PREDIKSI KETEPATAN WAKTU KELULUSAN MAHASISWA MENGGUNAKAN ALGORITMA K-NEAREST NEIGHBOR

Rizzandy Rasyid [a1,*], Nurhayati [b,2]

a1Program Studi Teknik Informatika, STMIK Bina Mulia Palu, Jl. Soeprapto No. 38, Besusu Tengah, Kec. Palu Timur, Kota Palu, Sulawesi Tengah 94111, Indonesia
b1 Program Studi Sistem Informasi Uin Datokarama Palu, Jl. Diponegoro, Kec. Palu Barat, Kota Palu, Sulawesi Tengah 94111, Indonesia

INFO ARTIKEL

Histori Artikel
Pengajuan
Diperbaiki
Diterima

Kata Kunci
Data Mining
K-Nearest Neighbor
Klasifikasi
Prediksi Kelulusan
Mahasiswa

ABSTRAK ()

Ketepatan waktu kelulusan mahasiswa merupakan indikator penting dalam menilai kualitas dan efektivitas perguruan tinggi. Penelitian ini bertujuan untuk memprediksi ketepatan waktu kelulusan mahasiswa di Universitas Islam Negeri Datokarama Palu menggunakan algoritma *K-Nearest Neighbor* (KNN). Data yang digunakan terdiri dari 60 rekam data mahasiswa angkatan 2015-2018 dari Fakultas Ushuluddin Adab dan Dakwah, dengan atribut Indeks Prestasi Kumulatif (IPK) semester 1 hingga 5 sebagai variabel independen dan status kelulusan sebagai variabel dependen. Pengujian dilakukan menggunakan dua metode validasi yaitu *split validation* dan *cross validation* dengan implementasi menggunakan *Jupyter Notebook* dan bahasa pemrograman Python. Hasil penelitian menunjukkan bahwa metode *split validation* menghasilkan akurasi sebesar 88% secara konsisten, sedangkan *cross validation* menghasilkan akurasi bervariasi antara 86% hingga 93%. Keterbatasan jumlah data menyebabkan *split validation* menjadi pilihan yang lebih stabil untuk penelitian ini.

1. Pendahuluan

Pendidikan merupakan usaha strategis dalam mempersiapkan sumber daya manusia untuk mencapai masa depan yang diinginkan serta meningkatkan kualitas kehidupan. Konstitusi mengamanatkan bahwa seluruh warga negara Indonesia berhak memperoleh pendidikan yang layak dan bermutu. Dengan pendidikan, manusia dapat memperluas ilmu pengetahuan dan wawasan [1].

Di tingkat perguruan tinggi, keberhasilan institusi dapat diukur melalui peningkatan kualitas pengajar dan fasilitas yang diberikan kepada mahasiswa. Kedua faktor tersebut berdampak pada peningkatan jumlah mahasiswa baru setiap tahun ajaran. Namun, tidak semua mahasiswa menyelesaikan studinya tepat waktu, sehingga diperlukan upaya prediktif untuk mengantisipasi ketidakpastian kelulusan. Banyak perguruan tinggi yang belum memanfaatkan teknologi pengolahan data untuk memprediksi waktu kelulusan mahasiswa, padahal dengan prediksi tersebut universitas dapat menentukan kuota mahasiswa baru setiap tahun dengan lebih mudah dan efektif [2].

Universitas Islam Negeri Datokarama Palu merupakan salah satu perguruan tinggi terbaik di Kota Palu yang memiliki 4 fakultas, yaitu Fakultas Ekonomi dan Bisnis Islam, Fakultas Tarbiyah dan Ilmu Keguruan, Fakultas Ushuluddin Adab dan Dakwah, serta Fakultas Syariah. Namun, UIN Datokarama Palu belum menerapkan teknik pengolahan data untuk memprediksi waktu kelulusan mahasiswanya [3].

Data mining adalah proses untuk mendapatkan informasi yang berguna dari basis data berukuran besar yang perlu diekstraksi agar menjadi informasi baru dan dapat membantu dalam pengambilan keputusan [4]. Salah satu teknik dalam *data mining* adalah klasifikasi, yaitu proses penemuan model yang menggambarkan dan membedakan kelas data atau konsep yang bertujuan agar bisa digunakan untuk memprediksi kelas dari objek yang kelasnya tidak diketahui [5].

Algoritma *K-Nearest Neighbor* (KNN) merupakan salah satu algoritma *machine learning* yang digunakan untuk masalah klasifikasi dan regresi. Algoritma ini bekerja berdasarkan prinsip bahwa objek-objek yang serupa cenderung berada dekat satu sama lain. Dengan kata lain, objek yang memiliki fitur atau karakteristik yang mirip akan cenderung termasuk dalam kategori yang sama atau memiliki nilai yang mirip [6].

