

Implementasi Algoritma Naïve Bayes untuk Memprediksi Kemampuan Pemrograman Mahasiswa Teknik Informatika Menggunakan Dataset Kuesioner

Ide Kristiani Zega^{*1}, Nabila Medina², Debora Aprillia S³, Yennimar⁴

^{1,2,3,4}Teknik Informatika, Universitas Prima Indonesia, Indonesia
Email: ¹kristianni.zega08@gmail.com, ²nabilamedina0602@gmail.com,
³deboraapril12204@gmail.com, ⁴yennimar@unprimdn.ac.id

Abstrak

Permintaan tenaga kerja di bidang pemrograman semakin meningkat, sementara banyak mahasiswa yang menghadapi kesulitan dalam menguasai keterampilan pemrograman. Penelitian ini bertujuan untuk memprediksi kemampuan pemrograman mahasiswa Teknik Informatika menggunakan algoritma Naïve Bayes, dengan mempertimbangkan pemahaman dasar algoritma pemrograman sebagai parameter. Model dikembangkan dan diuji menggunakan 210 data training dan 90 data testing. Hasil pengujian menunjukkan akurasi model sebesar 100%, dengan prediksi "Mampu" sesuai kenyataan sebanyak 82 data, sedangkan prediksi "Tidak Mampu" yang sesuai dengan kenyataan sebanyak 8 data. Tidak ditemukan kesalahan prediksi pada kategori "Mampu" dan "Tidak Mampu". Precision dan recall masing-masing mencapai 100%, mengindikasikan bahwa model ini sangat efektif dalam mengklasifikasikan mahasiswa sebagai "Mampu" dan "Tidak Mampu". Penelitian ini berkontribusi pada pengembangan metode berbasis data untuk mengevaluasi kemampuan pemrograman mahasiswa, memberikan wawasan penting bagi perbaikan kurikulum dan penilaian pendidikan di bidang Teknik Informatika.

Kata kunci: kemampuan pemrograman, naïve bayes, prediksi, teknik informatika

Implementation of the Naïve Bayes Algorithm to Predict Programming Skills of Informatics Engineering Students Using Questionnaire Dataset

Abstract

The demand for skilled programming professionals continues to rise, while many students face challenges in mastering programming skills. This study aims to predict the programming capabilities of Informatics Engineering students using the Naïve Bayes algorithm, with a focus on fundamental understanding of programming algorithms as a parameter. The model was developed and tested using 210 training data points and 90 testing data points. The results showed an accuracy of 100%, with 82 correct "Capable" predictions and 8 correct "Not Capable" predictions. No errors were found in either the "Capable" or "Not Capable" categories. Both precision and recall reached 100%, indicating the model's high effectiveness in classifying students as "Capable" or "Not Capable." This study contributes to the development of data-driven methods for evaluating students' programming abilities, offering valuable insights for curriculum improvement and educational assessment in Informatics Engineering.

Keywords: informatics engineering, naïve bayes, prediction, programming skills

1. PENDAHULUAN

Melihat kebutuhan tenaga kerja di bidang pemrograman di masa depan, lebih banyak orang dengan kemampuan pemrograman yang memadai dibutuhkan [1]. Mahasiswa yang memiliki keterampilan atau kemampuan dalam pemrograman yang baik memiliki peluang untuk sukses di dunia kerja yang semakin kompetitif. Untuk memiliki kemampuan pemrograman tersebut mahasiswa atau dalam hal ini disebut sebagai seorang programmer harus menguasai beberapa keterampilan khusus termasuk algoritma, logika, bahasa pemrograman, dan beberapa pengetahuan matematika [2]. Dengan mempelajari logika pemrograman yang disusun dalam algoritma terstruktur, seseorang dapat memecahkan masalah dalam pemrograman dengan lebih cepat dan efisien [3].

Algoritma adalah pendekatan sistematis yang melibatkan urutan langkah yang digunakan untuk memecahkan masalah dan membuat keputusan [4]. Makna algoritma serta penanganan masalah secara metodis,

serta dalam latihan pemrograman algoritma sering dianggap selaku dasar pemikiran dalam memutuskan program yang hendak dibuat [5]. Pemrograman merupakan proses penulisan, pengujian, pembaruan, dan pemeliharaan kode yang membangun program komputer [6]. Tujuan pemrograman adalah untuk memecahkan masalah-masalah menggunakan komputer dengan cepat dan akurat [7]. Berdasarkan hal ini, penting dilakukannya penelitian untuk memprediksi kemampuan pemrograman mahasiswa untuk mengetahui seberapa dalam pemahaman dasar algoritma pemrograman yang dimiliki mahasiswa.

Naïve Bayes merupakan algoritma yang bersifat klasikal, sederhana, dan independent [8]. Metode ini menggunakan pendekatan probabilitas dan statistik dalam proses pengklasifikasian. Naïve Bayes dapat memprediksi kemungkinan di masa depan dengan berdasarkan data pengalaman sebelumnya [9].

