

MODEL ALGORITMA K-NEAREST NEIGHBOR (K-NN) DAN NAÏVE BAYES UNTUK PREDIKSI KELULUSAN MAHASISWA

Ryan Situmorang, Woro Isti Rahayu, Rd Nuraini Siti Fathonah

Program Studi Sarjana Terapan Teknik Informatika, Universitas Logistik Bisnis Internasional
Jl. Sari Asih No.54, Kota Bandung, Jawa Barat, Indonesia
ryanface290300@gmail.com

ABSTRAK

Jurnal ini membahas penggunaan dua model algoritma machine learning, yaitu K-Nearest Neighbor (K-NN) dan Naïve Bayes, untuk memprediksi kelulusan mahasiswa. K-NN adalah algoritma yang digunakan untuk klasifikasi dan regresi, sementara Naïve Bayes adalah algoritma yang digunakan untuk klasifikasi. Kedua algoritma ini dapat digunakan untuk memprediksi kelulusan mahasiswa dengan memanfaatkan data historis, seperti nilai ujian, absensi, dan faktor-faktor lain yang mempengaruhi kelulusan. Dalam artikel ini, akan dijelaskan secara rinci bagaimana model-model ini bekerja dan bagaimana cara mengimplementasikannya untuk memprediksi kelulusan mahasiswa.

Kata kunci: K-Nearest Neighbor, Kelulusan, Naïve Bayes

1. PENDAHULUAN

Lembaga pendidikan adalah universitas yang memberikan pengajaran akademik kepada mahasiswa. Mahasiswa, juga dikenal sebagai kelompok masyarakat, memiliki kualitas intelektual yang luas jika dibandingkan dengan kelompok usia mereka yang bukan siswa atau kelompok usia lain di bawah mereka. Intelektual mahasiswa memungkinkan mereka untuk menghadapi dan mencari kesulitan secara sistematis yang akan diimplementasikan dalam kehidupan sehari-hari agar merasa nyaman di tempat kerja [1]. Siap dalam menghadapi tantangan persaingan global dan dapat mengembangkan sumber daya manusia menjadi kompeten merupakan peran yang besar bagi pendidikan tinggi [2].

Menurut [3], Mahasiswa memperoleh sekolah pascasarjana setelah menyelesaikan program pendidikan sesuai dengan standar kelulusan yang ditetapkan oleh program studi. Lulusan unggul memiliki karakteristik kompetensi akademik pembelajaran, antara lain hard skill dan softskill, yang terkandung dalam fasilitas yang berkualitas dan ditunjukkan dengan kinerja lulusan masyarakat dalam disiplin ilmu pekerjaan dan keilmuannya.

Kelulusan mahasiswa adalah salah satu faktor penilaian paling penting dalam proses akreditasi lembaga perguruan tinggi. Setiap universitas memiliki nilai tersendiri bagi lulusan mahasiswa, sehingga proses akreditasi dalam institusi akan lebih terbantu jika mahasiswa lulus tepat waktu setiap semesternya [4].

Dalam memperkirakan waktu kelulusan mahasiswa maka penggunaan metode *data mining* diusulkan pada penelitian ini. *Data mining* adalah jenis eksplorasi dan analisis data di mana pola dan fitur diekstraksi secara otomatis dari volume data yang sangat besar.

Model *k-nearest neighbor* dan *naïve bayes* akan digunakan sebagai pendekatan *data mining*. Hal ini dimaksudkan agar dengan menggunakan strategi ini, akan memungkinkan untuk mengatasi masalah yang muncul dengan meramalkan apakah setiap mahasiswa akan lulus tepat waktu atau tidak, serta membantu pengakreditasi yang terjadi di Program Studi yang ditempuh.

2. TINJAUAN PUSTAKA

2.1. Naïve Bayes

Salah satu algoritma pembelajaran induktif yang paling sukses dan efisien untuk *machine learning* serta *data mining* adalah *Naive Bayes*. Kinerja *naïve bayes* yang kompetitif dalam proses pengklasifikasi, bahkan dengan asumsi ketergantungan Atribut (tidak ada hubungan antara atribut[5]. Berikut ini adalah rumus untuk teorema *naïve bayes*:

$$P(H_j|X_i) = \frac{P(H_j)P(X_i|H_j)}{P(X_i)}$$

Keterangan:

