

**[CODE STRESS DETECTION] : ANALISIS DAMPAK BEBAN  
PROGRAMMING TERHADAP KESEHATAN MENTAL MAHASISWA  
TI MENGGUNAKAN METODE NAIVE BAYES**

**Silvia Dini Nuraini<sup>1</sup>, Tasya S<sup>2</sup>, Bayu Syabana<sup>3</sup>, Untung Rohwadi<sup>4</sup>, Sigit Alsyukuri  
Wibawa<sup>5</sup>**

<sup>1,2,3,4,5</sup>Universitas Bina Sarana Informatika

Email: [silviadinin@gmail.com](mailto:silviadinin@gmail.com)<sup>1</sup>, [tasyzesica05@gmail.com](mailto:tasyazesica05@gmail.com)<sup>2</sup>, [bayusyabana86@gmail.com](mailto:bayusyabana86@gmail.com)<sup>3</sup>,  
[untung.unr@bsi.ac.id](mailto:untung.unr@bsi.ac.id)<sup>4</sup>, [sigit.stb@bsi.ac.id](mailto:sigit.stb@bsi.ac.id)<sup>5</sup>

**ABSTRAK**

Kesehatan mental mahasiswa merupakan aspek penting dalam pendidikan tinggi, khususnya pada mahasiswa Teknik Informatika yang sering menghadapi aktivitas pemrograman dengan tingkat kompleksitas tinggi. Beban tugas yang besar, tekanan waktu, serta kesulitan logika dalam pemrograman dapat memicu stres yang berdampak pada kondisi psikologis dan performa akademik mahasiswa. Penelitian ini bertujuan untuk menganalisis dampak beban programming terhadap tingkat stres mahasiswa serta mengembangkan aplikasi berbasis web yang mampu mendeteksi tingkat stres secara otomatis menggunakan metode Naive Bayes. Data diperoleh melalui interaksi pengguna pada aplikasi, meliputi durasi coding, intensitas aktivitas pemrograman, frekuensi error, dan waktu istirahat. Data tersebut kemudian diklasifikasikan ke dalam tiga kategori tingkat stres, yaitu rendah, sedang, dan tinggi. Metode Naive Bayes dipilih karena efisien dalam melakukan klasifikasi berbasis probabilitas. Hasil pengujian menunjukkan bahwa sistem mampu mendeteksi tingkat stres mahasiswa dengan tingkat akurasi yang cukup baik, sehingga aplikasi ini dapat digunakan sebagai alat bantu pemantauan stres secara awal. Diharapkan penelitian ini dapat meningkatkan kesadaran mahasiswa terhadap kondisi kesehatan mental selama menjalani aktivitas pemrograman.

**Kata Kunci :** Deteksi Stres, Kesehatan Mental, Mahasiswa Ti, Beban Programming, Naive Bayes, Aplikasi Web.

**ABSTRACT**

*Student mental health is a crucial aspect of higher education, particularly for Information Technology (IT) students, who frequently encounter complex programming tasks. Heavy workloads, time pressures, and logical programming difficulties can trigger stress, impacting students' psychological well-being and academic performance. This study aims to analyze the impact of programming stress on students' stress levels and develop a web-based application capable of automatically detecting stress levels using the Naive Bayes method. Data was obtained through user interaction within the application, including coding duration, programming activity intensity, error frequency, and rest periods. The data was then classified into three stress levels: low, medium, and high. The Naive Bayes method was chosen due to its efficiency in probability-based classification. Test results indicate that the system is able to detect students' stress levels with a fairly high level of accuracy, enabling this application to*

*be used as an early stress monitoring tool. It is hoped that this research will increase students' awareness of mental health during programming activities.*

**Keywords:** Stress Detection, Mental Health, IT Students, Programming Stress, Naive Bayes, Web Applicatio.

---

## PENDAHULUAN

Kesehatan mental merupakan salah satu faktor penting yang berpengaruh terhadap keberhasilan akademik mahasiswa, khususnya mahasiswa Teknik Informatika yang kerap menghadapi tekanan tinggi dalam aktivitas pemrograman. Kompleksitas tugas, tuntutan penyelesaian proyek dalam waktu terbatas, serta tingkat kesulitan logika yang tinggi sering kali memicu stres berkepanjangan. Kondisi ini dapat berdampak pada menurunnya konsentrasi, motivasi belajar, dan capaian akademik mahasiswa. Apabila tidak dikelola dengan baik, stres tersebut berpotensi berkembang menjadi gangguan psikologis seperti kecemasan dan kelelahan emosional yang menghambat proses belajar.

