



Department of Digital Business

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

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

Vol. 4 No. 3 (2025) pp: 873-883

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

## Klasifikasi Keterlambatan Pembayaran SPP Santri Menggunakan Algoritma *K-Nearest Neighbor* di Pesantren Al Fajar Tegal

Misbahu Surur<sup>1</sup>, Nugroho Adhi Santoso<sup>2</sup>, Bayu Aji Santoso<sup>3</sup>

<sup>1</sup>Teknik Informatika, STMIK YMI Tegal

<sup>2,3</sup>Sistem Informasi, STMIK YMI Tegal

<sup>1</sup>misbahsurur261@gmail.com, <sup>2</sup>nugrohoadhisantoso@stmik-tegal.ac.id\*, <sup>3</sup>bayu@stmik-tegal.ac.id

### Abstrak

Penelitian ini bertujuan untuk merancang sistem klasifikasi keterlambatan pembayaran SPP menggunakan metode *K-Nearest Neighbor* (*K-NN*) di Pondok Pesantren Al-Fajar Tegal. Permasalahan keterlambatan pembayaran sering kali berdampak pada pengelolaan keuangan pesantren, sehingga dibutuhkan sistem pendukung yang mampu mengidentifikasi potensi keterlambatan sejak dini. Dalam penelitian ini, dilakukan serangkaian pengujian dengan variasi rasio data pelatihan dan pengujian, yaitu 60:40, 70:30, 80:20, dan 90:10. Evaluasi performa model dilakukan menggunakan metrik akurasi, precision, recall, *F1-score*, serta validasi silang (*cross-validation*). Hasil terbaik diperoleh pada rasio 60:40 dengan nilai  $K = 1$ , yang menghasilkan akurasi sebesar 92,31%. Sementara itu, melalui *cross-validation*, nilai  $K = 3$  menunjukkan performa yang paling stabil dengan tingkat kesalahan rendah dan akurasi maksimal sebesar 90,91%. Model terbukti mampu mengenali pola keterlambatan secara konsisten, bahkan dalam skenario data pelatihan yang lebih sedikit. Temuan ini menunjukkan bahwa metode *K-NN* efektif dalam mengklasifikasikan santri berdasarkan kemungkinan keterlambatan pembayaran SPP. Sistem ini memiliki potensi besar untuk dikembangkan sebagai alat bantu administrasi keuangan pesantren dalam memberikan peringatan dini dan meningkatkan efisiensi pengelolaan keuangan secara keseluruhan.

*Kata kunci:* Klasifikasi, Pembayaran SPP, *K-Nearest Neighbor*, *Cross-Validation*

### 1. Latar Belakang

Pondok Pesantren Al Fajar Tegal, yang berlokasi di Dukuh Babakan, Desa Jatimulya, Kecamatan Lebaksiu, Kabupaten Tegal, merupakan lembaga pendidikan Islam yang mengintegrasikan pendidikan agama dan umum. Keberlangsungan operasional pesantren sangat bergantung pada pembayaran Sumbangan Pembinaan Pendidikan (SPP) oleh santri. Dana ini digunakan untuk pembangunan infrastruktur, perbaikan sarana prasarana, peningkatan fasilitas pendukung, dan pemenuhan kebutuhan operasional harian. Namun, keterlambatan pembayaran SPP sering terjadi, dan pada tahun ajaran 2023/2024 tercatat sejumlah santri yang mengalami keterlambatan pembayaran. Kondisi ini berpotensi mengganggu kelancaran administrasi dan proses pembelajaran.

Keterlambatan pembayaran Sumbangan Pembinaan Pendidikan (SPP) tidak hanya berdampak pada aspek keuangan, tetapi juga berimplikasi pada keberlangsungan kegiatan akademik [1]. Santri yang menunggak sering kali terkendala dalam mengurus administrasi tepat waktu, sehingga dapat mempengaruhi keikutsertaan dalam ujian atau kegiatan akademik lainnya. Masalah ini memerlukan solusi yang mampu memberikan peringatan dini kepada pihak pengelola, sehingga dapat dilakukan tindakan pencegahan untuk menjaga stabilitas keuangan dan kelancaran operasional.

Prediksi keterlambatan pembayaran menjadi strategi efektif untuk mengantisipasi risiko dan meminimalkan dampak negatif terhadap operasional pesantren [2],[3]. Dengan sistem prediksi yang tepat, pengelola dapat mengidentifikasi santri yang berpotensi terlambat membayar sebelum jatuh tempo, memberikan peringatan atau pendekatan khusus, serta mengatur strategi penagihan yang lebih terarah. Pendekatan ini mendukung manajemen keuangan yang proaktif dan mengurangi potensi gangguan pada kegiatan belajar mengajar.

Beberapa penelitian sebelumnya telah memanfaatkan metode klasifikasi berbasis *machine learning* dengan menerapkan algoritma *K-Nearest Neighbor (KNN)* untuk mendeteksi keterlambatan pembayaran, tetapi sebagian besar dilakukan di sekolah umum dan belum banyak diterapkan di lingkungan pesantren yang memiliki karakteristik data berbeda. Selain itu, masih ada penelitian yang belum mengoptimalkan evaluasi model untuk memastikan kestabilan prediksi pada berbagai variasi pembagian data. Kesenjangan ini menjadi peluang untuk mengembangkan model prediksi yang sesuai dengan kondisi sosial-ekonomi santri di pesantren.

