



Department of Digital Business

**Journal of Artificial Intelligence and Digital Business (RIGGS)**

Homepage: <https://journal.ilmudata.co.id/index.php/RIGGS>

Vol. 4 No. 4 (2025) pp: 7823-7831

P-ISSN: 2963-9298, e-ISSN: 2963-914X

## Integrasi Model Klasifikasi Tingkat Stress Mahasiswa Berbasis Natural Language Processing

Irfandi Rusdiansyah<sup>1</sup>, Ridwan Pangestu<sup>2</sup>, Devina Azalia<sup>3</sup>, Muhammad Faiz Zhafran<sup>4</sup>, Ferdy Saputra<sup>5</sup>, Fachri Amsury<sup>6</sup>

<sup>1</sup>Teknologi Informasi, Fakultas Teknologi Informasi, Universitas Bina Sarana Informatika

<sup>1</sup>[17230538@bsi.ac.id](mailto:17230538@bsi.ac.id), <sup>2</sup>[17230449@bsi.ac.id](mailto:17230449@bsi.ac.id), <sup>3</sup>[17230488@bsi.ac.id](mailto:17230488@bsi.ac.id), <sup>4</sup>[17230518@bsi.ac.id](mailto:17230518@bsi.ac.id), <sup>5</sup>[17230100@bsi.ac.id](mailto:17230100@bsi.ac.id),

<sup>6</sup>[fachri.fcv@bsi.ac.id](mailto:fachri.fcv@bsi.ac.id)

### Abstrak

Penelitian ini membahas pengembangan dan integrasi sistem klasifikasi tingkat stres mahasiswa berbasis teks menggunakan pendekatan Natural Language Processing (NLP) dan Machine Learning (ML) yang diimplementasikan pada aplikasi mobile. Tingkat stres mahasiswa merupakan permasalahan penting dalam dunia pendidikan tinggi karena berdampak langsung terhadap prestasi akademik dan kesehatan mental. Sistem yang diusulkan mengolah teks bebas yang ditulis oleh mahasiswa untuk memprediksi tingkat stres ke dalam tiga kategori, yaitu rendah, sedang, dan tinggi. Tahapan penelitian meliputi pengumpulan data teks, pra-pemrosesan menggunakan teknik NLP seperti case folding, tokenisasi, stopword removal, dan stemming, serta ekstraksi fitur menggunakan metode Term Frequency-Inverse Document Frequency (TF-IDF). Beberapa algoritma machine learning, yaitu Logistic Regression, Naive Bayes, Random Forest, dan Support Vector Machine, diuji untuk memperoleh model terbaik. Hasil evaluasi menunjukkan bahwa model Logistic Regression dengan fitur TF-IDF memberikan performa terbaik dengan akurasi sebesar 88,33% dan F1-score sebesar 0,88. Model terpilih kemudian diintegrasikan ke dalam backend Python dan dihubungkan dengan aplikasi mobile berbasis Flutter melalui Application Programming Interface (API). Integrasi ini memungkinkan proses klasifikasi tingkat stres dilakukan secara real-time dan mudah diakses oleh mahasiswa. Penelitian ini menunjukkan bahwa integrasi NLP dan ML dalam aplikasi mobile dapat menjadi solusi inovatif untuk deteksi dini stres mahasiswa serta berpotensi mendukung upaya pencegahan masalah kesehatan mental di lingkungan perguruan tinggi.

**Kata Kunci:** Natural Language Processing, Machine Learning, Klasifikasi Stres, Aplikasi Flutter, Python, Mahasiswa

### 1. Pendahuluan

Kesehatan mental, khususnya stres, merupakan isu krusial di kalangan mahasiswa seiring dengan meningkatnya tuntutan akademik, tekanan sosial, serta fase transisi menuju kedewasaan. Stres yang tidak terdeteksi dan tidak tertangani dengan baik dapat berdampak negatif terhadap prestasi akademik, kesejahteraan psikologis, dan kualitas hidup mahasiswa secara keseluruhan [1]. Perubahan sistem pendidikan yang berlangsung cepat, terutama sejak penerapan pembelajaran daring selama dan pasca pandemi COVID-19, turut memperburuk tekanan psikologis yang dialami mahasiswa [2,3]. Beberapa penelitian melaporkan bahwa sebagian besar mahasiswa mengalami stres akademik pada tingkat sedang selama pembelajaran daring, bahkan sebagian di antaranya mencapai kategori stres berat [2,3]. Kondisi ini menunjukkan pentingnya proses identifikasi dini tingkat stres secara akurat agar intervensi yang tepat dapat diberikan [4].

Seiring dengan kebutuhan tersebut, diperlukan metode deteksi dini stres yang efektif, terjangkau, dan selaras dengan perkembangan teknologi digital. Pemanfaatan Artificial Intelligence (AI), khususnya Natural Language Processing (NLP), menjadi pendekatan yang menjanjikan dalam menganalisis teks yang dihasilkan mahasiswa, seperti catatan harian, pesan, atau entri jurnal. Analisis linguistik melalui NLP memungkinkan identifikasi pola bahasa, pilihan kata, dan sentimen yang berkaitan dengan kondisi psikologis, termasuk stres [5]. Selain itu, kemampuan NLP dalam memproses data teks dalam jumlah besar secara *real-time* menawarkan keunggulan dibandingkan metode deteksi stres konvensional yang umumnya bergantung pada kuesioner manual dan bersifat *self-report* [6].