Mahasiswa Teknik Informatika cenderung memiliki risiko stres yang lebih besar karena aktivitas pemrograman menuntut kemampuan analisis dan pemecahan masalah secara terus-menerus. Kesalahan kode yang berulang serta beban proyek yang besar sering menimbulkan rasa frustrasi, sulit fokus, dan penurunan performa akademik. Kondisi ini menunjukkan pentingnya adanya sarana yang mampu membantu mahasiswa mengenali tingkat stres sejak dini, sehingga upaya penanganan dapat dilakukan lebih cepat sebelum berdampak lebih luas terhadap kesehatan mental dan prestasi belajar.

Berdasarkan permasalahan tersebut, penelitian ini mengembangkan sebuah aplikasi web Code Stress Detection yang bertujuan untuk mendeteksi tingkat stres mahasiswa berdasarkan aktivitas pemrograman. Metode Naive Bayes digunakan dalam penelitian ini karena kemampuannya dalam melakukan klasifikasi berbasis probabilitas secara efisien. Melalui aplikasi ini, mahasiswa diharapkan dapat memantau kondisi stres secara mandiri, meningkatkan kesadaran terhadap kesehatan mental, serta mengelola beban belajar dengan lebih seimbang selama proses pemrograman. Dengan demikian, sistem yang dikembangkan dapat berperan sebagai sarana deteksi awal dalam upaya pencegahan stres berlebih pada mahasiswa Teknik Informatika.

## Rumusan Masalah

Berdasarkan uraian latar belakang di atas, maka rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana merancang dan mengimplementasikan aplikasi web Code Stress Detection yang dapat mendeteksi tingkat stres mahasiswa Teknik Informatika berdasarkan aktivitas pemrograman yang dilakukan?
2. Bagaimana penerapan metode Naive Bayes dalam melakukan klasifikasi tingkat stres berdasarkan data aktivitas pemrograman?
3. Seberapa akurat hasil klasifikasi yang dihasilkan oleh metode Naive Bayes dalam mendeteksi tingkat stres mahasiswa?

### **Batasan Masalah**

Dalam upaya memfokuskan penelitian ini agar pembahasan lebih terarah dan terukur, maka ditetapkan beberapa batasan masalah sebagai berikut:

1. Penelitian ini difokuskan pada mahasiswa Program Studi Teknik Informatika/Teknologi Informasi yang sedang menempuh mata kuliah pemrograman sebagai responden penelitian, sehingga hasil analisis merepresentasikan tingkat stres yang muncul akibat aktivitas pemrograman akademik.
2. Sistem yang dikembangkan berupa sistem pendekripsi stres berbasis web bernama Code Stress Detection, yang berfungsi untuk mendekripsi tingkat stres mahasiswa berdasarkan data aktivitas pemrograman, seperti durasi penggeraan tugas, jumlah kesalahan (error), dan frekuensi perbaikan kode.
3. Metode klasifikasi yang digunakan dalam penelitian ini adalah Naive Bayes, yang digunakan untuk mengelompokkan tingkat code stress mahasiswa ke dalam tiga kategori, yaitu stres rendah, stres sedang, dan stres tinggi.
4. Sistem dikembangkan menggunakan bahasa pemrograman PHP dengan framework CodeIgniter 3 serta didukung oleh basis data MySQL. Penelitian ini tidak membahas aspek psikologis secara klinis, melainkan berfokus pada analisis tingkat stres berdasarkan aktivitas pemrograman yang dilakukan mahasiswa.

### **Tujuan Penelitian**

Tujuan dari penelitian ini adalah untuk mengembangkan sebuah sistem pendekripsi stres berbasis web bernama Code Stress Detection yang mampu mendekripsi tingkat stres mahasiswa

Teknik Informatika berdasarkan aktivitas pemrograman dengan menggunakan metode Naive Bayes.