Penelitian oleh R. Annisa, dan A. Sasongko dengan judul “Prediksi Nilai Akademik Mahasiswa Menggunakan Algoritma Naïve Bayes” menyimpulkan bahwa penerapan algoritma naïve bayes dengan menggunakan perhitungan probabilitas dan data statistik sebelumnya untuk memprediksi data di masa depan menghasilkan *accuracy* 96,24%, *precision* 95,76%, dan *recall* 100% [10] (Prediksi Nilai Akademik Mahasiswa Menggunakan Algoritma Naïve Bayes). Penelitian lain yang telah dilakukan oleh Putri Ramadani et al dengan judul “Penerapan Metode Naïve Bayes dalam Memprediksi Kepuasan Mahasiswa Terhadap Cara Pengajaran Dosen” menyimpulkan bahwa Naïve Bayes adalah metode yang layak digunakan pengambilan keputusan dalam prediksi. Dengan menggunakan metode Naïve Bayes dengan 574 data (574 training dan 574 testing), hasilnya menunjukkan bahwa 397 data (69,078%) menyatakan "Sangat Puas" dan 177 data (30,798%) menyatakan "Puas" [11]. Dengan demikian, berdasarkan hasil ulasan penelitian sebelumnya, maka implementasi algoritma naïve bayes dalam penelitian ini menjadi solusi yang efektif sebagai metode prediksi kemampuan pemrograman mahasiswa.

Berikut ini diuraikan keterbaruan dari penelitian ini melalui *literature review* penelitian terdahulu yang berkaitan dengan implementasi algoritma Naïve Bayes, antara lain:

1. Menurut M Riski Qisthiano, Tri Basuki Kurniawan, Edi Surya Negara, Muhammad Akbar (2021) dalam penelitian yang berjudul “Pengembangan Model untuk Prediksi Tingkat Kelulusan Mahasiswa Tepat Waktu dengan Metode Naïve Bayes” mendapatkan hasil kinerja algoritma naïve bayes dengan data uji yang didapat memiliki tingkat akurasi sebesar 0.810 dengan menggunakan Algoritma Naïve Bayes. Untuk nilai *precision* Kelas Tepat Waktu memiliki nilai “0.81” sedangkan Tidak Tepat Waktu sebesar “0.81”. Untuk nilai *recall* Kelas Tepat Waktu memiliki nilai “0.82” sedangkan kelas Tidak Tepat Waktu sebesar “0.80” [12].
2. Menurut Nur Mahar Aji, Vihi Atina, dan Nugroho Arif Sudiboyo (2023) dalam penelitian yang berjudul “Pemodelan Prediksi Kelulusan Mahasiswa Dengan Metode Naïve Bayes Di UNIBA” mendapatkan hasil kinerja algoritma Naïve Bayes dengan akurasi ketepatan prediksi sebesar 85%, dengan nilai *precision* terlambat 0.42, *precision* tepat 0.95, *recall* terlambat 0.65, *recall* tepat 0.88, *f1-score* terlambat 0.51, *f1-score* tepat 0.91. [13]. Menurut Entin Sutinah, Nani Agustina, dan Martini (2023) dalam penelitian yang berjudul “Algoritma Naïve Bayes untuk Prediksi Keberhasilan Mahasiswa pada Mata Kuliah Praktikum”, dengan menggunakan 130 sampel dari hasil kuesioner yang diberikan kepada mahasiswa dengan 20 pertanyaan, mendapatkan hasil kinerja algoritma Naïve Bayes dengan akurasi sebesar 100%, *presisi* 100%, dan *recall* 100% [14].
3. Menurut Angga Aditya Permana, Rohmat Taufiq, Rachmat Destriana, Aliya Nur’aini (2024) dalam penelitian yang berjudul “Implementasi Algoritma Naïve Bayes Untuk Prediksi Kelulusan Mahasiswa” mendapatkan hasil kinerja algoritma naïve bayes dengan akurasi sebesar 89%, *presisi* sekitar 88% untuk kelas 0 dan 89% untuk kelas 1, dan *recall* mencapai sekitar 85% untuk kelas 0 dan 91% untuk kelas 1 [15].
4. Menurut Agustinus A. Botara, dan Ahmad Sabri (2024) dalam penelitian yang berjudul “Prediksi Masa Studi Mahasiswa Menggunakan Algoritma Naïve Bayes pada Universitas Halmahera” mendapatkan hasil akurasi sebesar 82,25% [16].

2. METODE PENELITIAN

2.1. Jenis Penelitian

Pada penelitian ini, digunakan jenis penelitian kuantitatif yang bertujuan untuk mengukur dan menganalisis data numerik secara objektif. Tujuan dari penelitian kuantitatif adalah untuk menguji hipotesis dan menentukan hubungan atau pola dari data yang dikumpulkan. Dalam hal ini, penelitian berfokus pada implementasi algoritma Naïve Bayes untuk memprediksi kemampuan pemrograman mahasiswa jurusan Teknik Informatika berdasarkan data kuesioner yang dikumpulkan dari 300 mahasiswa.

