori Lemah berisiko salah terlabeli sebagai Sedang atau Pintar hanya karena kalah jumlah pemungutan suara (*voting*) dari tetangga di sekitarnya.

3. Pencarian Nilai *K* yang Optimal (Melalui Eksperimen):
   Untuk mengatasi dilema tersebut, pencarian nilai *K* yang optimal wajib dilakukan melalui serangkaian prosedur pengujian langsung pada *dataset* yang digunakan karena performanya sangat bergantung pada karakteristik data. Sejalan dengan eksperimen komparatif yang dilakukan oleh Amri dkk. (2023), nilai *K* harus dicari secara empiris dengan menguji berbagai variasi angka ganjil (seperti 1, 3, 5, 7, dan seterusnya) guna menghindari terjadinya hasil pemungutan suara yang seimbang (*tie-voting*) [11]. Berdasarkan temuan dari Febrian dkk. (2025), variasi nilai *K* tersebut nantinya dievaluasi menggunakan metrik akurasi, *precision*, *recall*, dan *F1-score* untuk menemukan titik angka *K* yang paling seimbang dan memberikan performa terbaik [17].



Gambar 2.2. Ilustrasi Klasifikasi Spasial Siswa Baru dengan Nilai K = 5

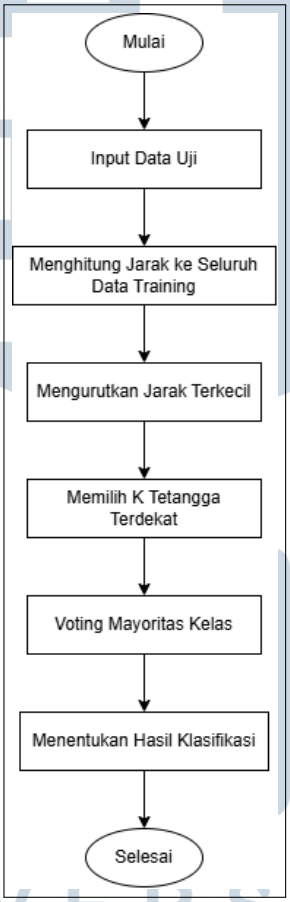
Sebagai bentuk implementasi setelah rumus *Euclidean Distance* dijabarkan, contoh konkret proses perhitungan jarak dan pengelompokan spasial dengan $K = 5$ disajikan pada Gambar 2.2. Ilustrasi tersebut memetakan sebaran data pada ruang dua dimensi berdasarkan dua fitur non-kognitif, yaitu Jam Belajar Harian (jam) pada sumbu X dan Durasi Tidur (jam) pada sumbu Y. Titik-titik data siswa historis direpresentasikan menggunakan tiga warna sesuai label kelas aktual, yakni merah untuk kategori Lemah, jingga untuk kategori Sedang, dan hijau untuk kategori Pintar. Sementara itu, satu profil siswa baru yang belum terklasifikasi diproyeksikan ke ruang fitur yang sama dan ditandai dengan ikon bintang berwarna hitam.

Melalui penerapan metrik jarak menggunakan persamaan *Euclidean Distance*, model KNN menghitung jarak antara data baru tersebut dan seluruh sampel historis, kemudian memilih lima tetangga terdekat yang berada dalam radius lingkaran imajiner berwarna biru. Berdasarkan hasil penapisan pada radius tersebut, komposisi tetangga yang diperoleh terdiri atas tiga siswa kategori Sedang, satu siswa kategori Pintar, dan satu siswa kategori Lemah. Selanjutnya, melalui mekanisme *majority voting*, kategori Sedang menjadi kelas dengan jumlah suara terbanyak. Dengan demikian, sistem menetapkan bahwa siswa baru tersebut

diklasifikasikan ke dalam kategori Sedang.

E Alur Kerja Algoritma KNN

Alur kerja sistem yang dikembangkan pada penelitian ini digambarkan melalui diagram blok pada Gambar 2.3. Diagram tersebut menunjukkan tahapan pemrosesan data mulai dari proses masukan dataset hingga menghasilkan rekomendasi pembelajaran personal. Secara umum, alur kerja sistem terdiri atas lima tahapan utama sebagai berikut.



Gambar 2.3. Diagram Alur Kerja dan Arsitektur Konseptual Algoritma KNN

1. **Tahap Input Dataset dan Seleksi Fitur**

   Proses diawali dengan memasukkan dataset ke dalam sistem. Selanjutnya dilakukan seleksi fitur untuk mempertahankan atribut yang relevan terhadap proses klasifikasi serta menghapus atribut yang tidak diperlukan. Berdasarkan penelitian Azizah dkk. (2022), proses seleksi fitur juga bertujuan menghindari terjadinya *data leakage*, yaitu kondisi ketika informasi

yang berkaitan langsung dengan label target ikut digunakan selama proses pelatihan sehingga dapat menghasilkan performa model yang tidak mencerminkan kondisi sebenarnya [12].