Lalu poin-poinnya tetap:

1. Menganalisis faktor-faktor penyebab stres yang timbul selama proses pemrograman pada mahasiswa Teknik Informatika.
2. Menerapkan algoritma Naive Bayes dalam sistem pendekripsi stres untuk mengklasifikasikan tingkat stres mahasiswa ke dalam kategori rendah, sedang, dan tinggi.
3. Mengevaluasi kinerja sistem dalam mendekripsi tingkat stres mahasiswa berdasarkan nilai akurasi klasifikasi yang dihasilkan oleh algoritma Naive Bayes.

## **TINJAUAN PUSTAKA**

Kesehatan mental mahasiswa merupakan isu penting di era digital, dengan stres akademik memengaruhi kesejahteraan psikologis dan performa belajar jika tidak ditangani. Mahasiswa Teknik Informatika rentan stres dari tugas pemrograman analitis, termasuk frekuensi error dan deadline ketat, yang menyebabkan penurunan motivasi, gangguan tidur, kecemasan, dan performa akademik. Stres didefinisikan sebagai respon terhadap tekanan eksternal yang melebihi adaptasi individu.

Penelitian deteksi stres menggunakan metode seperti SVM untuk suara/ekspresi wajah dan Decision Tree untuk data digital, tapi terbatas oleh kompleksitas data dan perangkat tambahan. Naive Bayes dipilih karena efisiensinya dalam klasifikasi probabilistik dengan akurasi tinggi, cocok untuk pola stres dari aktivitas pemrograman seperti durasi coding dan frekuensi error.

Masih ada kesenjangan dalam sistem praktis terintegrasi dengan aktivitas programming mahasiswa TI tanpa perangkat tambahan. Penelitian ini mengisi celah tersebut dengan aplikasi web Code Stress Detection berbasis Naive Bayes untuk deteksi stres otomatis dan mandiri.

## **METODE PENELITIAN**

Metode yang dilakukan penulis dalam penelitian ini dalam mendapatkan data yang dibutuhkan sebagai berikut:

- a. Studi Literatur

Pengumpulan data dilakukan dengan meninjau berbagai sumber seperti jurnal, artikel ilmiah, buku, dan penelitian terdahulu yang berkaitan dengan stres akademik,

beban pemrograman, serta metode klasifikasi menggunakan Naive Bayes. Studi literatur ini digunakan sebagai dasar teori dan pendukung perancangan sistem.

b. Kuesioner

Data primer diperoleh melalui penyebaran kuesioner kepada mahasiswa aktif Program Studi Informatika/Teknologi Informasi di Universitas BSI. Kuesioner ini berisi pernyataan terkait gejala stres yang dialami mahasiswa saat mengerjakan atau menghadapi tugas pemrograman. Data hasil pengisian kuesioner kemudian digunakan sebagai bahan analisis serta menjadi dataset utama dalam proses pelatihan model Naive Bayes.

## **Populasi dan Sampel**

1. Populasi

Populasi dalam penelitian ini adalah mahasiswa yang sedang menempuh pendidikan pada Program Studi Informatika atau Teknologi Informasi dari berbagai perguruan tinggi. Populasi tersebut dipilih karena mahasiswa di bidang informatika umumnya memiliki intensitas aktivitas pemrograman yang tinggi, seperti pengeroaan tugas coding, proyek, dan praktikum, yang berpotensi menimbulkan stres akademik (*code stress*). Berdasarkan data yang diperoleh, jumlah populasi dalam penelitian ini adalah sebanyak 80 mahasiswa.

2. Sampel

Pengambilan sampel dalam penelitian ini dilakukan menggunakan teknik probability sampling, yaitu metode pengambilan sampel yang memberikan kesempatan yang sama bagi setiap anggota populasi untuk terpilih sebagai responden. Jenis probability sampling yang digunakan adalah simple random sampling, di mana responden dipilih secara acak dari seluruh anggota populasi tanpa memperhatikan latar belakang tertentu.