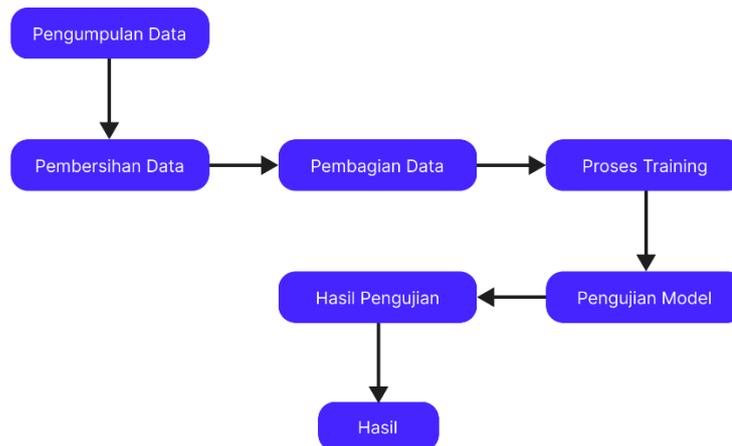
2.2. Prosedur Kerja

Prosedur kerja yang dilakukan dalam penelitian ini dapat dijelaskan sebagai berikut:

1. Studi Pustaka

Mempelajari jurnal internasional dan nasional yang khususnya berhubungan dengan kemampuan pemrograman, pemahaman dasar algoritma pemrograman serta melakukan prediksi dengan implementasi algoritma Naïve Bayes.

2. Kerangka Kerja yang Diusulkan



Gambar 1. Kerangka Kerja

- Pengumpulan Data: Dataset yang digunakan diperoleh dari 300 mahasiswa semester 4 dari program studi Teknik Informatika, Universitas Prima Indonesia. Data ini dikumpulkan melalui kuesioner berbasis Google Form yang terdiri dari 50 pertanyaan, mencakup aspek pemahaman algoritma dasar, pengalaman pemrograman, dan hasil akademik.
- Pembersihan Data: Proses pembersihan data dilakukan untuk memastikan data yang digunakan konsisten dan tidak ada yang hilang serta bebas dari duplikasi.
- Pembagian Dataset: Dataset dibagi menjadi dua bagian:
 - Data Training: 70% dari total dataset yang digunakan untuk melatih model.
 - Data Testing: 30% dari sisa dataset yang akan digunakan untuk menguji model.
- Proses Training: Pada proses training, model dilatih menggunakan Data Training untuk menemukan pola atau hubungan antara fitur dan label yang ada pada data.
- Pengujian Model: Model Testing kemudian digunakan untuk menguji bagaimana model tersebut bekerja pada data yang tidak pernah dilihat sebelumnya, yaitu untuk mengukur akurasi dan performa model.
- Hasil Pengujian: Pada tahap pengujian model, dilakukan analisis untuk menghitung nilai akurasi, precision, dan recall sebagai metrik utama dalam menilai performa model.
- Hasil: Hasil akhir diperoleh melalui perbandingan antara hasil pengujian yang dihitung secara manual dengan hasil pengujian menggunakan perangkat lunak RapidMiner.

2.3. Algoritma Naïve Bayes

Naïve Bayes adalah teknik untuk memprediksi kemungkinan di masa depan berdasarkan dari pengalaman sebelumnya dengan mengacu pada probabilitas dan statistik [17]. Rumus perhitungan naïve bayes adalah sebagai berikut[18]:

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

Keterangan:

- X : Data kelas tidak diketahui
- H : Hipotesis data merupakan suatu kelas spesifik
- P(H|X) : Kondisi X (posteriori probabilitas)
- P(H) : Probabilitas hipotesis H (prior probabilitas)
- P(X|H) : Probabilitas X berdasarkan kondisi pada hipotesis H
- P(X) : Probabilitas X.

3. HASIL DAN PEMBAHASAN

3.1. Dataset Penelitian

3.1.1. Pengumpulan Data

Data penelitian ini diperoleh dari responden yang merupakan mahasiswa jurusan Teknik Informatika semester 4 Universitas Prima Indonesia. Total responden berjumlah 300 mahasiswa.

Tabel 1. Responden

No	Nim	Jurusan	Semester
1	223303030413	Teknik Informatika	4
2	223303030421	Teknik Informatika	4
3	223303030444	Teknik Informatika	4
4	223303030406	Teknik Informatika	4
5	233303033004	Teknik Informatika	4
6	223303030446	Teknik Informatika	4
7	223303030436	Teknik Informatika	4
8	223303030447	Teknik Informatika	4
9	223303030431	Teknik Informatika	4
...			
300	223303030624	Teknik Informatika	4

Pengumpulan data dilakukan dengan menggunakan kuesioner yang terdiri dari 50 pertanyaan dan dibagi menjadi 5 parameter. Masing-masing pertanyaan dikategorikan sebagai A1, A2 dan seterusnya sampai A50. Kuesioner dibuat dalam bentuk *google-form* dan dibagikan secara *online*.