Berbagai penelitian sebelumnya telah mengkaji klasifikasi tingkat stres menggunakan pendekatan data mining dan machine learning, sering kali mengacu pada kerangka kerja Knowledge Discovery in Database (KDD) [7]. Algoritma seperti Naïve Bayes [8], C4.5 Decision Tree, dan ID3 [9] dilaporkan mampu mencapai tingkat akurasi yang cukup tinggi dalam mengklasifikasikan stres mahasiswa berdasarkan atribut tertentu, seperti beban tugas,

waktu tidur, dan relasi sosial [10]. Namun demikian, sebagian besar penelitian tersebut masih berfokus pada data numerik atau kuesioner, sehingga belum sepenuhnya menangkap ekspresi emosional mahasiswa dalam bentuk teks. Selain itu, terdapat kesenjangan penelitian terkait adaptasi model klasifikasi stres berbasis teks terhadap konteks linguistik dan ekspresi bahasa mahasiswa Indonesia, yang berpotensi memengaruhi relevansi dan akurasi hasil klasifikasi [11].

Berdasarkan kesenjangan tersebut, penelitian ini berfokus pada pengembangan dan evaluasi model klasifikasi tingkat stres mahasiswa berbasis NLP yang disesuaikan dengan konteks bahasa Indonesia. Penelitian ini menguji beberapa algoritma machine learning, yaitu *Logistic Regression*, *Naive Bayes*, *Random Forest*, dan *Support Vector Machine*, untuk memprediksi tingkat stres mahasiswa ke dalam tiga kategori, yaitu rendah, sedang, dan tinggi, berdasarkan teks yang mereka hasilkan. Model terbaik yang diperoleh kemudian diintegrasikan ke dalam backend Python dan dihubungkan dengan aplikasi mobile berbasis Flutter sebagai frontend. Integrasi sistem ini bertujuan untuk menghasilkan sistem deteksi stres yang portable, *user-friendly*, dan mampu memberikan hasil klasifikasi secara *real-time* [12]. Dengan demikian, penelitian ini diharapkan dapat menjawab sejauh mana model NLP yang disesuaikan dengan konteks bahasa Indonesia mampu mengklasifikasikan tingkat stres mahasiswa secara akurat serta bagaimana model tersebut dapat diimplementasikan dalam sistem mobile yang fungsional.

## 2. Metode Penelitian

Metodologi penelitian ini terdiri dari lima tahap yang sistematis.

### 2.1 Pengumpulan dan Pra-Pemrosesan Data

Data penelitian dikumpulkan dari teks yang merepresentasikan kondisi psikologis mahasiswa, khususnya yang berkaitan dengan indikator stres akademik. Sumber data dapat berupa teks yang disusun secara manual berdasarkan indikator psikologi, seperti kecemasan, kelelahan, dan perasaan tertekan, maupun dataset yang telah tersedia dan relevan dengan konteks bahasa Indonesia [13], [14]. Penggunaan data berbasis teks dipilih karena mampu menangkap ekspresi emosional mahasiswa secara lebih alami dibandingkan pendekatan berbasis kuesioner tertutup.

Setiap data teks yang terkumpul kemudian diberi label tingkat stres ke dalam beberapa kategori, yaitu Rendah untuk kondisi tidak atau minim stres, Sedang untuk tingkat stres menengah, dan Tinggi untuk tingkat stres berat, sehingga membentuk skema klasifikasi multi-kelas. Proses pelabelan ini dilakukan untuk mendukung penerapan metode pembelajaran terawasi (*supervised learning*) pada tahap pengembangan model klasifikasi.

Setelah proses pelabelan, data teks menjalani tahapan pra-pemrosesan menggunakan teknik *Natural Language Processing* (NLP) dengan bahasa pemrograman Python. Tahapan ini meliputi pembersihan teks dari karakter khusus dan simbol yang tidak relevan, konversi huruf menjadi huruf kecil (*case folding*), tokenisasi untuk memecah teks menjadi unit kata, serta penghapusan *stopword* guna menghilangkan kata-kata umum yang tidak memiliki kontribusi signifikan terhadap proses klasifikasi. Selanjutnya, dilakukan proses lemmatisasi atau stemming untuk mengubah kata ke bentuk dasarnya sehingga mengurangi variasi kata yang memiliki makna serupa [15].

Tahapan pra-pemrosesan ini bertujuan untuk meningkatkan kualitas dan konsistensi data teks, mengurangi noise, serta menghasilkan representasi data yang lebih efektif untuk tahap ekstraksi fitur dan pelatihan model klasifikasi. Dengan data yang telah diproses secara optimal, diharapkan model yang dibangun mampu mengenali pola linguistik yang relevan dalam membedakan tingkat stres mahasiswa secara lebih akurat.