Penentuan jumlah sampel dilakukan menggunakan rumus Slovin, yang digunakan untuk menentukan ukuran sampel berdasarkan jumlah populasi dan tingkat kesalahan yang dapat ditoleransi. Rumus Slovin yang digunakan dalam penelitian ini adalah sebagai berikut:

$$n = \frac{N}{1 + N \times e^2}$$

dengan keterangan:

n = jumlah sampel

N = jumlah populasi

e = tingkat kesalahan (error tolerance)

Berdasarkan data penelitian, diketahui bahwa jumlah populasi (N) adalah 80 mahasiswa. Dengan tingkat kesalahan (e) sebesar 6% (0,06), maka perhitungan jumlah sampel adalah sebagai berikut:

$$\begin{aligned} n &= \frac{80}{1 + 80(0,06)^2} \\ n &= \frac{80}{1 + 80(0,0036)} \\ n &= \frac{80}{1 + 0,288} \\ n &= \frac{80}{1,288} \\ n &= 62,11 \approx 60 \end{aligned}$$

Berdasarkan perhitungan menggunakan rumus Slovin dengan jumlah populasi 80 mahasiswa dan tingkat kesalahan 6%, diperoleh jumlah sampel sebesar 62 responden yang kemudian dibulatkan menjadi 60 responden.

## Alur Penelitian

Penelitian ini dilakukan karena banyak mahasiswa Teknologi Informasi yang sering mengalami tekanan akibat beban programming, seperti tugas coding, proyek, dan deadline yang padat, yang dapat berdampak pada kesehatan mental mereka. Penelitian diawali dengan studi pustaka untuk memahami konsep kesehatan mental mahasiswa, *code stress*, serta metode klasifikasi yang relevan. Data dikumpulkan melalui kuesioner yang dibagikan kepada mahasiswa TI untuk mengetahui tingkat beban programming dan kondisi mental yang dirasakan. Data yang terkumpul kemudian melalui tahap pengolahan dengan memeriksa validitas data dan mengubahnya ke dalam bentuk numerik agar dapat diproses. Metode Naive Bayes dipilih karena sederhana dan efektif dalam melakukan klasifikasi data berbasis

kuesioner, kemudian diterapkan dengan membagi data menjadi data latih dan data uji untuk mengelompokkan tingkat *code stress* ke dalam kategori rendah, sedang, dan tinggi. Hasil klasifikasi dievaluasi menggunakan confusion matrix untuk mengetahui tingkat akurasi metode yang digunakan. Penelitian ini diharapkan dapat membantu mahasiswa menyadari kondisi stres akibat aktivitas programming serta menjadi bahan evaluasi bagi pihak akademik, dengan catatan bahwa sistem ini bersifat sebagai deteksi awal dan bukan diagnosis medis.

### **Pre-processing Data**

Tahap pre-processing data dilakukan terhadap data yang diperoleh dari pengisian kuesioner berbasis web pada aplikasi Code Stress Detection. Website ini digunakan sebagai media pengumpulan data, di mana mahasiswa mengisi beberapa parameter yang berkaitan dengan aktivitas programming dan kondisi fisik sehari-hari. Data yang dikumpulkan terdiri dari jam coding per hari, jumlah bug yang ditemukan, jumlah deadline yang dihadapi, serta jam tidur per hari.

#### **1) Pemeriksaan dan Pembersihan Data (Data Cleaning)**

Pada tahap awal, dilakukan pemeriksaan terhadap data yang masuk dari website untuk memastikan seluruh data yang diinput oleh pengguna bersifat lengkap dan valid. Data yang tidak terisi, bernilai kosong, atau berada di luar batas wajar tidak digunakan dalam proses pengolahan data. Tahap ini bertujuan untuk menghindari kesalahan dalam proses klasifikasi serta menjaga kualitas data yang akan diproses menggunakan metode Naive Bayes.

#### **2) Penentuan Atribut Data**

Data yang telah lolos tahap pembersihan kemudian ditetapkan sebagai atribut penelitian. Atribut yang digunakan merupakan indikator utama dalam menentukan tingkat *code stress* mahasiswa, yaitu:

1. Jam coding per hari
2. Jumlah bug yang ditemukan
3. Jumlah deadline
4. Jam tidur per hari

Keempat atribut tersebut dipilih karena memiliki hubungan langsung dengan beban programming dan kondisi mental mahasiswa.

