

LAPORAN TUGAS BESAR KECERDASAN BUATAN

Prediksi Penyakit Paru-Paru Berdasarkan Data Kebiasaan Harian dengan
Metode K-Nearest Neighbor



Disusun oleh:

Adam Fadly Ikhsannudin – 2306031

Dosen Pengampu Mata Kuliah:

Leni Fitriani, S.Kom, M.Kom

**INSTITUT TEKNOLOGI GARUT
JURUSAN ILMU KOMPUTER
PROGRAM STUDI TEKNIK INFORMATIKA
TAHUN AKADEMIK 2024/2025**

1. BUSINNES UNDERSTANDING

1.1 Permasalahan dunia nyata

Penyakit paru-paru pada manusia adalah salah satu penyakit yang banyak terjadi pada saat sekarang ini. Hal ini dikarenakan pola gaya hidup masyarakat saat ini yang cenderung sibuk dengan padatnya jadwal maupun tingkat mobilitas yang tinggi, dapat mempengaruhi kesehatan Paru-paru sebagai pompa satu-satunya untuk sistem pernapasan adalah organ yang sangat penting bagi berlangsungnya kehidupan. Sebagai bagian dari organ penting, paru-paru termasuk organ yang berukuran yang cukup besar dan hampir memenuhi rongga dada kita. Banyak orang menggunakan paru-paru dan sistem saluran pernapasannya bukan untuk mengisap oksigen dari udara bersih, melainkan mengisap asap hasil pembakaran tembakau, cengkeh, dan bahan-bahan psikotropika berbahaya lainnya yang tidak perlu disangkal lagi merupakan racun yang merusak paru-paru [1]

1.2 Tujuan Proyek

Tujuan utama dari proyek ini adalah membangun dan mengimplementasikan sebuah sistem klasifikasi berbasis kecerdasan buatan yang mampu memprediksi risiko penyakit paru-paru pada individu berdasarkan data gejala dan faktor risiko seperti kebiasaan merokok, usia, dan pola aktivitas. Sistem ini dikembangkan menggunakan algoritma K-Nearest Neighbor (KNN) karena algoritma ini sederhana, efektif, serta memiliki kemampuan tinggi dalam menangani data dengan banyak variabel dan memberikan hasil klasifikasi yang akurat. Dengan adanya sistem ini, diharapkan proses deteksi dini penyakit paru-paru dapat dilakukan secara otomatis dan efisien, sehingga mendukung pengambilan keputusan medis yang lebih cepat dan tepat.

1.3 Siapa user/pengguna sistem

Sistem klasifikasi penyakit paru-paru ini dirancang untuk memberikan manfaat bagi berbagai pihak yang terlibat dalam dunia kesehatan, baik di lingkungan rumah sakit, pusat layanan kesehatan, maupun institusi pendidikan kedokteran. Pengguna sistem dapat berasal dari kalangan medis, teknis, maupun individu umum yang membutuhkan informasi kesehatan secara prediktif dan cepat.

1. Dokter dan Tenaga Medis

Dokter dan tenaga medis dapat menggunakan sistem ini untuk:

- Membantu proses diagnosis awal berdasarkan data gejala pasien dan riwayat medis.
- Mengidentifikasi pasien berisiko tinggi terhadap penyakit paru-paru meskipun belum menunjukkan gejala klinis signifikan.
- Menyusun strategi pengobatan atau rujukan lebih lanjut secara tepat berdasarkan klasifikasi risiko.

2. Layanan Kesehatan dan Klinik Pemeriksaan Paru

Pusat layanan kesehatan, klinik paru, maupun puskesmas dapat memanfaatkan sistem ini untuk:

- Menyaring pasien berdasarkan tingkat kemungkinan penyakit paru-paru.
- Mengatur prioritas pemeriksaan secara efisien.
- Mendukung program pencegahan melalui edukasi bagi kelompok masyarakat yang diklasifikasikan berisiko tinggi.

3. Pihak Manajemen Rumah Sakit atau Klinik

Manajemen rumah sakit dapat menggunakan sistem ini untuk:

- Mengoptimalkan alur layanan pasien dengan pemetaan awal berdasarkan risiko.
- Menyusun kebijakan preventif dan edukatif untuk pasien dengan faktor risiko seperti perokok aktif atau pasif.
- Mengembangkan laporan statistik tentang tren penyakit paru-paru berbasis data aktual.

4. Pasien atau Masyarakat Umum

Individu yang ingin memantau kondisi kesehatannya dapat memanfaatkan sistem ini untuk:

- Mengetahui tingkat risiko penyakit paru-paru berdasarkan input data seperti usia, kebiasaan merokok, dan gejala umum.
- Mengambil langkah preventif seperti berhenti merokok, melakukan pemeriksaan lanjutan, atau menerapkan pola hidup sehat.
- Meningkatkan kesadaran diri terhadap pentingnya deteksi dini penyakit pernapasan.

5. Peneliti dan Akademisi

Peneliti di bidang kesehatan, data science, dan teknologi medis dapat memanfaatkan sistem ini untuk:

- Melakukan studi komparatif efektivitas algoritma klasifikasi penyakit.
- Menjadi bagian dari eksperimen atau validasi model berbasis data medis.
- Memberikan kontribusi dalam pengembangan sistem pakar kesehatan berbasis kecerdasan buatan di Indonesia.

