



KLASIFIKASI STATUS PENYELESAIAN MASALAH KELISTRIKAN PELANGGAN PLN MENGGUNAKAN ALGORITMA K-NEAREST NEIGHBOAR (KNN)

Langgi Irawanti¹, Sriani²

^{1,2}Universitas Islam Negeri Sumatera Utara, Sains dan Teknologi, Ilmu Komputer

^{1,2}Jl. Lap. Golf, Kp. Tengah, Kec. Pancur Batu, Kab. Deli Serdang, Sumatera Utara 20353

e-mail : irawantilanggi@gmail.com

ABSTRAK

klasifikasi data pelanggan PT PLN Persero Medan berdasarkan atribut-atribut seperti petugas, tarif, daya, dan status pelanggan. Masalah yang dihadapi dalam penelitian ini adalah bagaimana mengidentifikasi pola yang ada dalam data pelanggan dan memprediksi status pelanggan dengan tingkat akurasi yang tinggi. Hasil implementasi menunjukkan bahwa algoritma KNN mampu mengidentifikasi pola data secara efektif, menghasilkan tingkat akurasi sebesar 70%. Akurasi ini menunjukkan bahwa model memiliki kemampuan yang cukup baik dalam memprediksi status pelanggan. Namun, meskipun hasil ini cukup memuaskan, masih terdapat ruang untuk perbaikan. Peningkatan akurasi dapat dilakukan melalui penyesuaian nilai k untuk menemukan parameter yang lebih optimal atau dengan menerapkan teknik pra-pemrosesan data yang lebih mendalam, seperti normalisasi dan penghapusan atribut yang kurang relevan. Penelitian ini juga menegaskan pentingnya evaluasi terhadap kinerja model dalam berbagai kondisi data, khususnya dalam menangani dataset yang lebih besar dan lebih kompleks. Dengan demikian, implementasi KNN dalam penelitian ini tidak hanya memberikan gambaran tentang penerapan metode klasifikasi dalam data mining, tetapi juga membuka peluang pengembangan lebih lanjut untuk meningkatkan keandalan dan efektivitas model.

Kata kunci : Klasifikasi, KNN, Data Mining, Akurasi.

ABSTRACT

This study focuses on classifying customer data of PT PLN Persero Medan based on attributes such as officer, tariff, power, and customer status. The main challenge in this research is how to identify patterns within customer data and predict customer status with a high level of accuracy. The implementation results demonstrate that the K-Nearest Neighbors (KNN) algorithm effectively identifies data patterns, achieving an accuracy rate of 70%. This accuracy indicates that the model is relatively proficient at predicting customer status. However, despite the satisfactory results, there is still room for improvement. Accuracy could be enhanced by adjusting the value of k to find a more optimal parameter or by applying more advanced data preprocessing techniques, such as normalization and eliminating irrelevant attributes. This research also emphasizes the importance of evaluating the model's performance under various data conditions, particularly when handling larger and more complex datasets. Therefore, the implementation of KNN in this study not only provides insights into the application of classification methods in data mining but also opens opportunities for further development to enhance the model's reliability and effectiveness.

Key word : Classifying, KNN, Data Mining, Accuracy.

1. PENDAHULUAN

Di era modern ini, listrik menjadi elemen yang sangat penting dalam kehidupan sehari-hari. Sebagian besar aktivitas manusia, baik skala individu maupun bisnis, sangat bergantung pada listrik (Abdullah et al., 2020). Berbagai peralatan membutuhkan daya listrik, sehingga penggunaannya harus disesuaikan dengan kebutuhan masing-masing pihak. Listrik juga merupakan salah satu sektor industri terbesar di

dunia, yang sangat vital bagi keberlangsungan aktivitas manusia maupun proses produksi di sektor industri (Rismayanti et al., 2024).

Di Indonesia, penyediaan dan pengelolaan listrik berada di bawah naungan PT. PLN. Dinamika kemajuan teknologi dan informasi mendorong penggunaan alat-alat berbasis listrik, yang secara langsung meningkatkan konsumsi energi. Akibatnya, rumah tangga mengharapkan akses jaringan listrik yang andal dan tanpa gangguan untuk mendukung



kelancaran aktivitas sehari-hari (Abu-Hamdeh et

Peningkatan konsumsi listrik di rumah tangga juga berdampak pada besarnya tarif listrik yang harus dibayarkan (Pasaribu, 2022). Pemerintah berupaya mengontrol harga listrik melalui kebijakan penyesuaian tarif dan pemberian subsidi tepat sasaran. PLN, sebagai penyedia layanan listrik, berupaya meningkatkan kualitas pelayanan kepada konsumen, salah satunya melalui pengelompokan data keluhan pelanggan. Metode ini bertujuan untuk menentukan status keluhan, sehingga dalam situasi serupa (klasifikasi tertentu), dapat ditemukan pola yang konsisten.

Salah satu metode yang digunakan untuk menganalisis dan mengelompokkan data keluhan pelanggan adalah algoritma KNN (K-Nearest Neighbor). Metode ini mampu membuat pola penyebaran data melalui aplikasi seperti ACMT (Aplikasi Catat Meter). Data keluhan pelanggan yang telah diolah akan direkam dalam dokumen khusus yang dikenal dengan Berita Acara (BA).

Pendapatan PLN diperoleh dari penjualan listrik kepada konsumen, baik di sektor industri maupun rumah tangga. Pengukuran konsumsi listrik dilakukan melalui dua cara, yaitu meter pascabayar dan prabayar (Karina et al., 2022). Pada meter pascabayar, pembacaan dilakukan setiap bulan, sementara meter prabayar memungkinkan pelanggan membeli token listrik sesuai kebutuhan. Untuk meter pascabayar, konsumsi energi dihitung berdasarkan pembacaan langsung pada meteran pelanggan.

Algoritma KNN digunakan untuk mengklasifikasikan data berdasarkan jarak terdekat antara objek data (Cholil et al., 2021). Penentuan nilai K yang optimal sangat penting dalam algoritma ini, karena nilai K yang terlalu besar dapat mengurangi efek noise, tetapi juga dapat menyebabkan batas antar kelas menjadi tidak jelas. Algoritma ini, yang bersifat non-parametrik, merupakan salah satu metode paling sederhana dalam data mining (Widaningsih et al., 2024). Meski efektif pada dataset kecil, KNN memiliki kelemahan jika diterapkan pada dataset besar dan kompleks, terutama dalam hal efisiensi waktu (Septria et al., 2022).