Penelitian terdahulu telah menerapkan berbagai algoritma untuk prediksi kelulusan mahasiswa. Devi Heryana (2019) menggunakan algoritma *Naïve Bayes* untuk memprediksi kelulusan mahasiswa Pendidikan Matematika UIN Raden Intan Lampung dengan akurasi 74,67% [7]. I Made Budi Adnyana (2015) menerapkan metode *Random Forest* untuk prediksi lama studi mahasiswa dengan *overall accuracy* 83,54% [8]. Arif Pratana dkk. (2018) mengimplementasikan *Support Vector Machine* (SVM) untuk prediksi ketepatan waktu kelulusan dengan akurasi terbaik 80,55% [9]. Penelitian ini berbeda dengan menggunakan algoritma KNN yang belum banyak diterapkan untuk kasus serupa di lingkungan Universitas Islam Negeri Datokarama Palu.

Berdasarkan latar belakang tersebut, penelitian ini bertujuan untuk memprediksi ketepatan waktu kelulusan mahasiswa di Universitas Islam Negeri Datokarama Palu menggunakan algoritma *K-Nearest Neighbor*. Penelitian ini dibatasi pada Fakultas Ushuluddin Adab dan Dakwah dengan data mahasiswa angkatan 2015, 2016, 2017, dan 2018 [10]. Hasil penelitian ini diharapkan dapat memberikan informasi berupa data yang dapat digunakan untuk memprediksi lamanya studi mahasiswa, menjadi dasar pengambilan keputusan dalam penilaian masa belajar mahasiswa, serta digunakan sebagai perbandingan bagi peneliti lain dalam menerapkan *data mining* di lingkungan perguruan tinggi [11].

2. Metode penelitian

Penelitian ini dilaksanakan di Universitas Islam Negeri Datokarama Palu, Fakultas Ushuluddin Adab dan Dakwah, pada bulan Juli hingga Desember 2023. Metode yang digunakan adalah metode kuantitatif dengan tipe penelitian prediktif, yaitu penelitian yang bertujuan memprediksi atau memperkirakan kejadian di masa depan berdasarkan analisis data yang ada saat ini [11][12].

2.1 Desain Penelitian

Penelitian ini menggunakan pendekatan kuantitatif yang mengutamakan angka-angka mulai dari proses pengumpulan data hingga interpretasi. Metode penelitian kuantitatif diartikan sebagai bagian dari serangkaian investigasi sistematis terhadap fenomena dengan

mengumpulkan data untuk kemudian diukur dengan teknik statistik matematika atau komputasi [13].

Tipe penelitian yang digunakan adalah penelitian prediktif, yaitu penelitian yang bertujuan memprediksi dan memperkirakan apa yang akan terjadi di masa depan berdasarkan hasil analisis data saat ini. Penelitian prediktif dilakukan melalui penelitian yang bersifat korelasional dan kecenderungan, di samping mendapatkan korelasi antara dua atau lebih variabel juga dihitung regresinya [14].

2.2 Populasi dan Sampel

Populasi dalam penelitian ini adalah alumni Universitas Islam Negeri Datokarama Palu, Fakultas Ushuluddin Adab dan Dakwah angkatan tahun 2015 sampai dengan angkatan tahun 2018. Sampel penelitian ini adalah data kelulusan mahasiswa yang berjumlah 60 rekam data, dengan komposisi 31 mahasiswa lulus tepat waktu dan 29 mahasiswa lulus tidak tepat waktu [15].

Tabel 1. Contoh Data IPK Mahasiswa

S1	S2	S3	S4	S5	Keterangan
3.88	3.79	3.82	3.94	3.90	Lulus Tepat Waktu
3.94	3.88	3.75	3.80	3.88	Lulus Tepat Waktu
3.40	3.39	3.75	3.32	3.38	Lulus Tidak Tepat Waktu
3.56	3.16	3.61	3.75	3.63	Lulus Tidak Tepat Waktu
3.54	3.70	3.62	3.57	3.72	Lulus Tidak Tepat Waktu

Data yang digunakan terdiri dari Indeks Prestasi Kumulatif (IPK) semester 1 hingga semester 5 sebagai variabel independen, dan status kelulusan (tepat waktu/tidak tepat waktu) sebagai variabel dependen. Setiap angkatan berisi 15 data mahasiswa [16].

2.3 Algoritma K-Nearest Neighbor

Algoritma *K-Nearest Neighbor* (KNN) adalah salah satu algoritma *machine learning* yang digunakan untuk masalah klasifikasi dan regresi. Algoritma ini bekerja berdasarkan prinsip bahwa objek-objek yang serupa cenderung berada dekat satu sama lain. KNN mengikuti strategi "induk burung" untuk menentukan di mana data baru akan ditempatkan. Algoritma ini mengasumsikan bahwa sesuatu yang serupa akan ada di dekat atau di antara tetangga, artinya data yang serupa berdekatan satu sama lain [17].