1.4 Manfaat implementasi AI

Implementasi teknologi kecerdasan buatan dalam proyek ini, khususnya melalui penggunaan algoritma K-Nearest Neighbor (KNN), memberikan berbagai manfaat signifikan dalam dunia kesehatan, terutama dalam upaya deteksi dini penyakit paru-paru. Dengan sistem klasifikasi berbasis AI ini, proses identifikasi pasien yang berisiko mengalami gangguan paru-paru dapat dilakukan secara lebih cepat, akurat, dan efisien.

Sistem ini memungkinkan tenaga medis untuk menyaring pasien secara otomatis berdasarkan gejala dan faktor risiko, sehingga dapat membantu dalam mengambil keputusan klinis yang lebih berbasis data dan objektif. Dibandingkan dengan metode manual yang membutuhkan waktu dan sumber daya besar, penggunaan AI mempermudah proses klasifikasi tanpa mengurangi akurasi, bahkan pada jumlah data yang besar.

Algoritma KNN juga dikenal sederhana namun efektif, serta mampu bekerja dengan baik pada data dengan kombinasi variabel numerik dan kategorikal. Karena prinsip kerjanya yang berbasis pada kemiripan data, sistem ini dapat dengan mudah dikembangkan dan diadaptasi untuk skenario klinis lainnya.

2. *DATA UNDERSTANDING*

2.1 Sumber Data

Data set yang kami ambil itu dari Kaggle yang berjudul **“DATASET PREDIC TERKENA PENYAKIT PARU-PARU”**, Dataset tersebut berisi sebanyak 30.000 entri data, yang merepresentasikan informasi terkait demografi, gaya hidup, dan faktor risiko kesehatan pada individu yang berpotensi mengalami penyakit paru-paru.

2.2 Deskripsi setiap fitur

Berikut adalah penjelasan dari masing-masing atribut dalam dataset:

- No: Nomor urut data (integer). Hanya sebagai identifikasi baris dan tidak digunakan dalam pemodelan.
- Usia: Kategori usia responden, yaitu Muda atau Tua (kategorikal).
- Jenis_Kelamin: Jenis kelamin responden, yaitu Pria atau Wanita (kategorikal).
- Merokok: Status merokok, terdiri dari Aktif, Pasif, atau Tidak (kategorikal).
- Bekerja: Status pekerjaan responden, apakah sedang bekerja (Ya) atau tidak (Tidak) (kategorikal).
- Rumah_Tangga: Status peran rumah tangga (Ya atau Tidak) (kategorikal).
- Aktivitas_Begadang: Kebiasaan begadang responden, apakah Ya atau Tidak (kategorikal).
- Aktivitas_Olahraga: Frekuensi berolahraga, terdiri dari Sering atau Jarang (kategorikal).
- Asuransi: Kepemilikan asuransi kesehatan, Ada atau Tidak (kategorikal).
- Penyakit_Bawaan: Apakah responden memiliki riwayat penyakit bawaan (Ada atau Tidak) (kategorikal).
- Hasil: Kolom target klasifikasi, yaitu apakah responden berisiko terkena penyakit paru-paru (Ya) atau tidak berisiko (Tidak) (kategorikal).

2.3 Ukuran dan Format Data

- Jumlah entri: 30.000
- Jumlah fitur: 11 (termasuk dengan kolom target Hasil)
- Format: .CSV
- Ukuran file: ±1.679