**3) Pengelompokan Data (Categorization)**

Data mentah yang diperoleh dari input numerik selanjutnya dikelompokkan ke dalam beberapa kategori berdasarkan rentang nilai tertentu. Pengelompokan ini bertujuan untuk menyederhanakan data dan mempermudah proses perhitungan probabilitas pada metode Naive Bayes.

**Tabel Kategori Atribut****a. Jam Coding per Hari**

Rentang Nilai	Kategori
0 – 4 jam	Rendah
5 – 8 jam	Sedang
>8 jam	Tinggi

**b. Jumlah Bug Ditemukan**

Rentang Nilai	Kategori
0 – 5	Sedikit
6 - 15	Sedang
>15	Banyak

**c. Deadline**

Rentang Nilai	Kategori
0 - 2	Sedikit
3 - 5	Sedang
>5	Banyak

**d. Jam Tidur per Hari**

Rentang Nilai	Kategori
<5 jam	Kurang
5 – 7 jam	Cukup
>7 jam	Baik

**4) Transformasi Data (Numerical Encoding)**

Setelah data dikelompokkan, setiap kategori diubah ke dalam bentuk numerik agar dapat diproses oleh algoritma Naive Bayes. Transformasi ini dilakukan untuk menyeragamkan format data dan mempermudah perhitungan probabilitas.

Tabel Transformasi Data

Kategori	Nilai Numerik
Rendah / Sedikit / Baik	1
Sedang / Cukup	2
Tinggi / Banyak / Kurang	3

#### 5) Penentuan Label Kelas (Code Stress Level)

Data yang telah ditransformasikan kemudian digunakan untuk menentukan label kelas tingkat code stress. Tingkat code stress diklasifikasikan ke dalam tiga kelas, yaitu:

1. Stress Rendah
2. Stress Sedang
3. Stress Tinggi

Label kelas ini berfungsi sebagai output dalam proses klasifikasi menggunakan metode Naive Bayes.

#### 6) Pembagian Data

Tahap akhir dari pre-processing data adalah pembagian data menjadi data latih dan data uji. Data latih digunakan untuk membangun model Naive Bayes, sedangkan data uji digunakan untuk menguji performa model dalam mengklasifikasikan tingkat code stress mahasiswa.

### Evaluasi Model (Confusion Matrix)

Evaluasi model dilakukan untuk menilai kemampuan metode Naive Bayes dalam mengklasifikasikan tingkat code stress mahasiswa Teknologi Informasi berdasarkan data kuesioner berbasis web. Evaluasi ini bertujuan untuk mengetahui tingkat ketepatan hasil klasifikasi yang dihasilkan oleh model.

#### 1) Confusion Matrix

Evaluasi model pada penelitian ini menggunakan confusion matrix, yaitu metode evaluasi yang membandingkan hasil prediksi sistem dengan data aktual. Confusion matrix digunakan

untuk mengetahui jumlah data yang berhasil diklasifikasikan dengan benar serta data yang mengalami kesalahan klasifikasi pada setiap kelas code stress.

Dalam penelitian ini, tingkat code stress dibagi ke dalam tiga kelas, yaitu:

1. Stress Rendah
2. Stress Sedang
3. Stress Tinggi

Tabel confusion matrix yang digunakan dalam evaluasi model disajikan sebagai berikut:

Aktual \ Prediksi	Stress Rendah	Stress Sedang	Stress Tinggi
Stress Rendah	a	b	c
Stress Sedang	d	e	f
Stress Tinggi	g	h	i

Keterangan:

- a, e, i menunjukkan jumlah data yang berhasil diklasifikasikan dengan benar sesuai dengan kelas aktualnya.
- b, c, d, f, g, h menunjukkan jumlah data yang mengalami kesalahan klasifikasi, yaitu data yang diprediksi tidak sesuai dengan kelas aktualnya.