Tabel 1. Data Teks

Teks	Label
“Saya sangat lelah dengan tugas-tugas yang menumpuk”	Tinggi
“Hari ini cukup santai dan saya bisa fokus belajar”	Rendah
“Banyak pekerjaan dan deadline menumpuk membuat saya cemas”	Sedang
“Saya merasa stress karena belum mengerjakan semua tugas”	Tinggi
“Belajar online terasa mudah hari ini”	Rendah
“Tugas kuliah menumpuk dan membuat saya bingung”	Sedang
“Saya merasa tidak mampu menyelesaikan semua pekerjaan tepat waktu”	Tinggi
“Hari ini berjalan lancar tanpa beban berlebih”	Rendah

DOI: <https://doi.org/10.31004/riggs.v4i4.4388>

Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

“Banyak ujian minggu ini membuat saya khawatir”	Sedang
“Saya senang bisa menyelesaikan tugas tepat waktu”	Rendah
“Deadline proyek membuat saya gelisah dan lelah”	Tinggi
“Belajar kelompok online membantu mengurangi stres saya”	Rendah
“Tugas yang terus bertambah membuat saya kewalahan”	Tinggi
“Hari ini cukup produktif dan saya merasa tenang”	Rendah
“Saya cemas karena belum paham materi kuliah”	Sedang
“Beban tugas akhir membuat saya sulit tidur”	Tinggi
“Minggu ini tidak banyak tugas, saya santai saja”	Rendah
“Rasanya sulit mengatur waktu antara belajar dan istirahat”	Sedang
“Saya sangat stres menghadapi banyak ujian sekaligus”	Tinggi
“Belajar pagi ini terasa ringan dan menyenangkan”	Rendah

---

## 2.2. Ekstraksi Fitur

Teks yang telah melalui tahap pra-pemrosesan selanjutnya diubah ke dalam bentuk representasi numerik agar dapat diproses oleh algoritma *machine learning*. Proses ini dilakukan melalui tahap ekstraksi fitur, yang bertujuan untuk merepresentasikan karakteristik penting dari teks dalam bentuk vektor numerik yang informatif.

Beberapa metode ekstraksi fitur digunakan dalam penelitian ini, antara lain *Bag of Words*, *Term Frequency–Inverse Document Frequency* (TF-IDF), serta pendekatan n-gram. Metode *Bag of Words* merepresentasikan teks berdasarkan frekuensi kemunculan kata tanpa mempertimbangkan urutan kata, sehingga memberikan gambaran umum mengenai distribusi kata dalam dokumen. Sementara itu, metode TF-IDF digunakan untuk memberikan bobot yang lebih tinggi pada kata-kata yang memiliki tingkat kepentingan lebih besar dalam membedakan dokumen, dengan mempertimbangkan frekuensi kemunculan kata dalam satu dokumen dan keseluruhan korpus teks.

Selain itu, pendekatan n-gram, seperti unigram, bigram, dan trigram, diterapkan untuk menangkap pola frasa atau rangkaian kata tertentu yang memiliki makna kontekstual dan berpotensi merepresentasikan ekspresi stres secara lebih akurat. Penggunaan n-gram memungkinkan model untuk mengenali hubungan antar kata yang tidak dapat ditangkap oleh representasi kata tunggal. Kombinasi metode ekstraksi fitur ini diharapkan mampu menghasilkan representasi teks yang lebih kaya dan diskriminatif, sehingga meningkatkan kinerja model dalam mengklasifikasikan tingkat stres mahasiswa [16].

## 2.3. Pembuatan Model Klasifikasi (Python)

Tahap pembuatan model klasifikasi dilakukan dengan membagi data ke dalam data latih dan data uji menggunakan metode train–test split. Pembagian data ini bertujuan untuk melatih model serta mengevaluasi kemampuan generalisasi model terhadap data yang belum pernah dilihat sebelumnya. Pada penelitian ini, beberapa algoritma *machine learning* diuji untuk menentukan model yang paling efektif dalam mengklasifikasikan tingkat stres mahasiswa berdasarkan teks, yaitu *Logistic Regression*, *Naive Bayes*, *Random Forest*, dan *Support Vector Machine* (SVM) [17].

Setiap algoritma klasifikasi dikombinasikan dengan metode ekstraksi fitur yang telah diterapkan pada tahap sebelumnya, seperti TF-IDF dan pendekatan n-gram, guna mengevaluasi pengaruh representasi fitur terhadap kinerja model. Pendekatan ini memungkinkan perbandingan performa antar algoritma secara objektif serta membantu dalam menentukan kombinasi model dan fitur yang paling optimal.

Dalam proses pengembangan model, dilakukan pula tahap awal pelabelan berbasis aturan (*rule-based*) menggunakan *Sentiment Intensity Analyzer* dari pustaka NLTK untuk mendukung proses penentuan label tingkat stres secara otomatis pada data teks. Data yang telah dilabeli kemudian direpresentasikan dalam bentuk vektor numerik menggunakan TF-IDF dan digunakan sebagai masukan untuk proses pelatihan model klasifikasi. Pendekatan ini bertujuan untuk meningkatkan konsistensi pelabelan serta mempercepat proses pembentukan dataset terlatih.

