



Implementasi Algoritma Klasifikasi K-Nearest Neighbor (KNN) Untuk Klasifikasi Level Stres Mahasiswa

Devi Yunita^{1*}, Firmansyah Surya Kusuma², Alpian Maulana², Alana Zatayumni²,
Ihdaudin², Diski Bima Pradana²

^{1,2}Fakultas Ilmu Komputer, Program Studi Teknik Informatika, Universitas Pamulang, Tangerang Selatan, Indonesia

Email: ^{1*}dosen00846@unpam.ac.id, ²firmanysyahsuryakusuma@gmail.com

(* : coresponding author)

Abstrak – Stress pada mahasiswa merupakan masalah yang dapat memengaruhi performa akademik, termasuk nilai *Grade Point Average* (GPA). Penelitian ini bertujuan untuk mengklasifikasikan tingkat stress mahasiswa menggunakan algoritma klasifikasi *K-Nearest Neighbor* (KNN). Dataset yang digunakan terdiri dari 2.000 data mahasiswa, yang diambil dari platform Kaggle, mencakup variabel-variabel seperti jam belajar perhari, jam ekstrakurikuler perhari, jam tidur perhari, jam aktivitas sosial perhari, jam aktifitas fisik perhari, nilai *Grade Point Average* (GPA) dan hasil kuesioner tingkat stress. Data diolah melalui proses praproses, termasuk normalisasi dan pembagian data menjadi *training* dan *testing* dengan rasio 80:20. Model KNN dikembangkan untuk mengklasifikasikan tingkat stress ke dalam kategori rendah, sedang, dan tinggi. Hasil pengujian menggunakan model KNN dengan nilai $k = 3$, dengan tujuan mengetahui tingkat akurasi yang lebih baik dari algoritma *K-Nearest Neighbor*. Setelah dilakukan pengujian didapatkan hasil yang dari algoritma *K-Nearest Neighbor* dengan menggunakan nilai $k = 3$, yaitu dengan tingkat akurasi mencapai 93%, dengan metrik evaluasi berupa *precision*, *recall*, dan *F1-score* yang juga menunjukkan performa yang baik. Penelitian ini dapat di simpulkan bahwa algoritma *K-Nearest Neighbor* efektif untuk digunakan dalam klasifikasi tingkat stress mahasiswa. Hasil ini dapat menjadi dasar untuk pengembangan sistem deteksi dini tingkat stres yang dapat diintegrasikan ke dalam platform pendidikan.

Kata Kunci: Aktifitas Mahasiswa; Klasifikasi; K-Nearest Neighbor

Abstract – Stress in students is a problem that can affect academic performance, including *Grade Point Average* (GPA). This study aims to classify the stress level of college students using the *K-Nearest Neighbor* (KNN) classification algorithm. The dataset used consists of 2,000 student data, taken from the Kaggle platform, including variables such as study hours per day, extracurricular hours per day, sleep hours per day, social activity hours per day, physical activity hours per day, *Grade Point Average* (GPA) scores and stress level questionnaire results. The data was preprocessed, including normalization and division of data into training and testing with a ratio of 80:20. The KNN model was developed to classify stress levels into low, medium and high categories. The test results use the KNN model with a value of $k = 3$, with the aim of knowing the level of accuracy that is better than the *K-Nearest Neighbor* algorithm. After testing, the results obtained from the *K-Nearest Neighbor* algorithm using the value of $k = 3$, namely with an accuracy rate of 93%, with evaluation metrics in the form of *precision*, *recall*, and *F1-score* which also show good performance. This research can be concluded that the *K-Nearest Neighbor* algorithm is effective for use in classifying student stress levels. This result can be the basis for the development of an early stress detection system that can be integrated into the education platform.

Keywords: Student Activity; Classification; K-Nearest Neighbor

1. PENDAHULUAN

Stress pada siswa menjadi salah satu isu penting dalam dunia pendidikan, karena dapat memengaruhi performa akademik, termasuk nilai *Grade Point Average* (GPA) (Condroningtyas & Marsofiyanti, 2024). Mahasiswa mengalami stress sebagai tuntutan kehidupan akademik yang harus dijalani, termasuk aktivitas diluar akademik diantaranya bersosialisasi dan menyesuaikan diri dengan teman sesama mahasiswa dimana memiliki karakteristik dan latar belakang yang berbeda, mengembangkan bakat dan minat melalui kegiatan – kegiatan non akademis, dan bekerja sama untuk menambah uang saku (Govaerst & Gregoire, 2004). Kondisi tersebut menjadikan stressor bagi mahasiswa. Sehingga menghasilkan mahasiswa yang mampu bersaing di era yang penuh dengan perkembangan informasi.