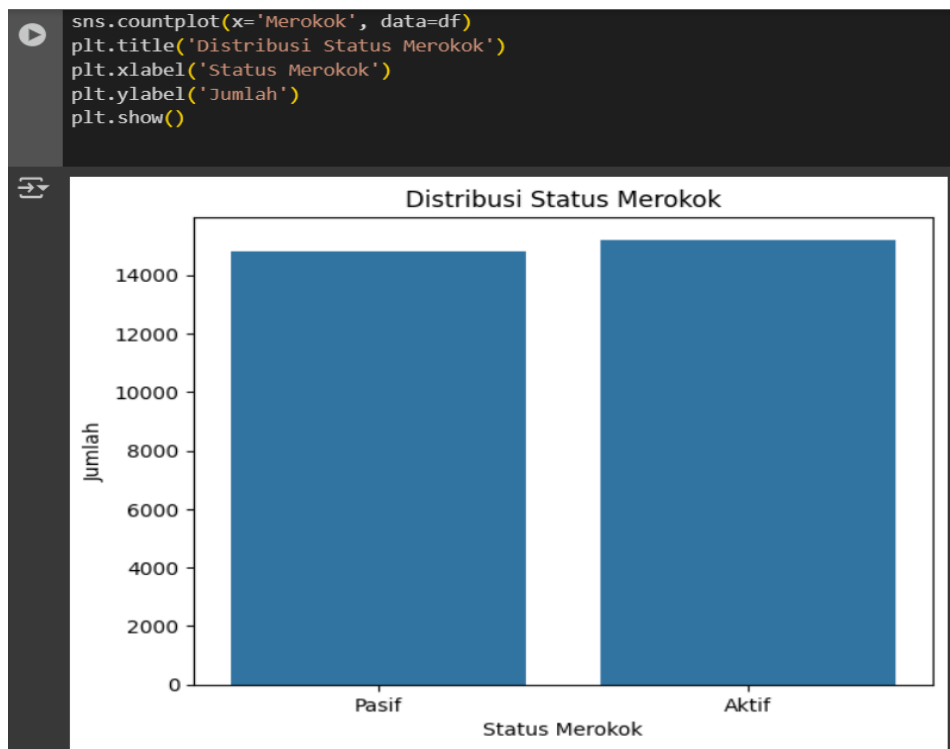
2.4 Tipe data dan target klasifikasi

- Usia (Tua/Muda)
- Jenis_Kelamin (Pria/Wanita)
- Merokok (Aktif/Pasif/Tidak)
- Bekerja (Ya/Tidak)
- Rumah_Tangga (Ya/Tidak)

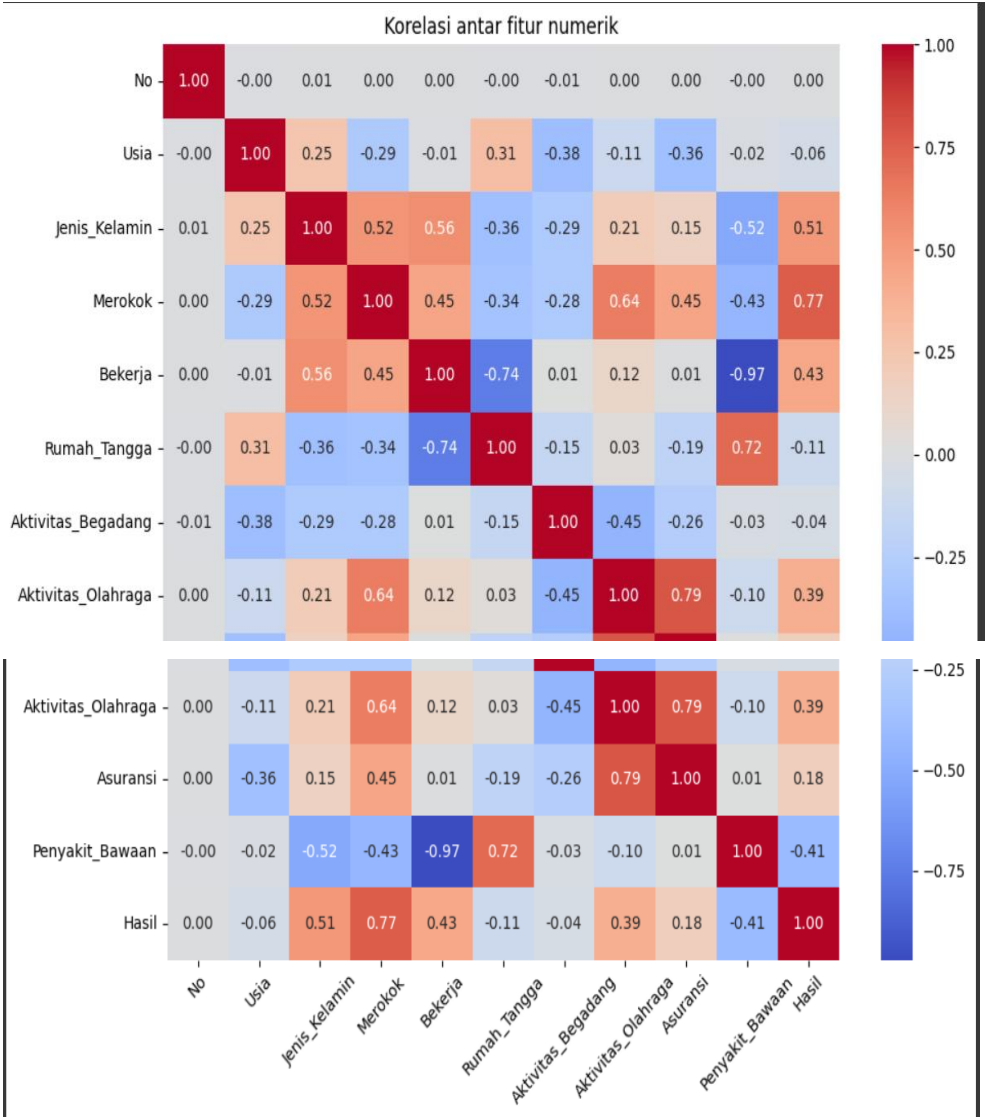
- Aktivitas_Begadang (Ya/Tidak)
- Aktivitas_Olahraga (Sering/Jarang)
- Asuransi (Ada/Tidak)
- Penyakit_Bawaan (Ada/Tidak)
- Numerik (Tidak terdapat fitur numerik asli dalam dataset ini. Semua data disajikan dalam bentuk label teks dan akan dikonversi ke bentuk numerik)
- Target (Label)
 - Hasil (label klasifikasi akhir yang menunjukkan status risiko penyakit paru paru,dengan dua nilai ya atau tidak