2. **Tahap Pra-pemrosesan Data (*Data Preprocessing*)**
   Setelah fitur ditentukan, data diproses agar siap digunakan oleh algoritma KNN. Tahapan ini meliputi *Label Encoding* untuk mengubah data kategorikal menjadi bentuk numerik serta normalisasi menggunakan metode *MinMaxScaler*. Mengacu pada penelitian Wahyuni dan Kurniawan (2025), normalisasi diperlukan agar seluruh fitur berada pada rentang nilai yang sama sehingga tidak ada atribut yang memiliki pengaruh lebih besar dalam perhitungan jarak [1].

3. **Tahap Partisi Data (*Data Splitting*)**
   Data yang telah dipra-pemroses kemudian dibagi menjadi data latih (*training data*) dan data uji (*testing data*) menggunakan metode *stratified split*. Pembagian ini dilakukan agar proporsi setiap kelas tetap terjaga pada kedua kelompok data, sehingga proses pelatihan dan evaluasi model dapat dilakukan secara lebih representatif.

4. **Tahap Pelatihan dan Evaluasi Model KNN**
   Data latih digunakan untuk membangun model *K-Nearest Neighbor* (KNN) dengan menghitung kedekatan antar data menggunakan jarak *Euclidean*. Setelah model terbentuk, data uji digunakan untuk menghasilkan prediksi dan mengevaluasi performa model menggunakan metrik seperti akurasi, *precision*, *recall*, *F1-score*, serta *confusion matrix*. Tahap ini bertujuan untuk mengetahui kemampuan model dalam mengklasifikasikan kemampuan siswa.

5. **Tahap *Rule Engine* dan Output Rekomendasi**
   Hasil klasifikasi dari model KNN selanjutnya menjadi masukan bagi *knowledge-based rule engine*. Pada tahap ini, sistem membandingkan nilai setiap indikator siswa dengan nilai rata-rata (*threshold*) pada kelompok hasil klasifikasinya. Berdasarkan hasil perbandingan tersebut, sistem menghasilkan rekomendasi pembelajaran yang sesuai dengan kondisi masing-masing siswa. Mengacu pada penelitian Atalla dkk. (2023), integrasi antara model *machine learning* dan aturan berbasis *If-Then* memungkinkan hasil klasifikasi diubah menjadi rekomendasi yang lebih mudah diterapkan sebagai dasar pengambilan keputusan [3].

F Kelebihan dan Kekurangan Algoritma *K-Nearest Neighbor* (KNN)

Sebagai salah satu algoritma yang matang dalam bidang *machine learning*, *K-Nearest Neighbor* memiliki karakteristik operasional yang khas. Karakteristik ini melahirkan sejumlah keunggulan sekaligus batasan teknis tertentu yang perlu dipahami secara mendalam agar implementasinya pada data pendidikan dapat berjalan secara optimal.

F.1 Kelebihan Algoritma *K-Nearest Neighbor* (KNN):

Kelebihan utama dari algoritma KNN terletak pada kesederhanaan konsepnya yang membuatnya sangat mudah untuk diimplementasikan ke dalam program tanpa memerlukan fase pelatihan model yang rumit. Berdasarkan pendapat dari Hidayatullah dkk. (2022), KNN merupakan algoritma non-parametrik, yang berarti metode ini tidak memerlukan asumsi teoretis apa pun mengenai bentuk distribusi data dasar yang digunakan [9]. Keunggulan ini membuat KNN sangat tangguh dan adaptif ketika dihadapkan pada data multidimensi riil yang memiliki pola hubungan tidak linear. Sejalan dengan pendapat dari Muhaimin dkk. (2024), KNN juga terbukti sangat efektif untuk menangani kasus klasifikasi multi-kelas (*multi-class classification*), seperti pengelompokan kemampuan belajar siswa ke dalam tiga tingkatan kelas yang berbeda secara simultan [14]. Selain itu, karena seluruh data latih disimpan langsung di dalam memori, model KNN dapat diperbarui dengan sangat mudah dan dinamis tanpa harus melatih ulang seluruh sistem dari awal setiap kali ada data siswa baru yang masuk.

F.2 Kekurangan Algoritma *K-Nearest Neighbor* (KNN):

Di balik kesederhanaannya, KNN memiliki kekurangan utama berupa tingginya beban komputasi dan memori pada saat fase pengujian (*testing time*), terutama jika *dataset* yang diolah berukuran sangat besar. Hal ini terjadi karena komputer dipaksa untuk menghitung jarak geometris satu per satu ke seluruh baris data pelatihan setiap kali ingin memprediksi satu data baru. Mengacu pada keterbatasan yang diidentifikasi oleh Azizah dkk. (2022), KNN juga memiliki sensitivitas yang sangat tinggi terhadap keberadaan data pencilan (*outliers*) atau data *noise* yang dapat mengacaukan penentuan *voting* mayoritas tetangga. Selain itu, algoritma ini sangat sensitif terhadap perbedaan skala atau satuan antar fitur yang digunakan. Jika fitur akademik seperti nilai matematika berskala puluhan

(0–100) langsung dihitung bersama fitur non-kognitif seperti durasi tidur yang hanya berskala satuan jam (1–10), maka fitur dengan skala angka yang lebih besar akan mendominasi perhitungan jarak dan membuat prediksi menjadi tidak adil. Oleh karena itu, penerapan teknik transformasi data seperti normalisasi *MinMaxScaler* menjadi tahapan wajib yang tidak boleh dilewatkan agar seluruh fitur gabungan memiliki bobot pengaruh yang setara di dalam ruang keputusan.