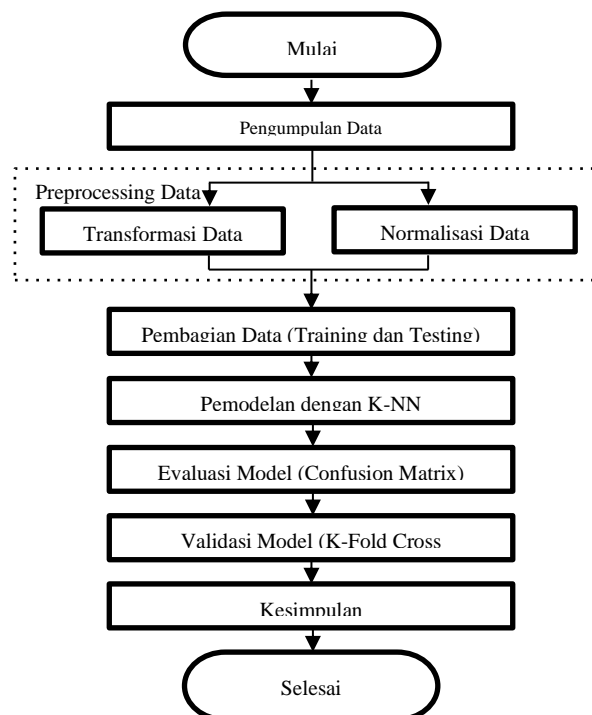
*K-Nearest Neighbor (KNN)* merupakan algoritma klasifikasi berbasis pembelajaran *instance-based* yang menentukan kelas suatu data baru dengan mengacu pada sejumlah tetangga terdekat (*k-nearest neighbors*) dari data tersebut dalam ruang fitur [4],[5]. Penentuan kedekatan biasanya dihitung menggunakan ukuran jarak seperti *Euclidean Distance*, *Manhattan Distance*, atau *Minkowski Distance*. KNN tidak memerlukan proses pelatihan model yang kompleks, karena seluruh data latih digunakan sebagai acuan dalam proses klasifikasi [6],[7]. Keunggulan KNN terletak pada kemampuannya menangani data dengan distribusi yang tidak diketahui serta kemudahan implementasi, sementara kelemahannya adalah sensitif terhadap skala data, nilai *k* yang dipilih, dan keberadaan *noise* pada dataset [8]. Oleh karena itu, normalisasi data dan pemilihan parameter *k* yang tepat sangat penting untuk memperoleh hasil prediksi yang optimal [9].

Kebaruan dari penelitian ini terletak pada penerapan algoritma *K-Nearest Neighbor (KNN)* yang dioptimalkan untuk memprediksi keterlambatan pembayaran SPP di Pondok Pesantren Al Fajar Tegal dengan mempertimbangkan atribut spesifik seperti pekerjaan dan penghasilan orang tua santri. Penelitian ini juga mengintegrasikan evaluasi *cross-validation* untuk menguji kestabilan model pada berbagai skenario pembagian data, sehingga model tidak hanya mengutamakan akurasi tetapi juga konsistensi performa.

Tujuan penelitian ini adalah mengembangkan model prediksi keterlambatan pembayaran SPP menggunakan metode KNN yang disesuaikan dengan konteks Pondok Pesantren Al Fajar Tegal. Evaluasi model dilakukan menggunakan metrik *akurasi*, *precision*, *recall*, dan *F1-score*, serta diuji kestabilannya melalui *cross-validation*. Hasil penelitian diharapkan menjadi dasar pengambilan keputusan bagi pengelola pesantren dalam melakukan tindakan preventif, menjaga stabilitas keuangan, dan memastikan kelancaran kegiatan akademik.

## 2. Metode Penelitian

Penelitian ini menggunakan pendekatan kuantitatif dengan metode klasifikasi berbasis algoritma *K-Nearest Neighbor (KNN)* untuk memprediksi keterlambatan pembayaran Sumbangan Pembinaan Pendidikan (SPP) di Pondok Pesantren Al Fajar Tegal, alur penelitian dapat dilihat pada gambar 1.



Gambar 1. Alur Penelitian

Gambar 1. Menunjukkan bahwa penelitian ini dimulai dengan pengumpulan data yang relevan, di mana data mengenai keterlambatan pembayaran SPP di Pondok Pesantren Al Fajar Tegal dikumpulkan dari bagian keuangan Pesantren Al Fajar Tegal. Setelah data terkumpul, langkah selanjutnya adalah persiapan data yang meliputi transformasi, dan normalisasi untuk memastikan kualitas data yang optimal. Kemudian, data yang telah disiapkan dibagi menjadi dua bagian: data latih dan data uji, untuk memungkinkan evaluasi model yang adil. Selanjutnya, algoritma *K-Nearest Neighbor* (K-NN) diterapkan pada data latih untuk membangun model prediksi keterlambatan pembayaran. Selanjutnya model dievaluasi menggunakan *confusion matrix* untuk mengukur akurasi dan efektivitas prediksi, memberikan gambaran yang jelas tentang kinerja model, dan yang terakhir hasil akurasi di validasi menggunakan *K-Fold Cross Validation* Prediksi keterlambatan pembayaran SPP Menggunakan Metode *K-Nearest Neighbor*. Dataset yang digunakan terdiri dari data historis pembayaran SPP santri tahun ajaran 2023/2024. Setiap entri memiliki atribut seperti pekerjaan ayah, pekerjaan ibu, penghasilan ayah, penghasilan ibu, dan label keterlambatan (tepat atau terlambat).

### 2.1. Pengumpulan Data

Pengumpulan data adalah proses mengumpulkan informasi atau data yang relevan untuk analisis. Dalam penelitian ini, data yang dikumpulkan meliputi atribut seperti penghasilan ayah, penghasilan ibu, pekerjaan ayah, dan pekerjaan ibu, serta label pembayaran yaitu tepat dan terlambat bayar. Data yang diperoleh yaitu sebanyak 65 data santri.