Confusion matrix ini digunakan sebagai dasar dalam perhitungan nilai akurasi model serta untuk menganalisis kesalahan klasifikasi yang terjadi pada masing-masing kelas code stress.

## 2) Perhitungan Akurasi

Akurasi digunakan untuk mengukur tingkat ketepatan model dalam mengklasifikasikan seluruh data uji. Nilai akurasi diperoleh dari perbandingan jumlah prediksi yang benar dengan total data uji yang digunakan.

$$\text{Akurasi} = \frac{\text{Jumlah prediksi benar}}{\text{Total data uji}}$$

Nilai akurasi yang tinggi menunjukkan bahwa metode Naive Bayes memiliki kemampuan yang baik dalam mendeteksi tingkat code stress mahasiswa berdasarkan data yang tersedia.

### 3) Presisi dan Recall

Selain akurasi, evaluasi model juga dilakukan menggunakan nilai presisi dan recall untuk menilai kualitas prediksi pada masing-masing kelas code stress. Presisi menunjukkan tingkat ketepatan model dalam memprediksi suatu kelas tertentu, sedangkan recall menunjukkan kemampuan model dalam mengenali seluruh data aktual pada kelas tersebut. Penggunaan presisi dan recall bertujuan untuk memastikan bahwa model tidak hanya akurat secara keseluruhan, tetapi juga mampu mengklasifikasikan setiap tingkat code stress dengan baik.

### 4) Analisis dan Interpretasi Hasil Evaluasi

Berdasarkan hasil evaluasi menggunakan confusion matrix, metode Naive Bayes menunjukkan performa yang cukup baik dalam mengklasifikasikan tingkat code stress mahasiswa. Sebagian besar data uji dapat diprediksi sesuai dengan kelas aktualnya, meskipun masih terdapat beberapa kesalahan klasifikasi, khususnya pada kelas stress sedang dan stress tinggi yang memiliki karakteristik data yang saling berdekatan.

Hasil evaluasi ini menunjukkan bahwa metode Naive Bayes layak digunakan sebagai metode deteksi awal code stress mahasiswa. Namun demikian, hasil yang diperoleh tidak dimaksudkan sebagai diagnosis medis, melainkan sebagai alat bantu untuk memberikan gambaran awal mengenai kondisi stres mahasiswa berdasarkan aktivitas pemrograman yang mereka lakukan.

## **HASIL DAN PEMBAHASAN**

### 1. Hasil Klasifikasi Tingkat Code Stress

Berdasarkan penerapan metode Naive Bayes terhadap data uji yang telah melalui tahap pre-processing, diperoleh hasil klasifikasi tingkat code stress mahasiswa Teknologi Informasi ke dalam tiga kategori, yaitu stress rendah, stress sedang, dan stress tinggi. Hasil klasifikasi ini menunjukkan bahwa aktivitas pemrograman yang dijalani mahasiswa memiliki keterkaitan dengan kondisi kesehatan mental yang mereka alami.

Sebagian besar data uji dapat diklasifikasikan dengan tepat oleh sistem. Namun demikian, masih terdapat beberapa data yang mengalami kesalahan klasifikasi, khususnya pada kelas stress sedang. Hal ini disebabkan oleh karakteristik kelas stress sedang yang

berada di antara stress rendah dan stress tinggi, sehingga memiliki pola data yang relatif berdekatan.

## 2. Pembahasan Hasil Evaluasi Model

Hasil evaluasi menggunakan confusion matrix menunjukkan bahwa metode Naive Bayes mampu mengklasifikasikan tingkat code stress mahasiswa dengan tingkat akurasi yang cukup baik. Nilai akurasi yang diperoleh mengindikasikan bahwa model dapat mengenali pola hubungan antara beban programming dan kondisi mental mahasiswa berdasarkan indikator yang digunakan dalam penelitian.

Kesalahan klasifikasi yang terjadi umumnya disebabkan oleh kemiripan nilai atribut antar kelas, terutama pada responden dengan jam coding dan jumlah deadline yang berada pada batas kategori tertentu. Kondisi ini menunjukkan bahwa tingkat stres mahasiswa tidak bersifat kaku, melainkan berada pada suatu spektrum yang dapat berubah tergantung pada situasi akademik yang dihadapi.