Tujuan algoritma KNN adalah untuk mengidentifikasi tetangga terdekat dari titik kueri yang diberikan, sehingga dapat menetapkan label kelas ke titik tersebut. Ada beberapa hal yang harus diperhatikan dalam algoritma KNN:

1. **Menentukan Metrik Jarak**: Untuk menentukan titik data mana yang terdekat dengan titik kueri tertentu, jarak antara titik kueri dan titik data lain perlu dihitung. Metrik jarak membantu dalam pembentukan batasan keputusan yang mengarahkan kueri partisi ke kelas berbeda.
2. **Mendefinisikan Nilai K**: Nilai K pada algoritma KNN mendefinisikan berapa banyak tetangga yang akan diperiksa untuk menentukan klasifikasi titik kueri tertentu [18].

Perhitungan jarak antara dua titik pada algoritma KNN menggunakan metode *Euclidean Distance* dengan rumus:

$$d(x_1, x_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2}$$

Jika terdapat lebih dari satu dimensi, rumus yang digunakan adalah:

$$d = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2 + (y_{1i} - y_{2i})^2 + \cdots}$$

Kelebihan algoritma KNN adalah mudah diterapkan, mudah beradaptasi, dan memiliki sedikit *hyperparameter*. Kekurangannya adalah tidak berfungsi dengan baik pada dataset berukuran besar, kurang cocok untuk dimensi tinggi, dan perlu penskalaan fitur serta sensitif terhadap *noise data*, *missing data*, dan *outliers* [19].

2.4 Metode Validasi

Penelitian ini menggunakan dua teknik validasi untuk mengevaluasi kinerja model [20]:

2.4.1 Split Validation

Split validation adalah pembagian dataset menjadi dua bagian utama, satu bagian untuk melatih model (data pelatihan) dan bagian lainnya untuk menguji atau mengevaluasi model (data pengujian). Pada penelitian ini, data dibagi dengan perbandingan 40% untuk data pelatihan dan 60% untuk data pengujian, yaitu 6 data untuk pelatihan dan 9 data untuk pengujian dari total 15 data mahasiswa per angkatan.

2.4.2 Cross Validation

Cross validation adalah teknik evaluasi model yang digunakan untuk mengukur kinerja model *machine learning*. Cross validation melibatkan pembagian dataset menjadi beberapa subset yang kemudian digunakan untuk melatih dan menguji model dalam beberapa iterasi. Penelitian ini menggunakan 5 lipatan (*n_splits=5*) dengan pengacakan (*shuffle=true*) sebelum pembagian data.

2.5 Alat dan Bahan Penelitian

Penelitian ini menggunakan beberapa alat dan bahan sebagai berikut [21]:

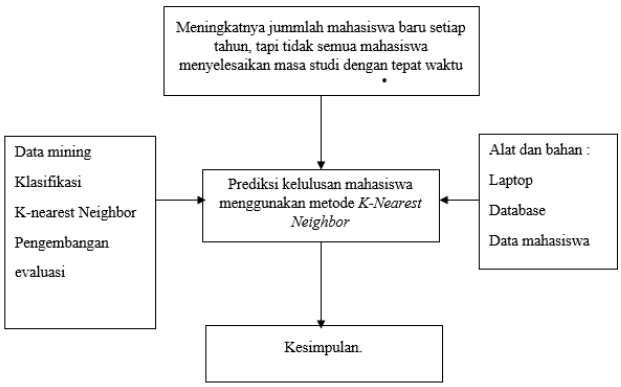
1. Data rekam mahasiswa meliputi IPK dari semester 1 sampai dengan semester 5
2. Aplikasi *Jupyter Notebook* sebagai *tools* pengolahan data
3. Bahasa pemrograman Python dengan *library*:
    - **Pandas**: untuk manipulasi, pembersihan, dan persiapan data
    - **NumPy**: untuk *scientific computing* dan pembentukan objek *N-Dimensional array*
    - **Matplotlib**: untuk menampilkan grafik atau visualisasi data
    - **Scikit-learn**: untuk implementasi algoritma KNN dan evaluasi model

2.6 Prosedur Penelitian

Prosedur penelitian yang dilakukan mengikuti langkah-langkah sistematis sebagai berikut [22]:

1. **Pengumpulan Data**: Data dikumpulkan dari bagian kemahasiswaan Universitas Islam Negeri Datokarama Palu melalui metode dokumentasi.
2. **Pra-pemrosesan Data**: Data yang telah dikumpulkan dibersihkan dan dipersiapkan untuk pengolahan lebih lanjut.
3. **Pembagian Data**: Data dibagi menjadi data pelatihan dan data pengujian menggunakan metode *split validation* dan *cross validation*.
4. **Normalisasi Data**: Data dinormalisasi menggunakan *StandarScaler* dari *scikit-learn* untuk memastikan setiap fitur memiliki skala yang sama.
5. **Implementasi KNN**: Algoritma KNN diimplementasikan dengan nilai K=2 (2 tetangga terdekat) menggunakan metrik jarak *Minkowski* dengan p=2 (*Euclidean Distance*).
6. **Evaluasi Model**: Model dievaluasi menggunakan *confusion matrix* dan *classification report* untuk mengukur akurasi prediksi.