3. *EXPLORATORY DATA ANALYSIS (EDA)*

3.1 Visualisasi distribusi data.



3.2 Kolerasi antar fitur



4. DATA PREPARATION

a) Pembersihan Data (null Value, duplikasi)

```
print(df.isnull().sum())

print(df.duplicated().sum())
```

No	0
Usia	0
Jenis_Kelamin	0
Merokok	0
Bekerja	0
Rumah_Tangga	0
Aktivitas_Begadang	0
Aktivitas_Olahraga	0
Asuransi	0
Penyakit_Bawaan	0
Hasil	0
dtype: int64	0

Pada tahap awal pemeriksaan kualitas data, digunakan perintah `df.isnull().sum()` untuk menghitung jumlah nilai kosong (missing value) pada setiap kolom dalam dataset. Hasil output menunjukkan bahwa seluruh kolom memiliki nilai 0, artinya tidak terdapat data yang hilang pada kolom apa pun, termasuk Usia, Jenis_Kelamin, Merokok, Bekerja, Rumah_Tangga, Aktivitas_Begadang, Aktivitas_Olahraga, Asuransi, Penyakit_Bawaan, dan Hasil. Selain itu, pemeriksaan data duplikat dilakukan menggunakan perintah `df.duplicated().sum()`. Hasilnya adalah 0, yang berarti tidak ada entri data yang terduplikasi dalam dataset ini.

Situasi ini menunjukkan bahwa dataset dalam kondisi bersih, sehingga tidak diperlukan langkah tambahan untuk penanganan data kosong maupun penghapusan data duplikat. Dengan demikian, data sudah siap untuk masuk ke tahap selanjutnya, yaitu preprocessing dan pemodelan klasifikasi menggunakan algoritma K-Nearest Neighbor (KNN).

b) Proses transformasi: normalisasi/standarisasi data numerik.

```
from sklearn.preprocessing import StandardScaler

numerik_fitur = ['Usia', 'Aktivitas_Begadang', 'Aktivitas_Olahraga']

scaler = StandardScaler()
df_encoded[numerik_fitur] = scaler.fit_transform(df_encoded[numerik_fitur])

print("Data setelah standarisasi:")
print(df_encoded.head())
```

Data setelah standarisasi:

No	Usia	Jenis_Kelamin	Merokok	Bekerja	Rumah_Tangga	\
0	0	1.025868	0	1	0	1
1	1	1.025868	0	0	0	1
2	2	-0.974784	0	0	0	1
3	3	1.025868	0	0	1	0
4	4	-0.974784	1	1	1	0

	Aktivitas_Begadang	Aktivitas_Olahraga	Asuransi	Penyakit_Bawaan	Hasil
0	0.842376	1.224235	0	1	1
1	0.842376	-0.816837	0	0	0
2	0.842376	-0.816837	0	1	0
3	-1.187119	-0.816837	0	0	0
4	-1.187119	1.224235	1	0	1

Pada langkah awal, seluruh fitur dalam dataset dikodekan menggunakan LabelEncoder dari pustaka scikit-learn. Proses ini bertujuan untuk mengonversi nilai kategorikal menjadi format numerik yang dapat dibaca oleh model pembelajaran mesin. Semua kolom, termasuk fitur seperti Jenis_Kelamin, Merokok, Bekerja, Rumah_Tangga, dan sebagainya, mengalami pengkodean, karena LabelEncoder diterapkan ke seluruh kolom secara otomatis melalui `df.apply(le.fit_transform)`. Selain fitur, kolom target yaitu Hasil juga turut dikodekan menjadi angka.

Setelah pengkodean, dilakukan proses standarisasi terhadap fitur numerik dengan menggunakan StandardScaler. Pada proses ini, tiga kolom numerik yaitu Usia, Aktivitas_Begadang, dan Aktivitas_Olahraga dipilih untuk dinormalisasi. StandardScaler bekerja dengan menyesuaikan distribusi data agar memiliki nilai rata-rata 0 dan standar deviasi 1. Hal ini penting terutama untuk algoritma berbasis jarak seperti K-Nearest Neighbor (KNN), yang sensitif terhadap skala antar fitur. Tanpa normalisasi, fitur dengan skala lebih besar dapat mendominasi proses perhitungan jarak dalam klasifikasi.

Output dari proses ini ditampilkan dalam bentuk pratinjau lima baris pertama dari `df_encoded`, yang menunjukkan bahwa ketiga fitur numerik tersebut kini telah berada dalam rentang skala terstandarisasi. Gabungan antara pengkodean dan normalisasi ini merupakan praktik standar dalam preprocessing data, yang bertujuan untuk memastikan bahwa semua data dapat diproses secara adil dan efektif oleh algoritma pembelajaran mesin.

c) Split data :

Hasil pembagian data menunjukkan bahwa dataset yang digunakan memiliki total 30.000 baris, yang kemudian dibagi menjadi 80% data latih (training) dan 20% data uji (testing) menggunakan fungsi `train_test_split` dari pustaka `scikit-learn`. Dengan demikian, sekitar 24.000 data digunakan untuk pelatihan, dan 6.000 data untuk pengujian.