## 3. Pengaruh Beban Programming terhadap Kesehatan Mental

Berdasarkan hasil klasifikasi, mahasiswa dengan jam coding yang tinggi, jumlah bug yang banyak, serta tekanan deadline yang padat cenderung berada pada kategori stress sedang hingga stress tinggi. Sebaliknya, mahasiswa dengan jam coding yang lebih terkontrol dan waktu istirahat yang cukup umumnya berada pada kategori stress rendah.

Temuan ini menunjukkan bahwa beban programming yang berlebihan dapat berdampak langsung pada kondisi kesehatan mental mahasiswa, khususnya dalam bentuk kelelahan mental dan tekanan psikologis. Oleh karena itu, pengelolaan waktu belajar serta keseimbangan antara aktivitas akademik dan istirahat menjadi faktor penting dalam upaya menurunkan tingkat code stress.

## 4. Keterkaitan Hasil dengan Metode Naive Bayes

Penerapan metode Naive Bayes dalam penelitian ini terbukti efektif untuk melakukan deteksi awal tingkat code stress mahasiswa. Metode ini mampu memanfaatkan data kuesioner berbasis web untuk menghasilkan klasifikasi yang relatif akurat dan mudah dipahami.

Keunggulan metode Naive Bayes terletak pada kesederhanaan proses perhitungan dan kemampuannya dalam menangani data berbasis kategori. Hal tersebut menjadikan

metode ini sesuai digunakan pada penelitian yang berfokus pada analisis kondisi mental mahasiswa berdasarkan aktivitas pemrograman sehari-hari.

### 5. Implikasi Penelitian

Hasil penelitian ini dapat dimanfaatkan sebagai bahan evaluasi bagi mahasiswa maupun pihak akademik dalam memahami dampak beban programming terhadap kesehatan mental. Sistem Code Stress Detection yang dikembangkan dapat berfungsi sebagai alat bantu deteksi awal untuk meningkatkan kesadaran mahasiswa terhadap kondisi stres yang mereka alami.

Meskipun demikian, hasil yang diperoleh dari sistem ini tidak dimaksudkan sebagai diagnosis medis. Oleh karena itu, apabila ditemukan tingkat stres yang tinggi, mahasiswa tetap disarankan untuk mendapatkan pendampingan dari tenaga profesional di bidang kesehatan mental

## KESIMPULAN DAN SARAN

### Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa beban programming memiliki pengaruh terhadap kesehatan mental mahasiswa Teknologi Informasi. Aktivitas seperti jam coding yang tinggi, banyaknya bug yang ditemukan, tekanan deadline, serta kurangnya waktu tidur dapat meningkatkan tingkat code stress yang dialami mahasiswa. Hal ini menunjukkan bahwa aktivitas akademik di bidang pemrograman perlu mendapat perhatian tidak hanya dari sisi akademik, tetapi juga dari sisi kesehatan mental.

Penerapan metode Naive Bayes dalam penelitian ini mampu mengklasifikasikan tingkat code stress mahasiswa ke dalam kategori stress rendah, sedang, dan tinggi dengan hasil yang cukup baik. Hasil evaluasi menggunakan confusion matrix menunjukkan bahwa metode Naive Bayes dapat digunakan sebagai metode klasifikasi yang efektif untuk deteksi awal code stress berdasarkan data kuesioner berbasis web.

Website Code Stress Detection yang dibangun juga berhasil digunakan sebagai media pengumpulan data sekaligus penyampaian hasil analisis kepada mahasiswa. Sistem ini dapat membantu mahasiswa untuk lebih menyadari kondisi mental yang mereka alami akibat beban

programming. Namun, hasil yang diberikan oleh sistem bersifat sebagai deteksi awal dan tidak dapat dijadikan sebagai diagnosis medis.

### Saran

Berdasarkan hasil penelitian yang telah dilakukan, terdapat beberapa saran yang dapat diberikan untuk pengembangan selanjutnya. Pertama, penelitian selanjutnya dapat menambahkan lebih banyak indikator yang berkaitan dengan kondisi mental mahasiswa, seperti tingkat kelelahan emosional, tekanan akademik non-programming, atau faktor lingkungan belajar. Dengan penambahan indikator tersebut, hasil klasifikasi diharapkan dapat menjadi lebih akurat.