2.1.4 Algoritma *Decision Tree*

Algoritma *Decision Tree* (Pohon Keputusan) merupakan salah satu metode klasifikasi *supervised learning* yang memetakan *dataset* ke dalam struktur visual menyerupai pohon keputusan untuk menghasilkan aturan-aturan klasifikasi yang terstruktur.

A Definisi dan Konsep Dasar

Secara konseptual, *Decision Tree* mengubah sekumpulan data mentah menjadi aturan keputusan berbentuk hierarki yang merepresentasikan hubungan logis antar fitur prediktor. Mengacu pada penelitian Fauzan dkk. (2024), algoritma ini bekerja secara efektif dalam mengekstrak pola dari data akademik untuk memetakan capaian belajar siswa berdasarkan kombinasi nilai mata pelajaran [15]. Sejalan dengan pendapat dari Muraina dkk. (2022), fleksibilitas metode ini memungkinkannya untuk mengolah data bertipe numerik maupun kategorikal sekaligus menyajikan alur keputusan yang transparan sehingga memudahkan tenaga pendidik dalam menginterpretasikan hasil klasifikasi model [18].

B Cara Kerja (*Splitting, Node, Leaf, Branch*):

Cara kerja algoritma ini didasarkan pada strategi pembagian data secara rekursif (*recursive partitioning*) dari atas ke bawah. Proses dimulai pada *root node* (simpul akar) yang merepresentasikan keseluruhan *dataset* awal, di mana atribut paling informatif dipilih untuk melakukan *splitting* (pemecahan data) menjadi beberapa cabang (*branch*). Setiap *branch* bertindak sebagai jalur penghubung yang mengarahkan data menuju *internal node* (simpul internal) untuk mengevaluasi kondisi fitur berikutnya secara spesifik. Proses pemisahan ini akan terus berjalan secara bertahap hingga mencapai titik akhir di *leaf node* (simpul daun). Berdasarkan pemodelan struktur dari Novianto dkk. (2023), *leaf node* merupakan bagian

terminal yang sudah tidak bisa dipecah lagi karena telah memuat label keputusan atau kelas target akhir dari objek data yang diuji [10].

C Konsep *Gini Impurity*

Dalam program pemodelan ini, penentuan atribut terbaik untuk melakukan *splitting* dikonfigurasi menggunakan kriteria *Gini Impurity*. Metrik ini digunakan untuk mengukur tingkat ketidakmurnian atau heterogenitas elemen data di dalam sebuah simpul, dengan rentang nilai operasional berkisar antara 0 (simpul murni sempurna di mana seluruh data berasal dari satu kelas) hingga maksimal 0,5 untuk klasifikasi biner seimbang. Persamaan matematis untuk menghitung nilai *Gini Impurity* dinyatakan sebagai berikut:

$$Gini = 1 - \sum_{i=1}^{n} (p_i)^2 \tag{2.2}$$

Di mana p_i melambangkan proporsi nilai probabilitas kemunculan data dari kelas ke-i di dalam simpul tersebut. Berdasarkan temuan dari Febrian dkk. (2025), proses pemilihan atribut terbaik untuk pembagian cabang ini secara terstruktur bertujuan untuk mengurangi dimensi data berlebih, menghindari kelebihan informasi yang tidak perlu, serta mengoptimalkan efisiensi kinerja model [17].

D Kelebihan dan Kekurangan

Kelebihan utama dari algoritma *Decision Tree* terletak pada kemampuannya menghasilkan model yang transparan, mudah dipahami dalam bentuk aturan logis sederhana, serta tidak membutuhkan proses pra-pemrosesan data yang rumit seperti normalisasi skala nilai fitur. Mengacu pada temuan Fatah dan Navizah (2026), keunggulan interpretasi visual ini sangat membantu institusi pendidikan dalam merancang kebijakan strategis karena pola dan faktor penentu keputusan tersaji secara intuitif [19]. Namun, keterbatasan utama dari metode ini adalah kerentanannya yang tinggi terhadap masalah *overfitting*, terutama jika struktur pohon dibiarkan tumbuh terlalu dalam dan kompleks demi mengikuti variasi acak data latih. Berdasarkan analisis dari Kord dkk. (2025), pohon keputusan juga bersifat tidak stabil (*unstable classifier*), di mana perubahan kecil pada sebaran data latih dapat mengubah struktur pohon secara drastis dan memengaruhi akurasi

generalisasinya [13].