Metode ini menghitung jarak terdekat untuk memprediksi label kelas dari data baru. Pada praktiknya, salah satu tantangan utama dalam pembacaan meteran listrik adalah sering kali nomor meteran pelanggan tidak terbaca di aplikasi yang tersedia. Hal ini menjadi salah satu penyebab utama perselisihan antara PLN dan pelanggan, karena PLN tidak dapat mendeteksi masalah yang dialami konsumen secara tepat waktu.

Dengan penerapan algoritma KNN, PLN dapat mengelompokkan data keluhan pelanggan secara lebih efisien, sehingga status penyelesaian masalah

al., 2021).

pelanggan dapat ditentukan dengan lebih mudah. Data yang telah terklasifikasi kemudian dapat diberikan kepada petugas lapangan untuk ditindaklanjuti melalui pemeriksaan langsung (Feriawan et al., 2024).

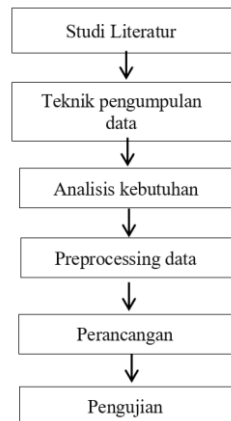
Seperti pada penelitian sebelumnya Pada penelitian sebelumnya yang dilakukan oleh elamin dkk yang berjudul Photovoltaic Output Power Forecast Using Artificial Neural Networks membahas tentang peramalan daya photovoltaic menggunakan metode feed forward neural network berdasarkan data suhu sekitar dan radiasi matahari. penelitian ini dilaksanakan dengan metode KNN dengan Tujuan agar dapat mengetahui daya listrik PLTS on grid pada rumah tinggal menggunakan metode k-NNDcNN berdasarkan data meteorologi untuk lima jam kedepan, yang mana hasilnya akan dibandingkan dengan data aktual (Elamim et al., 2018).

Berdasarkan uraian di atas, penelitian ini akan mengkaji penerapan data mining dengan menggunakan algoritma KNN untuk pengelompokan keluhan pelanggan di PT. PLN (Persero) Medan. Penelitian ini berjudul "Klasifikasi Status Penyelesaian Masalah Kelistrikan Pelanggan PLN Menggunakan Algoritma KNN". Dengan memanfaatkan data status yang tersedia, klasifikasi menggunakan algoritma KNN diharapkan dapat membantu dalam menentukan status penyelesaian keluhan pelanggan terkait masalah kelistrikan. Hasil pengelompokan ini memungkinkan pihak PLN untuk dengan cepat dan mudah mengetahui apakah keluhan pelanggan telah berstatus selesai atau belum selesai, sehingga memudahkan pelaksanaan tindak lanjut, seperti penyisiran langsung ke rumah pelanggan.

Tujuan dari penelitian ini adalah mengelompokkan data status pelanggan PLN pascabayar di wilayah Helvetia berdasarkan data yang tersedia. Pengelompokan ini bertujuan untuk membagi pelanggan ke dalam beberapa kelompok dengan karakteristik yang serupa.

2. METODE PENELITIAN

Tahap-tahap penelitian ini bisa dilihat pada gambar 1 yang terdiri dari pengumpulan data, analisis kebutuhan, preprosesing data, dan penerapan.



Gambar 1. Kerangka Penelitian

1. Studi Literatur

Studi literatur dilakukan untuk menghimpun informasi dari berbagai sumber, seperti jurnal dan karya ilmiah lainnya, yang relevan dengan topik yang sedang dibahas oleh penulis.

2. Teknik Pengumpulan Data

Metode pengumpulan data dalam penelitian ini dengan cara Wawancara. Proses wawancara akan dilakukan dengan Bapak Dedy Evandry Bangun, selaku Manajer Unit Pelayanan Pelanggan (ULP) Helvetia. Melalui wawancara dengan pihak PLN, data seperti demografi, riwayat penggunaan listrik, serta informasi lain yang mendukung penelitian akan dikumpulkan.

3. Analisis Kebutuhan

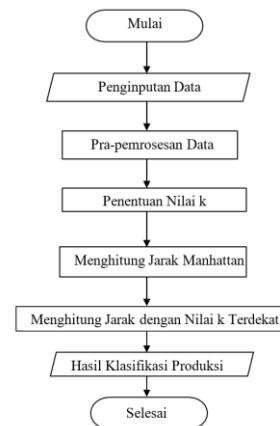
Fokus utama dalam penelitian ini adalah data pelanggan PLN, yang mencakup profil pelanggan, nilai daya, nomor meteran, serta status pelanggan. Selain itu, algoritma KNN menjadi komponen utama yang digunakan dalam penelitian ini. Oleh karena itu, diperlukan pemahaman mendalam tentang cara kerja algoritma KNN dan kemampuan untuk mengimplementasikannya menggunakan bahasa pemrograman tertentu, yaitu Python.

4. Preprocessing Data

Preprocessing atau pembersihan data merupakan tahap pemilihan atribut-atribut yang relevan untuk digunakan dalam analisis. Dalam data tersebut, terdapat beberapa atribut, seperti nilai daya, nomor meteran, dan keterangan status. Ketiga atribut ini nantinya akan menjadi penentu dalam hasil penyelesaian status kelistrikan pelanggan PLN.