Penelitian ini penting untuk dilakukan karena stress menjadi masalah kesehatan yang meningkat belakangan ini. Dengan memahami pola faktor aktifitas yang mempengaruhi tingkat

stress mahasiswa dan menggunakan metode data mining. penelitian ini, diharapkan dapat dikembangkan sebuah sistem deteksi dini tingkat stress mahasiswa yang dapat diintegrasikan ke dalam platform pendidikan, sehingga memungkinkan intervensi yang lebih efektif dalam mendukung kesehatan mental dan pencapaian akademik mahasiswa.

2. METODE

2.1 Data Mining

Data mining adalah gabungan dua kata bahasa inggris *Data* yang berarti data dan *Mining* yang berarti menambang. Data mining adalah proses pengumpulan dan pengolahan data yang bertujuan untuk mengekstrak informasi penting pada data. Proses pengumpulan dan ekstraksi informasi tersebut dapat dilakukan menggunakan perangkat lunak dengan bantuan perhitungan statistika, matematika, ataupun teknologi *Artificial Intelligence* (AI). (Rony, 2021).

2.2. Klasifikasi

Menurut KBBI klasifikasi merupakan kata serapan dari bahasa Belanda *classificatie* yang sendirinya berasal dari bahasa Prancis *classification*. Istilah ini menunjuk kepada sebuah metode untuk menyusun data secara sistematis atau menurut beberapa aturan atau kaidah yang telah ditetapkan. Klasifikasi adalah proses pengelompokan benda berdasarkan ciri – ciri persamaan dan perbedaan pada benda.

2.2. Algoritma K-Nearest Neighbor (KNN)

K-Nearest Neighbor (KNN) di kembangkan pertama kali oleh Evelyn Fix dan Joseph Hodges pada tahun 1951. *K-Nearest Neighbor* (KNN) adalah metode klasifikasi terhadap objek baru berdasarkan *data training* yang memiliki jarak tetangga terdekat (*nearest neighbor*) dengan objek baru tersebut (Bahharuddin, 2019). Berikut ini adalah langkah – langkah algoritma KNN :

1. Menentukan nilai K. Penentuan nilai K ini tidak ada rumus pastinya. Namun satu tips yang dapat dipertimbangkan, yakni jika kelas berjumlah genap maka sebaiknya nilai K-nya ganjil, sebaliknya jika kelas berjumlah ganjil maka sebaiknya nilai K-nya genap.
2. Melakukan perhitungan jarak terdekat (eulidean distance) terhadap masing masing objek data yang diberikan. Rumus untuk menghitung jarak terdekat berikut ini :

$$d_i = \sqrt{(x_{ki} - x_{kj})^2 + (x_{ki} - x_{kj})^2 + \dots + (x_{ki} - x_{kj})^2}$$

Keterangan :

d_i = jarak *eulidean*

x_{ki} = *data training* ke-1

x_{kj} = *data testing* ke-1

3. Urutkan jarak-jarak yang telah dihitung dan pilih k data terdekat.
4. Dari k tetangga terdekat, lakukan voting untuk menentukan kelas dari data baru tersebut. Kelas yang paling banyak dipilih oleh tetangga akan menjadi prediksi kelas.
5. Tetapkan kelas dengan jumlah suara terbanyak sebagai kelas data baru.

2.2. Algoritma K-Nearest Neighbor (KNN)

Adalah tabel yang digunakan untuk mengevaluasi performa model dengan membandingkan prediksi model terhadap data sebenarnya. Tabel ini memberikan gambaran detail tentang bagaimana model bekerja pada setiap kelas dalam data, bukan hanya berdasarkan akurasi keseluruhan.

Dalam mengevaluasi performance algoritma dari Machine Learning (khususnya supervised learning), menggunakan acuan Confusion Matrix. Confusion Matrix merepresentasikan prediksi dan kondisi sebenarnya(aktual) dari data yang dihasilkan oleh algoritma. Berdasarkan Confusion Matrix, bisa menentukan Accuracy, Precission, Recall dan Specificity.