E Posisi di Penelitian: Baseline Pembanding Pertama

Di dalam penelitian skripsi ini, algoritma *Decision Tree* ditempatkan secara strategis sebagai baseline pembanding pertama untuk mengevaluasi kinerja dari algoritma utama yang diusulkan, yaitu *K-Nearest Neighbor* (KNN). Melalui skema pengujian komparatif dengan *dataset* yang sama, metrik evaluasi seperti akurasi, *precision*, *recall*, dan *F1-score* dari pohon keputusan akan dijadikan standar tolak ukur dasar (*benchmark*). Evaluasi ini bertujuan untuk membuktikan secara empiris apakah pendekatan klasifikasi berbasis kedekatan spasial antar tetangga (KNN) mampu memberikan performa dan kestabilan yang lebih unggul dalam memetakan potensi akademik siswa dibandingkan pendekatan berbasis aturan hierarki terstruktur milik *Decision Tree*.

2.1.5 Algoritma *Naive Bayes*

Algoritma *Naive Bayes* merupakan salah satu metode klasifikasi berbasis statistika yang memanfaatkan *Teorema Bayes* untuk menghitung probabilitas keanggotaan suatu objek ke dalam kelas tertentu.

A Definisi dan Konsep Dasar

Naive Bayes adalah sebuah metode klasifikasi data statistik yang dapat memprediksi probabilitas terkait keanggotaan suatu kelas. Mengacu pada penelitian Fauzan dkk. (2024), penerapan metode klasifikasi berbasis probabilitas ini sangat efektif dalam mengevaluasi prestasi belajar siswa karena mampu mengolah sekumpulan atribut data akademik secara efisien untuk menghasilkan keputusan yang cepat [15]. Lalu, berdasarkan temuan dari Novianto dkk. (2023), algoritma ini bekerja dengan menghitung nilai probabilitas bersyarat dari setiap kelas target terhadap kombinasi fitur yang ada, lalu memilih kelas dengan nilai probabilitas tertinggi sebagai keputusan akhir model [10]

B *Teorema Bayes* secara Sederhana

Dasar pilar teoretis utama dari algoritma ini dibangun di atas *Teorema Bayes* yang mengukur probabilitas terjadinya suatu peristiwa berdasarkan informasi atau

kondisi awal yang relevan. Sejalan dengan pendapat dari Amri dkk. (2023), rumus matematika dalam klasifikasi ini digunakan untuk menghitung probabilitas posterior dari suatu kelas target setelah bukti-bukti dari fitur prediktor dimasukkan ke dalam perhitungan. Persamaan matematis dari Teorema Bayes dirumuskan secara sederhana sebagai berikut:

$$P(H|X) = \frac{P(X|H) \cdot P(H)}{P(X)} \tag{2.3}$$

Di mana $P(H|X)$ merepresentasikan probabilitas posterior dari hipotesis atau kelas H jika diberikan bukti fitur X. Nilai $P(X|H)$ melambangkan *likelihood* atau probabilitas bersyarat kemunculan fitur X pada kelas H, sedangkan $P(H)$ bertindak sebagai probabilitas *prior* awal dari kelas H, dan $P(X)$ adalah probabilitas marjinal prediktor dari keseluruhan data.

C Jenis *Naive Bayes* (*Gaussian Naive Bayes*)

Pemilihan varian dari algoritma *Naive Bayes* harus disesuaikan secara presisi dengan tipe karakteristik dari fitur atau variabel yang terdapat di dalam *dataset*. Mengacu pada metodologi pemodelan dalam Ahmed (2024), variasi data di lingkungan pendidikan sering kali memuat variabel numerik kontinu yang merepresentasikan nilai ujian ataupun metrik interaksi siswa [5]. Oleh karena itu, penelitian ini mengonfigurasikan varian *Gaussian Naive Bayes*. Berdasarkan temuan dari Fatah dan Navizah (2026), model *Gaussian Naive Bayes* bekerja dengan asumsi bahwa sebaran data numerik kontinu pada setiap fitur mengikuti kurva distribusi normal (*Gaussian*), di mana nilai rata-rata (*mean*) dan standar deviasi dari masing-masing fitur dihitung guna mengestimasi probabilitas bersyarat objek data baru secara akurat [19].

D Kelebihan dan Kekurangan

Kelebihan utama dari algoritma *Naive Bayes* terletak pada tingkat efisiensi komputasinya yang sangat tinggi, kecepatan proses pelatihan yang singkat, serta kebutuhan memori yang relatif kecil. Berdasarkan temuan dari Fatah dan Navizah (2026), algoritma berbasis probabilitas ini terbukti menawarkan kestabilan dan efektivitas yang jauh lebih baik, terutama ketika diterapkan pada data dengan distribusi kelas yang tidak seimbang (imbalanced data) [19]. Namun, sejalan

dengan pendapat dari Fauzan dkk. (2024), kekurangan mendasar dari metode ini terletak pada asumsi independensi yang naif, di mana algoritma menganggap seluruh fitur prediktor berdiri sendiri dan tidak saling memengaruhi satu sama lain [15]. Pada realitas data pendidikan, variabel akademik siswa kerap kali memiliki ketergantungan implisit, sehingga pengabaian korelasi antar fitur ini berpotensi memengaruhi akurasi prediksi model.