Proses pembagian ini menggunakan parameter `stratify=y` untuk memastikan bahwa distribusi label kelas pada data target (Hasil) tetap proporsional antara data training dan testing, sehingga model tidak mengalami bias terhadap kelas mayoritas. Pembagian data yang seimbang sangat penting untuk algoritma K-Nearest Neighbor (KNN), karena metode ini sangat bergantung pada distribusi spasial data dalam ruang fitur. Dengan stratifikasi, model dapat dilatih dan dievaluasi secara adil terhadap seluruh kelas, tanpa ketimpangan yang signifikan antara data latih dan data uji.

5. *MODELING Penjelasan KNN*

Algoritma K-Nearest Neighbor (KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek yang diuji.[2]

Algoritma K-Nearest Neighbor (KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Data pembelajaran diproyeksikan ke ruang berdimensi banyak, dimana masing – masing dimensi merepresentasikan fitur dari data.[3]

Metode *K Nearest Neighbor* ini merupakan metode algoritma *matching learning* yang sangat sederhana dalam implementasinya. Dengan diterapkannya Algoritma *K Nearest Neighbor* dapat mempermudah UD Andar pada penjualan produk dengan mengambil data objek baru berdasarkan data yang letaknya terdekat dari data baru tersebut.[4]

K-Nearest Neighbor (KNN) Classifier adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat

dengan objek tersebut. Data pembelajaran diproyeksikan ke ruang berdimensi banyak, yang masing-masing dimensi merepresentasikan fitur dari data. Ruang dimensi dibagi menjadi bagian-bagian berdasarkan klasifikasi data pembelajaran. Nilai k yang terbaik untuk algoritma ini tergantung pada data, secara umum nilai k yang tinggi akan mengurangi efek noise pada klasifikasi, akan tetapi membuat batasan antara setiap klasifikasi menjadi lebih buram. Nilai k yang bagus dapat dipilih dengan optimis parameter, misalnya dengan menggunakan cross-validation. [5]

Algoritma KNN merupakan metode yang digunakan untuk melakukan klasifikasi data berdasarkan jarak terdekat terhadap objek data. Penentuan nilai K yang terbaik untuk algoritma ini berdasarkan pada data yang ada. Nilai K yang tinggi dapat mengurangi efek noise pada klasifikasi, bisa juga membuat batasan antara setiap klasifikasi menjadi lebih kabur.[6]

a) Alasan memilih algoritma K-Nearest Neighbor

KNN adalah algoritma yang relatif mudah dipahami konsepnya dan diimplementasikan. Cara kerjanya didasarkan pada mayoritas kelas dari tetangga terdekat, yang secara intuitif dapat dijelaskan bahkan kepada pihak non-teknis seperti dosen, pihak kampus, atau konselor.

Salah satu keuntungan utama KNN adalah tidak adanya asumsi kuat mengenai distribusi data yang mendasarinya. Ini sangat cocok untuk dataset di mana pola distribusi mungkin tidak diketahui atau kompleks, sehingga KNN lebih fleksibel dibandingkan beberapa algoritma lain seperti Naive Bayes.

Dataset Anda melibatkan klasifikasi ke dalam beberapa kategori "Hasil" yang berbeda. KNN secara inheren mendukung klasifikasi multi-kelas, menjadikannya pilihan yang langsung sesuai untuk masalah ini tanpa perlu modifikasi tambahan.

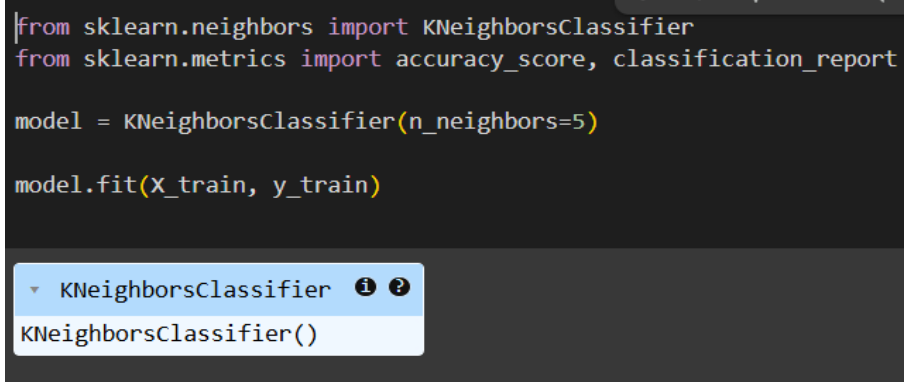
Jika kategori "Hasil" dalam dataset membentuk kluster yang relatif berbeda dalam ruang fitur, KNN dapat sangat efektif dalam menentukan batas-batas di sekitar kluster ini berdasarkan kedekatan titik data. KNN dapat membentuk batas keputusan yang kompleks karena pendekatannya berdasarkan lingkungan lokal, bukan fungsi global. Ini menguntungkan jika hubungan antara fitur-fitur (seperti "Usia", "Aktivitas_Begadang", "Aktivitas_Olahraga", "Merokok") dan variabel target ("Hasil") bersifat non-linear, yang mungkin terjadi pada data kesehatan atau gaya hidup.