Tabel 1. Data Pembayaran SPP Tepat Dan Terlambat

No	NIS	Nama	Kelas	Gender	Penghasilan Ayah	Penghasilan Ibu	Pekerjaan Ayah	Pekerjaan Ibu	Label
1	A1	Annisa	Reguler	P	Rp. 2,000,000 - Rp. 4,999,999	Rp. 2,000,000 - Rp. 4,999,999	Karyawan Swasta	PNS	Tepat
2	A2	Andini	Reguler	P	Rp. 2,000,000 - Rp. 4,999,999	Rp. 2,000,000 - Rp. 4,999,999	Karyawan Swasta	Pedagang	Tepat
3	A3	Royanah	Reguler	P	Rp. 2,000,000 - Rp. 4,999,999	Rp. 2,000,000 - Rp. 4,999,999	Buruh Harian Lepas	Petani/Perkebunan	Tepat
4	A4	Bunga	Reguler	P	Rp. 2,000,000 - Rp. 4,999,999	Rp. 2,000,000 - Rp. 4,999,999	PNS	Pedagang	Tepat
5	A5	Mentari	Reguler	P	Rp. 2,000,000 - Rp. 4,999,999	Rp. 2,000,000 - Rp. 4,999,999	Buruh Harian Lepas	Petani/Perkebunan	Tepat
6	A6	Sukhem	Reguler	P	Rp. 2,000,000 - Rp. 4,999,999	Rp. 2,000,000 - Rp. 4,999,999	Karyawan Swasta	Pedagang	Tepat
7	A7	Silpiyati	Reguler	P	Rp. 2,000,000 - Rp. 4,999,999	Rp. 2,000,000 - Rp. 4,999,999	Buruh Harian Lepas	Petani/Perkebunan	Tepat
8	A8	Epitamala	Reguler	P	Rp. 1.000.000 - Rp. 1.999,000	Rp. 1.000.000 - Rp. 1.999,000	Buruh Harian Lepas	Buruh Harian Lepas	Terlambat
9	A9	Hani	Reguler	P	Rp. 1.000.000 - Rp. 1.999,000	Rp. 1.000.000 - Rp. 1.999,000	Pedagang	Ibu Rumah Tangga	Terlambat
10	A10	Hikmah	Reguler	P	Rp. 1.000.000 - Rp. 1.999,000	Rp. 1.000.000 - Rp. 1.999,000	Buruh Harian Lepas	Petani/Perkebunan	Terlambat
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
65	A65	Fajar	Tahfidz	L	Rp. 5,000,000 - Rp. 6,000,000	Kurang dari Rp. 500,000	PNS	Wiraswasta	Tepat

### 2.2. Persiapan Data

Persiapan data merupakan tahapan penting yang bertujuan untuk memastikan bahwa dataset yang digunakan siap dan layak diproses dalam pemodelan. Proses ini diawali dengan pembersihan data, yaitu mengidentifikasi dan menangani nilai yang hilang (*missing values*) dengan teknik imputasi yang sesuai, baik menggunakan nilai rata-

rata, median, maupun modus tergantung pada jenis atributnya. Selanjutnya, dilakukan konversi data ke format yang sesuai, seperti mengubah atribut kategorik menjadi bentuk numerik menggunakan metode *encoding* agar dapat diproses oleh algoritma. Pada tahap ini juga dilakukan normalisasi atau standarisasi atribut numerik untuk menyamakan skala data, sehingga setiap fitur memiliki pengaruh yang seimbang dalam proses perhitungan jarak pada algoritma *K-Nearest Neighbor*. Selain itu, data juga diperiksa untuk mendeteksi adanya *outlier* yang berpotensi mengganggu kinerja model, serta memastikan konsistensi tipe data dan format penulisan. Dengan melalui tahapan persiapan data yang komprehensif ini, kualitas dataset dapat terjaga sehingga model KNN dapat menghasilkan prediksi yang lebih akurat dan andal [10].

### 2.3. Pembagian Data

Pembagian data adalah proses membagi dataset menjadi subset untuk pelatihan (*training*) dan pengujian (*testing*). Biasanya, data dibagi dalam rasio tertentu, seperti 60% untuk pelatihan dan 40% untuk pengujian, untuk memastikan model dapat dievaluasi dengan data yang belum pernah dilihat sebelumnya. Dalam penelitian ini menggunakan percobaan rasio 60:40, 70:30, 80:20 dan 90:10. Dari beberapa percobaan yang telah dilakukan, rasio pembagian data 60:40 terbukti menghasilkan akurasi tertinggi, yaitu sebesar 92,31%. Artinya, saat 60% data digunakan untuk melatih model dan 40% sisanya untuk menguji, model *K-Nearest Neighbor* memberikan performa prediksi yang paling optimal dibandingkan rasio lainnya. Hasil ini mengindikasikan bahwa penggunaan proporsi data pelatihan dan pengujian yang seimbang mampu meningkatkan kinerja model serta memberikan evaluasi yang lebih akurat dan stabil. Oleh karena itu, pada tahap selanjutnya, pengujian dan evaluasi model akan difokuskan menggunakan rasio 60:40 agar memperoleh hasil yang lebih konsisten.