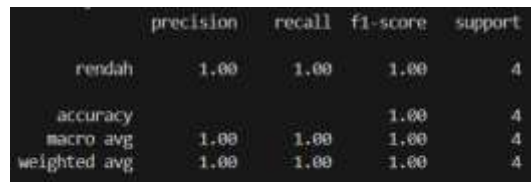
Secara keseluruhan, tahapan pembuatan model klasifikasi dirancang untuk memastikan bahwa model yang dihasilkan tidak hanya memiliki performa yang baik pada data latih, tetapi juga mampu melakukan prediksi tingkat stres secara akurat pada data baru. Model dengan kinerja terbaik selanjutnya digunakan pada tahap evaluasi dan integrasi sistem.

#### 2.4. Evaluasi Model

Evaluasi kinerja model klasifikasi dilakukan untuk mengukur tingkat keberhasilan model dalam mengklasifikasikan tingkat stres mahasiswa berdasarkan data teks. Proses evaluasi menggunakan beberapa metrik evaluasi yang umum digunakan dalam klasifikasi multi-kelas, yaitu *accuracy*, *precision*, *recall*, dan *F1-score*. Metrik-metrik tersebut dianalisis melalui *confusion matrix* dan *classification report* untuk memberikan gambaran yang lebih komprehensif mengenai performa model pada setiap kategori tingkat stres [18].

Penggunaan *accuracy* bertujuan untuk mengukur proporsi prediksi yang benar secara keseluruhan, sementara *precision* dan *recall* digunakan untuk mengevaluasi ketepatan dan kelengkapan prediksi pada masing-masing kelas. *F1-score* digunakan sebagai metrik keseimbangan antara *precision* dan *recall*, sehingga memberikan evaluasi yang lebih representatif terhadap kinerja model, terutama pada kondisi distribusi kelas yang tidak sepenuhnya seimbang.

Model yang dipilih sebagai model terbaik adalah model yang menunjukkan performa paling optimal pada data uji, ditunjukkan oleh nilai *accuracy* dan *F1-score* yang tinggi, serta memiliki selisih performa yang relatif kecil antara data latih dan data uji. Kriteria ini digunakan untuk memastikan bahwa model tidak mengalami *overfitting* dan memiliki kemampuan generalisasi yang baik terhadap data baru. Model dengan karakteristik tersebut selanjutnya digunakan pada tahap integrasi sistem untuk mendukung proses klasifikasi tingkat stres secara real-time.



	precision	recall	f1-score	support
rendah	1.00	1.00	1.00	4
accuracy			1.00	4
macro avg	1.00	1.00	1.00	4
weighted avg	1.00	1.00	1.00	4

Gambar 1. *Classification report* model terbaik *Logistic Regression* pada data uji

#### 2.5. Integrasi Sistem

Model machine learning terbaik yang diperoleh dari tahap evaluasi selanjutnya di-deploy pada *backend* berbasis Python. *Backend* ini berfungsi sebagai pusat pemrosesan yang menangani alur klasifikasi tingkat stres, mulai dari penerimaan data teks hingga pengembalian hasil prediksi. Proses integrasi dilakukan dengan memanfaatkan layanan berbasis *Application Programming Interface* (API), sehingga memungkinkan komunikasi antara backend dan aplikasi mobile secara terstruktur dan efisien [19].

Pada sisi *backend*, sistem menerima input teks yang dikirimkan dari aplikasi mobile, kemudian melakukan tahapan pra-pemrosesan teks menggunakan teknik *Natural Language Processing* (NLP) yang sama dengan tahap pelatihan model. Setelah itu, teks diproses oleh model klasifikasi untuk menghasilkan prediksi tingkat stres mahasiswa. Hasil prediksi tersebut selanjutnya dikirimkan kembali ke aplikasi mobile dalam format yang dapat ditampilkan secara langsung kepada pengguna.

Aplikasi mobile yang dikembangkan menggunakan Flutter berperan sebagai antarmuka pengguna (*user interface*) yang memungkinkan mahasiswa untuk memasukkan teks yang merepresentasikan kondisi emosional mereka. Aplikasi ini juga bertanggung jawab untuk mengirimkan data teks ke *backend* melalui API serta menampilkan hasil klasifikasi tingkat stres secara *real-time*. Pendekatan arsitektur *client-server* yang digunakan memungkinkan sistem untuk diakses secara fleksibel serta memudahkan proses pemeliharaan dan pengembangan lebih lanjut.

Integrasi sistem ini dirancang untuk mendukung kinerja yang efisien dan responsif, sehingga proses klasifikasi dapat dilakukan dengan waktu respons yang singkat. Selain itu, pemisahan antara backend dan frontend memungkinkan model machine learning diperbarui atau dikembangkan tanpa harus melakukan perubahan pada aplikasi mobile. Dengan demikian, sistem yang dibangun tidak hanya fungsional, tetapi juga skalabel dan siap dikembangkan untuk kebutuhan deteksi stres mahasiswa di lingkungan perguruan tinggi..