1. Akurasi

Akurasi menghitung persentase prediksi yang benar dibandingkan dengan keseluruhan jumlah data. Rumus untuk menentukan akurasi sebagai berikut :

$$Akurasi = Specificity \frac{P}{(P + N)} + Specificity \frac{N}{(P + N)}$$

2. Presisi

Presisi adalah metrik evaluasi yang mengukur seberapa baik model memprediksi kelas positif. Presisi menunjukkan proporsi prediksi positif yang benar dibandingkan dengan semua prediksi positif. Rumus untuk menentukan presisi sebagai berikut :

$$Presisi = \frac{TP}{TP + FP}$$

3. Recall

Recall adalah metrik evaluasi yang mengukur kemampuan model untuk mendeteksi semua data yang benar-benar termasuk dalam kelas positif. Dengan kata lain, recall menunjukkan seberapa banyak dari total kasus positif yang berhasil diprediksi dengan benar. Rumus untuk menentukan recall sebagai berikut :

$$Presisi = \frac{TP}{TP + FN}$$

4. *F1-Score atau F-Measure.*

F1-Score adalah metrik evaluasi yang digunakan untuk menyeimbangkan presisi dan recall dalam satu nilai tunggal. F1-Score sangat berguna ketika Anda ingin mempertimbangkan baik keakuratan prediksi positif (presisi) maupun kelengkapan deteksi positif (recall), terutama dalam dataset dengan distribusi kelas yang tidak seimbang.

$$F1 - Score = 2 \times \frac{presisi \times recall}{presisi + recall}$$

Keterangan:

TP : *True Positive*

TN : *True Negative*

FP : *False Positive*

FN : *False Negative*

3. ANALISA DAN PEMBAHASAN

3.1 Dataset

Pada penelitian ini menerapkan data mining untuk klasifikasi level stress mahasiswa berdasarkan 6 aktivitas, yakni: jam belajar perhari (Sty), jam ekstrakurikuler per hari (Ext), jam tidur perhari (Slp), jam aktivitas sosial perhari (Scl), jam aktivitas fisik perhari(Phyac), nilai rata-rata (GPA).

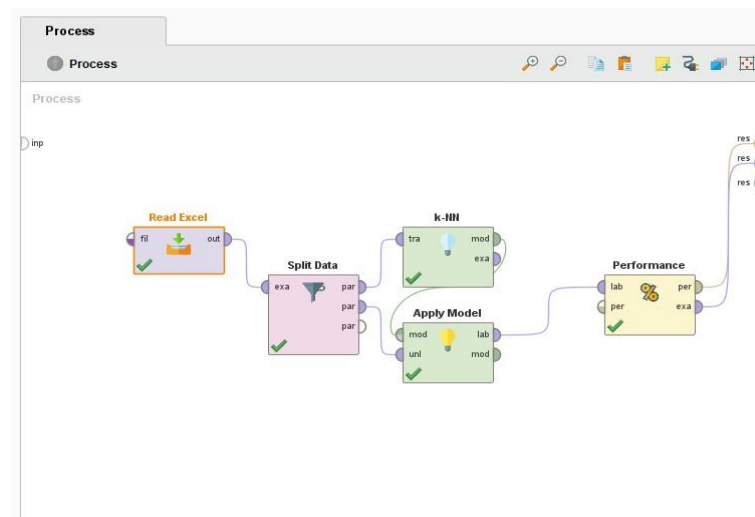
Dataset berisi sebanyak 2000 sampel data yang akan di klasifikasi menggunakan algoritma KNN. Dataset pada table 1 akan di split menjadi *data training* 80% dan *data testing* 20%.

Table 1. Dataset

StudentID	Sty	Ext	Slp	Scl	PhyAc	GPA	LevelStress
1	6.9	3.8	8.7	2.8	1.8	2.99	Moderate
2	5.3	3.5	8	4.2	3	2.75	Low
3	5.1	3.9	9.2	1.2	4.6	2.67	Low
4	6.5	2.1	7.2	1.7	6.5	2.88	Moderate
5	8.1	0.6	6.5	2.2	6.6	3.51	High
...	...	...	...	...	...	...	...
2000	9	1.7	7.3	3.1	2.9	3.58	High

3.2 Data Mining

Proses klasifikasi data mining dengan algoritma KNN untuk menganalisis Level Stress mahasiswa terhadap kegiatan keseharian dan nilai rata rata yang di peroleh menggunakan aplikasi rapidminer.



Gambar 1. Proses Klasifikasi Menggunakan Rapidminer

Berdasarkan Gambar 1, proses klasifikasi menggunakan beberapa operator yang terdapat pada rapidmier. Operator yang digunakan tersebut memiliki fungsi masing – masing. Tahapan pada proses klasifikasi sebagai berikut

1. Data Selection

Proses pertama yang dilakukan memasukan operator *read excel* yang berfungsi membaca file excel aktifitas mahasiswa dalam sehari. Langkah selanjutnya mengimpor data.

Pada tahap impor data, sistem akan menampilkan preview data sebelum memasukkannya. Tujuannya adalah untuk memeriksa kembali apakah informasi yang dimasukkan sudah benar. Berikan role id pada StudentID dan role label pada LevelStress. Kemudian klik Finish untuk

Import Data - Format your columns.