5. Perancangan

Berdasarkan hasil analisis, penulis akan mengumpulkan data untuk pengelompokan berdasarkan kriteria (variabel) yang telah ditentukan. Kriteria (variabel) yang digunakan dalam penelitian ini meliputi nilai daya dan nomor meteran. Data yang diperoleh akan dimasukkan ke dalam Microsoft Excel untuk kemudian diolah menggunakan Python. Flowchart yang telah dirancang dapat dilihat pada gambar 2 :



Gambar 2. Perancangan Algoritma KNN

6. Pengujian

Pengujian hasil dilakukan dengan menggunakan Python sebagai alat untuk mengimplementasikan algoritma KNN. Data yang telah diproses dan dipilih melalui tahapan preprocessing dimasukkan ke dalam program Python untuk dilakukan klasifikasi. Python digunakan karena memiliki berbagai pustaka yang mendukung pemrosesan data dan penerapan algoritma machine learning, seperti scikit-learn. Selanjutnya, model KNN diuji untuk menentukan tingkat akurasi dalam mengklasifikasikan status penyelesaian masalah kelistrikan berdasarkan variabel yang telah ditentukan, seperti nilai daya dan nomor meteran pelanggan. Hasil pengujian akan memberikan gambaran sejauh mana algoritma KNN dapat diandalkan dalam menyelesaikan masalah klasifikasi pada konteks kelistrikan pelanggan PLN.

3. HASIL DAN PEMBAHASAN

Penelitian ini berfokus pada klasifikasi masalah kelistrikan pelanggan dengan menggunakan metode K-Nearest Neighbors (KNN). Metode ini bekerja dengan menghitung jarak terdekat antara data pelanggan dan data referensi untuk menentukan kategori masalah yang



tepat. Atribut yang digunakan dalam penelitian ini mencakup Petugas 1, Petugas 2, tarif, daya, dan status. Pendekatan ini dianggap efektif karena dapat memanfaatkan kedekatan data dalam ruang multidimensi untuk menghasilkan prediksi yang akurat.

Proses penelitian melibatkan pengumpulan dan analisis data pelanggan berdasarkan atribut-atribut tersebut, seperti kombinasi petugas yang menangani, biaya layanan, daya listrik, dan status penyelesaian masalah. Data ini selanjutnya diproses menggunakan algoritma KNN untuk menemukan pola atau hubungan yang mendasari klasifikasi masalah kelistrikan. Hasil dari penelitian ini diharapkan dapat membantu penyedia layanan listrik dalam mengidentifikasi dan menyelesaikan permasalahan pelanggan dengan lebih cepat dan efisien.

A. Deskripsi Data

Data dalam penelitian ini diperoleh dari PT. PLN (Persero) ULP Helvetia Medan, dengan total jumlah data sebanyak 1000 catatan. Data ini mencakup berbagai atribut seperti Petugas 1, Petugas 2, Tarif, Daya, dan Status. Untuk mendukung analisis dan klasifikasi, data dibagi menjadi dua kelompok, yaitu 80% sebagai data pelatihan (training) untuk melatih model. 20% sebagai data pengujian (testing) untuk mengevaluasi akurasi model.

Pembagian data ini dirancang agar model dapat melakukan prediksi dengan baik pada data baru. Data yang beragam dan representatif dari permasalahan listrik pelanggan diharapkan memberikan kontribusi signifikan untuk meningkatkan efisiensi dan efektivitas layanan di PT. PLN (Persero) ULP Helvetia Medan. Berikut paparan data pada tabel 1.

Table 1. Data Masalah Kelistrikan PLN

| NO | PTGS 1 | PTGS 2 | TARIF | DAYA | STATUS |
|----|---------------|---------------|-------|------|---------------|
| 1 | Iskandar | Syahrudin | B1t | 5500 | Belum Selesai |
| 2 | Sutarji | Erwin.Nst | R1t | 2200 | Selesai |
| 3 | Pandapotan | Muldiono | R1mt | 900 | Belum Selesai |
| 4 | G. Nainggolan | Didi Darmanto | R1mt | 900 | Selesai |
| 5 | Pandapotan | Muldiono | R1mt | 900 | Selesai |
| 6 | Pandapotan | Muldiono | R1mt | 900 | Belum Selesai |
| 7 | Erwin Nst | Sutarji | R1mt | 900 | Belum Selesai |
| 8 | Erwin Nst | Sutarji | R1t | 900 | Selesai |
| 9 | Heri | Sukardi | R1t | 2200 | Selesai |
| 10 | Erwin Nst | Sutarji | R1mt | 900 | Belum |

| | | | | | |
|-----|---------------|-----------|------|------|---------------|
| ... | ... | ... | ... | ... | Selesai |
| ... | ... | ... | ... | ... | ... |
| ... | ... | ... | ... | ... | ... |
| 790 | Gonnaget | Kardi | R3 | 7700 | Belum Selesai |
| 791 | Gonnaget | Kardi | R1t | 2200 | Belum Selesai |
| 792 | Gonnaget | Kardi | R2t | 5500 | Selesai |
| 793 | Pandapotan | Fadhli | R1t | 2200 | Selesai |
| 794 | Iskandar | Syahrudin | R1t | 900 | Selesai |
| 795 | G. Nainggolan | Sukardi | R2 | 5500 | Belum Selesai |
| 796 | Pandapotan | Fadhli | R2 | 5500 | Belum Selesai |
| 797 | Fadhli | Dapot | R2 | 5500 | Belum Selesai |
| 798 | Dayat | Karyadi | R1mt | 900 | Belum Selesai |
| 799 | Dayat | Karyadi | R3 | 7700 | Belum Selesai |
| 800 | Syahrudin | Iskandar | R1mt | 900 | Belum Selesai |

B. Transformasi Data

Setelah data diperoleh, langkah selanjutnya adalah mengonversi data tersebut ke dalam format numerik. Proses ini bertujuan untuk mempermudah pengolahan data dan analisis menggunakan algoritma K-Nearest Neighbors (KNN), yang memerlukan input berupa angka untuk menghitung

Selain itu, langkah ini juga membantu menghilangkan ambiguitas dalam interpretasi nilai atribut, sehingga menghasilkan model klasifikasi yang lebih terstruktur. Dapat dilihat pada tabel 2.