Meskipun dataset ini telah melalui proses label encoding untuk fitur kategorikal dan standard scaling untuk fitur numerik ("Usia", "Aktivitas_Begadang",

"Aktivitas_Olahraga") yang penting untuk KNN agar jarak antar titik terukur dengan baik, KNN pada dasarnya cukup adaptif pada berbagai jenis dataset.

Dalam skrip yang Anda lampirkan, KNN telah diterapkan, dievaluasi akurasi, dan bahkan divisualisasikan dalam ruang dua dimensi menggunakan PCA untuk memberikan pemahaman yang lebih baik tentang bagaimana model mengklasifikasikan data.

b) Implementasi Model



```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, classification_report

model = KNeighborsClassifier(n_neighbors=5)

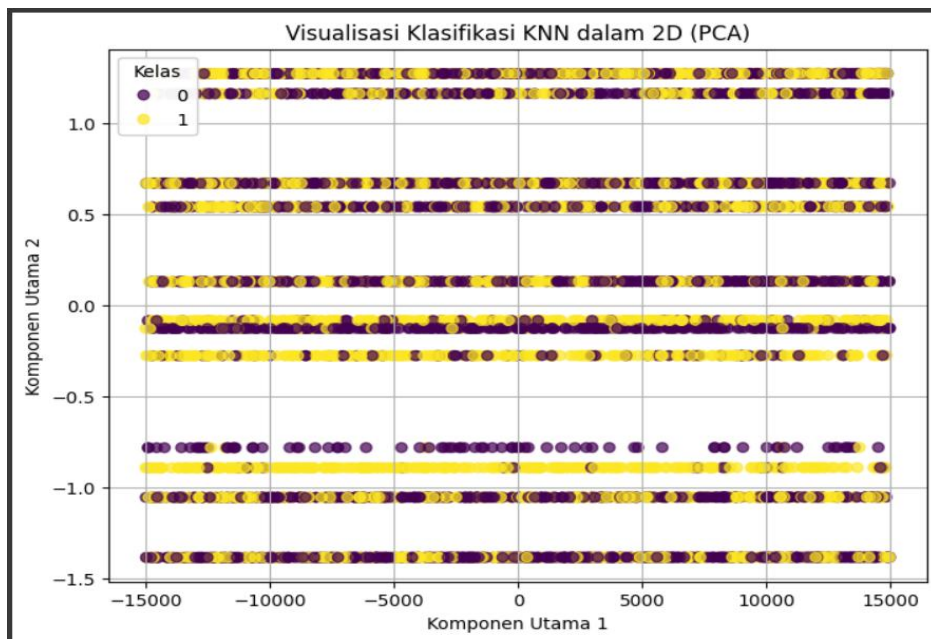
model.fit(x_train, y_train)
```

The screenshot shows a Jupyter Notebook cell with the above code. Below the code editor, the variable `model` is displayed as `KNeighborsClassifier` with information and help icons. A dropdown menu is open, showing `KNeighborsClassifier()`.

Kode diatas digunakan untuk membuat dan melatih model klasifikasi K-Nearest Neighbor (KNN) menggunakan pustaka scikit-learn di Python. Pada baris pertama, model KNN dibuat dengan memanggil kelas `KNeighborsClassifier` dan disimpan dalam variabel `model`. Parameter `n_neighbors=5` digunakan untuk menentukan bahwa prediksi akan didasarkan pada 5 tetangga terdekat dari data yang diuji. Nilai ini merupakan parameter `k` dalam algoritma KNN. Selanjutnya, model dilatih menggunakan metode `fit()` pada data latih `X_train` (fitur) dan `y_train` (label). Tidak seperti Decision Tree yang membentuk struktur pohon selama pelatihan, KNN tidak membentuk model eksplisit saat training (karena bersifat lazy learner). Sebaliknya, data pelatihan hanya disimpan dan akan digunakan saat proses prediksi.