Format your columns.

☐ Replace errors with missing values ⓘ

	Extracurric...	SleepHours...	SocialHour...	PhysicalAct...	GPA	StressLevel
	real	real	real	real	real	polynomial label
1	3.800	8.700	2.800	1.800	2.990	Moderate
2	3.500	8.000	4.200	3.000	2.750	Low
3	3.900	9.200	1.200	4.600	2.670	Low
4	2.100	7.200	1.700	6.500	2.880	Moderate
5	0.600	6.500	2.200	6.600	3.510	High
6	2.100	8.000	0.300	7.600	2.850	Moderate
7	0.700	5.300	5.700	4.300	3.080	High
8	1.800	5.600	3.000	5.200	3.200	High
9	3.600	6.300	4.000	4.900	2.820	Low
10	0.700	9.800	4.500	1.300	2.760	Moderate
11	3.600	8.000	2.500	0.200	3.430	High
12	1.100	9.100	2.100	4.800	2.970	Moderate

no problems.

Previous Finish Cancel

Gambar 2. Hasil Import Data

2. Transformasi Data

Pada proses klasifikasi data akan di bagi dua (split data) menjadi data *training* dan data *testing* menggunakan operator split data. Dengan membagi data sebesar 80% data *training* dan 20% data *testing*.

Edit Parameter List: partitions

Edit Parameter List: **partitions**
The partitions that should be created.

ratio

0.8

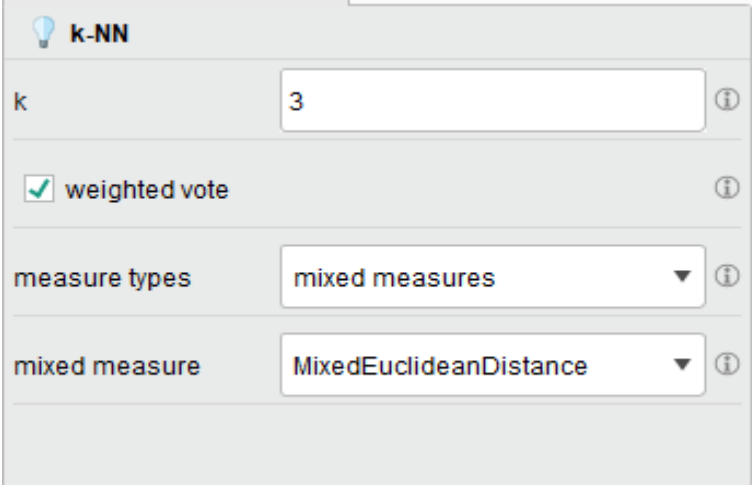
0.2

Add Entry Remove Entry OK Cancel

Gambar 3. Proses Split Data

3. Menentukan nilai K

Setelah dataset sudah diolah, kemudian dataset diproses menggunakan operator KNN dengan menentukan nilai parameter $K = 3$



k-NN

k: 3

☒ weighted vote

measure types: mixed measures

mixed measure: MixedEuclideanDistance

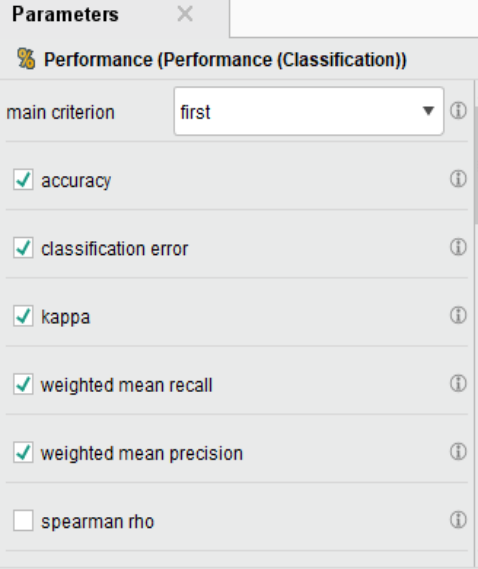
Gambar 4. Menentukan nilai k

4. Apply Model

Operator apply model digunakan untuk menaplikasikan model data yang telah dilatih menggunakan *data training* pada *data testing*

5. Performance (Classification)

Operator utama yang menghitung berbagai metrik performa klasifikasi. Operator ini mengambil label kelas aktual dan label kelas yang diprediksi sebagai input dan menghasilkan tabel yang merangkum metrik performa (Vaibhav, 2023).