Tabel 2. Pertanyaan Kuesioner

Parameter	Kategori	Pertanyaan
P1	A1	Seberapa baik Anda dalam memahami apa itu Algoritma dalam pemrograman?
	A2	Seberapa baik Anda dalam memahami jenis-jenis algoritma seperti algoritma sekuensial (<i>sequential</i>), algoritma percabangan (<i>branching/selection</i>), dan algoritma perulangan (<i>iteration</i>)?
	A3	Seberapa baik Anda dalam memahami dua jenis algoritma dasar pemrograman yaitu algoritma <i>searching</i> dan algoritma <i>sorting</i> ?
	A4	Seberapa baik Anda dalam memahami cara penulisan algoritma seperti <i>Deskriptif</i> , <i>Flowchart</i> dan <i>Pseudocode</i> ?
	A5	Seberapa baik Anda dalam memahami fungsi dan kegunaan simbol-simbol pada <i>Flowchart</i> ?
	A6	Seberapa baik Anda dalam menganalisis masalah dan merancang solusi menggunakan algoritma pemrograman?
	A7	Seberapa baik Anda dalam menyelesaikan masalah atau <i>study case</i> menggunakan <i>Flowchart</i> ?
	A8	Seberapa baik Anda dalam menyelesaikan masalah atau <i>study case</i> menggunakan <i>Algoritma Pseudocode</i> ?
P2	A9	Seberapa baik Anda dalam memahami apa itu program dan pemrograman?
	A10	Seberapa baik Anda dalam memahami konsep dasar pemrograman, seperti sintaks, variabel, dan struktur kontrol?
	A11	Seberapa baik Anda dalam memahami tipe-tipe data pemrograman (seperti <i>int</i> , <i>float</i> , <i>char</i> , <i>void</i>)?
	A12	Seberapa baik Anda memahami perbedaan antara tipe data primitif (<i>int</i> , <i>float</i> , <i>boolean</i>) dan tipe data kompleks (<i>list</i> , <i>dictionary</i> , <i>set</i>)?
	A13	Seberapa baik Anda dalam memahami konsep pengulangan (<i>looping</i>) dalam pemrograman?
	A14	Seberapa baik Anda dalam memahami konsep pengambilan keputusan (<i>if-else</i>) dalam pemrograman?
	A15	Seberapa baik Anda dalam memahami konsep modularisasi (<i>function</i> , dan <i>class</i>) dalam bahasa pemrograman?
	A16	Seberapa baik Anda dalam memahami jenis-jenis <i>function</i> yaitu; <i>Built-In Function</i> dan