Gambar 1. Alur Penelitian



Gambar 1 menunjukkan langkah-langkah sistematis dalam proses penelitian, mulai dari pengumpulan data hingga evaluasi hasil prediksi.

3. Hasil dan Analisis

Pada bagian ini dijelaskan hasil penelitian dan pada saat yang sama diberikan analisis dan diskusi yang komprehensif. Hasil disajikan dalam bentuk tabel untuk memudahkan pembaca memahami temuan penelitian [15][16].

3.1 Hasil Pengujian dengan Split Validation

Pengujian menggunakan *split validation* dilakukan pada setiap angkatan dengan pembagian 6 data untuk pelatihan dan 9 data untuk pengujian dari total 15 data mahasiswa per angkatan. Implementasi dilakukan menggunakan *Jupyter Notebook* dengan bahasa pemrograman Python.

Langkah-langkah pengujian menggunakan *split validation* adalah sebagai berikut:

1. **Memuat data**: Data mahasiswa dalam format CSV dimuat menggunakan *library* pandas.
2. **Memisahkan variabel**: Variabel independen (IPK semester 1-5) dipisahkan dari variabel dependen (status kelulusan).
3. **Membagi data**: Data dibagi menjadi data *training* sebesar 40% (6 data) dan data *testing* sebesar 60% (9 data) menggunakan fungsi *train_test_split* dari *scikit-learn*.
4. **Normalisasi data**: Data dinormalisasi menggunakan *StandarScaler* untuk memastikan setiap fitur memiliki skala yang sama.
5. **Implementasi KNN**: Algoritma KNN diimplementasikan dengan jumlah tetangga (*neighbors*) sebanyak 2 menggunakan fungsi *KneighborsClassifier*.
6. **Prediksi**: Model melakukan prediksi pada data *testing*.
7. **Evaluasi**: Hasil prediksi dievaluasi menggunakan *confusion matrix* dan *classification report*.

1. Gambar Penggunaan Library Pandas dan NumPy

```
import pandas as pd
import numpy as np
```

Gambar 1 menunjukkan import library pandas dan numpy yang digunakan untuk manipulasi, pembersihan, dan persiapan data. Pandas berfokus pada manipulasi data, sedangkan numpy digunakan untuk scientific computing dan pembentukan objek N-Dimensional array.

2. Gambar Data dari Angkatan 18.cvs

```
dataset = pd.read_csv("angkatan18.csv", sep = ";")
```

```
dataset.head(10)

#Jika lebih dari 4 tahun dikategorikan predikit tidak tepat waktu atau dilambangkan angka nol "0",
#sedangkan jika kurang dari 4 tahun dikategorikan tepat waktu atau dilambangkan dengan angka satu "1"
```

	S1	S2	S3	S4	S5	Tepat Waktu
0	4.00	3.82	3.90	3.88	4.00	1
1	3.80	4.00	3.92	3.85	3.90	1
2	3.75	3.88	3.85	3.90	3.79	1
3	3.69	3.77	3.80	3.75	3.82	1
4	3.80	3.55	3.78	3.82	3.75	1
5	3.65	3.80	3.79	4.00	3.92	1
6	3.55	3.20	3.55	3.82	3.65	0
7	3.76	3.50	3.42	3.65	3.75	0
8	3.55	3.44	3.57	3.72	3.82	0
9	3.19	3.56	3.68	3.73	3.60	0

Gambar 2 menampilkan data mahasiswa angkatan 18 dalam format CSV yang telah dimuat menggunakan library pandas dengan nama 'dataset'. Data ini berisi IPK semester 1 sampai 5 dan keterangan kelulusan.

3. Gambar Tipe Data Setiap Kolom

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15 entries, 0 to 14
Data columns (total 6 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   S1           15 non-null     float64
 1   S2           15 non-null     float64
 2   S3           15 non-null     float64
 3   S4           15 non-null     float64
 4   S5           15 non-null     float64
 5   Tepat Waktu  15 non-null     int64
dtypes: float64(5), int64(1)
memory usage: 848.0 bytes
```

Gambar 3 menunjukkan tipe data yang digunakan pada setiap kolom dalam dataset untuk memastikan kompatibilitas saat pemrosesan data.

4. Gambar Menghilangkan Variabel "Tepat Waktu"