Table 2. Transformasi data

| N o | PTG S 1 | PTG S 2 | TARI F | DAY A | STATU S |
|-----|---------|---------|--------|-------|---------|
| 1 | 1 | 8 | 1 | 7 | 0 |
| 2 | 2 | 5 | 2 | 4 | 1 |
| 3 | 3 | 12 | 3 | 2 | 0 |
| 4 | 4 | 13 | 3 | 2 | 1 |
| 5 | 3 | 12 | 3 | 2 | 1 |
| 6 | 3 | 12 | 3 | 2 | 0 |
| 7 | 5 | 2 | 3 | 2 | 0 |
| 8 | 5 | 2 | 2 | 2 | 1 |
| 9 | 6 | 10 | 2 | 4 | 1 |
| 10 | 5 | 2 | 3 | 2 | 0 |
| ... | ... | ... | ... | ... | ... |
| ... | ... | ... | ... | ... | ... |
| ... | ... | ... | ... | ... | ... |
| 790 | 20 | 19 | 9 | 8 | 0 |
| 791 | 20 | 19 | 2 | 4 | 0 |
| 792 | 20 | 19 | 4 | 7 | 1 |
| 793 | 3 | 7 | 2 | 4 | 1 |
| 794 | 1 | 8 | 2 | 2 | 1 |



| | | | | | |
|-----|---|----|----|---|---|
| 795 | 4 | 10 | 12 | 7 | 0 |
| 796 | 3 | 7 | 12 | 7 | 0 |
| 797 | 7 | 7 | 12 | 7 | 0 |

| | | | | | |
|-----|----|---|---|---|---|
| 798 | 11 | 9 | 3 | 2 | 0 |
| 799 | 11 | 9 | 9 | 8 | 0 |
| 800 | 8 | 1 | 3 | 2 | 0 |

C. Normalisasi Data

Setelah data diubah menjadi bentuk numerik, langkah berikutnya adalah menormalisasi data. Normalisasi dilakukan untuk menyamakan skala antar atribut sehingga perhitungan jarak pada algoritma K-Nearest Neighbors (KNN) menjadi lebih akurat. Tanpa normalisasi, atribut dengan nilai yang memiliki rentang lebih besar dapat mendominasi perhitungan jarak, yang dapat menyebabkan bias pada hasil klasifikasi.

Sebagai gambaran, proses normalisasi data menggunakan metode Min-Max Scaler dapat dilihat pada nilai di kolom pertama dan baris pertama dalam dataset. Berikut adalah contoh penerapannya untuk memberikan pemahaman yang lebih jelas mengenai cara skala data diubah ke rentang tertentu.

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

$$X' = \frac{1 - 1}{20 - 1} = 0$$

Data yang telah melalui proses normalisasi juga akan menjadi data *training* dan 20% sisanya akan digunakan untuk pengujian (*testing*) yang disajikan dalam bentuk tabel. Dapat dilihat pada tabel 3.

Table 3. Normalisasi Data

| No | PTGS 1 | PTGS 2 | TARIF | DAYA | STATUS |
|-----|---------|---------|---------|---------|--------|
| 1 | 0 | 0,36842 | 0 | 0,4 | 0 |
| 2 | 0,05263 | 0,21053 | 0,08333 | 0,2 | 1 |
| 3 | 0,10526 | 0,57895 | 0,16667 | 0,06667 | 0 |
| 4 | 0,15789 | 0,63158 | 0,16667 | 0,06667 | 1 |
| 5 | 0,10526 | 0,57895 | 0,16667 | 0,06667 | 1 |
| 6 | 0,10526 | 0,57895 | 0,16667 | 0,06667 | 0 |
| 7 | 0,21053 | 0,05263 | 0,16667 | 0,06667 | 0 |
| 8 | 0,21053 | 0,05263 | 0,08333 | 0,06667 | 1 |
| 9 | 0,26316 | 0,47368 | 0,08333 | 0,2 | 1 |
| 10 | 0,21053 | 0,05263 | 0,16667 | 0,06667 | 0 |
| ... | ... | ... | ... | ... | ... |
| ... | ... | ... | ... | ... | ... |
| ... | ... | ... | ... | ... | ... |
| 795 | 0,15789 | 0,47368 | 0,91667 | 0,4 | 0 |
| 796 | 0,10526 | 0,31579 | 0,91667 | 0,4 | 0 |
| 797 | 0,31579 | 0,31579 | 0,91667 | 0,4 | 0 |
| 798 | 0,52632 | 0,42105 | 0,16667 | 0,06667 | 0 |
| 799 | 0,52632 | 0,42105 | 0,66667 | 0,46667 | 0 |
| 800 | 0,36842 | 0 | 0,16667 | 0,06667 | 0 |

Berikut ini adalah data yang digunakan untuk pengujian (data testing) yang akan digunakan untuk menguji performa model K-Nearest Neighbors

Table 4. Data Uji

| NO | PTGS 1 | PTGS 2 | TARIF | DAYA | STATUS |
|----|---------|---------|---------|---------|--------|
| 1 | 0 | 0 | 0,16667 | 0,06667 | ? |
| 2 | 0,42105 | 0,52632 | 0,16667 | 0,06667 | ? |
| 3 | 0,31579 | 0,10526 | 0,16667 | 0,06667 | ? |
| 4 | 0,10526 | 0,31579 | 1 | 0,06667 | ? |
| 5 | 0,15789 | 0,63158 | 0,16667 | 0,06667 | ? |

Setelah data dibagi menjadi data pelatihan dan pengujian, langkah berikutnya adalah menentukan nilai K dalam algoritma K-Nearest Neighbors (KNN). Nilai K mengacu pada jumlah tetangga terdekat yang akan digunakan sebagai acuan dalam proses klasifikasi.

Menentukan Jumlah K Pada penelitian ini, nilai K ditetapkan sebesar 7. Pemilihan ini didasarkan pada hasil evaluasi yang menunjukkan bahwa mempertimbangkan 7 tetangga terdekat memberikan performa klasifikasi yang paling optimal. Dengan memilih K=7, model dapat memperhitungkan informasi dari 7 data terdekat untuk menentukan kategori yang paling sesuai. Strategi ini membantu menghindari risiko overfitting sekaligus menjaga keseimbangan antara tingkat akurasi dan kemampuan generalisasi model pada data baru.