Kedua, metode klasifikasi yang digunakan dapat dikembangkan dengan membandingkan Naive Bayes dengan metode lain, seperti K-Nearest Neighbor atau Support Vector Machine, untuk mengetahui metode yang memiliki performa terbaik dalam mendeteksi code stress. Selain itu, jumlah data responden juga dapat diperbanyak agar model yang dihasilkan memiliki kemampuan generalisasi yang lebih baik.

Ketiga, sistem berbasis web yang telah dibangun dapat dikembangkan lebih lanjut dengan menambahkan fitur rekomendasi atau saran penanganan awal sesuai dengan tingkat code stress yang terdeteksi. Dengan adanya fitur tersebut, sistem tidak hanya berfungsi sebagai alat deteksi, tetapi juga sebagai media edukasi bagi mahasiswa dalam menjaga kesehatan mental selama menjalani aktivitas programming.

### DAFTAR PUSTAKA

- Yusof, N., & Rahman, A. A. (2018). Academic workload and stress among computer science students. *International Journal of Education*, 10(3), 45–52.
- Shah, H., & Patel, S. (2020). Impact of programming assignments on student stress levels. *Journal of Computer Science Education*, 28(1), 1–10.
- Wijaya, A., & Nugroho, R. (2021). Analisis beban tugas pemrograman terhadap stres mahasiswa TI. *Jurnal Teknologi Informasi*, 9(2), 87–95.
- Putri, R. A., & Lestari, S. (2019). Analisis tingkat stres akademik pada mahasiswa. *Jurnal Psikologi*, 15(2), 123–131.
- Rahmawati, L., & Handayani, S. (2021). Analisis stres akademik mahasiswa berbasis kuesioner. *Jurnal Psikologi Pendidikan*, 10(2), 89–97.\

Misra, R., & McKean, M. (2000). College students' academic stress and its relation to anxiety, time management, and leisure satisfaction. *American Journal of Health Studies*, 16(1), 41–51.

Firmansyah, R., & Kurniawan, D. (2022). Implementasi algoritma Naive Bayes dalam sistem pendukung keputusan berbasis web. *Jurnal Sistem Informasi*, 8(1), 33–41.

Manajemen, J., Informasi, S., Widyawati, N. G., Khasanah, M., & Rasywir, E. (2022). Prediksi Tingkat Stress Pada Mahasiswa Universitas Dinamika Bangsa Jambi Dalam Melakukan Perkuliahan Metode Hybrid Menggunakan Algoritma Naive Bayes *Jurnal Manajemen Teknologi dan Sistem Informasi ( JMS )*. 1(April), 99–110.

Risa, D. F., Pradana, F., Bachtiar, F. A., & Brawijaya, U. (2021). IMPLEMENTASI METODE NAÏVE BAYES UNTUK MENDETEKSI STRES SISWA IMPLEMENTATION OF THE NAÏVE BAYES METHOD FOR DETECTING STUDENT. 8(6). <https://doi.org/10.25126/jtiik.202184372>

Risiko, D., Dan, D., Mahasiswa, K., Informatika, T., & Malikussaleh, U. (2015). SENASTIKA Universitas Malikussaleh. 1–7.

Susanti, L., Studi, P., Informatika, T., Miner, R., & Bayes, N. (2024). KLASIFIKASI TINGKAT STRESS PADA MAHASISWA PERKULIAHAN METODE HYBRID MENGGUNAKAN. 8(3), 243–248.

Moningka, N., Hafidurrohman, M., & R, W. A. T. (2023). Klasifikasi Mental Mahasiswa Menggunakan Metode Machine Learning. 1(2).

Rekayasa, T., Lunak, P., Komputer, T., & Medan, P. N. (2024). IMPLEMENTASI METODE NAÏVE BAYES DALAM SISTEM DETEKSI KESEHATAN MENTAL MAHASISWA TERHADAP ANXIETY DISORDER. 449–464