		User-Defined Function dalam pemrograman?
	A17	Seberapa baik Anda dalam memahami perbedaan parameter dan argumen dalam pemrograman?
	A18	Seberapa baik Anda dalam memahami apa itu Pemrograman Berorientasi Objek (OOP)?
	A19	Seberapa baik Anda dalam memahami konsep dasar Pemrograman Berorientasi Objek (OOP), seperti class, object, method, inheritance, polimorfisme, dan enkapsulasi?
	A20	Seberapa baik Anda dalam memahami apa itu abstract dalam Pemrograman Berorientasi Objek (OOP)?
P3	A21	Seberapa baik Anda dalam memahami apa itu Data dalam pemrograman?
	A22	Seberapa baik Anda dalam memahami apa itu Struktur Data dalam pemrograman?
	A23	Seberapa baik Anda dalam memahami perbedaan tipe data dalam Pascal (Tipe data sederhana, Type data terstruktur, Tipe data Pointer)?
	A24	Seberapa baik Anda dalam memahami metode pengurutan data (sorting) yang terdiri dari Bubble Sort, Selection Sort, Insertion Sort, dan Quick Sort?
	A25	Seberapa baik Anda dalam memahami tipe-tipe struktur data yang dapat digunakan dalam menyimpan data di komputer seperti Array, Linked List, Stack, Queue, Tree, Graph, Hash Table?
	A26	Seberapa baik Anda dalam memahami tipe array seperti array satu dimensi dan array dua dimensi??
	A27	Seberapa baik Anda dalam memahami apa itu Basis Data?
	A28	Seberapa baik Anda dalam memahami Struktur Dasar Basis Data seperti data, information, table, column, query, dan record?
	A29	Seberapa baik Anda dalam memahami Operasi Dasar Basis Data seperti CREATE, DROP, UPDATE, DELETE, INSERT, QUERY?
	A30	Seberapa baik Anda dalam memahami dan menggunakan SQL (Structured Query Language) dalam mengakses dan mengelola data?
P4	A31	Seberapa baik Anda dalam memahami apa itu bahasa pemrograman?
	A32	Seberapa baik Anda dalam memahami perbedaan bahasa pemrograman (bahasa mesin, bahasa assembly, bahasa tingkat tinggi)?
	A33	Seberapa baik Anda dalam memahami dan menggunakan bahasa pemrograman C, seperti penggunaan pointer, manipulasi string, penanganan file, atau pemrograman berbasis memori?
	A34	Seberapa baik Anda dalam memahami dan menggunakan bahasa pemrograman Javascript, seperti penggunaan variabel, fungsi, loop, kondisional, dan manipulasi DOM?
	A35	Seberapa baik Anda dalam memahami dan menggunakan bahasa pemrograman PHP, seperti penggunaan variabel, fungsi, loop, kondisional, dan pengelolaan database?
	A36	Seberapa baik Anda dalam memahami dan menggunakan bahasa pemrograman Phyton, seperti penggunaan variabel, fungsi, loop, kondisional, dan manipulasi string?
	A37	Seberapa baik Anda dalam memahami dan menggunakan bahasa pemrograman Dart, seperti penggunaan variabel, fungsi, loop, kondisional, dan pemrograman berbasis objek?
P5	A38	Seberapa baik Anda dalam mengimplementasikan (menerapkan) konsep algoritma pemrograman seperti pengurutan, pencarian, pemrosesan string, atau pengelolaan struktur data (seperti linked list, tree, dan graph) dalam bentuk kode program?
	A39	Seberapa baik Anda dalam mendeklarasikan dan menggunakan variabel untuk melakukan penyimpanan dan manipulasi data?
	A40	Seberapa baik Anda dalam melakukan implementasi struktur kontrol seperti if-else, loop, dan switch-case dalam bentuk kode program?
	A41	Seberapa baik Anda dalam mengimplementasikan (menerapkan) function untuk tugas-tugas seperti menghitung nilai tertentu, melakukan manipulasi data, atau menyelesaikan masalah spesifik dalam bentuk kode program?
	A42	Seberapa baik Anda dalam mengimplementasikan (menerapkan) konsep Pemrograman Berorientasi Objek (OOP) seperti inheritance, polimorfisme, dan enkapsulasi serta membuat class, object, method dalam bentuk kode program?
	A43	Seberapa baik Anda dalam mengimplementasikan (menerapkan) Abstract (baik Abstract Class dan Abstract Method) dalam bentuk kode program?

A44	Seberapa baik Anda dalam mengimplementasikan (menerapkan) kueri SQL dalam mengelola struktur database (pembuatan tabel dan indeks), pengelompokan data, dan pengurutan data dalam bentuk kode program?
A45	Seberapa baik Anda dalam mengimplementasikan (menerapkan) array baik array satu dimensi dan dua dimensi dalam bentuk kode program?
A46	Seberapa baik Anda dalam mengimplementasikan (menerapkan) suatu program sederhana (seperti menghitung bilangan prima, mengurutkan array, mengelola struktur data sederhana atau program sederhana lainnya) dalam bahasa pemrograman C?
A47	Seberapa baik Anda dalam mengimplementasikan (menerapkan) suatu program sederhana (seperti menghitung bilangan prima, mengurutkan array, mengelola struktur data sederhana atau program sederhana lainnya) dalam bahasa pemrograman Javascript?
A48	Seberapa baik Anda dalam mengimplementasikan (menerapkan) suatu program sederhana (seperti menghitung bilangan prima, mengurutkan array, mengelola struktur data sederhana atau program sederhana lainnya) dalam bahasa pemrograman PHP?
A49	Seberapa baik Anda dalam mengimplementasikan (menerapkan) suatu program sederhana (seperti menghitung bilangan prima, mengurutkan array, mengelola struktur data sederhana atau program sederhana lainnya) dalam bahasa pemrograman Python?
A50	Seberapa baik Anda dalam mengimplementasikan suatu program sederhana (seperti menghitung bilangan prima, mengurutkan array, membuat antarmuka pengguna yang responsif atau program sederhana lainnya) dalam bahasa pemrograman Dart?

3.1.2. Pembersihan Data

Sebelum data digunakan untuk proses *training* dan pengujian, dilakukan tahap pembersihan data untuk memastikan data sudah lengkap. Setiap parameter mencakup sejumlah pertanyaan kemudian diajukan kepada responden dengan menggunakan skala likert 5 poin, yaitu: Sangat Baik (SB), Baik (B), Cukup (C), Kurang (K), Sangat Kurang (SK), yang kemudian dikonversi menjadi nilai numerik seperti yang tertera pada Tabel 3.

Tabel 3. Skala Penilaian

Skala Penilaian (Likert)	Nilai
SB	5
B	4
C	3
K	2
SK	1

Tabel 3. Skala Penilaian digunakan untuk menggantikan kategori teks menjadi angka yang dapat memudahkan proses *training* dan pengujian. Berikut adalah hasil konversi jawaban responden dalam nilai numerik seperti yang tertera pada Tabel 4.