$P(H_j|X_i)$: peluang pada kelas j saat ada munculnya kata

(H_j) : peluang munculnya sebuah kelas j

$P(X_i|H_j)$: peluang kata i masuk kedalam kelas j

(X_i) : peluang munculnya suatu kelas

2.2. K-Nearest Neighbor

K-Nearest Neighbor merupakan salah satu algoritma yang bekerja melakukan prediksi dengan pembelajaran terawasi dimasa hasil *querin instance* berdasarkan mayoritas kategoriya[6]. Adapun rumus dari kedekatan jarak menggunakan permasaan sebaiaig berikut:

$$d = \sqrt{(X1 - Y1)^2 + (X2 - Y2)^2 + \dots}$$

Keterangan:

d = *Euclidean Distance*

X1 = Data Pertama

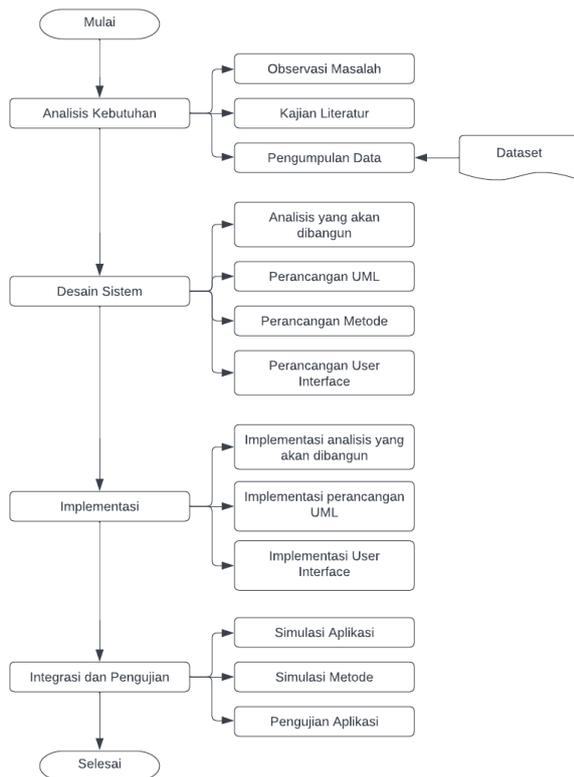
Y1 = Data Baru Pertama

2.3. Jupyter Notebook

Jupyter notebook (file format ipynb) adalah dokumen yang dibuat oleh Aplikasi jupyter notebook yang berisi kode komputer serta rich text element seperti paragraf, persamaan matematika, gambar, dan tautan [7].

3. METODE PENELITIAN

Penelitian ini dilakukan secara mantap dan sistematis. Berikut langkah-langkah penelitian ini:



Gambar 1. Metodologi penelitian

3.1. Analisis Kebutuhan yang Digunakan

Dalam penelitian ini, teknik data mining disarankan untuk digunakan dalam memperkirakan tanggal kelulusan siswa.

a. Observasi

Pada titik ini, penulis harus menilai kesulitan yang terlibat dalam mengidentifikasi masalah, menetapkan tujuan, dan menuai hasil dari penelitian mereka. Hal ini dilakukan pada tahap identifikasi masalah dengan mengkaji keadaan dan situasi saat ini yang menjadi fokus utama masalah dalam penelitian ini. Dengan melihat hal-hal yang perlu dicapai dalam penelitian ini dan hal-hal yang dapat diperoleh peneliti dan orang lain darinya, hal ini dilakukan pada tahap tujuan penelitian dan keunggulan penelitian.

b. Kajian literatur

Kajian literatur yang relevan akan dilakukan pada titik ini sehubungan dengan topik penelitian.

c. Pengumpulan data

Data tentang mahasiswa akan dikumpulkan pada penelitian ini langsung dari Program Studi D-4 Teknik Informatika.