```
x = dataset.drop(["Tepat Waktu"], axis = 1)
x.head(10)
```

	S1	S2	S3	S4	S5
0	4.00	3.82	3.90	3.88	4.00
1	3.80	4.00	3.92	3.85	3.90
2	3.75	3.88	3.85	3.90	3.79
3	3.69	3.77	3.80	3.75	3.82
4	3.80	3.55	3.78	3.82	3.75
5	3.65	3.80	3.79	4.00	3.92
6	3.55	3.20	3.55	3.82	3.65
7	3.76	3.50	3.42	3.65	3.75
8	3.55	3.44	3.57	3.72	3.82
9	3.19	3.56	3.68	3.73	3.60

```
y= dataset["Tepat Waktu"]
y.head(15)
```

```
0     1
1     1
2     1
3     1
4     1
5     1
6     0
7     0
8     0
9     0
10    0
11    0
12    0
13    0
14    0
Name: Tepat Waktu, dtype: int64
```

Gambar 4 menunjukkan proses pemilihan variabel independen yaitu nilai IPK semester 1 sampai 5 dengan menghilangkan kolom "Tepat Waktu" dari tabel, serta menentukan variabel dependennya.

5. Gambar Pembagian Jumlah Data Training dan Testing

```
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.60)
```

```
print(f"Jumlah data pelatihan: {len(x_train)}, Jumlah data pengujian: {len(x_test)}, Jumlah Keseluruhan data :{len(x)}")
```

```
Jumlah data pelatihan: 6, Jumlah data pengujian: 9, Jumlah Keseluruhan data :15
```

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
scaler.fit(x_train)

x_train = scaler.transform(x_train)
x_test = scaler.transform(x_test)
```

Gambar 5 menunjukkan pembagian data menjadi data training sebesar 40% (6 data) dan data testing sebesar 60% (9 data) menggunakan fungsi train_test_split. Selanjutnya dilakukan normalisasi menggunakan StandarScaler.

6. Gambar Penggunaan KNN

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier (n_neighbors=2)
```

Gambar 6 menunjukkan implementasi algoritma KNN menggunakan fungsi KneighborsClassifier dengan jumlah neighbors sebanyak 2 untuk melakukan klasifikasi.

7. Gambar Memasukkan Data Training dan Melakukan Prediksi

```
knn.fit(x_train, y_train)
```

```
KNeighborsClassifier(n_neighbors=2)
```

```
y_pred = knn.predict(x_test)
df = pd.DataFrame({"Prediksi": y_pred})
print(df)
```

```
   Prediksi
0         1
1         0
2         0
3         1
4         0
5         1
6         0
7         0
8         0
```

Gambar 7 menunjukkan proses memasukkan data training pada fungsi KNN dan melakukan prediksi, kemudian menampilkan hasil prediksi menggunakan fungsi dari library pandas.

8. Gambar Hasil Probabilitas

```
knn.predict_proba(x_test)
```

```
array([[0. , 1. ],
       [1. , 0. ],
       [1. , 0. ],
       [0. , 1. ],
       [1. , 0. ],
       [0. , 1. ],
       [1. , 0. ],
       [0.5, 0.5],
       [1. , 0. ]])
```

Gambar 8 menampilkan hasil perhitungan probabilitas menggunakan predict_proba dari KNN.

Tabel 1. Confusion Matrix Split Validation

	Prediksi Tepat Waktu	Prediksi Tidak Tepat Waktu
Aktual Tepat Waktu	5	0
Aktual Tidak Tepat Waktu	1	3

15. Gambar Confusion Matrix Cross Validation

```
: report = classification_report(y, y_pred_cv)
  print("Classification Report:")
  print(report)

Classification Report:
              precision    recall  f1-score   support

           0       0.83      0.83      0.83         6
           1       0.89      0.89      0.89         9

    accuracy                           0.87        15
   macro avg       0.86      0.86      0.86        15
weighted avg       0.87      0.87      0.87        15


: from sklearn.metrics import accuracy_score
  akurasi = accuracy_score(y, y_pred_cv)
  print ("Tingkat akurasi : %d persen" %(akurasi*100))

Tingkat akurasi : 86 persen
```

Gambar 15 menampilkan total confusion matrix dari cross validation yang digabungkan dari setiap lipatan. Hasil prediksi benar berjumlah 5 dengan selisih kesalahan 1, dan prediksi salah sebanyak 1 dengan selisih kesalahan berjumlah 8.

16. Gambar Hasil Akurasi Cross Validation

```
: from sklearn.metrics import accuracy_score
  akurasi = accuracy_score(y, y_pred_cv)
  print ("Tingkat akurasi : %d persen" %(akurasi*100))