Menghitung Jarak Antar Data Penghitungan jarak antar data dilakukan menggunakan rumus Euclidean Distance, yang merupakan metode umum pada algoritma KNN untuk mengukur kedekatan antar data dalam ruang fitur. Rumus ini bekerja dengan menghitung selisih nilai pada setiap atribut, mengkuadratkan selisih tersebut, menjumlahkan hasilnya, lalu mengambil akar kuadrat dari jumlah tersebut.

$$D(ij) = \sqrt{(x_{1i} - x_{1j})^2 + (x_{2i} - x_{2j})^2 + \dots + (x_{ki} - x_{kj})^2}$$

Dengan menggunakan metode ini, jarak antara data uji dan data pelatihan dapat dihitung, sehingga tetangga terdekat dapat diidentifikasi untuk mendukung proses klasifikasi.

D. Pengujian Data

(0; 0; 0,16667; 0,06667)

$$d1 = \sqrt{(0-0)^2 + (0-0,36842)^2 + (0,16667-0)^2 + (0,06667-0,4)^2}$$



$$d1 = 0,524043008$$

$$d2 = \sqrt{(0 - 0,05263)^2 + (0 - 0,21053)^2 + (0,16667 - 0,08333)^2 + (0,06667 - 0,2)^2}$$

$$d2 = 0,267983661$$

$$d3 = \sqrt{(0 - 0,10526)^2 + (0 - 0,57895)^2 + (0,16667 - 0,16667)^2 + (0,06667 - 0,06667)^2}$$

$$d3 = 0,588440966$$

$$d4 = \sqrt{(0 - 0,15789)^2 + (0 - 0,63158)^2 + (0,16667 - 0,16667)^2 + (0,06667 - 0,06667)^2}$$

$$d4 = 0,65101655$$

$$d5 = \sqrt{(0 - 0,10526)^2 + (0 - 0,57895)^2 + (0,16667 - 0,16667)^2 + (0,06667 - 0,06667)^2}$$

$$d5 = 0,588440966$$

$$d6 = \sqrt{(0 - 0,10526)^2 + (0 - 0,57895)^2 + (0,16667 - 0,16667)^2 + (0,06667 - 0,06667)^2}$$

$$d6 = 0,588440966$$

$$d7 = \sqrt{(0 - 0,21053)^2 + (0 - 0,05263)^2 + (0,16667 - 0,16667)^2 + (0,06667 - 0,06667)^2}$$

$$d7 = 0,217008751$$

$$d8 = \sqrt{(0 - 0,21053)^2 + (0 - 0,05263)^2 + (0,16667 - 0,08333)^2 + (0,06667 - 0,06667)^2}$$

$$d8 = 0,23246151$$

$$d9 = \sqrt{(0 - 0,26316)^2 + (0 - 0,47368)^2 + (0,16667 - 0,08333)^2 + (0,06667 - 0,2)^2}$$

$$d9 = 0,56422369$$

$$d10 = \sqrt{(0 - 0,21053)^2 + (0 - 0,05263)^2 + (0,16667 - 0,16667)^2 + (0,06667 - 0,06667)^2}$$

$$d10 = 0,217008751$$

Dari perhitungan menggunakan persamaan Euclidean Distance di atas, diperoleh hasil jarak antara setiap data uji dengan data latih. Hasil perhitungan jarak ini kemudian akan ditampilkan dalam bentuk tabel untuk memudahkan analisis lebih lanjut. Tabel tersebut mencakup informasi tentang jarak setiap data uji terhadap semua data latih yang ada, yang selanjutnya akan digunakan untuk menentukan k-nearest neighbors atau tetangga terdekat yang berpengaruh terhadap proses klasifikasi dalam metode KNN. Seperti yang terlihat pada tabel 5.

Table 5. Jarak Antara Data (Data Ke-1)

| No | Jarak Antar Data | Status |
|-----|------------------|--------|
| 1 | 0,524043008 | 0 |
| 2 | 0,267983661 | 1 |
| 3 | 0,588440966 | 0 |
| 4 | 0,65101655 | 1 |
| 5 | 0,588440966 | 1 |
| 6 | 0,588440966 | 0 |
| 7 | 0,217008751 | 0 |
| 8 | 0,23246151 | 1 |
| ... | ... | ... |
| ... | ... | ... |
| 793 | 0,368137795 | 1 |
| 794 | 0,377728543 | 1 |
| 795 | 0,960682509 | 0 |
| 796 | 0,8856703 | 0 |
| 797 | 0,934374409 | 0 |
| 798 | 0,674014722 | 0 |
| 799 | 0,929675129 | 0 |
| 800 | 0,36842 | 0 |

Setelah jarak antar data dihitung, langkah berikutnya adalah mengurutkan nilai-nilai jarak tersebut dari yang terkecil hingga yang terbesar. Proses ini bertujuan untuk mempermudah identifikasi k-nearest neighbors (k tetangga terdekat), yang akan menjadi acuan dalam menentukan klasifikasi data uji. Mengingat nilai k yang digunakan dalam penelitian ini adalah 7, maka tujuh data dengan jarak terkecil akan dipilih sebagai referensi. Pemilihan ini memberikan wawasan yang lebih jelas tentang keterkaitan data uji dengan data pelatihan yang paling relevan, sehingga proses klasifikasi dapat dilakukan dengan lebih akurat dan dapat diandalkan.

Table 6. Pengurutan Jarak Data Uji Ke-1

| No | Jarak Antar Data | Status |
|-----|------------------|--------|
| 197 | 0,189214461 | 0 |
| 7 | 0,217008751 | 0 |
| 10 | 0,217008751 | 0 |
| 8 | 0,23246151 | 1 |
| 122 | 0,241830331 | 0 |
| 2 | 0,267983661 | 1 |
| 77 | 0,268371203 | 1 |
| 652 | 0,31579 | 0 |
| 173 | 0,320145656 | 1 |
| ... | ... | ... |
| ... | ... | ... |
| 783 | 1,381629121 | 1 |



| | | |
|-----|-------------|---|
| 615 | 1,401887587 | 1 |
| 715 | 1,401887587 | 1 |
| 768 | 1,419705144 | 1 |

Dari hasil perhitungan jumlah kemunculan Masing-masing kelas di antara 7 tetangga terdekat yang dipilih berdasarkan jarak terkecil, diketahui bahwa kelas 0 muncul sebanyak 4 kali, sementara kelas 1 muncul sebanyak 3 kali. Berdasarkan hasil tersebut, dapat disimpulkan bahwa data uji pertama diklasifikasikan ke dalam kelas 0, yang menunjukkan status "Belum Selesai."