3.2. Desain Sistem

Desain sistem dalam penelitian ini, tidak hanya tampilan saja tetapi yang dimaksud desain dalam metode ini adalah desain sistem yang meliputi:

a. Analisis proses yang akan dibangun

Analisis tersebut akan menjadi aplikasi gambaran alur yang nantinya akan dimasukkan dalam aplikasi prediksi kelulusan Mahasiswa. Nilai akurasi algoritma K-Nearest Neighbor dan Naive Bayes juga dapat dibandingkan dalam Aplikasi ini.

b. Perancangan Unified Modelling Language (UML)

Ini sering dibuat ketika aplikasi sedang dibangun. Desain UML diperlukan dalam penelitian ini untuk memahami ruang lingkup yang akan dibangun serta keseluruhan proses bisnis. Unified Modelling Language (UML) yang akan dibangun dirancang menggunakan UML berorientasi objek yang terdiri dari diagram use case dan diagram aktivitas.

c. Perancangan Algoritma K-NN dan Naive Bayes

Ada metode, K-NN dan Naive Bayes, dalam aplikasi yang akan dibangun untuk memecahkan permasalahan yang telah dianalisis dalam sistem yang disarankan.

3.3. Proses Implementasi

Pendekatan ini mencakup simulasi algoritma K-NN dan Naive Bayes untuk menentukan apakah metode yang digunakan dapat diimplementasikan dalam aplikasi yang akan dibangun. Teknik ini akan diimplementasikan dengan memasukkan algoritma metode yang digunakan dalam aplikasi yang akan dibuat.

3.4. Integrasi dan Pengujian

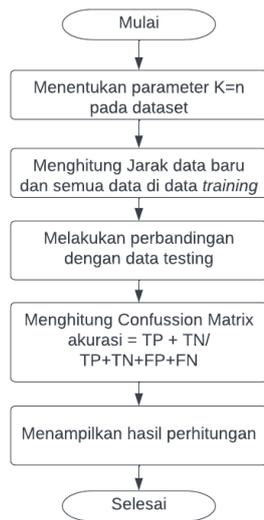
Setelah keseluruhan untuk atau modul yang diproduksi dan diuji dalam tahap implementasi, tahap ini mengintegrasikan sistem secara keseluruhan. Setelah proses integrasi selesai, sistem akan diperiksa dan diuji untuk menemukan potensi masalah dan kesalahan sistem. Selenium digunakan dalam teknik whitebox, di mana keakuratan perangkat lunak yang diuji hanya diverifikasi berdasarkan output yang dihasilkan dari data atau keadaan input yang disediakan untuk fungsi saat ini., terlepas dari bagaimana prosedur untuk mencapai hasil itu diimplementasikan.

Pada tahap ini, penulis menguji aplikasi yang dibangun untuk memastikan bahwa sistem berfungsi dengan baik.

3.5. Penerapan Metode yang Digunakan

Tahap ini terdiri dari simulasi algoritma yang digunakan dalam penelitian untuk menentukan apakah metode tersebut dapat diimplementasikan dengan benar atau tidak. Algoritma yang digunakan adalah K-NN dan Naive Bayes, yang diikuti dengan klasifikasi data dan perbandingan nilai akurasi menggunakan algoritma K-NN dan Naive Bayes.

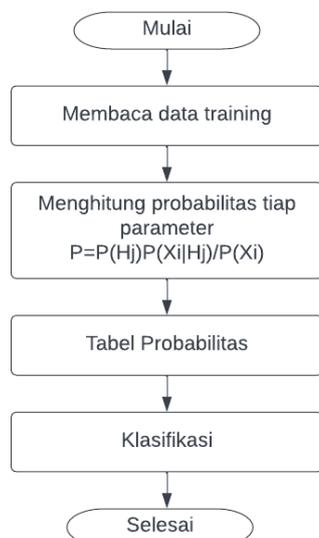
3.6. Algoritma K-Nearest Neighbor



Gambar 2. Metode K-Nearest Neighbor

Penerapan pendekatan K-NN dilakukan dalam penelitian ini dengan menemukan nilai parameter K. Jarak antara data baru dan semua data dalam set pelatihan kemudian dihitung. Setelah itu, perbandingan dengan data pengujian dilakukan, dan *confusion matrix* dihasilkan. Setelah itu, aplikasi akan menampilkan data yang diproyeksikan.

3.7. Algoritma Naive Bayes



Gambar 3. Metode Naive Bayes

Pendekatan Nave Bayes digunakan dalam penelitian ini untuk membaca data pelatihan terlebih dahulu. Setelah itu, kemungkinan setiap parameter dihitung, diikuti oleh perhitungan kesamaan kosinus. Akibatnya, akurasi akan dibandingkan selanjutnya dengan pendekatan K-NN.