Tingkat akurasi : 86 persen
```

Gambar 16 menunjukkan hasil perhitungan keakuratan prediksi menggunakan cross validation yang menghasilkan akurasi sebesar 86%.

Berdasarkan *confusion matrix* pada Tabel 2 dan Gambar 15, hasil prediksi menunjukkan data yang diprediksi dengan benar berjumlah 5 dengan selisih kesalahan 1, dan prediksi yang salah sebanyak 1 dengan selisih kesalahan berjumlah 8. Hasil ini menunjukkan bahwa *cross*

validation menghasilkan distribusi kesalahan yang berbeda dibandingkan *split validation*, di mana terjadi peningkatan kesalahan pada kelas "Tepat Waktu" (dari 0 menjadi 1) dan peningkatan kebenaran pada kelas "Tidak Tepat Waktu" (dari 3 menjadi 8).

Perhitungan akurasi:

$$\text{Akurasi} = \frac{5+8}{5+8+1+1} \times 100\% = \frac{13}{15} \times 100\% = 86{,}67\%$$

Hasil akurasi yang diperoleh adalah sebesar **86%** [20][21]. Penurunan akurasi dari 88% menjadi 86% menunjukkan bahwa *cross validation* memberikan estimasi yang lebih konservatif terhadap kinerja model dibandingkan *split validation*

Hasil Cross Validation pada Angkatan 2015 dan 2017:

Pengujian menggunakan *cross validation* pada dua dataset angkatan yang berbeda menunjukkan hasil yang bervariasi [22]. Variasi ini terjadi karena perbedaan distribusi data antar angkatan yang mempengaruhi proses pelatihan dan pengujian pada setiap lipatan.

Tabel 3. Perbandingan Akurasi Cross Validation Angkatan 2015 dan 2017

Angkatan	**Akurasi**
2015	93%
2017	86%

Pada angkatan 2015, diperoleh akurasi sebesar **93%**, sedangkan pada angkatan 2017 diperoleh akurasi sebesar **86%**. Hasil ini menunjukkan bahwa *cross validation* lebih sensitif terhadap karakteristik data setiap angkatan dibandingkan *split validation* [23].

17. Gambar Tingkat Akurasi Angkatan 2015

```
from sklearn.metrics import accuracy_score
akurasi = accuracy_score(y, y_pred_cv)
print ("Tingkat akurasi : %d persen" %(akurasi*100))
```

```
Tingkat akurasi : 93 persen
```

Gambar 17 menunjukkan tingkat akurasi pada angkatan 2015 menggunakan cross validation sebesar 93%.

18. Gambar Tingkat Akurasi Angkatan 2017

```
from sklearn.metrics import accuracy_score
akurasi = accuracy_score(y, y_pred_cv)
print ("Tingkat akurasi : %d persen" %(akurasi*100))
```