E Posisi di Penelitian: *Baseline* Pembanding Kedua

Di dalam susunan kerangka penelitian skripsi ini, algoritma *Naive Bayes* ditempatkan secara strategis bertindak sebagai baseline pembanding kedua setelah algoritma *Decision Tree*. Mengacu pada skema komparasi yang dijalankan oleh Novianto dkk. (2023), pengujian performa komparatif multi-algoritma sangat diperlukan sebagai standar tolak ukur dasar (*benchmark*) untuk mengonfirmasi keandalan model utama [10]. Melalui skema pengujian ini, kualitas dari model berbasis probabilitas teoretis (*Naive Bayes*) akan diadu langsung dengan model utama *K-Nearest Neighbor* (KNN) menggunakan indikator metrik berupa akurasi, *precision*, *recall*, dan *F1-score*. Evaluasi komparatif ini bertujuan untuk membuktikan secara empiris apakah pendekatan spasial berbasis kedekatan jarak antar-tetangga (KNN) mampu memberikan performa yang lebih adaptif dan presisi dalam memetakan potensi kemampuan belajar siswa.

2.1.6 *Preprocessing Data*

Tahapan *preprocessing data* (pra-pemrosesan data) merupakan fase krusial dalam *Educational Data Mining* (EDM) yang bertujuan untuk membersihkan, mentransformasikan, dan menyiapkan data mentah agar memiliki format yang optimal sebelum dimasukkan ke dalam model *machine learning*.

A Label Encoding

Label Encoding adalah teknik pra-pemrosesan data yang digunakan untuk mengubah data kategorikal atau nilai bertipe teks (*string*) menjadi representasi nilai numerik berupa angka integer diskret (seperti 0, 1, 2, dan seterusnya). Mengacu pada penelitian Novianto dkk. (2023), transformasi data kategorikal menjadi representasi numerik menggunakan *Label Encoding* sangat krusial dilakukan agar

sistem komputer dapat membaca, memproses, dan melakukan kalkulasi matematis terhadap variabel prediktor yang awalnya berbentuk teks [10].

Di dalam dataset yang dipakai, fitur *parent_education_level* (tingkat pendidikan orang tua) dan *study_environment* (lingkungan belajar siswa) merupakan variabel kategorikal non-numerik yang memuat informasi kualitatif penting terkait latar belakang siswa. Berdasarkan temuan dari Arrohman dkk. (2025), data demografis siswa seperti latar belakang pendidikan orang tua memiliki pengaruh implisit terhadap performa akademik siswa, namun fiturnya wajib melalui proses pengkodean terlebih dahulu agar dapat diolah oleh algoritma klasifikasi [20]. Sejalan dengan pendapat dari Amri dkk. (2023), algoritma berbasis spasial dan matematika seperti *K-Nearest Neighbor* (KNN) sama sekali tidak dapat menghitung jarak geometris apabila fitur di dalam radius ruang keputusan masih bertipe teks (*string*). Oleh karena itu, penerapan *Label Encoding* pada variabel *parent_education_level* dan *study_environment* menjadi sebuah kewajiban adaptif di dalam program agar nilai kedekatan antar-karakteristik siswa dapat dihitung secara akurat melalui rumus jarak.

Proses pengubahan nilai teks menjadi angka integer diskret ini berjalan secara berurutan sesuai urutan alfabetis atau urutan kemunculan data uniknya. Sebagai contoh konkret, visualisasi konversi nilai kualitatif pada fitur *parent_education_level* di dalam program diatur sebagai berikut:

1. *Bachelor* akan diubah menjadi nilai numerik 0.
2. *High School* akan diubah menjadi nilai numerik 1.
3. *Master* akan diubah menjadi nilai numerik 2.

B Normalisasi *MinMaxScaler*

Normalisasi *MinMaxScaler* adalah teknik pra-pemrosesan data numerik yang digunakan untuk mentransformasikan nilai-nilai fitur asli ke dalam skala rentang baru yang seragam tanpa mengubah struktur distribusi asli dari data tersebut. Mengacu pada penelitian dari Wahyuni dan Kurniawan (2025), penerapan metode transformasi ini sangat penting dilakukan guna menyetarakan kontribusi dari seluruh variabel prediktor numerik di dalam *dataset* sebelum diproses oleh model klasifikasi [1].