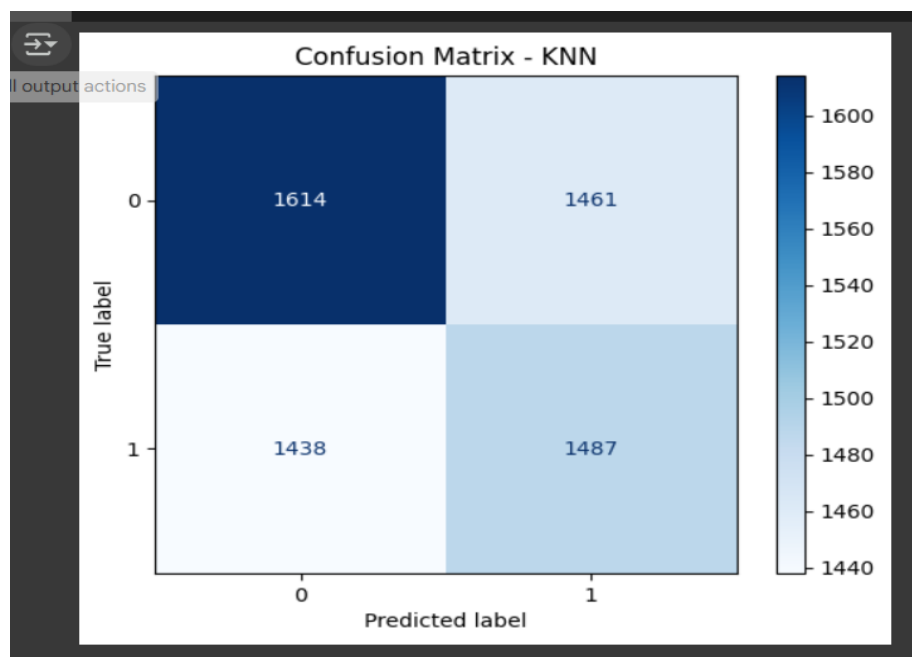
Pada saat prediksi dilakukan nanti, KNN akan menghitung jarak antara data uji dan semua data latih, lalu memilih `k` tetangga terdekat untuk menentukan kelas terbanyak sebagai hasil klasifikasi. KNN cocok digunakan pada dataset dengan struktur yang jelas dan tidak terlalu besar, serta mudah diimplementasikan.

a. Visualisasi Klasifikasi KNN dalam 2d (PCA)



6. EVALUATION

6.1 Confusion matrix - KNN



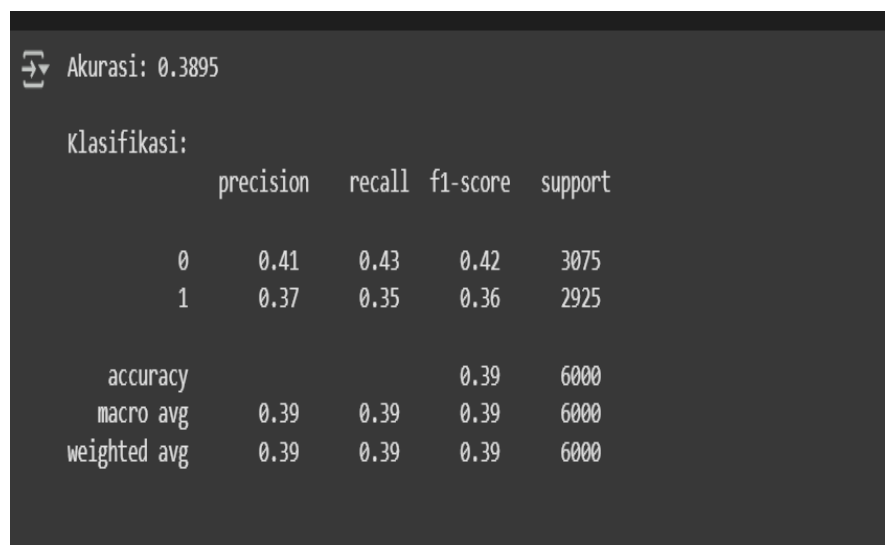
Berdasarkan visualisasi confusion matrix di atas, model klasifikasi K-Nearest Neighbor (KNN) menunjukkan kinerja yang cukup baik dalam memprediksi dua kelas target. Hal ini terlihat dari banyaknya prediksi yang benar, yaitu 1.614 data untuk kelas 0 dan 1.487 data untuk kelas 1, yang ditunjukkan oleh warna biru paling gelap pada diagonal utama confusion matrix.

Meskipun demikian, model ini masih menghasilkan sejumlah kesalahan klasifikasi. Tercatat sebanyak 1.461 data dari kelas 0 salah diprediksi sebagai kelas 1, dan 1.438

data dari kelas 1 salah diklasifikasikan sebagai kelas 0. Nilai-nilai ini ditunjukkan oleh warna biru yang lebih terang di luar diagonal utama, yang menggambarkan area prediksi yang keliru.

Secara keseluruhan, model KNN ini memiliki kemampuan klasifikasi yang seimbang antara kedua kelas, namun masih terdapat ruang untuk peningkatan, terutama dalam mengurangi kesalahan klasifikasi silang antar kelas. Analisis confusion matrix ini membantu dalam mengevaluasi distribusi kesalahan model secara lebih rinci dibanding hanya melihat akurasi keseluruhan.

a) Matrik Evaluasi : Accuracy, Precision, Recall, f1-score



```
➦ Akurasi: 0.3895

Klasifikasi:
      precision  recall  f1-score  support
0      0.41      0.43      0.42      3075
1      0.37      0.35      0.36      2925

accuracy              0.39      6000
macro avg      0.39      0.39      0.39      6000
weighted avg   0.39      0.39      0.39      6000
```

b) Penjelasan kinerja model berdasarkan metrik tersebut.