```
Tingkat akurasi : 86 persen
```

Gambar 18 menunjukkan tingkat akurasi pada angkatan 2017 menggunakan cross validation sebesar 86%.

Pada angkatan 2015, diperoleh akurasi sebesar **93%**, sedangkan pada angkatan 2017 diperoleh akurasi sebesar **86%**. Hasil ini menunjukkan bahwa *cross validation* lebih sensitif terhadap karakteristik data setiap angkatan dibandingkan *split validation* [23]. Perbedaan akurasi sebesar 7% ini mengindikasikan bahwa kualitas data (sebaran IPK dan status kelulusan) pada setiap angkatan memiliki tingkat homogenitas yang berbeda.

Berdasarkan *confusion matrix* pada Tabel 1, hasil prediksi menunjukkan data yang diprediksi dengan benar berjumlah 5 dengan selisih salah 0, dan prediksi yang salah sebanyak 1 dengan perbandingan kesalahan berjumlah 3.

Perhitungan akurasi menggunakan rumus:

$$\text{Akurasi} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \times 100\%$$

$$\text{Akurasi} = \frac{5 + 3}{5 + 3 + 0 + 1} \times 100\% = \frac{8}{9} \times 100\% = 88{,}89\%$$

Hasil akurasi yang diperoleh adalah sebesar **88%**. Hasil ini konsisten untuk semua angkatan yang diuji, menunjukkan stabilitas metode *split validation* pada penelitian ini [17][18].

3.2 Hasil Pengujian dengan Cross Validation

Pengujian menggunakan *cross validation* dilakukan dengan membagi dataset menjadi 5 lipatan (*n_splits=5*) dan menggunakan pengacakan (*shuffle=true*) sebelum pembagian data [19].

Langkah-langkah pengujian menggunakan *cross validation* adalah sebagai berikut:

1. **Membuat KFold**: Objek KFold dibuat dengan n_splits=5 dan shuffle=true.
2. **Normalisasi data**: Data dinormalisasi menggunakan *StandarScaler* untuk menghitung nilai *mean* dan deviasi standar dari fitur dalam dataset, kemudian mengaplikasikan penskalaan menggunakan rumus z-score:

$$z = \frac{x - \mu}{\sigma}$$

Keterangan:

   - *x* = nilai asli
   - μ = mean
   - σ = standar deviasi

3. **Implementasi KNN**: Objek KNN dibuat dengan jumlah tetangga (*neighbors*) sebanyak 2 dan metrik jarak *Minkowski* dengan p=2 (*Euclidean Distance*).
4. **Prediksi**: Prediksi dilakukan menggunakan *cross_val_predict*.
5. **Evaluasi**: Hasil prediksi dievaluasi menggunakan *confusion matrix* yang digabungkan dari setiap lipatan.

Tabel 2. Confusion Matrix Cross Validation

	Prediksi Tepat Waktu	Prediksi Tidak Tepat Waktu
Aktual Tepat Waktu	5	1
Aktual Tidak Tepat Waktu	1	8

Berdasarkan *confusion matrix* pada Tabel 2, hasil prediksi menunjukkan data yang diprediksi dengan benar berjumlah 5 dengan selisih kesalahan 1, dan prediksi yang salah sebanyak 1 dengan selisih kesalahan berjumlah 8.

Perhitungan akurasi:

$$\text{Akurasi} = \frac{5+8}{5+8+1+1} \times 100\% = \frac{13}{15} \times 100\% = 86{,}67\%$$

Hasil akurasi yang diperoleh adalah sebesar **86%** [20][21].

Hasil Cross Validation pada Angkatan 2015 dan 2017:

Pengujian menggunakan *cross validation* pada dua dataset angkatan yang berbeda menunjukkan hasil yang bervariasi [22].

Tabel 3. Perbandingan Akurasi Cross Validation Angkatan 2015 dan 2017

Angkatan	**Akurasi**
2015	93%
2017	86%

Pada angkatan 2015, diperoleh akurasi sebesar **93%**, sedangkan pada angkatan 2017 diperoleh akurasi sebesar **86%**. Hasil ini menunjukkan bahwa *cross validation* lebih sensitif terhadap karakteristik data setiap angkatan dibandingkan *split validation* [23].

Hasil Split Validation pada Angkatan 2016 dan 2018:

Tabel 4. Hasil Akurasi Split Validation

Angkatan	**Akurasi**
2016	88%
2018	88%

Pada perhitungan menggunakan *split validation*, jika jumlah data *training*, data *testing*, dan *neighbor* sama, hasil akan tetap sama yaitu **88%** [24].

3.3 Analisis Perbandingan

Berdasarkan hasil pengujian dari 4 angkatan dan 2 metode validasi data dalam melakukan prediksi menggunakan metode K-Nearest Neighbor, terdapat beberapa hal yang sangat berpengaruh dalam menentukan hasil prediksi dari dataset yang digunakan pada penelitian ini [25].

Tabel 5. Perbandingan Hasil Akurasi Split Validation dan Cross Validation

Angkatan	**Split Validation**	**Cross Validation**
2015	88%	93%
2016	88%	-

2017	88%	86%
2018	88%	-

Pada penggunaan metode validasi menggunakan *split validation*, banyaknya data yang digunakan sebagai data *testing* dan data *training* sangat berpengaruh pada akurasi prediksi. Jumlah pada penelitian ini menggunakan 6 data sebagai data *training* dan 9 data sebagai data *testing*, dengan hasil tingkat akurasi tertinggi 88%. Sedangkan jika menggunakan data testing lebih atau kurang dari 9 akan menghasilkan tingkat akurasi yang hanya 10-50% [26].

Sedangkan ketika menggunakan *cross validation*, menghitung dua data set yang berbeda saja dapat memiliki hasil atau tingkat keakurasian yang berbeda-beda. Contoh ketika menghitung dataset angkatan tahun 2015 dan 2017, dari dua dataset tersebut menghasilkan tingkat akurasi yang berbeda yaitu 93% dan 86%. Padahal pada perhitungan menggunakan *split validation*, jika jumlah data *training*, data *testing*, dan *neighbor* yang sama, hasil akan tetap sama yaitu 88% [27].