Proses penskalaan ulang nilai ini dilakukan dengan menghitung selisih antara nilai data asli dengan nilai minimum pada fitur tersebut, kemudian dibagi

dengan total rentang antara nilai maksimum dan nilai minimumnya. Persamaan matematis untuk menghitung nilai kriteria *MinMaxScaler* dirumuskan sebagai berikut:

$$X_{\text{norm}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} \tag{2.4}$$

Di mana X_{norm} melambangkan nilai hasil transformasi baru, X merepresentasikan nilai data asli yang akan dikonversi, X_{min} merupakan nilai minimum dari fitur tersebut, dan X_{max} menyatakan nilai maksimum dari fitur terkait.

Penerapan *MinMaxScaler* memegang peranan yang sangat krusial, khususnya pada algoritma yang mengandalkan kalkulasi jarak geometris seperti *K-Nearest Neighbor*. Berdasarkan temuan dari Hidayatullah dkk.(2022), algoritma KNN sangat bergantung pada perhitungan kedekatan spasial antar-titik data, sehingga perbedaan rentang angka yang mencolok antarfitur dapat mendistorsi objektivitas ruang keputusan model [9]. Sejalan dengan pendapat dari Azizah dkk.(2022), apabila tahapan normalisasi ini dilewatkan, fitur numerik yang memiliki skala rentang nilai besar (misalnya nilai ujian berskala puluhan hingga ratusan) akan mendominasi (*feature dominance*) seluruh kalkulasi jarak *Euclidean* dan mengabaikan pengaruh dari fitur prediktor yang berskala kecil (misalnya durasi interaksi berskala satuan jam) [12]. Oleh karena itu, normalisasi ini wajib diimplementasikan demi meniadakan bias dominasi fitur tersebut agar seluruh variabel memiliki bobot pengaruh yang setara.

Melalui kalkulasi batas nilai tersebut, seluruh dimensi data numerik kontinu yang digunakan di dalam program akan disatukan ke dalam standarisasi matriks koordinat yang setara. Mengacu pada pengujian pemodelan data dari Arrohman dkk. (2025), hasil akhir dari implementasi teknik *MinMaxScaler* ini secara presisi akan memetakan seluruh variasi nilai fitur numerik ke dalam batas skala rentang nilai yang konstan, yaitu antara 0 hingga 1 ($0 \leq X_{\text{norm}} \leq 1$) [20].

C Train-Test Split

Train-Test Split merupakan implementasi dari metode *Hold-Out Validation*, yaitu teknik evaluasi model yang bekerja dengan cara membagi dataset tunggal secara statis menjadi bagian terpisah yang saling bebas (*mutually exclusive*), yaitu data latih (*training data*) dan data uji (*testing data*). Mengacu pada penelitian Amri dkk. (2023), metode pembagian data ini digunakan untuk memastikan bahwa model yang dilatih dapat diuji objektivitasnya menggunakan data baru yang belum pernah

dilihat sebelumnya selama fase pelatihan, sehingga tingkat kemampuan generalisasi model dapat diukur secara adil [11].

Di dalam program penelitian ini, dataset dibagi secara proporsional dengan menetapkan rasio sebesar 80% sebagai data latih (*training set*) dan 20% sebagai data uji (testing set). Berdasarkan temuan dari Febrian dkk. (2025), alokasi porsi data latih sebesar 80% bertujuan untuk menyediakan volume informasi dan pola variasi data yang cukup melimpah bagi algoritma guna menyusun ruang keputusan yang matang, sementara sisa 20% data uji sudah sangat ideal dan valid untuk bertindak sebagai instrumen pengevaluasi performa akhir model klasifikasi [17].

Pada saat pembagian data dijalankan, program menyertakan konfigurasi parameter `stratify` yang merujuk pada label target variabel (*y*). Sejalan dengan analisis metodologi dari Novianto dkk.(2023), fungsi stratifikasi ini sangat krusial untuk menjaga agar persentase atau proporsi sebaran kelas target pada data latih dan data uji tetap seimbang dan identik dengan proporsi pada *dataset* asli [10]. Melalui mekanisme ini, risiko terjadinya ketidakseimbangan sebaran kelas (*class imbalance*) pada salah satu sub-data dapat dihindari, sehingga model terhindar dari bias klasifikasi kelompok mayoritas.

untuk mengunci proses pengacakan baris data pada saat pembagian dilakukan, program mengonfigurasikan nilai parameter `random_state` sebesar 42. Berdasarkan metodologi eksperimen dalam Fauzan dkk.(2024), penetapan nilai benih (*seed*) acak yang konstan ini bertujuan untuk menjamin sifat reproduksibilitas (*reproducibility*) dari eksperimen [15]. Artinya, setiap kali program dijalankan ulang oleh siapa pun, sistem akan menghasilkan pembagian baris data latih dan data uji yang persis sama, sehingga fluktuasi metrik evaluasi murni dipengaruhi oleh perubahan parameter algoritma dan bukan karena faktor kebetulan dari perubahan variasi sampel data.