### 2.4. Pemodelan Algoritma *K-Nearest Neighbor* (KNN)

Tahap pemodelan melibatkan pencarian pola atau informasi menarik dalam data yang telah dipilih berdasarkan proses persiapan data. Teknik yang digunakan untuk model prediksi adalah algoritma *K-Nearest Neighbor*, di mana metode ini menggunakan atribut yang diinisialisasi sebagai nilai *k*, yaitu jumlah tetangga yang dijadikan acuan dalam algoritma [11]. Prosesnya adalah sebagai berikut:

- a. Menentukan nilai *k*
- b. Hitung Jarak (*d*) test data dan train data menggunakan rumus *Eucliden Distance*

$$Eucliden\ Distance = \sqrt{\sum_{i=1}^p (a_k - b_k)^2} \quad (1)$$

- c. Sorting (*d*) dari yang terkecil sampai terbesar
- d. Klasifikasi test data mayoritas

### 2.5. Evaluasi *Confusion Matrix*

Evaluasi menggunakan *confusion matrix* adalah metode untuk menilai kinerja model klasifikasi [12]. *Confusion matrix* menunjukkan jumlah prediksi benar dan salah untuk setiap kelas, memungkinkan perhitungan metrik seperti *akurasi*, *precision*, *recall*, dan *f1-score*, yang memberikan gambaran komprehensif tentang kinerja model [13].

Tabel 2. Confision Matrix

		<i>Predicted</i>	
		Negatif	Positif
<i>Aktual</i>	Negatif	TP	FP
	Positif	FN	TN

Keterangan:

TP (*True Positive*) adalah jumlah data poin berlabel yes yang teridentifikasi dengan benar.

TN (*True Negative*) adalah jumlah data poin berlabel no yang teridentifikasi dengan benar.

FP (*False Positive*) adalah jumlah data poin berlabel no yang sebenarnya teridentifikasi sebagai yes.

FN (*False Negative*) adalah jumlah data poin berlabel yes yang sebenarnya teridentifikasi sebagai no.

Nilai akurasi dapat dihitung dengan rumus sebagai berikut:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

### 2.6. Validasi *K-Fold Cross Validation*

Untuk memastikan keandalan model klasifikasi yang digunakan dalam penelitian ini, diterapkan teknik *K-Fold Cross Validation* sebagai metode validasi [14]. Teknik ini digunakan untuk mengevaluasi performa model secara lebih objektif dengan membagi data menjadi beberapa bagian (fold), sehingga seluruh data memiliki kesempatan yang sama untuk menjadi data latih dan data uji [15]. Dalam penelitian ini, digunakan konfigurasi *10-Fold Cross Validation*, yang berarti data dibagi menjadi 10 bagian, dan proses pelatihan serta pengujian dilakukan sebanyak 10 kali dengan rotasi fold yang berbeda-beda.

## 3. Hasil dan Diskusi

### 3.1. Implementasi *K-Nearest Neighbor* dengan Python

```
# Langkah 1: Import library yang dibutuhkan
import pandas as pd

# Langkah 2: Membaca file CSV
df = pd.read_csv('iss.csv')

# Langkah 3: Menampilkan 5 data teratas untuk mengecek
df.head()
```

Gambar 3. Source Code Mengimport dan Menampilkan Dataset

Gambar 3 merupakan source code pemrograman Python untuk proses penginputan data yang akan diuji. Pertama yaitu mengimport dataset yang sudah disiapkan untuk dilakukan pengujian data.

No	Kelas	Gender	Penghasilan Ayah	Penghasilan Ibu	Pekerjaan Ayah	Pekerjaan Ibu	Label	
0	1	Reguler	P	Rp. 2,000,000 - Rp. 4,999,999	Rp.2,000,000 - Rp. 4,999,999	Karyawan Swasta	PNS	Tepat
1	2	Reguler	P	Rp. 2,000,000 - Rp. 4,999,999	Rp.2,000,000 - Rp. 4,999,999	Karyawan Swasta	Pedagang	Tepat
2	3	Reguler	P	Rp. 2,000,000 - Rp. 4,999,999	Rp. 2,000,000 - Rp. 4,999,999	Buruh Harian Lepas	Petani/Perkebunan	Tepat
3	4	Reguler	P	Rp. 2,000,000 - Rp. 4,999,999	Rp. 2,000,000 - Rp. 4,999,999	PNS	Pedagang	Tepat
4	5	Reguler	P	Rp. 2,000,000 - Rp. 4,999,999	Rp. 2,000,000 - Rp. 4,999,999	Buruh Harian Lepas	Petani/Perkebunan	Tepat

Gambar 4. Hasil Import Dataset

Gambar 4 tampilan data setelah dilakukan inputan data, dari dataset yang sudah ada terdapat enam atribut yaitu kelas, gender, penghasilan ayah, penghasilan ibu, pekerjaan ayah, pekerjaan ibu, dan label target. Jumlah data yang digunakan adalah 65. Kemudian sebelum data di proses yaitu dilakukan normalisasi data, proses normalisasi data dapat dilihat pada gambar 5.

```
from sklearn.preprocessing import MinMaxScaler

# Inisialisasi scaler
scaler = MinMaxScaler()

# Daftar fitur yang akan dinormalisasi
fitur_normalisasi = ['Kelas', 'Gender', 'Penghasilan Ayah', 'Penghasilan Ibu',
                    'Pekerjaan Ayah', 'Pekerjaan Ibu']

# Terapkan normalisasi
df_encoded[fitur_normalisasi] = scaler.fit_transform(df_encoded[fitur_normalisasi])

# Lihat hasil normalisasi
df_encoded.head()
```

Gambar 5. Source Code Normalisasi Data

Berdasarkan source code pada gambar 5, dapat dilihat hasil normasilsasi yang dilakukan dapat dilihat pada gambar 6 sebagai berikut.