Tabel 4. Data Numerik

Resp	A1	A2	A3	A4	A5	A6	A7	A8	...	A49	A50
R1	3	3	3	3	3	3	3	3		3	3
R2	5	5	5	3	3	3	5	5		5	5
R3	4	4	4	3	4	4	4	4		4	4
R4	4	4	5	5	5	4	4	4		4	4
R5	4	4	4	4	4	4	4	3		4	4
R6	3	3	4	4	4	3	3	2		3	3
R7	3	3	3	3	3	3	3	3		3	3
R8	2	2	3	3	3	3	3	2		2	2
...											
R299	4	4	4	4	4	4	4	4		4	4
R300	4	3	4	5	5	4	5	3		4	2

3.1.3. Pembagian Data

Data dibagi menjadi 70% untuk data training dan 30% untuk data testing. Kolom skor total diperoleh dengan menjumlahkan setiap jawaban dari masing-masing responden. Nilai terendah yang didapatkan adalah 100 sementara nilai tertinggi adalah 260. Untuk menentukan rentang penilaian setiap kategori berdasarkan skor total, skor-skor ini kemudian dikelompokkan menjadi 5 kategori: ≤ 116 (Sangat Kurang), ≤ 135 (Kurang), ≤ 150 (Cukup), ≤ 180 (Baik), ≤ 260 (Sangat Baik).

Klasifikasi data training dibagi menjadi dua kategori setelah kategori yang diketahui yaitu Mampu dan Tidak Mampu. Ketentuannya adalah: jika kategori Sangat Baik, Baik, dan Cukup maka diklasifikasikan sebagai Mampu, sedangkan jika kategori Kurang dan Sangat Kurang maka diklasifikasikan sebagai Tidak Mampu. Terdapat sebanyak 210 data responden dengan 197 responden dinyatakan Mampu dan 13 responden dinyatakan Tidak Mampu untuk data training, seperti yang terlihat pada Tabel 5.

Tabel 5. Data Training

Resp	A1	...	A50	Skor Total	Kategori	Klasifikasi
R1	3		3	169	Baik	Mampu
R2	5		5	234	Sangat Baik	Mampu
R3	4		4	212	Sangat Baik	Mampu
R4	4		4	199	Sangat Baik	Mampu
...						
R210	4		4	214	Sangat Baik	Mampu

Data testing terdiri dari 90 data yang kemudian dicari klasifikasinya seperti yang terlihat pada Tabel 6.

Tabel 6. Data Testing

Resp	A1	A50	Klasifikasi
R211	B		K	?
R212	B		B	?
R213	B		B	?
R214	B		C	?
...				
R300	SK		K	?

3.1.4. Proses Training (Perhitungan Manual dengan Metode Naïve Bayes)

1) Melakukan perhitungan jumlah probabilitas sebelumnya

Terdapat 210 data training, data Mampu sebanyak 197 dan data Tidak Mampu sebanyak 13. Maka didapat nilai dari probabilitas pada responden, yaitu:

$$P(R | M) = 197/210 = 0,938$$

$$P(R | TM) = 13/210 = 0,061$$

2) Melakukan perhitungan probabilitas untuk setiap parameter

Setiap parameter dari parameter I sampai dengan parameter V dihitung probabilitas menggunakan kategori A1 sampai A50.

a) Parameter I: Dasar Algoritma kategori A1

$$P(SB | M) = 22/197 = 0,1117$$

$$P(SB | TM) = 0/13 = 0$$

$$P(B | M) = 107/197 = 0,5431$$

$$P(B | TM) = 1/13 = 0,0769$$

$$P(C | M) = 57/197 = 0,2893$$

$$P(C | TM) = 4/13 = 0,3077$$

$$P(K | M) = 11/197 = 0,0558$$

$$P(K | TM) = 8/13 = 0,6154$$

$$P(SK | M) = 0/197 = 0$$

$$P(SK | TM) = 0/13 = 0$$

precision: 100.00% (positive class: Tidak Mampu)

	true Mampu	true Tidak Mampu	class precision
pred. Mampu	82	0	100.00%
pred. Tidak Mampu	0	8	100.00%
class recall	100.00%	100.00%	

Gambar 4. Nilai Precision

4) **Recall**

Hasil pengujian menunjukkan bahwa nilai recall adalah 100%, seperti yang ditunjukkan pada Gambar 5. Nilai ini menggambarkan efektivitas model dalam mengidentifikasi data yang sebenarnya termasuk dalam kategori positif.

recall: 100.00% (positive class: Tidak Mampu)

	true Mampu	true Tidak Mampu	class precision
pred. Mampu	82	0	100.00%
pred. Tidak Mampu	0	8	100.00%
class recall	100.00%	100.00%	

Gambar 5. Nilai Recall

5) **Performance Vector**

Hasil pengujian dengan RapidMiner menunjukkan bahwa model Naïve Bayes yang dibuat memiliki performa yang sangat baik, dengan nilai akurasi, precision, dan recall masing-masing sebesar 100%. Hal ini mengindikasikan bahwa model mampu memprediksi semua data tanpa kesalahan, baik dalam mengklasifikasikan kelas positif maupun negatif.