2.1.7 Evaluasi Model

Tahapan evaluasi model merupakan fase krusial dalam penelitian berbasis *data mining* untuk mengukur tingkat keberhasilan, keandalan, dan efektivitas dari algoritma klasifikasi yang telah dilatih. Mengacu pada penelitian Hidayatullah dkk. (2022), instrumen evaluasi performa model klasifikasi perlu diukur menggunakan berbagai metrik penilaian yang komprehensif agar hasil pengujian terhadap kemampuan prediksi sistem bersifat valid, objektif, dan terukur [9].

1. **Confusion Matrix**

Confusion Matrix adalah tabel kontingensi berdimensi 2×2 (untuk klasifikasi biner) yang digunakan untuk mendokumentasikan serta membandingkan akumulasi hasil prediksi yang dilakukan oleh algoritma dengan label aktual data yang sebenarnya. Sejalan dengan pendapat dari Muhaimin dkk. (2024), pemetaan hasil klasifikasi ke dalam matriks ini sangat mempermudah peneliti untuk menganalisis secara mendalam mengenai letak dan jenis kesalahan prediksi yang dilakukan oleh algoritma terhadap setiap kelas target [14]. Struktural dari tabel matriks keputusan tersebut digambarkan pada Tabel 2.1.

Tabel 2.1. Struktur Matriks Kontingensi (*Confusion Matrix*)

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

Penjelasan mengenai komponen utama di dalam matriks tersebut dijabarkan sebagai berikut:

(a) *True Positive* (TP): Jumlah data aktual positif yang berhasil diprediksi dengan benar oleh model sebagai kelas positif.

(b) *True Negative* (TN): Jumlah data aktual negatif yang berhasil diprediksi dengan benar oleh model sebagai kelas negatif.

(c) *False Positive* (FP): Jumlah data aktual negatif yang salah diprediksi oleh model sebagai kelas positif (*Error* Tipe I).

(d) *False Negative* (FN): Jumlah data aktual positif yang salah diprediksi oleh model sebagai kelas negatif (*Error* Tipe II).

2. **Akurasi (Accuracy)**

Akurasi merupakan metrik paling mendasar yang merepresentasikan rasio total prediksi yang dilakukan dengan benar oleh model (baik positif maupun negatif) terhadap keseluruhan volume sampel data pengujian yang ada. Berdasarkan temuan dari Wahyuni dkk. (2025), akurasi menjadi indikator utama yang paling sering digunakan untuk memberikan gambaran global mengenai tingkat kematangan dan keandalan sistem klasifikasi dalam mengelompokkan data secara tepat [1]. Persamaan matematis untuk menghitung nilai akurasi dituliskan sebagai berikut:

$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.5}$$

3. **Presisi (Precision)**

   Presisi atau *positive predictive value* adalah metrik yang mengukur tingkat ketepatan antara jumlah data yang benar-benar positif dengan total seluruh data yang diprediksi sebagai kelas positif oleh model, dengan fokus utama meminimalkan kesalahan *False Positive*. Mengacu pada penelitian Azizah dkk. (2022), nilai presisi memegang peranan penting dalam domain pendidikan untuk memastikan bahwa siswa yang diidentifikasi oleh model masuk ke dalam klasifikasi tertentu benar-benar memenuhi kriteria objektif, sehingga meminimalkan salah sasaran intervensi [12]. Persamaan matematis presisi dirumuskan sebagai berikut:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.6}$$

4. **Recall (Sensitivity)**

   Recall atau sensitivitas adalah metrik yang mengukur keberhasilan model dalam menjaring atau menemukan kembali seluruh sampel data yang secara aktual termasuk dalam kategori positif, dengan fokus meminimalkan kesalahan *False Negative*. Sejalan dengan analisis metodologi dari Jasri dkk. (2024), tingkat *recall* yang tinggi memastikan bahwa sistem tidak melewatkan objek atau siswa yang secara riil membutuhkan perhatian khusus atau penanganan dini dari institusi pendidikan [16]. Persamaan matematis *recall* dituliskan sebagai berikut:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.7}$$

5. **F1-Score**

   F1-Score merupakan metrik evaluasi yang menghitung nilai rata-rata harmonik (*harmonic mean*) yang mengombinasikan komponen presisi dan *recall* ke dalam satu nilai tunggal. Berdasarkan temuan dari Fuazan dkk. (2024), metrik *F1-score* bertindak sebagai tolak ukur evaluasi yang lebih solid dan objektif ketika peneliti membutuhkan kepastian bahwa performa presisi dan *recall* di dalam model klasifikasi telah berada pada titik keseimbangan

yang optimal [15]. Persamaan matematis *F1-score* dirumuskan sebagai berikut:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.8}$$

6. **Weighted Average**

   Di dalam kasus nyata dunia pendidikan, ketersediaan sampel data antar-kelas target kerap kali memuat sebaran distribusi yang tidak merata, seperti jumlah siswa yang bermasalah akademik berjumlah jauh lebih sedikit dibandingkan siswa berprestasi. Jika evaluasi model hanya menggunakan metode rata-rata biasa (*macro average*), nilai metrik dari kelas minoritas akan diperlakukan setara dengan kelas mayoritas tanpa memedulikan volume datanya, sehingga berpotensi menghasilkan interpretasi performa yang bias.