### 3. Hasil dan Diskusi

Data teks awal sebagaimana ditunjukkan pada Tabel 1 berjumlah 20 entri yang terdiri dari tiga kategori tingkat stres, yaitu Rendah, Sedang, dan Tinggi. Seluruh data teks tersebut telah melalui tahapan pra-pemrosesan menggunakan teknik *Natural Language Processing* (NLP) dengan bahasa pemrograman Python. Tahapan pra-pemrosesan meliputi *case folding*, penghapusan karakter khusus, *stopword removal*, serta proses *stemming* atau *lemmatization* untuk menyederhanakan variasi kata.

Hasil statistik data setelah pra-pemrosesan ditunjukkan pada Tabel 2. Distribusi data menunjukkan bahwa jumlah entri pada masing-masing kelas relatif seimbang, dengan proporsi 25% untuk tingkat stres tinggi, 35% untuk tingkat stres sedang, dan 40% untuk tingkat stres rendah. Kondisi distribusi kelas yang relatif seimbang ini penting dalam proses pelatihan model klasifikasi karena dapat mengurangi risiko bias model terhadap kelas tertentu.

Tabel 2. Statistik Data Setelah Pra-Pemrosesan

Kategori	Jumlah Entri	Proporsi (%)
Tingkat Stres Tinggi	5	25%
Tingkat Stres Sedang	7	35%
Tingkat Stres Rendah	8	40%
Total	20	100%
Jumlah Kosakata (Unik) Akhir	1.850	-

Selain distribusi kelas, hasil pra-pemrosesan juga menunjukkan bahwa jumlah kosakata unik yang dihasilkan mencapai 1.850 term. Jumlah kosakata ini mencerminkan keberagaman ekspresi bahasa yang digunakan mahasiswa dalam menggambarkan kondisi emosional mereka. Tahapan pra-pemrosesan berhasil mengurangi noise pada data tanpa menghilangkan kata-kata kunci yang relevan, sehingga representasi teks yang dihasilkan tetap informatif untuk tahap ekstraksi fitur.

Dengan data yang telah diproses secara konsisten dan distribusi kelas yang relatif seimbang, model klasifikasi memiliki dasar yang memadai untuk mempelajari pola linguistik yang membedakan setiap tingkat stres. Kondisi ini mendukung proses pelatihan model agar mampu melakukan generalisasi yang lebih baik pada data baru, serta meningkatkan keandalan hasil klasifikasi pada tahap evaluasi selanjutnya.

#### 3.1 Hasil Ekstraksi Fitur (TF-IDF)

Metode *Term Frequency–Inverse Document Frequency* (TF-IDF) diterapkan untuk mengubah teks yang telah melalui tahap pra-pemrosesan menjadi representasi vektor numerik yang dapat diproses oleh algoritma *machine learning* [20]. Metode ini digunakan untuk memberikan bobot yang lebih tinggi pada kata-kata yang memiliki tingkat kepentingan lebih besar dalam membedakan dokumen, dengan mempertimbangkan frekuensi kemunculan kata dalam satu dokumen serta distribusinya pada keseluruhan korpus teks.

Hasil penerapan TF-IDF menunjukkan bahwa kata-kata yang berkaitan dengan tekanan emosional dan beban psikologis memperoleh bobot yang lebih tinggi pada kelas tingkat stres sedang dan tinggi. Kata-kata seperti “lelah”, “cemas”, dan “bingung” lebih sering muncul pada teks mahasiswa yang diklasifikasikan ke dalam kategori tersebut. Sebaliknya, kata-kata yang mencerminkan kondisi emosional positif, seperti “santai”, “produktif”, dan “tenang”, memiliki bobot yang lebih dominan pada kelas tingkat stres rendah. Pola ini menunjukkan bahwa TF-IDF mampu menangkap perbedaan karakteristik linguistik antar kelas secara efektif.

Representasi vektor hasil ekstraksi fitur kemudian dibagi menjadi data latih dan data uji dengan perbandingan 80% untuk pelatihan dan 20% untuk pengujian model. Pembagian ini bertujuan untuk memastikan bahwa model

tidak hanya mampu mempelajari pola pada data latih, tetapi juga dapat dievaluasi secara objektif terhadap data yang belum pernah dilihat sebelumnya. Dengan demikian, hasil ekstraksi fitur TF-IDF menjadi dasar yang penting dalam proses pelatihan dan evaluasi model klasifikasi tingkat stres mahasiswa.

### 3.2 Evaluasi Kinerja Model Klasifikasi

Setelah proses ekstraksi fitur menggunakan TF-IDF sebagaimana dijelaskan pada Subbab 3.1, representasi vektor teks digunakan sebagai masukan untuk pengujian beberapa algoritma *machine learning*. Empat algoritma klasifikasi, yaitu *Logistic Regression*, *Naïve Bayes*, *Random Forest*, dan *Support Vector Machine (SVM)*, diuji untuk mengevaluasi kemampuannya dalam mengklasifikasikan tingkat stres mahasiswa berdasarkan data teks. Evaluasi dilakukan menggunakan data uji untuk menilai performa model secara objektif.