E. Confusion Matrix

a) Belum Selesai

1. True Positive (TP): 127 contoh yang benar-benar "Belum Selesai" tapi diprediksi sebagai "Belum Selesai".
2. False Negative (FN): 11 contoh yang sebenarnya "Belum Selesai" tapi diprediksi sebagai "Selesai".
3. False Positive (FP): 49 contoh yang sebenarnya "Selesai" tapi diprediksi sebagai "Belum Selesai".
4. True Negative (TN): Semua contoh yang bukan "Belum Selesai" (total=13) yang diprediksi bukan sebagai "Belum Selesai".
5. Precision (Presisi): Persentase prediksi "Belum Selesai" yang benar dari semua prediksi "Belum Selesai".

$$Precision = \frac{TP}{TP + FP} = \frac{127}{127 + 49} = 0,721590909$$

$$Precision = 72,15\%$$

6. Recall (Sensitivitas): Persentase contoh "Belum Selesai" yang berhasil diprediksi dengan benar dari semua contoh yang sebenarnya "Belum Selesai".

$$Recall = \frac{TP}{TP + FN} = \frac{127}{127 + 11} = 0,920289855$$

$$Recall = 92,02\%$$

b) Selesai

1. True Positive (TP): 13 contoh yang benar-benar "Selesai" tapi diprediksi sebagai "Selesai".

| | | |
|-----|-------------|---|
| 792 | 1,419705144 | 1 |
| 737 | 1,472623104 | 0 |
| 790 | 1,519049017 | 0 |

2. False Negative (FN): 49 contoh yang sebenarnya "Selesai" tapi diprediksi sebagai "Belum Selesai".
3. False Positive (FP): 11 contoh yang sebenarnya "Belum Selesai" tapi diprediksi sebagai "Selesai".
4. True Negative (TN): Semua contoh yang bukan "Selesai" (total = 127) yang diprediksi bukan sebagai "Selesai".
5. Precision (Presisi): Persentase prediksi "Selesai" yang benar dari semua prediksi "Selesai".

$$Precision = \frac{TP}{TP + FP} = \frac{13}{13 + 11} = 0,541666667$$

$$Precision = 54,16\%$$

6. Recall (Sensitivitas): Persentase contoh "Selesai" yang berhasil diprediksi dengan benar dari semua contoh yang sebenarnya "Selesai".

$$Recall = \frac{TP}{TP + FN} = \frac{13}{13 + 49} = 0,209677419$$

$$Recall = 20,96\%$$

F. Akurasi

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% = \frac{13 + 127}{13 + 127 + 11 + 49} \times 100\% = 70\%$$

$$Akurasi = \frac{140}{200} \times 100\% = 70\%$$

Akurasi model yang diperoleh dalam penelitian ini adalah sebesar 70%, yang menunjukkan bahwa model mampu melakukan klasifikasi dengan tingkat keakuratan tersebut. Nilai akurasi ini diperoleh dari perbandingan jumlah prediksi yang benar terhadap total data yang diuji.

G. Hasil Pengujian

Pada tahap hasil, akan ditampilkan implementasi metode K-Nearest Neighbors (KNN) menggunakan bahasa pemrograman Python. Implementasi ini mencakup seluruh proses, mulai dari persiapan dataset, normalisasi data, pembagian dataset menjadi data latih dan data uji, hingga penerapan algoritma KNN untuk melakukan



klasifikasi.

H. Penerapan *K-Nearest Neighbors* Pada Python

1. Import Library

Berikut adalah gambar yang menunjukkan kode Python untuk mengimpor library yang digunakan dalam implementasi algoritma *K-Nearest Neighbors*.

```
#import library
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix, accuracy_score, classification_report
from sklearn.preprocessing import MinMaxScaler
import seaborn as sns
```

Gambar 3. Import Library

2. Import Data

Berikut ini merupakan kode Python yang digunakan untuk mengimpor data PLN ke dalam program. Proses ini dilakukan sebagai langkah awal untuk mempersiapkan data yang akan digunakan dalam analisis menggunakan metode *K-Nearest Neighbors* (KNN).

```
#import data
df = pd.read_csv('dataPLN.csv')
df
```

| | PETUGAS 1 | PETUGAS 2 | TARIF | DAYA | STATUS |
|-----|-----------|-----------|-------|------|---------------|
| 0 | 1 | 8 | 1 | 7 | BELUM SELESAI |
| 1 | 2 | 5 | 2 | 4 | SELESAI |
| 2 | 3 | 12 | 3 | 2 | BELUM SELESAI |
| 3 | 4 | 13 | 3 | 2 | SELESAI |
| 4 | 3 | 12 | 3 | 2 | SELESAI |
| ... | ... | ... | ... | ... | ... |
| 995 | 11 | 9 | 3 | 2 | SELESAI |
| 996 | 4 | 10 | 6 | 3 | SELESAI |
| 997 | 4 | 10 | 6 | 3 | BELUM SELESAI |
| 998 | 4 | 10 | 6 | 3 | SELESAI |
| 999 | 7 | 10 | 6 | 1 | SELESAI |

1000 rows x 5 columns

Gambar 4. Import Data

3. Info Data

Berikut ini adalah kode Python yang digunakan untuk memeriksa informasi dasar dari data PLN yang telah diimpor. Langkah ini bertujuan untuk memahami struktur dataset, seperti jumlah baris dan kolom, tipe data pada setiap kolom, serta jumlah data yang kosong atau tidak terisi. Informasi ini penting untuk memastikan bahwa dataset siap digunakan untuk analisis lebih lanjut dengan metode *K-Nearest Neighbors* (KNN).

```
#melihat info data
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 5 columns):
#   Column      Non-Null Count  Dtype
---  -
0   PETUGAS 1   1000 non-null   int64
1   PETUGAS 2   1000 non-null   int64
2   TARIF       1000 non-null   int64
3   DAYA       1000 non-null   int64
4   STATUS      1000 non-null   object
dtypes: int64(4), object(1)
memory usage: 39.2+ KB
```