Kelas	Gender	Penghasilan Ayah	Penghasilan Ibu	Pekerjaan Ayah	Pekerjaan Ibu	Label	
0	0.0	1.0	0.666667	1.00	0.333333	0.666667	0
1	0.0	1.0	0.666667	1.00	0.333333	0.777778	0
2	0.0	1.0	0.666667	0.75	0.066667	0.888889	0
3	0.0	1.0	0.666667	0.75	0.533333	0.777778	0
4	0.0	1.0	0.666667	0.75	0.066667	0.888889	0

Gambar 6. Hasil Normalisasi Data

Kemudian dilanjutkan ke proses pembagian data menjadi x dan y dapat dilihat pada gambar 7.

```
# Memisahkan fitur dan label
X = df_encoded.drop('Label', axis=1) # semua kolom kecuali Label
y = df_encoded['Label']             # hanya kolom Label

# Menampilkan dimensi hasil pemisahan
print("Shape X:", X.shape)
print("Shape y:", y.shape)
```

Gambar 7. Source Code Membagi Atribut X dan Y

Gambar 7 terdapat variabel x dan y, dimana x memiliki atribut data yaitu *feature\_columns* dan variabel y memiliki atribut *target\_column*. Selanjutnya yaitu proses pemodelan algoritma *K-Nearest Neighbor*. Berikut adalah proses menerapkan algoritma *K-Nearest Neighbor* dapat dilihat pada gambar 8.

```
# Inisialisasi list untuk menyimpan hasil
error_rate_knn = []
k_value         = []
pred_knn       = []
acc_knn        = []

# Rentang nilai K (jumlah tetangga)
k_range = range(1, 21)

# Looping untuk setiap nilai K
for i in k_range:
    print('Proses Neighbor ke-', i)
    k_value.append(i)

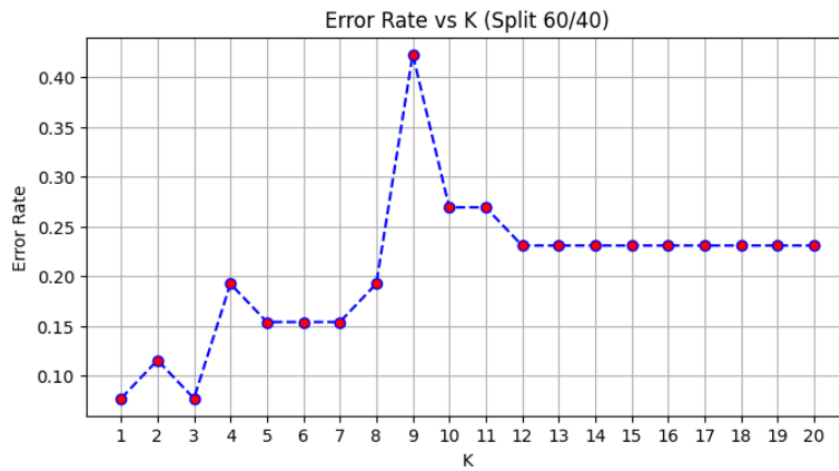
    # Inisialisasi dan latih model
    knn = KNeighborsClassifier(n_neighbors=i)
    knn.fit(X_train, y_train)

    # Prediksi
    pred_i = knn.predict(X_test)
    pred_knn.append(pred_i)

    # Hitung error rate dan akurasi
    error_rate_knn.append(np.mean(pred_i != y_test))
    acc_knn.append(accuracy_score(y_test, pred_i))
    print('Selesai')
```

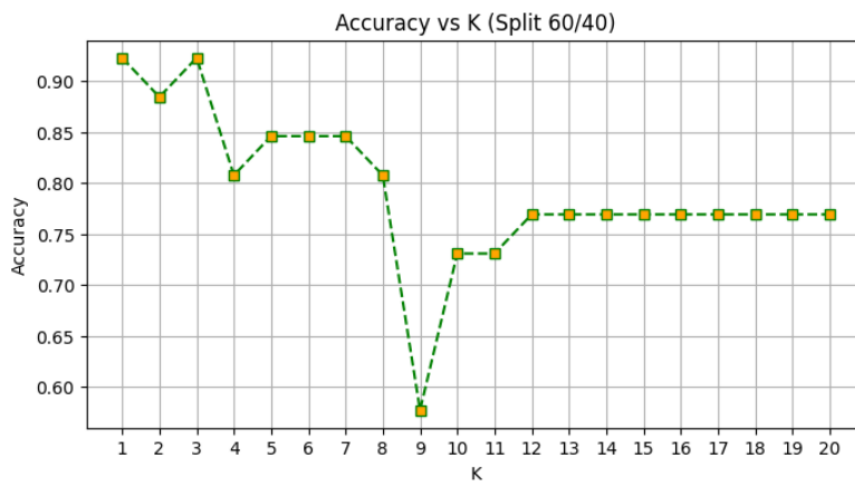
Gambar 8. Source Code Proses Menerapkan Algoritma KNN

*Source code* pada Gambar 8 digunakan untuk mengevaluasi performa algoritma *K-Nearest Neighbor* (KNN) dengan berbagai nilai K (1–20). Prosesnya meliputi inisialisasi list untuk menyimpan prediksi, akurasi, *error rate*, dan nilai K, kemudian melatih model KNN pada setiap nilai K dan melakukan prediksi pada data uji. Setiap hasil prediksi dihitung akurasinya dan *error rate*-nya, lalu disimpan untuk dibandingkan. Langkah ini memungkinkan penentuan nilai K dengan akurasi terbaik dan paling stabil, yang kemudian diperoleh melalui perintah pada Gambar 9.



Gambar 9. Hasil Minimum Error pada K di Split 60:40

Berdasarkan Gambar 9, grafik *Error Rate vs K* pada pembagian data 60:40 menunjukkan *error rate* terendah sekitar 8% pada  $K=1$ , menandakan kinerja terbaik pada nilai  $K$  kecil. *Error rate* meningkat tajam pada  $K=9$  hingga di atas 40%, lalu menurun dan stabil di kisaran 23% pada  $K=13-20$ . Hal ini menegaskan pentingnya pemilihan  $K$  optimal agar model tidak terlalu sensitif terhadap noise atau terlalu umum (*over-smoothing*). Nilai *error rate* maksimum dan  $K$  terbaik dihitung menggunakan source code pada Gambar 10.