4. HASIL DAN PEMBAHASAN

Pembahasan pada bab ini akan menggunakan *alat jupyter notebook* untuk mendapatkan perbandingan dan hasil dari prediksi dengan menggunakan dua metode, yaitu *k-nearest neighbor* dan *naive bayes*.

4.1. K-Nearest Neighbor

Dalam menggunakan metode *k-nearest neighbor*, dibutuhkan package yang digunakan seperti *numpy*, *pandas*, *matplotlib*, *pyplot* serta *seaborn*. untuk impor *package* yang dibutuhkan menggunakan *sitaks* seperti berikut

```

#Import package yang akan digunakan
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn import preprocessing
    
```

Gambar 4. Import package

Setelah semua *package* telah dimasukkan, langkah selanjutnya adalah memasukkan dataset yang digunakan. Dengan *syntax pd.read_csv(dataset)*. Setelah dataset dimasukkan, langkah selanjutnya adalah untuk memisahkan variabel x dan variabel y dengan *syntax* sebagai berikut :

```

x_data = df.iloc[:, [8,9,10,11]].values
y_data = df.iloc[:,12].values
    
```

Gambar 5. Penentuan variabel

Setelah penentuan variabel x dan y, data dibagi menjadi data *testing* dan data *training*. Untuk 20% dari total data dalam penelitian ini, data pengujian digunakan. Setelah pembagian data menjadi dua jenis, query dinormalisasi sebagai berikut:

```

sc = StandardScaler()
x_data_train = sc.fit_transform(x_data_train)
x_data_test = sc.fit_transform(x_data_test)
    
```

Gambar 6. Query Normalisasi

Setelah semua data telah dinormalisasi, langkah selanjutnya ialah dengan melakukan prediksi dengan query

```

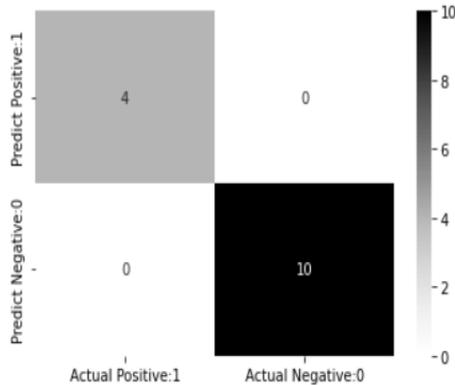
knn=KNeighborsClassifier(n_neighbors=1)
knn.fit(x_data_train,y_data_train)
predik1 = knn.predict(x_data_test)
    
```

Gambar 7. Prediksi KNN

Dan hasil dari prediksi KNN yang dilakukan sebagai berikut:

```
array([1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0], dtype=int64)
```

Setelah mengetahui hasil prediksinya, kemudian lanjut untuk menampilkan konfusi matriks sebagai berikut:



Gambar 8, Konfusi matriks

4.2. Naïve Bayes

Untuk metode *naïve bayes* ada tambahan *package* dari *sklearn* untuk metode *naïve bayes* nya. Untuk syntax di *jupyter notebook* sebagai berikut:

```
from sklearn.naive_bayes import GaussianNB
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder # Mengubah Data Bersifat Karakt
from sklearn.preprocessing import MinMaxScaler # Mengubah Nilai 0-1
from sklearn.model_selection import train_test_split # Memisahkan Data Trainin
from sklearn.metrics import classification_report, confusion_matrix # Mengukur
import seaborn as sns # Melihat Grafik
```

Gambar 9. Import package

Setelah semua *package* sudah dimasukkan, selanjutnya menghapus variabel nama dan status dari dataset sehingga menyisakan dataset yang berisi data numerik. selanjutnya adalah dengan menampilkan banyaknya status 1 dengan status 0 dengan dengan *package* *seaborn* dengan query sebagai berikut:

```
sns.countplot(x='Status', data=df)
plt.show()
```

Gambar 10. Grafik banyak data sesuai status

Setelah mengetahui banyak data sesuai status, langkah selanjutnya adalah menormalisasi data dengan query sebagai berikut:

```
scaler = MinMaxScaler()
X_min_max = scaler.fit_transform(X_min_max)
X_min_max
```