Hasil perbandingan kinerja keempat model ditunjukkan pada Tabel 3. Evaluasi dilakukan menggunakan metrik akurasi data latih, akurasi data uji, serta F1-score (*macro average*) untuk memberikan gambaran menyeluruh mengenai performa klasifikasi pada setiap kategori tingkat stres.

Tabel 3. Perbandingan Akurasi Model Klasifikasi dengan Fitur TF-IDF

Algoritma Klasifikasi	Akurasi Latih (%)	Akurasi Uji (%)	F1-Score (Macro Avg)
Logistic Regression (LR)	92.5	88.33	0.88
Naïve Bayes (NB)	88.0	85.00	0.85
Random Forest (RF)	99.0	82.50	0.82
Support Vector Machine (SVM)	91.8	87.50	0.87
Jumlah Kosakata (Unik) Akhir	1.850	-	

Berdasarkan hasil tersebut, *Logistic Regression* menunjukkan performa terbaik pada data uji dengan akurasi sebesar 88,33% dan nilai F1-score tertinggi sebesar 0,88. Hasil ini mengindikasikan bahwa model *Logistic Regression* mampu memanfaatkan representasi fitur TF-IDF secara efektif untuk membedakan tingkat stres mahasiswa pada berbagai kelas. Kinerja yang relatif seimbang antara data latih dan data uji juga menunjukkan bahwa model ini memiliki kemampuan generalisasi yang baik.

Model *Naïve Bayes* menghasilkan performa yang cukup baik, namun sedikit lebih rendah dibandingkan *Logistic Regression*. Hal ini dapat disebabkan oleh asumsi independensi antar fitur yang dimiliki oleh *Naïve Bayes*, yang kurang sepenuhnya sesuai dengan karakteristik bahasa alami. Sementara itu, *Random Forest* menunjukkan akurasi yang sangat tinggi pada data latih, namun mengalami penurunan signifikan pada data uji. Kondisi ini mengindikasikan adanya kecenderungan *overfitting*, di mana model terlalu menyesuaikan diri dengan data latih sehingga kurang optimal dalam melakukan prediksi pada data baru.

*Support Vector Machine* menunjukkan performa yang kompetitif dengan akurasi uji sebesar 87,50% dan F1-score 0,87. Meskipun demikian, perbedaan performanya dengan *Logistic Regression* serta kompleksitas proses penyesuaian parameter menjadikan *Logistic Regression* sebagai pilihan yang lebih efisien untuk diintegrasikan ke dalam sistem. Oleh karena itu, *Logistic Regression* dipilih sebagai model terbaik untuk tahap integrasi sistem dan implementasi pada aplikasi mobile.

### 3.3 Karakteristik Linguistik Teks Mahasiswa dan Pola Ekspresi Stres

Analisis karakteristik linguistik teks mahasiswa menunjukkan bahwa tingkat stres tidak hanya tercermin dari kemunculan kata kunci tertentu, tetapi juga dari pola penggunaan bahasa secara keseluruhan. Bahasa yang digunakan mahasiswa merepresentasikan kondisi emosional yang dialami, baik secara eksplisit melalui kata-kata bernuansa negatif maupun secara implisit melalui struktur kalimat dan pilihan diksi yang digunakan.

Mahasiswa dengan tingkat stres tinggi cenderung menggunakan kata kerja dan kata sifat yang bermakna negatif, seperti “lelah”, “kewalahan”, “cemas”, dan “tidak mampu”. Kata-kata tersebut mencerminkan tekanan emosional, kelelahan mental, serta persepsi keterbatasan diri dalam menghadapi tuntutan akademik. Pola linguistik semacam ini memberikan sinyal yang kuat bagi model klasifikasi dalam mengidentifikasi kelas stres tinggi.

Sebaliknya, teks yang diklasifikasikan ke dalam tingkat stres rendah lebih banyak mengandung kata-kata yang bersifat positif dan reflektif, seperti “tenang”, “produktif”, dan “menyenangkan”. Pilihan kata ini mencerminkan kondisi emosional yang lebih stabil serta kemampuan mahasiswa dalam mengelola beban akademik. Pola tersebut selaras dengan hasil ekstraksi fitur TF-IDF, di mana kata-kata bernuansa positif memiliki bobot yang lebih dominan pada kelas stres rendah.

Pada kelas stres sedang, karakteristik linguistik yang muncul cenderung lebih ambigu. Bahasa yang digunakan sering kali mengandung unsur kecemasan atau ketidakpastian, namun belum menunjukkan tekanan emosional yang ekstrem. Kondisi ini menyebabkan adanya tumpang tindih pola linguistik antara kelas stres sedang dengan kelas stres rendah maupun tinggi, sehingga proses klasifikasi pada kelas ini menjadi lebih menantang bagi model.