Gambar 10. Hasil Akurasi dengan K Value (60:40)

Berdasarkan Gambar 10, grafik *Accuracy vs K Value* pada pembagian data 60:40 menunjukkan akurasi tertinggi sekitar 93% pada  $K=1$  dan  $K=3$ , menandakan kinerja terbaik saat menggunakan sedikit tetangga. Akurasi menurun tajam pada  $K=9$  hingga 58%, lalu stabil di kisaran 76%–77% untuk  $K>10$ . Hal ini menegaskan pentingnya pemilihan  $K$  optimal, dengan  $K$  kecil seperti 1 atau 3 sebagai pilihan terbaik dalam skenario ini.

### 3.2. Evaluasi Model

Tahap ini merupakan proses evaluasi terhadap algoritma KNN dengan menggunakan *confusion matrix* untuk menilai kinerja model, khususnya tingkat akurasi. Sebelum evaluasi dilakukan, model KNN digunakan terlebih dahulu untuk memprediksi data uji yang telah disiapkan. Selanjutnya, dilakukan evaluasi melalui *confusion matrix* yang menghasilkan metrik seperti *accuracy*, *precision*, *recall*, dan *f1-score*. Proses prediksi menggunakan KNN ditunjukkan pada Gambar 9.

```
# Prediksi ulang dengan model terbaik (jika belum dilakukan)
best_prediction_knn = knn_best.predict(X_test)

# Hitung metrik evaluasi
accuracy_neigh = accuracy_score(y_test, best_prediction_knn)
precision_neigh = precision_score(y_test, best_prediction_knn, average='weighted')
recall_neigh = recall_score(y_test, best_prediction_knn, average='weighted')
f1_score_neigh = f1_score(y_test, best_prediction_knn, average='weighted')

# Tampilkan hasil
print('Accuracy :', round(accuracy_neigh * 100, 2), '%')
print('Precision :', round(precision_neigh * 100, 2), '%')
print('Recall :', round(recall_neigh * 100, 2), '%')
print('F1-score :', round(f1_score_neigh * 100, 2), '%')
```

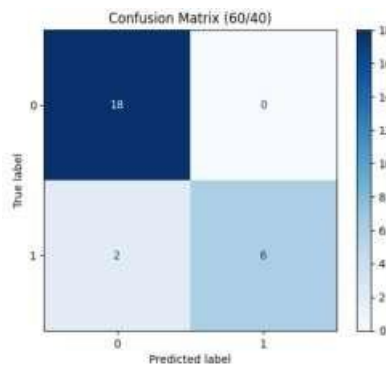
Gambar 11. Source Code Prediksi Data KNN

Berdasarkan hasil dari Gambar 11 maka menghasilkan akurasi dari pemodelan algoritma *K-Nearest Neighbor* yang dibuat sebelumnya, dapat dilihat pada gambar 12.

```
Accuracy : 90.0 %
Precision : 91.33 %
Recall : 90.0 %
F1-score : 89.52 %
```

Gambar 12. Hasil Pemodelan Algoritma KNN

Visualisasi *confusion matrix* untuk algoritma KNN ditampilkan pada Gambar 13. Melalui grafik tersebut, dapat diperoleh informasi mengenai jumlah *True Positive* (TP), *False Positive* (FP), *False Negative* (FN), serta *True Negative* (TN) yang dihasilkan dari proses klasifikasi.



Gambar 13. Hasil Inisialisasi Model KNN dengan K Terbaik

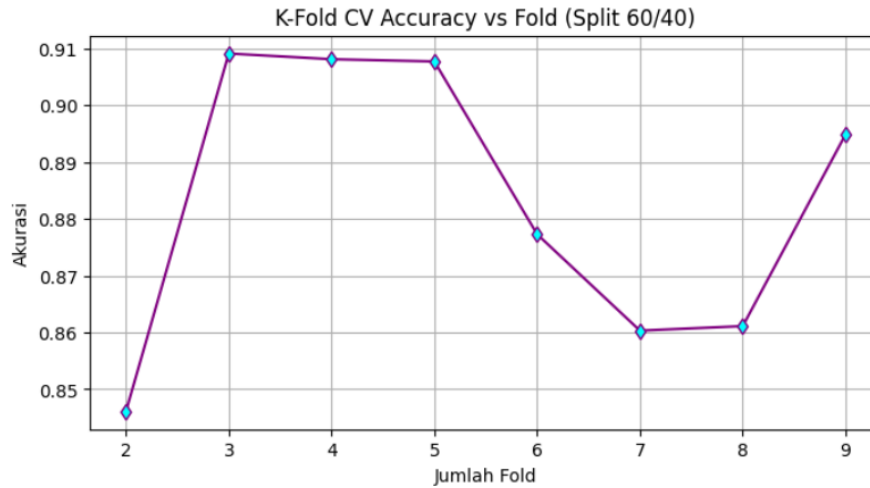
Berdasarkan gambar 13 menunjukkan hasil klasifikasi model *K-Nearest Neighbor* (KNN) pada rasio 60:40, di mana model berhasil mengklasifikasikan 18 data dengan benar sebagai kelas "Tepat" (*True Negative*) dan 6 data sebagai kelas "Terlambat" (*True Positive*). Terdapat 2 data yang salah diklasifikasikan dari "Terlambat" menjadi "Tepat" (*False Negative*), sementara tidak ada kesalahan dari "Tepat" ke "Terlambat" (*False Positive* = 0). Visualisasi ini mengindikasikan bahwa model memiliki kinerja klasifikasi yang baik pada skenario pembagian data 60% latih dan 40% uji.