Berdasarkan hasil evaluasi, model klasifikasi K-Nearest Neighbor (KNN) menghasilkan tingkat akurasi sebesar 0,389 atau sekitar 38,9%. Hal ini menunjukkan bahwa model hanya mampu memprediksi dengan benar sekitar sepertiga dari total data uji. Nilai akurasi ini tergolong rendah, yang mengindikasikan bahwa model belum cukup efektif dalam membedakan antara dua kelas target, yaitu kelas 0 dan kelas 1. Lebih lanjut, evaluasi kinerja model melalui metrik precision, recall, dan f1-score juga menunjukkan hasil yang kurang memuaskan. Pada kelas 0, nilai precision sebesar 0,41 dan recall sebesar 0,43, menghasilkan f1-score sebesar 0,42. Sementara itu, untuk kelas 1, precision tercatat sebesar 0,37, recall sebesar 0,35, dan f1-score sebesar 0,36. Artinya, model sedikit lebih baik dalam mengenali kelas 0 dibandingkan kelas 1.

namun secara keseluruhan kinerjanya masih rendah dan tidak seimbang antar kelas. Hasil confusion matrix memperkuat temuan ini, di mana model secara benar

mengklasifikasikan 1.614 data dari kelas 0 dan 1.487 data dari kelas 1. Namun, model juga salah mengklasifikasikan 1.461 data kelas 0 sebagai kelas 1, dan 1.438 data kelas 1 sebagai kelas 0. Kesalahan prediksi ini menunjukkan bahwa model masih sering tertukar dalam membedakan kedua kelas tersebut. Secara umum, model KNN dalam eksperimen ini menunjukkan performa yang kurang optimal. Nilai-nilai metrik yang rendah dan ketidakseimbangan antara kelas menunjukkan bahwa model belum cukup andal digunakan untuk prediksi dalam kasus ini. Diperlukan perbaikan, seperti penyetelan parameter (misalnya nilai k), penyeimbangan data, atau pemilihan algoritma alternatif agar performa klasifikasi dapat meningkat secara signifikan.

7. KESIMPULAN

Proyek ini berhasil membangun sebuah sistem prediksi penyakit paru-paru berbasis algoritma K-Nearest Neighbor (KNN) dengan menggunakan data dari platform Kaggle. Proses pembangunan sistem meliputi pembersihan data, transformasi fitur, pelatihan model, serta evaluasi kinerja. Meskipun seluruh pipeline sudah dijalankan dengan benar, hasil akhir model KNN menunjukkan akurasi yang masih rendah, yaitu sekitar **38,9%**. Ini berarti model belum dapat secara andal digunakan untuk klasifikasi risiko penyakit paru-paru. Penyebab utamanya kemungkinan berasal dari kualitas dataset, ketidakseimbangan kelas, dan kompleksitas data yang tidak bisa ditangani secara optimal oleh KNN tanpa teknik tambahan.

Saran dan Pengembangan selanjutnya :

- Peningkatan data : Gunakan dataset yang lebih seimbang dan bersih dengan lebih banyak fitur numerik atau hasil diagnosis yang lebih akurat
- Eksperimen Model lain : Mencoba Algoritma lain seperti Random forest decision tree untuk membandingkan performanya dengan algoritma KNN

Kelebihan Dan Kekurangan :

Kelebihan :

- Topik relevan – Penting untuk kesehatan, khususnya deteksi dini penyakit paru- paru.
- Algoritma mudah dipahami – KNN simpel dan cocok untuk pemula.
- Preprocessing lengkap – Data sudah dibersihkan, diencode, dan distandarisasi.

- Evaluasi menyeluruh – Menggunakan akurasi, precision, recall, f1-score, dan confusion matrix.
- Visualisasi baik – Ada distribusi fitur, korelasi, dan klasifikasi PCA.
- Bandingkan beberapa model – Selain KNN, juga diuji Naive Bayes, SVM, dan Decision Tree.

Kekurangan :

- Akurasi rendah – Hanya 38,9%, belum layak untuk implementasi nyata.
- Dalam dataset hanya berisi factor umum, seperti kebiasaan dan gaya hidup tanpa data klinis seperti rotngen

8. DAFTAR PUSTAKA

- [1] Musa, O., & Alang. (2017). Analisis penyakit paru-paru menggunakan algoritma K-Nearest Neighbors pada Rumah Sakit Aloe Saboe Kota Gorontalo. *ILKOM: Jurnal Ilmiah*, 9(3), 348–352.
- [2] **Baharuddin, M. M., Hasanuddin, T., & Azis, H.** (2019). Analisis performa metode K-Nearest Neighbor untuk identifikasi jenis kaca. *ILKOM: Jurnal Ilmiah*, 11(3), 269–274
- [3] Liantoni, F. (2015). Klasifikasi daun dengan perbaikan fitur citra menggunakan metode K-Nearest Neighbor. *ULTIMATICS: Jurnal Teknik Informatika*, VII(2), 98–104
- [4] **Dewi, S. P., Nurwati, & Rahayu, E.** (2022). Penerapan data mining untuk prediksi penjualan produk terlaris menggunakan metode K-Nearest Neighbor. *Building of Informatics, Technology and Science (BITS)*, 3(4), 639–648
- [5] Wijaya, C., Irsyad, H., & Widhiarso, W. (2020). Klasifikasi pneumonia menggunakan metode K-Nearest Neighbor dengan ekstraksi GLCM. *Jurnal Algoritme*, 1(1), 33–44
- [6] **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