Parameters

Performance (Performance (Classification))

main criterion: first

☒ accuracy

☒ classification error

☒ kappa

☒ weighted mean recall

☒ weighted mean precision

☐ spearman rho

Gambar 4. Parameter Performance (Classification)

3.3 Evaluasi

Hasil evaluasi kinerja algoritma KNN dalam mengklasifikasi level stress mahasiswa terhadap aktivitas keseharian. Evaluasi dilakukan menggunakan confusion matrix yang di tunjukan pada gambar 5.



accuracy: 93.00%

	true Moderate	true Low	true High	class precision
pred. Moderate	123	6	6	91.11%
pred. Low	3	51	2	91.07%
pred. High	9	2	198	94.74%
class recall	91.11%	86.44%	96.12%	

Gambar 5. Parameter Performance (Classification)

Penelitian klasifikasi menggunakan algoritma *K-Nearest Neighbor* (KNN), menciptakan hasil dengan rincian sebagai berikut :

1. Nilai akurasi sebesar 93%.
2. Prediksi Moderate true 123.
3. Prediksi Moderate Low High 6.
4. Prediksi Moderate High True 6.
5. Prediksi Low True 51.
6. Prediksi Low Moderate True 3.
7. Prediksi Low High True 2.
8. Prediksi High True 198.
9. Prediksi High Moderate True 9.
10. Prediksi High Low True 2.
11. Dengan *class recall* Moderate sebesar 91.11%, Low 86.44%, dan High 96.12%.
Dengan *class precision* Moderate sebesar 91.11%, Low 91.07% dan High 94.74%.

4. KESIMPULAN

Penerapan algoritma *K-Nearest Neighbor* (KNN) dalam menentukan klasifikasi Level Stress Mahasiswa terhadap kegiatan kesehariannya dengan menggunakan 4 variabel aktivitas dan 1 variabel nilai rata rata mahasiswa seperti: belajar, kegiatan ekstrakurikuler, kegiatan sosial, aktifitas fisik, nilai GPA. Penelitian ini menggunakan 2000 sampel data dengan data *training* 80% dan data *testing* 20% yang sudah diproses melalui proses menggunakan aplikasi rapidminer dengan algoritma *K-Nearest Neighbor* (KNN) menggunakan nilai $k = 3$. Yang mendapatkan nilai akurasi sebesar 93%. Dengan hasil akurasi ini algoritma *K-Nearest Neighbor* (KNN).

Penelitian ini dinilai berhasil menerapkan metode KNN untuk melakukan klasifikasi level stress mahasiswa. Dengan ini algoritma KNN dapat digunakan mahasiswa untuk meminimalisir level stress yang akan dialami terhadap berbagai jadwal aktivitas yang dilakukan dalam sehari. Selain itu mahasiswa dapat mengorganisir jadwal.

REFERENCES

- Condroningtyas, S. A., & Marsofiyati, M. (2024). Pengaruh Tekanan Atau Stres Akademik Terhadap Prestasi Belajar Mahasiswa. *Cendikia: Jurnal Pendidikan Dan Pengajaran*, 2(6), 205–212. Retrieved from <https://jurnal.kolibi.org/index.php/cendikia/article/view/1697>
- Dewi Rahayu. (2024). Pengaruh Stres Akademik Terhadap Kinerja Siswa. Diambil dari <https://www.kompasiana.com/dewirahayu8474/67152c23ed6415244b5e7762/pengaruh-stres-akademik-terhadap-kinerja-siswa>
- Govarest, S & Gregoire, J. (2004). Stressfull academic situations : study on appraisil variables in adolescence. *British Journal of Clinical Psychology*, 54, 261 - 271.
- Bunga Dea Laraswati. (2022). Mengenal K-Nearest Neighbor dan Pengaplikasiannya. Diambil dari <https://blog.algorit.ma/k-nearest-neighbor/>
- Rony Setiawan. (2023). Apa itu Data Minint dan Bagaimana Metodenya. Diambil dari



JRIIN : Jurnal Riset Informatika dan Inovasi

Volume 2, No. 11 April Tahun 2025

ISSN 3025-0919 (media online)

Hal 2045-2053

<https://www.dicoding.com/blog/apa-itu-data-mining/>

M. M. Baharuddin, H. Azis, and T. Hasanuddin, (2019). Analisis Performa Metode *K-Nearest Neighbor* Untuk Identifikasi Jenis Kaca. *Ilk. J. Ilm.*, vol. 11, no. 3, pp. 269–274, 2019 doi: 10.33096/ilkom.v11i3.489.269-274.

Vaibhav R. (2023). Performance (Classification) proses in RapidMiner. Diambil dari <https://medium.com/@vaibhav1403/performance-classification-process-in-rapidminer-b674decacfa>.