### 3.3. Validasi Model

Untuk memastikan keandalan model, penelitian ini menggunakan 10-Fold Cross Validation yang membagi data menjadi 10 bagian, sehingga setiap bagian bergantian menjadi data latih dan uji. Proses ini dilakukan 10 kali dengan rotasi fold berbeda untuk evaluasi yang lebih objektif. Hasil *K-fold Cross Validation K-Nearest Neighbor Classifier* dengan K terbaik (K = 1) ditampilkan pada gambar 14.

DOI: <https://doi.org/10.31004/riggs.v4i3.2117>

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



Gambar 14. Hasil Validasi Model dengan *K-Fold Cross Validation*

Pada gambar 14. Menunjukkan hasil validasi model dengan *K-Fold Cross Validation* dengan K terbaik yaitu K=1. Hasil *K-Fold Cross Validation K-Nearest Neighbor Classifier* dengan K terbaik (K=1) ditampilkan pada tabel 3.

Tabel 3. Hasil *K-Fold Validation* KNN dengan K = 1

n_fold	Accuracy
2	0.84612
3	0.90909
4	0.90809
5	0.90769
6	0.87727
7	0.86032
8	0.86111
9	0.89484

Berdasarkan tabel 3 menunjukkan hasil evaluasi model *K-Nearest Neighbor (K-NN)* menggunakan metode *K-Fold Cross Validation* dengan jumlah lipatan (n\_folds) sebanyak 3. Hasilnya menunjukkan bahwa akurasi rata-rata dari model K-NN adalah 90,91%, yang berarti bahwa model mampu mengklasifikasikan data dengan tingkat ketepatan yang cukup tinggi.

Split Ratio	K Terbaik	Akurasi	Precision	Recall	F1-Score	Best CV Fold	CV Accuracy	
0	60/40	1	92.31%	93.08%	92.31%	91.96%	3	90.91%
1	70/30	1	90.0%	91.33%	90.0%	89.52%	3	90.91%
2	80/20	1	84.62%	87.41%	84.62%	82.82%	3	90.91%
3	90/10	7	85.71%	88.1%	85.71%	83.98%	9	83.13%

Gambar 15. Hasil Eksperimen Metode *K-Nearest Neighbor (K-NN)* dari 4 Split Data

Berdasarkan Gambar 15, rekap hasil eksperimen *K-Nearest Neighbor (K-NN)* menunjukkan bahwa pada split 60:40, nilai K terbaik adalah 1 dengan akurasi 92,31%, precision 93,08%, recall 92,31%, F1-score 91,96%, dan akurasi cross-validation tertinggi 90,91% pada fold ke-3. Split 70:30 dan 80:20 juga menghasilkan K=1 dengan akurasi masing-masing 90,0% dan 84,62%, sementara nilai cross-validation tetap stabil di 90,91%. Pada split 90:10, K terbaik berubah menjadi 7 dengan akurasi 85,71% dan cross-validation 83,13% pada fold ke-9. Secara

keseluruhan, K=1 konsisten memberikan performa terbaik pada sebagian besar split ratio, kecuali pada 90:10 di mana K=7 lebih optimal, dengan *cross-validation* menunjukkan kestabilan performa model.

### 3.4. Pembahasan

Penelitian ini merancang sistem prediksi keterlambatan pembayaran SPP berbasis *K-Nearest Neighbor (KNN)* di Pondok Pesantren Al-Fajar Tegal dengan *akurasi* 92,31%, *precision* 93,08%, *recall* 92,31%, *FI-score* 91,96%, dan *cross-validation* 90,91% (*n\_folds*=3). Penelitian sebelumnya menggunakan KNN dengan evaluasi 5-fold dan 10-fold, menghasilkan akurasi sekitar 85% dan *cross-validation* stabil di kisaran 82%, menegaskan pentingnya pemilihan K optimal dan pembagian data yang tepat. Keterbatasan penelitian sebelumnya adalah penggunaan data Kaggle yang belum merepresentasikan kompleksitas data lokal. Perbandingan hasil penelitian disajikan pada Tabel 3.

Tabel 4. Perbandingan Hasil Penelitian

Studi	Dataset	Model	K	Accuracy	AVG CV
Penelitian Sebelumnya	Harga Mobil Bekas	KNN	4	85%	82%
Penelitian Saat ini	Keterlambatan Pembayaran SPP	KNN	1	92.31%	90.91%

Berdasarkan Tabel 4, akurasi *K-Nearest Neighbor (KNN)* pada penelitian ini lebih tinggi dibanding penelitian sebelumnya. Dengan dataset 65 data administrasi pembayaran SPP santri Pondok Pesantren Al Fajar Tegal, yang terdiri dari dua kelas (“Tepat” dan “Terlambat”), KNN menghasilkan *akurasi* 92,31% serta *precision*, *recall*, dan *FI-score* yang menunjukkan performa baik. *Confusion matrix* mencatat 6 *True Positive*, 18 *True Negative*, 2 *False Positive*, dan 0 *False Negative*, menandakan model minim kesalahan negatif namun masih memiliki sedikit kelemahan dalam mendeteksi keterlambatan tertentu. Validasi *K-Fold Cross Validation* memberikan akurasi tertinggi 90,91% pada fold ke-3, menunjukkan kemampuan generalisasi yang baik. Secara keseluruhan, KNN memiliki kinerja tinggi dan stabil, sehingga berpotensi menjadi alat bantu efektif bagi pesantren untuk memantau dan mencegah keterlambatan pembayaran secara proaktif.