   Oleh karena itu, program penelitian ini mengonfigurasikan pendekatan *Weighted Average*. Mekanisme *Weighted Average* bekerja dengan cara menghitung nilai rata-rata dari presisi, *recall*, dan *F1-score* yang dibobotkan secara proporsional berdasarkan jumlah sampel asli (*support*) yang menduduki masing-masing kelas target. Mengacu pada temuan dari Fatah dan Navizah (2026), pemanfaatan perhitungan *weighted average* menjadi solusi metodologis yang sangat akurat untuk memperoleh penilaian performa model yang jujur, stabil, dan tepercaya ketika dihadapkan pada karakteristik sebaran data yang memiliki distribusi kelas tidak seimbang (*imbalanced data*) [19].

2.2 Perbandingan Penelitian Terdahulu *(State of the Art)*

Mengacu pada perkembangan metodologi dalam *Educational Data Mining* (EDM), pemetaan terhadap hasil eksperimen yang telah dilakukan oleh peneliti sebelumnya sangat krusial untuk mempertegas posisi kebaruan (*novelty*) dari penelitian ini. Berdasarkan temuan dari berbagai studi literatur, algoritma klasifikasi seperti *K-Nearest Neighbor* (KNN), *Decision Tree*, dan *Naive Bayes* secara konsisten menunjukkan efektivitas yang tinggi dalam mengolah data akademis maupun demografis siswa.

Sejalan dengan pendapat dari para peneliti terdahulu, variasi penggunaan kombinasi atribut dan penempatan model pembanding menjadi faktor utama penentu tingkat akurasi sistem. Oleh karena itu, guna memberikan visualisasi

komparatif yang jelas, rangkuman mengenai perbedaan mendasar antara penelitian terdahulu dengan penelitian skripsi yang diusulkan dijabarkan pada Tabel **??**.

Tabel 2.2. Perbandingan Penelitian Terdahulu

No	Penulis (Tahun)	Objek & Metode	Temuan Utama	Celah Penelitian (*Research Gap*)
1	Arrohman dkk. (2025)	Dataset *Students Performance*; KNN & *Random Forest*	*Random Forest* mencapai akurasi 100% dan KNN mencapai 90% dalam prediksi kelulusan [20].	Berfokus pada prediksi kelulusan umum, belum mengevaluasi pengaruh variabel lingkungan belajar secara spesifik.
2	Fatah & Navizah (2026)	Status Beasiswa; *Decision Tree* & *Naive Bayes*	*Naive Bayes* lebih unggul dengan akurasi 96,26% dibandingkan *Decision Tree* (76,44%) [19].	Terbatas pada klasifikasi kelayakan finansial penerima beasiswa, bukan pemetaan potensi kemampuan akademik siswa.
3	Febrian dkk. (2025)	Potensi Akademik Siswa SMP; KNN & *Decision Tree*	Komparasi kedua metode efektif memetakan potensi berdasarkan kombinasi nilai kognitif dan perilaku [17].	Subjek terbatas pada siswa tingkat SMP dan belum melibatkan model probabilitas kontinu seperti *Gaussian Naive Bayes*.

No	Penulis (Tahun)	Objek & Metode	Temuan Utama	Celah Penelitian (*Research Gap*)
4	Muhaimin dkk. (2024)	Prestasi Akademik; KNN	Klasifikasi prestasi berjalan akurat memanfaatkan nilai rapor dan poin pelanggaran kedisiplinan [14].	Hanya menerapkan metode tunggal KNN tanpa adanya algoritma *baseline* lain sebagai pembanding kinerja model.
5	Fauzan dkk. (2024)	Prestasi Belajar SMK; *Decision Tree* & *Naive Bayes*	Evaluasi prestasi belajar berhasil dipetakan murni menggunakan kombinasi nilai mata pelajaran kognitif [15].	Belum menguji algoritma berbasis kedekatan spasial/jarak (KNN) untuk mengoptimalkan objektivitas akurasi ruang keputusan.
6	Wahyuni & Kurniawan (2025)	Kemampuan Akademik; KNN Berbasis Web	Sistem berbasis web berhasil mengelompokkan kemampuan akademik untuk mendukung Kurikulum Merdeka [1].	Lebih menitikberatkan pada implementasi aplikasi sistem daripada analisis komparatif model terhadap data tidak seimbang (*imbalanced data*).
7	**Penelitian Ini (2026)**	**Potensi Akademik Siswa; KNN, DT, NB**	**Klasifikasi potensi akademik dengan optimasi fitur demografis dan lingkungan belajar siswa.**	**Mengisi celah dengan menguji model utama KNN terhadap dua baseline sekaligus mengevaluasi dampak fitur non-kognitif.**