Gambar 11 Syntax normalisasi

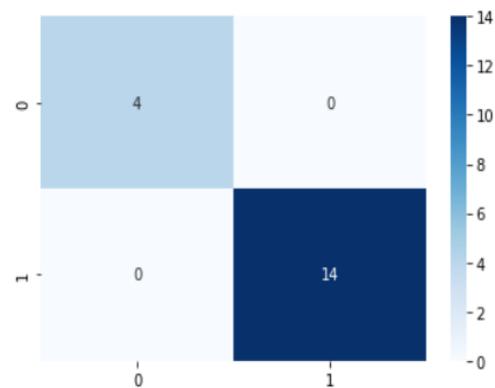
Setelah menormalisasikan data, selanjutnya membagi data menjadi data *testing* dan data *training*. Data *testing* menggunakan 26% dari data keseluruhan. Setelah sudah membagi menjadi dua

data, selanjutnya melakukan permodelan dengan model Gaussian. Model Gaussian data dituliskan dengan query sebagai berikut:

```
model = GaussianNB()
model = model.fit(X_train, y_train)
print(f'Akurasi/Score Pada Model Data. : {model.score(X, y)}')
print(f'Akurasi/Score Pada Model Training : {model.score(X_train, y_train)}')
print(f'Akurasi/Score Pada Model Testing : {model.score(X_test, y_test)}')
```

Gambar 12. Model Gaussian

Selanjutnya memprediksi dengan *naïve bayes* untuk mengetahui prediksi yang dibutuhkan. Setelah hasil prediksi, dilanjutkan dengan pengecekan konfusi matrik untuk mengetahui tingkat akurasi yang didapat. Konfusi matrik menggunakan *seaborn* untuk memvisualisasikannya dengan query sebagai berikut:



Gambar 13. Konfusi Matriks

Untuk *naïve bayes* memiliki tingkat akurasi sebesar 99,29% dimana lebih rendah dari prediksi yang dilakukan oleh metode sebelumnya yaitu *K-Nearest Neighbor*.

5. KESIMPULAN DAN SARAN

Dari penelitian diatas, dapat ditarik kesimpulan ialah hasil akurasi yang didapat dari metode *k-nearest neighbor* lebih besar di banding hasil prediksi dari metode *naïve bayes*. *K-nearest neighbor* mendapatkan hasil prediksi sebesar 100% dan sedangkan untuk *naïve bayes* mendapatkan hasil prediksi sebesar 99,29%. Adapun saran yang dapat disampaikan ialah dengan menggunakan metode lain sehingga mengetahui akurasi yang sempurna dalam prediksi kelulusan mahasiswa.

DAFTAR PUSTAKA

- [1] A. Rohman, "Model Algoritma K-nearest Neighbor untuk memprediksi kelulusan mahasiswa," *Foreign Aff.*, vol. 91, no. 5, pp. 1–9, 2012.
- [2] W. I. Sabilla and T. E. Putri, "233-240 Dokumen diterima pada 16 Oktober," *J. Komput. Terap.*, vol. 3, no. 2, pp. 233–240, 2017, [Online]. Available: <http://jurnal.pcr.ac.id>.
- [3] S. Rahmatullah, "Prediksi Tingkat Kelulusan Tepat Waktu Dengan Metode Naïve Bayes Dan

- K-Nearest Neighbor,” *J. Inf. dan Komput.*, vol. 7, no. 1, pp. 7–16, 2019, doi: 10.35959/jik.v7i1.118.
- [4] Y. Apridiansyah, N. D. M. Veronika, and E. D. Putra, “Prediksi Kelulusan Mahasiswa Fakultas Teknik Informatika Universitas Muhammadiyah Bengkulu Menggunakan Metode Naive Bayes,” *JSAI (Journal Sci. Appl. Informatics)*, vol. 4, no. 2, pp. 236–247, 2021, doi: 10.36085/jsai.v4i2.1701.
- [5] S. Syarli and A. Muin, “Metode Naive Bayes Untuk Prediksi Kelulusan (Studi Kasus: Data Mahasiswa Baru Perguruan Tinggi),” *J. Ilm. Ilmu Komput.*, vol. 2, no. 1, pp. 22–26, 2016.
- [6] I. Dwi *et al.*, “ANALISIS PERBANDINGAN METODE K-NEAREST NEIGHBOR DAN NEURAL,” vol. 7, no. 1, 2023.
- [7] D. Setiabudidaya, “Penggunaan Piranti Lunak Jupyter Notebook,” pp. 2–5, 2018, [Online]. Available: <https://osf.io/preprints/inarxiv/2h7q4/%0Ahttps://osf.io/preprints/inarxiv/2h7q4/download>.