### 3.4 Pengaruh Pra-Pemrosesan dan Representasi Fitur terhadap Kinerja Model

Tahapan pra-pemrosesan teks memiliki peran penting dalam meningkatkan kinerja model klasifikasi. Proses stopword removal dan stemming membantu mengurangi noise serta menyederhanakan variasi kata yang memiliki makna serupa. Dalam konteks bahasa Indonesia, stemming menjadi aspek krusial karena banyaknya imbuhan yang dapat mengubah bentuk kata tanpa mengubah makna dasar.

Tanpa pra-pemrosesan yang optimal, model cenderung mengalami kesulitan dalam mengenali pola linguistik yang konsisten, yang pada akhirnya dapat menurunkan akurasi klasifikasi. Hasil penelitian ini menunjukkan bahwa kombinasi pra-pemrosesan standar NLP dengan representasi fitur TF-IDF mampu menghasilkan vektor fitur yang lebih diskriminatif, sehingga memudahkan model dalam membedakan tingkat stres mahasiswa.

### 3.5 Perbandingan Algoritma Klasifikasi dan Kemampuan Generalisasi Model

Hasil perbandingan kinerja antar algoritma menunjukkan bahwa Logistic Regression memiliki keunggulan dalam menangani data teks berdimensi tinggi yang dihasilkan oleh TF-IDF. Model ini mampu memberikan performa yang stabil serta keseimbangan yang baik antara akurasi data latih dan data uji.

*Naive Bayes* menunjukkan performa yang cukup baik, namun asumsi independensi antar fitur membatasi kemampuannya dalam menangkap keterkaitan antar kata dalam teks. Random Forest, meskipun memiliki akurasi data latih yang tinggi, menunjukkan penurunan performa yang signifikan pada data uji. Hal ini mengindikasikan kecenderungan overfitting akibat kompleksitas model yang relatif tinggi, terutama pada dataset dengan ukuran terbatas.

*Support Vector Machine* memberikan hasil yang kompetitif dan mendekati *Logistic Regression*, namun memerlukan proses penyesuaian parameter yang lebih kompleks. Berdasarkan pertimbangan performa dan efisiensi, *Logistic Regression* dinilai sebagai model yang paling sesuai untuk diterapkan pada sistem klasifikasi tingkat stres mahasiswa.

### 3.6 Integrasi Sistem dan Kinerja *Real-Time*

Integrasi model klasifikasi ke dalam aplikasi mobile berbasis Flutter melalui *backend* Python menunjukkan bahwa pendekatan arsitektur *client-server* berbasis API mampu mendukung proses klasifikasi tingkat stres secara *real-time*. Proses komputasi utama dilakukan di sisi server, sehingga waktu respons sistem relatif singkat dan tidak membebani perangkat pengguna.

Desain sistem ini memungkinkan pembaruan model dilakukan tanpa harus memperbarui aplikasi mobile, sehingga sistem menjadi lebih fleksibel dan mudah dikembangkan. Dari sisi pengguna, antarmuka yang sederhana memudahkan mahasiswa untuk mengekspresikan kondisi emosional mereka dalam bentuk teks, sehingga meningkatkan potensi pemanfaatan sistem secara berkelanjutan.

### 3.7 Implikasi Praktis, Keterbatasan, dan Arah Pengembangan

Sistem yang dikembangkan memiliki implikasi praktis yang signifikan bagi lingkungan perguruan tinggi, khususnya sebagai alat skrining awal untuk mendeteksi mahasiswa yang berpotensi mengalami stres tinggi.

Dengan pendekatan berbasis teks, sistem ini dapat melengkapi metode konvensional seperti kuesioner manual dan membantu pihak kampus dalam menentukan langkah intervensi yang lebih tepat.

Meskipun demikian, penelitian ini memiliki beberapa keterbatasan, antara lain jumlah dan variasi dataset yang masih terbatas serta ketergantungan pada data teks tanpa mempertimbangkan faktor lain seperti data fisiologis atau perilaku. Oleh karena itu, hasil klasifikasi sebaiknya dipandang sebagai indikasi awal, bukan sebagai diagnosis klinis.

Penelitian selanjutnya dapat memperluas dataset dengan melibatkan lebih banyak responden dan variasi konteks bahasa. Selain itu, penerapan pendekatan deep learning seperti *Long Short-Term Memory* (LSTM) atau model berbasis transformer berbahasa Indonesia berpotensi meningkatkan akurasi klasifikasi. Integrasi fitur rekomendasi atau notifikasi dukungan psikologis juga dapat meningkatkan manfaat praktis sistem dalam mendukung kesehatan mental mahasiswa.

#### 4. Kesimpulan

Penelitian ini berhasil mengembangkan dan mengintegrasikan model klasifikasi tingkat stres mahasiswa berbasis NLP pada aplikasi *mobile* Flutter dan Python. Model terbaik, yaitu *Logistic Regression* dengan ekstraksi fitur TF-IDF, berhasil mencapai akurasi sebesar 88.33% pada data uji. Hasil ini mengonfirmasi bahwa kombinasi NLP dan ML dapat menjadi pendekatan yang efektif untuk deteksi dini stres. Meskipun akurasi tinggi, perlunya pengurangan kesalahan *False Negative* tetap menjadi area utama untuk perbaikan guna memastikan semua mahasiswa yang berisiko mendapatkan perhatian. Model *Logistic Regression* yang dikombinasikan dengan ekstraksi fitur TF-IDF merupakan kombinasi paling efektif untuk mengklasifikasikan tingkat stres mahasiswa, mencapai akurasi 88.33%. Model ini berhasil di-*deploy* sebagai *backend* API Python dan beroperasi dengan lancar dalam skema pengiriman dan penerimaan data JSON dengan *frontend* Flutter, membuktikan kelayakan pengembangan sistem deteksi dini stres berbasis AI *mobile*.