## 4. Kesimpulan

Berdasarkan hasil analisis dan implementasi, dapat disimpulkan bahwa model *K-Nearest Neighbor (K-NN)* menunjukkan akurasi tertinggi dalam memprediksi keterlambatan pembayaran SPP santri pada rasio pembagian data 60:40, yaitu sebesar 92,31%, yang menandakan proporsi ini paling optimal untuk pelatihan dan pengujian model. Secara keseluruhan, metode K-NN terbukti efektif dan andal dalam melakukan klasifikasi keterlambatan pembayaran berdasarkan fitur-fitur seperti kelas, gender, penghasilan orang tua, dan pekerjaan, sehingga dapat menjadi alat prediksi yang kuat bagi manajemen keuangan pesantren. Melalui penerapan *K-Fold Cross Validation* dengan k=3, diperoleh rata-rata akurasi 90,91% pada tiga rasio pembagian data (60:40, 70:30, dan 80:20) dengan K terbaik pada K=1, yang menunjukkan performa model K-NN cukup konsisten, tidak terlalu terpengaruh variasi pembagian data, dan memiliki kemampuan generalisasi yang baik pada data baru.

## Referensi

- [1] P. Nurhasanah and A. Ahmad, “Perancangan Aplikasi Pembayaran Sumbangan Pembinaan Pendidikan (SPP) Bulanan Santri pada Pondok Pesantren Syekh Hamzah Al-Fansury Kota Subulussalam,” *J. Manaj. Sist. Inf.*, vol. 4, no. 1, pp. 14–26, 2025.
- [2] W. Hidayat and M. J. Sodiq, “Implementasi Manajemen Risiko di Pondok Pesantren Miftahul Khoir Bandung,” *Al-Istimrar J. Ekon. Syariah*, vol. 2, no. 2, pp. 140–149, 2023.
- [3] A. Nova and P. N. Hayra, “PENERAPAN MANAJEMEN RISIKO UNTUK MEMINIMALISIR RISIKO KREDIT MACET PADA LISTRIK PASCA BAYAR (STUDI KASUS PADA PT. PLN ULP KOLAKA),” *J. Intelek Insa. Cendikia*, vol. 2, no. 2, pp. 2351–2357, 2025.
- [4] R. Nursyanti, G. Prakarsa, S. N. Rahmawati, and A. Andriana, “Implementasi K-Nearest Neighbor untuk Prediksi Harga Bibit Kelapa Sawit dengan Metode CRISP-DM,” *SISINFO J. Sist. Inf. dan Inform.*, vol. 7, no. 1, pp. 124–134, 2025.
- [5] B. Pikiran, C. Dibalas, and A. Better, “LACAK PESANAN,” 2024.
- [6] P. W. S. ST, M. Kom, D. U. M. T. M. Kuliah, A. H. D. Rinjani, and E. E. A. Lestari, “Komparasi Kinerja Algoritma K-Nearest Neighbors (KNN) dan Naive Bayes Dalam Klasifikasi Diagnosis Penyakit Diabetes”.
- [7] D. Ardiyansyah and N. Oktafiani, “Perbandingan metode pengukuran jarak pada k-nearest neighbour dalam klasifikasi data teks kardiovaskular,” *J. Inf. Syst. Manag. Digit. Bus.*, vol. 1, no. 2, pp. 116–122, 2024.
- [8] S. N. Bakri and L. S. Harahap, “Analisis klasifikasi Algoritma K-Nearest Neighbor (K-NN) pada struktur Daerah di Kota Medan,” *J. Ilmu Komput. dan Sist. Inf.*, vol. 4, no. 2, pp. 182–193, 2025.
- [9] R. Rahmadini, E. E. LorencisLubis, A. Priansyah, Y. RWN, and T. Meutia, “Penerapan data mining untuk memprediksi harga bahan pangan di Indonesia menggunakan algoritma K-Nearest Neighbor,” *J. Mhs. Akunt. Samudra*, vol. 4, no. 4, pp. 223–235, 2023.
- [10] R. N. Fitria, W. Sugianto, and A. C. Nur’aidha, “Prediksi Penyakit Diabetes Mellitus Tipe I dan Tipe II Menggunakan Metode KNN di Klinik Dharma Husada,” *Antigen J. Kesehat. Masy. dan Ilmu Gizi*, vol. 2, no. 3, pp. 39–53, 2024.

- [11] R. Sitepu and M. Manohar, "Implementasi Algoritma K-Nearest Neighbor Untuk Klasifikasi Pengajuan Kredit," *J. Sist. Informasi, Tek. Inform. dan Teknol. Pendidik.*, vol. 1, no. 2, pp. 49–56, 2022.
- [12] F. Nuraeni, D. Kurniadi, and M. H. Diazki, "Algoritma K-Nearest Neighbor pada Kasus Dataset Imbalanced untuk Klasifikasi Kinerja Karyawan Perusahaan," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 11, no. 3, pp. 557–568, 2024.
- [13] A. Fakhri, M. A. Hamzami, M. R. Hadiano, and N. I. S. Alifah, "Perbandingan Akurasi Algoritma C4. 5 dan K-NN Untuk Prediksi Kelulusan Mahasiswa Penerima Beasiswa," *J. Komput. Antart.*, vol. 3, no. 1, pp. 18–25, 2025.
- [14] R. Tuntun, K. Kusri, and K. Kusnawi, "Analisis Perbandingan Kinerja Algoritma Klasifikasi dengan Menggunakan Metode K-Fold Cross Validation," *J. Media Inform. Budidarma*, vol. 6, no. 4, p. 2111, 2022.
- [15] A. Nugroho and A. Amrullah, "Evaluasi Kinerja Algoritma K-NN Menggunakan K-Fold Cross Validation Pada Data Debitur KSP Galih Manunggal," *J. Inform. Teknol. dan Sains*, vol. 5, no. 2, pp. 294–300, 2023.