PerformanceVector

```

PerformanceVector:
accuracy: 100.00%
ConfusionMatrix:
True:  Mampu  Tidak Mampu
Mampu: 82      0
Tidak Mampu: 0      8
precision: 100.00% (positive class: Tidak Mampu)
ConfusionMatrix:
True:  Mampu  Tidak Mampu
Mampu: 82      0
Tidak Mampu: 0      8
recall: 100.00% (positive class: Tidak Mampu)
ConfusionMatrix:
True:  Mampu  Tidak Mampu
Mampu: 82      0
Tidak Mampu: 0      8
AUC (optimistic): 1.000 (positive class: Tidak Mampu)
AUC: 1.000 (positive class: Tidak Mampu)
AUC (pessimistic): 1.000 (positive class: Tidak Mampu)
    
```

Gambar 6. Performance Vector

4. **KESIMPULAN**

Hasil pengujian pada penelitian ini menunjukkan bahwa penerapan algoritma Naïve Bayes berhasil dilakukan untuk memprediksi kemampuan pemrograman mahasiswa Teknik Informatika berdasarkan pemahaman dasar algoritma pemrograman sebagai parameter. Hasil pengujian melibatkan 210 responden data training serta 90 responden data testing menunjukkan bahwa model dapat memprediksi dengan akurasi sebesar 100%. Hasil akhir yang diperoleh untuk prediksi Mampu dan kenyataannya benar Mampu sebanyak 82 data. Jumlah prediksi Mampu dan kenyataannya benar Tidak Mampu sebanyak 0 data. Jumlah prediksi Tidak Mampu dan kenyataannya benar Mampu sebanyak 0 data dan jumlah prediksi Tidak Mampu dengan kenyataannya benar Tidak Mampu sebanyak 8 data. Secara keseluruhan, model naïve bayes berhasil mencapai akurasi sebesar 100% dengan recall benar Mampu dan benar Tidak Mampu 100% serta precision prediksi

Mampu dan prediksi Tidak Mampu 100%. Hasil ini mengindikasikan bahwa algoritma Naïve Bayes sangat efektif dalam mengklasifikasikan kemampuan mahasiswa. Proses pembersihan data, konversi skala likert, hingga evaluasi model yang menyeluruh memberikan hasil yang dapat diandalkan.

Hasil penelitian ini dapat membantu institusi pendidikan, khususnya di bidang teknik informatika, dalam mengembangkan metode atau berupa alat bantu seperti aplikasi pembelajaran untuk mengevaluasi kemampuan pemrograman mahasiswa. Hal ini juga dapat digunakan untuk mengidentifikasi mahasiswa yang membutuhkan bimbingan tambahan sehingga dapat meningkatkan kualitas pembelajaran secara keseluruhan.

5. SARAN

Meskipun hasil penelitian ini menunjukkan performa model yang sangat baik, beberapa saran dapat diajukan untuk penelitian lanjutan:

1. Peningkatan Jumlah Dataset: Menggunakan dataset yang lebih besar dan lebih beragam untuk menguji model terhadap variasi pola data yang lebih kompleks.
2. Perbandingan Algoritma: Membandingkan kinerja algoritma Naïve Bayes dengan algoritma lain, seperti Decision Tree, Random Forest, atau Support Vector Machine, untuk melihat keunggulan setiap metode.
3. Faktor Eksternal: Mempertimbangkan faktor lain, seperti gaya belajar, tingkat partisipasi dalam kegiatan akademik, atau hasil praktikum, untuk meningkatkan akurasi prediksi.

Penelitian ini membuka peluang untuk pengembangan lebih lanjut dalam penerapan model machine learning di bidang pendidikan, khususnya dalam evaluasi kemampuan pemrograman mahasiswa supaya lebih memadai.