Keterbatasan jumlah data (60 rekam data) menjadi faktor penting dalam pemilihan metode validasi. Penelitian ini menunjukkan bahwa *split validation* merupakan pilihan yang lebih stabil untuk dataset terbatas, seperti yang dinyatakan oleh penelitian sebelumnya [28].

4. Conclusion

Berdasarkan penelitian yang berjudul "Prediksi Ketepatan Waktu Kelulusan Mahasiswa Universitas Islam Negeri Datokarama Palu Menggunakan Algoritma K-Nearest Neighbor" maka dari itu peneliti dapat menarik kesimpulan sebagai berikut:

1. Pada saat validasi data menggunakan *split validation*, banyaknya *neighbor*, data *training*, dan data *testing* dapat berpengaruh banyak pada tingkat akurasi dan selalu mendapatkan tingkat akurasi yang sama walaupun menggunakan data dari angkatan yang berbeda.
2. Sedangkan *cross validation* mendapat tingkat akurasi yang berbeda-beda jika menggunakan data dari angkatan yang berbeda walaupun n_splits pada kfold dan juga *neighbors* pada jumlah yang sama, tingkat akurasi akan tetap berbeda. Tidak seperti ketika menggunakan *split validation*.
3. Dikarenakan data yang didapatkan terbatas, *split validation* merupakan pilihan yang lebih stabil untuk penelitian menggunakan metode *K-Nearest Neighbors* kali ini.

References

[1] Widoyo H. Pentingnya Pendidikan Dalam Kehidupan. 2023. binus.ac.id.

[2] Marisa F, Maukar AL, Akhriza TM. Data Mining Konsep dan Penerapannya. Deepublish; 2021.

[3] Kononenko I, Kukar M. Machine Learning and Data Mining. Elsevier Science; 2007.

[4] Anwarudin W, Purnomosidi DP, Kristono D. Implementasi Algoritma K-Nearest Neighbor untuk Prediksi Kelulusan Mahasiswa. 2022:1-14.

[5] Heryana D. Data Mining untuk Memprediksi Kelulusan Mahasiswa Pendidikan Matematika UIN Raden Intan Lampung Menggunakan Naive Bayes. 2019:1-29.

[6] Adnyana IMB. Prediksi Lama Studi Mahasiswa dengan Metode Random Forest. 2015.

[7] Pratana A, Wihandika RC, Ratnawati DE. Implementasi Algoritme Support Vector Machine (SVM) untuk Prediksi Ketepatan Waktu Kelulusan Mahasiswa. 2018.

[8] Agwil W, Fransiska H, Hidayati N. Analisis Ketepatan Waktu Lulus Mahasiswa dengan Menggunakan Bagging CART. 2020;6.

[9] Fajri MS, Septian N, Sanjaya E. Evaluasi Implementasi Algoritma Machine Learning K-Nearest Neighbors (KNN) pada Data Spektroskopi Gamma Resolusi Rendah. 2020;3:9-14.

[10] Erdiansyah U, Lubis AI, Erwansyah K. Komparasi Metode K-Nearest Neighbor dan Random Forest Prediksi Akurasi Klasifikasi Pengobatan Penyakit Kutil. 2022;6:208-214.

[11] Nugroho YS. Data Mining Menggunakan Algoritma Naïve Bayes untuk Klasifikasi Kelulusan Mahasiswa Universitas Dian Nuswantoro:1-3.

[12] Purnomo D. Model Prototyping pada Pengembangan Sistem Informasi. 2017:1-8.

[13] Mohan N, Undeland TM, Robbins WP. Power Electronics. New York: John Wiley & Sons. 2005.

[14] Ward J, Peppard J. Strategic planning for Information Systems. Edisi 4. West Susse: John Willey & Sons Ltd. 2007.

[15] Casadei D, Serra G, Tani K. Implementation of a Direct Control Algorithm for Induction Motors Based on Discrete Space Vector Modulation. IEEE Transactions on Power Electronics. 2007; 15(4): 769-777.

[16] Calero C, Piatiini M, Pascual C, Serrano MA. Towards Data Warehouse Quality Metrics. Proceedings of the 3rd Intl. Workshop on Design and Management of Data Warehouses (DMDW). Interlaken. 2009; 39: 2-11.

[17] Yamin L, Wanming C. Implementation of Single Precision Floating Point Square Root on FPGAs. IEEE Symposium on FPGA for Custom Computing Machines. Napa. 2008: 226-232.

[18] Zade F, Talenta A. Editors. Advanced Fuzzy Control System. Yogyakarta: UAD Press. 2010.

[19] Arkanuddin M, Fadlil A, Sutikno T. A Neuro-Fuzzy Control for Robotic Application Based on Microcontroller. In: Krishnan R, Blaabjerg F. Editors. Advanced Control for Industrial Application. 2nd ed. London: Academic Press; 2006: 165-178.

[20] Rusdi M. A Novel Fuzzy ARMA Model for Rain Prediction in Surabaya. PhD Thesis. Surabaya: Postgraduate ITS; 2009.

[21] Ahmad LP, Hooper A. The Lower Switching Losses Method of Space Vector Modulation. CN103045489 (Patent). 2007.

[22] IEEE Standards Association. 1076.3-2009. IEEE Standard VHDL Synthesis Packages. New York: IEEE Press; 2009.

[23] James S, Whales D. The Framework of Electronic Goverment. U.S. Dept. of Information Technology. Report number: 63. 2005.

Foto 3x4

Rizzandy Rasyid lahir di Kota Palu, 27 November 2000. Memperoleh gelar Sarjana Teknik (S.Kom) dari Program Studi Teknik Informatika, Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Bina Mulia Palu pada tahun 2024. Saat ini berprofesi sebagai peneliti dan praktisi di bidang teknologi informasi. Minat penelitiannya adalah *data mining*, *machine learning*, dan sistem informasi akademik.

Alamat Email : rizzandyr27@gmail.com