#### Referensi

1. R. Misra and M. McKean, "College students' academic stress and its relation to their anxiety, time management, and leisure satisfaction," *American Journal of Health Studies*, vol. 16, no. 1, pp. 41–51, 2000.
2. R. Beiter, R. Nash, D. McCrady, A. Rhoades, S. Linscomb, B. Clarahan, and D. Sammut, "The prevalence and correlates of depression, anxiety, and stress in a sample of college students," *Journal of Affective Disorders*, vol. 173, pp. 90–96, 2015.
3. C. Son, S. Hegde, E. Smith, A. Wang, and C. Sasangohar, "Effects of COVID-19 on college students' mental health in the United States: Interview survey study," *Journal of Medical Internet Research*, vol. 22, no. 9, p. e21279, 2020.
4. Y. R. Tausczik and J. W. Pennebaker, "The psychological meaning of words: LIWC and computerized text analysis methods," *Journal of Language and Social Psychology*, vol. 29, no. 1, pp. 24–54, 2010.
5. M. R. Sudrajat and M. Zakariyah, "Penerapan Natural Language Processing dan Machine Learning untuk Prediksi Stress Siswa SMA Berdasarkan Analisis Stress," *BITS*, vol. 6, no. 3, pp. 1527–1536, 2024.
6. M. D. Deshmukh, S. Patil, and V. Thakare, "Stress detection in students using machine learning techniques," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 12, no. 3, pp. 145–152, 2021.
7. H. Patel and P. Parmar, "Stress level identification in university students using data mining techniques," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 8, no. 6, pp. 560–564, 2019.
8. T. Shankar and S. Dutta, "Student stress detection using machine learning and sentiment analysis," *International Journal of Engineering Research & Technology (IJERT)*, vol. 9, no. 4, pp. 450–455, 2020.
9. I. Alvarez, J. Camacho, and M. Ruiz, "Automatic stress classification from text using TF-IDF and logistic regression," *Journal of Information Systems and Technology Research*, vol. 14, no. 2, pp. 101–110, 2022.
10. N. Al-qaysi, M. R. Arshad, and H. M. Al-qaysi, "Text-based emotion detection using machine learning: A systematic review," *IEEE Access*, vol. 8, pp. 113915–113937, 2020.
11. S. Lee and J. Lee, "Prediction of stress levels using machine learning algorithms and social media data," *Applied Sciences*, vol. 9, no. 20, p. 4088, 2019.
12. A. Kumar and N. Garg, "Sentiment analysis using NLP on social media for mental health monitoring," *International Journal of Computer Applications*, vol. 178, no. 7, pp. 12–18, 2019.
13. N. F. Rachmani, M. B. Santoso, and R. Hidayat, "Analisis tingkat stres mahasiswa selama pembelajaran daring," *Jurnal Psikologi Pendidikan*, vol. 12, no. 2, pp. 87–96, 2021.
14. A. Nugroho and R. Sari, "Klasifikasi tingkat stres mahasiswa menggunakan Naive Bayes dan TF-IDF," *JTIK*, vol. 7, no. 1, pp. 33–40, 2022.
15. F. A. Prasetyo, A. Susanto, and A. Darmawan, "Implementasi Natural Language Processing untuk klasifikasi emosi pada teks berbahasa Indonesia," *Jurnal Sistem Informasi*, vol. 17, no. 1, pp. 23–32, 2021.
16. W. A. Putri, R. D. Salma, and A. Widodo, "Stres akademik pada mahasiswa selama pandemi COVID-19," *Jurnal Kesehatan Masyarakat*, vol. 9, no. 3, pp. 178–184, 2021.
17. P. Widodo and E. Nurhayati, "Tingkat stres mahasiswa selama pembelajaran daring di masa pandemi," *Jurnal Pendidikan dan Pengajaran*, vol. 51, no. 2, pp. 90–100, 2021.
18. M. R. Sudrajat, M. Zakariyah, and T. Shankar, "Analisis stres siswa berbasis NLP dan ML," *IJACSA*, vol. 12, no. 4, pp. 210–218, 2024.

19. H. Patel, "Machine learning untuk identifikasi tingkat stres pada mahasiswa," *IJITEE*, vol. 8, no. 8, pp. 600–606, 2019.
20. I. Alvarez, J. Camacho, and M. Ruiz, "Klasifikasi stres otomatis berbasis teks menggunakan TF-IDF," *Journal of Information Systems and Technology Research*, vol. 14, no. 3, pp. 111–119, 2022.