DAFTAR PUSTAKA

- [1] A. Çetinkaya and Ö. K. Baykan, "Prediction of middle school students' programming talent using artificial neural networks," *Eng. Sci. Technol. an Int. J.*, vol. 23, no. 6, pp. 1301–1307, 2020, doi: 10.1016/j.jestch.2020.07.005.
- [2] R. Harimurti, E. Ekohariadi, M. Munoto, I. G. P. Asto B, and E. T. Winanti, "Analysis of Programming Skills Concept in Developing Problem Solving Skills," *J. Pendidik. Teknol. dan Kejur.*, vol. 25, no. 1, pp. 43–51, 2019, doi: 10.21831/jptk.v25i1.22638.
- [3] Z. Zulkipli, "Hubungan antara Kemampuan Matematika dengan Keterampilan Pemrograman," *J. Bangkit Indones.*, vol. 12, no. 2, pp. 59–64, 2023, doi: 10.52771/bangkitindonesia.v12i2.251.
- [4] I. Tarsini and R. Anggraeni, "Explore flowchart and pseudocode concepts in algorithms and programming," *Indones. J. Multidiscip. Sci.*, vol. 3, no. 5, 2024, doi: 10.55324/ijoms.v3i5.807.
- [5] Y. G. Nengsih *et al.*, *Konsep Algoritma dan Pemrograman Mengenal Konsep Dasar dan Praktis dalam Bahasa Pascal dan C*. INDIE PRESS.
- [6] S. Sinaga and Y. Yahfizham, "Analisis Studi Literatur Pemahaman Mahasiswa Tentang Algoritma Pemrograman," vol. 2, no. 1, 2024, [Online]. Available: <https://doi.org/10.59841/saber.v2i1.603>
- [7] A. B. Chaudhuri, *Flowchart and Algorithm Basics*. Mercury Learning and Information.
- [8] S. Hartati and H. A. SAN, "Algoritma Naive Bayes untuk Prediksi Kelulusan Mahasiswa," *J. Cakrawala Inf.*, vol. 2, no. 2, pp. 42–50, 2022, doi: 10.54066/jci.v2i2.234.
- [9] M. Asfi and N. Fitrianiingsih, "Implementasi Algoritma Naive Bayes Classifier sebagai Sistem Rekomendasi Pembimbing Skripsi," *J. Nas. Inform. dan Teknol. Jar.*, vol. 5, pp. 45–50, 2020, [Online]. Available: <https://jurnal.uisu.ac.id/index.php/infotekjar/article/view/2536>
- [10] R. Annisa and A. Sasongko, "Prediksi Nilai Akademik Mahasiswa Menggunakan Algoritma Naïve Bayes," *JST (Jurnal Sains dan Teknol.)*, vol. 9, no. 1, pp. 1–10, 2020, doi: 10.23887/jstundiksha.v9i1.19488.
- [11] D. R. Sari, D. Hartama, I. S. Damanik, and A. Wanto, "Penerapan Metode Naive Bayes dalam Memprediksi Kepuasan Mahasiswa Terhadap Cara Pengajaran Dosen," *Pros. Semin. Nas. Ris. Inf. Sci.*, vol. 1, no. 2, p. 287, 2019, doi: 10.30645/senaris.v1i0.34.
- [12] M. R. Qisthiano, T. B. Kurniawan, E. S. Negara, and M. Akbar, "Pengembangan Model Untuk Prediksi Tingkat Kelulusan Mahasiswa Tepat Waktu dengan Metode Naive Bayes," *J. Media Inform. Budidarma*, vol. 5, no. 3, p. 987, 2021, doi: 10.30865/mib.v5i3.3030.
- [13] N. M. A. Mahar, Vihi Atina, and Nugroho Arif Sudibyo, "Pemodelan Prediksi Kelulusan Mahasiswa Dengan Metode Naïve Bayes Di Uniba," *J. Manaj. Inform. dan Sist. Inf.*, vol. 6, no. 2, pp. 148–158,

- 2023, doi: 10.36595/misi.v6i2.875.
- [14] E. Sutinah, N. Agustina, and Martini, "Algoritma Naive Bayes untuk Prediksi Keberhasilan Mahasiswa pada Mata Kuliah Praktikum," *J. Tek. Inform. Unika ST. Thomas*, vol. 08, no. 2, pp. 275–286, 2023.
- [15] A. A. Permana *et al.*, "Implementasi Algoritma Naive Bayes Untuk Prediksi Kelulusan Mahasiswa," vol. 13, no. 01, pp. 65–70, 2024.
- [16] A. A. Botara and A. Sabri, "Prediksi Masa Studi Mahasiswa Menggunakan Algoritma Naive Bayes pada Universitas Halmahera," vol. 08, no. 02, pp. 1–14, 2024.
- [17] C. R. Batuallo, A. Mewengkang, and D. R. Kaparang, "Klasifikasi Pencapaian Nilai Mahasiswa Berdasarkan Jenis Mata Kuliah Menggunakan Algoritma Naive Bayesian Classifisier," *Eduetik J. Pendidik. Teknol. Inf. dan Komun.*, vol. 3, no. 4, pp. 499–511, 2023, doi: 10.53682/edutik.v3i4.7606.
- [18] R. Angella, C. Walangare, and B. Sujatmiko, "Penerapan Algoritma Naive Bayes Dalam Sistem Pendukung Keputusan Pemilihan Peminatan Konsentrasi Berdasarkan Nilai Akademik Berbasis Web Pada Program Studi S1 Pendidikan Teknologi Informasi," *J. IT-EDU*, vol. 7, no. 1, pp. 74–83, 2022.