Gambar 5. Info Data

4. Menghitung Label

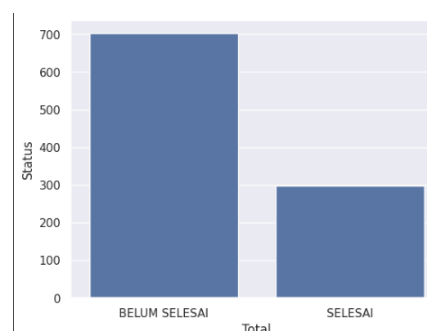
Berikut ini adalah kode Python yang digunakan untuk menghitung jumlah label pada atribut status dalam dataset PLN. Langkah ini bertujuan untuk mengetahui distribusi data pada setiap kategori label, seperti "Belum Selesai" dan "Selesai", guna memastikan keseimbangan data sebelum diterapkan pada algoritma *K-Nearest Neighbors* (KNN).

```
#menghitung label
df.STATUS.value_counts()
```

```
count
STATUS
BELUM SELESAI    702
SELESAI          298
dtype: int64
```

Gambar 6. Menghitung Label

Dan hasil tampilan Grafik terlihat pada Gambar 7.



Gambar 7. Grafik Label

5. Label Encoder

Berikut ini adalah kode Python untuk melakukan proses label encoding pada dataset PLN. Label encoding digunakan untuk mengonversi data



kategorikal menjadi data numerik agar dapat diproses oleh algoritma K-Nearest Neighbors (KNN).

```
#merubah label menjadi numerik
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
df['STATUS'] = label_encoder.fit_transform(df['STATUS'])
df.head()
```

| | PETUGAS 1 | PETUGAS 2 | TARIF | DAYA | STATUS |
|---|-----------|-----------|-------|------|--------|
| 0 | 1 | 8 | 1 | 7 | 0 |
| 1 | 2 | 5 | 2 | 4 | 1 |
| 2 | 3 | 12 | 3 | 2 | 0 |
| 3 | 4 | 13 | 3 | 2 | 1 |
| 4 | 3 | 12 | 3 | 2 | 1 |

Gambar 8. Label Encoder

6. Normalisasi Data

Berikut adalah kode Python untuk melakukan normalisasi data menggunakan metode Min-Max Scaler. Proses ini bertujuan untuk menyelaraskan skala data sehingga berada dalam rentang tertentu, biasanya antara 0 hingga 1. Normalisasi dilakukan agar setiap fitur memiliki pengaruh yang seimbang dalam perhitungan jarak antar data pada algoritma *K-Nearest Neighbors* (KNN).

```
#normalisasi data
from sklearn.preprocessing import MinMaxScaler
cols = df.columns
ms = MinMaxScaler()
X = ms.fit_transform(df)
X = pd.DataFrame(X, columns=[cols])
df_rounded = X.round(5)
df_rounded.to_csv('normalisasi langgi.csv', index=False)
print(df_rounded)
```

| | PETUGAS 1 | PETUGAS 2 | TARIF | DAYA | STATUS |
|---|-----------|-----------|---------|---------|--------|
| 0 | 0.00000 | 0.36842 | 0.00000 | 0.40000 | 0.0 |
| 1 | 0.05263 | 0.21053 | 0.08333 | 0.20000 | 1.0 |
| 2 | 0.10526 | 0.57895 | 0.16667 | 0.06667 | 0.0 |
| 3 | 0.15789 | 0.63158 | 0.16667 | 0.06667 | 1.0 |
| 4 | 0.10526 | 0.57895 | 0.16667 | 0.06667 | 1.0 |

Gambar 9. Normalisasi Data

7. Membagi Data

Berikut adalah contoh kode Python untuk membagi data menjadi data uji dan data latih menggunakan fungsi `train_test_split` dari pustaka `scikit-learn`. Pembagian ini dilakukan dengan proporsi 80% untuk data latih dan 20% untuk data uji, yang merupakan praktik umum dalam evaluasi model. Proses ini penting untuk memastikan bahwa model yang dibangun dapat diuji dengan data yang belum pernah dilihat sebelumnya, memberikan gambaran akurasi yang lebih realistis.

```
#membagi data training dan data testing
from pickle import NONE
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, shuffle=False)
print(f"x_train : {x_train.shape}")
print(f"y_train : {y_train.shape}")
print(f"x_test : {x_test.shape}")
print(f"y_test : {y_test.shape}")
```

```
x_train : (800, 4)
y_train : (800, 1)
x_test : (200, 4)
y_test : (200, 1)
```

Gambar 10. Membagi Data

8. Confusion Matrix

Berikut adalah contoh kode Python untuk menghitung dan menampilkan Confusion Matrix. Confusion matrix digunakan untuk menilai kinerja model klasifikasi dengan membandingkan prediksi model terhadap nilai sebenarnya. Matrix ini memberikan informasi tentang jumlah prediksi yang benar dan salah, serta klasifikasi yang tepat dan salah.

```
#confusion matrix
print(confusion_matrix(y_test, y_pred))
```

```
[[127 11]
 [ 49 13]]
```

Gambar 11. Confusion Matrix

9. Akurasi

Berikut adalah contoh kode Python untuk menghitung nilai recall, precision, dan akurasi dari model klasifikasi. Metrik ini digunakan untuk menilai sejauh mana model dapat mengklasifikasikan data dengan benar.

```
#Akurasi KNN
knn = KNeighborsClassifier(n_neighbors=7)
knn.fit(x_train, y_train)
y_pred = knn.predict(x_test)
KNN_acc = accuracy_score(y_pred, y_test)
print(classification_report(y_test, y_pred))
print("Akurasi KNN : {:.2f}%".format(KNN_acc*100))
```

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0.0 | 0.72 | 0.92 | 0.81 | 138 |
| 1.0 | 0.54 | 0.21 | 0.30 | 62 |
| accuracy | | | 0.70 | 200 |
| macro avg | 0.63 | 0.56 | 0.56 | 200 |
| weighted avg | 0.67 | 0.70 | 0.65 | 200 |

Akurasi KNN : 70.00%

Gambar 12. Akurasi Model

4. KESIMPULAN

Bagian Kesimpulan dari penelitian ini menunjukkan bahwa algoritma *K-Nearest Neighbors* (KNN) dapat diterapkan secara efektif untuk mengklasifikasikan data pelanggan PT PLN (Persero) Medan. KNN mampu mengelompokkan data berdasarkan atribut-atribut penting seperti petugas, tarif, daya, dan status pelanggan. Dengan memanfaatkan data latih yang memadai, algoritma



ini dapat memprediksi status pelanggan melalui perhitungan jarak terdekat antara data uji dan data latih. Hasil klasifikasi memberikan wawasan yang jelas mengenai pengelompokan pelanggan berdasarkan parameter yang tersedia.

Penerapan KNN dengan nilai $k = 7$ untuk memprediksi status pelanggan menunjukkan efektivitas algoritma ini dalam menganalisis dan mengklasifikasikan data berdasarkan pola yang ada. Algoritma ini menggunakan atribut-atribut yang tersedia untuk mengidentifikasi pola dan memprediksi status pelanggan dengan mengukur jarak antar data. Nilai k yang digunakan memungkinkan algoritma mempertimbangkan tujuh tetangga terdekat dalam menentukan hasil klasifikasi. Hasil implementasi membuktikan bahwa KNN dengan $k = 7$ mampu menghasilkan prediksi yang cukup akurat, meskipun evaluasi lebih lanjut diperlukan untuk meningkatkan performa model melalui eksplorasi parameter yang berbeda atau dataset yang lebih besar.

Model KNN yang digunakan berhasil mencapai akurasi sebesar 70% dalam mengklasifikasikan status pelanggan PT PLN (Persero) Medan. Meskipun akurasi ini cukup baik, masih terdapat peluang untuk peningkatan. Beberapa upaya perbaikan yang dapat dilakukan meliputi penyesuaian nilai k agar lebih optimal atau penerapan teknik pra-pemrosesan data yang lebih mendalam. Dengan demikian, meskipun hasil saat ini sudah memadai, evaluasi tambahan diperlukan untuk memastikan bahwa model dapat memberikan kinerja yang lebih andal dan efektif dalam berbagai kondisi data.

5. REFERENSI

- Abdullah, D., Prasetyo, M., Raharja, U., Ita Erlina, C., & Karim, A. (2020). *Sistem Informasi Pelayanan Dan Keluhan Pelanggan di PT.PLN* (1st ed.). CV. SEFA Bumi Persada. https://repository.unimal.ac.id/5594/1/BUKU_DAHLAN-MUHAJIR.pdf
- Abu-Hamdeh, N. H., Salilih, E. M., Alsulami, R. A., Rawa, M. J. H., Aljinaidi, A. A., Alazwari, M. A., Eltaher, M. A., Almitani, K. H., Abulkhair, H. A., Alnefaie, K. A., Abusorrah, A. M., & Sindi, H. F. (2021). The effects of incident solar radiation on the collector efficiency using coolant hybrid nanofluid via simulation of solar tower system with the parallel heat exchangers. *Journal of the Taiwan Institute of Chemical Engineers*, 124, 106–115. <https://doi.org/https://doi.org/10.1016/j.jtice.2021.04.063>
- Cholil, S. R., Handayani, T., Prathivi, R., & Ardianita, T. (2021). Implementasi Algoritma Klasifikasi K-Nearest Neighbor (KNN) Untuk Klasifikasi Seleksi Penerima Beasiswa. *IJCIT (Indonesian Journal on Computer and Information Technology)*, 6(2), 118–127. <https://doi.org/10.31294/ijcit.v6i2.10438>
- Elamim, A., Hartiti, B., Barhdadi, A., Haibaoui, A., Lfakir, A., & Thevenin, P. (2018). Photovoltaic output power forecast using artificial neural networks. *Journal of Theoretical and Applied Information Technology*, 96(15), 5116–5126. <https://centralesupelec.hal.science/hal-02434084>
- Feriawan, A. D., Musta'in, Munit, & Reknadi, D. B. (2024). Klasifikasi layanan pengaduan di smk nu 2 kedungpring lamongan menggunakan metode algoritma k-nn. *JURNAL KHATULISTIWA INFORMATIKA*, 12(2), 98–106. <https://ejournal.bsi.ac.id/ejurnal/index.php/khatulistiwa/article/view/24696>
- Karina, J. M., Anisa, S., & Hamdani. (2022). Studi Komparasi Kwh Meter Pascabayar Dengan Kwh Meter Prabayar Tentang Akurasi Pengukuran Terhadap Tarif Listrik Yang Bervarias. *Jurnal Darma Agung*, 30(1), 488–506. <https://ejournal.darmaagung.ac.id/index.php/jurnaluda/article/view/2230>
- Pasaribu, J. P. K. (2022). Dampak Dari Keterlibatan Menggunakan Listrik Pintar Dari Daya Beli Masyarakat Pada Ekonomi Rumah Tangga. *Management and Sustainable Development Journal*, 4(1). <https://doi.org/10.46229/msdj.v4i1.427>
- Rismayanti, Devayanti, B., & Fadillah, N. (2024). Persepsi masyarakat terhadap efektivitas pembangunan infrastruktur di kota makassar. *EDUSOS: Jurnal Edukasi Dan Ilmu Sosial*, 01(01), 16–20.
- Septria, P., Asnawati, A., & Fredricka, J. (2022). Comparison of the K-Nearest Neighbor Method and the Naive Bayes Method in Classification of Eligibility for Lending. *Jurnal Komputer, Informasi Dan Teknologi*, 2(2), 529–542. <https://doi.org/10.53697/jkomitek.v2i2.952>
- Widaningsih, S., Suheri, A., & Fauziyana, T. T. (2024). Penerapan Data Mining Menggunakan Algoritma K- Nearest Neighbor untuk Penentuan Penerimaan Proposal Hibah. *Jurnal Media Teknik Dan Sistem Industri*, 8(1), 9. <https://doi.org/10.35194/jmtsi.v8i1.2696>