

Implementation of the K-Nearest Neighbors Algorithm for Comprehensive Heart Attack Analysis and Prediction



DISUSUN OLEH:

KELOMPOK 02:

11321007 Angelica Manurung

11321011 Juan Munthe

11321021 Luana Breka Banjarnahor

11321031 Albert Aritonang

PROGRAM STUDI TEKNOLOGI INFORMASI

FAKULTAS VOKASI

2022/2023

DAFTAR ISI

DAFTAR ISI.....	2
DAFTAR GAMBAR.....	3
BAB I PENDAHULUAN.....	4
1.1 Latar Belakang	4
BAB II DASAR TEORI	5
2.1 Heart Attack	5
2.2 K-Nearest Neighbors.....	5
2.3 Confusion matrix.....	6
2.4 Flask Framework.....	7
BAB III ANALISIS DAN DESAIN.....	8
3.1 Pengumpulan Data	8
3.2 Preprocessing Data.....	9
3.2.1. Data Cleaning	9
3.2.2. Data Splitting.....	10
3.3 Modelling	11
BAB IV ARSITEKTUR dan IMPLEMENTASI.....	14
4.1 Arsitektur model.....	14
4.2 Deployment Model.....	15
4.3 Evaluasi Model.....	15
REFERENSI	18

DAFTAR GAMBAR

Gambar 1. Data Cleaning.....	10
Gambar 2. Data Splitting	11
Gambar 3. Code penemuan nilai K.....	12
Gambar 4. Hasil pencarian nilai tetangga terbaik	12
Gambar 5. Pencarian nilai K optimal untuk KNN.....	13
Gambar 6. Hasil Error Rate	13
Gambar 7. Arsitektur Model	14
Gambar 8. Tampilan Form Pengisian Kesehatan	16
Gambar 9. Hasil Prediksi	16
Gambar 10. Evaluasi Model	17

BAB I

PENDAHULUAN

1.1 Latar Belakang

Serangan jantung juga dikenal sebagai *infark miokard*, terjadi ketika aliran darah ke bagian jantung terhalang, biasanya oleh bekuan darah. Hal ini menyebabkan kerusakan pada otot jantung akibat kekurangan oksigen dan nutrisi [1]. Seiring meningkatnya serangan jantung (HA) dan kondisi terkaitnya, kekhawatiran akan kesehatan semakin tumbuh di kalangan banyak orang. Data medis menegaskan bahwa serangan jantung dan penyakit jantung (HD) menjadi penyebab utama kematian global sebelum pandemi Covid-19 mengambil alih perhatian dunia, termasuk Indonesia. [2]. Menurut *Global Burden of Disease dan Institute for Health Metrics and Evaluation (IHME)* dari 2014 hingga 2019, penyakit jantung merupakan penyebab utama kematian di Indonesia. Data Riset Kesehatan Dasar (Riskesdas) pada tahun 2013 dan 2018 menunjukkan bahwa kasus penyakit jantung meningkat dari 0,5% pada 2013 menjadi 1,5% pada 2018. Penyakit jantung juga membebani secara finansial, terbukti dari data BPJS Kesehatan tahun 2021 yang menunjukkan bahwa biaya kesehatan terbesar terkait dengan penyakit jantung, mencapai Rp. 7,7 triliun [3].

Dalam upaya untuk menghadapi kompleksitas masalah kesehatan terkait serangan jantung, pengembangan sistem yang dapat menganalisis dan memprediksi risiko serangan jantung menjadi semakin penting. Dengan mempertimbangkan informasi medis yang menunjukkan peningkatan kasus serangan jantung dan beban finansial yang signifikan akibat kondisi ini, perlunya solusi yang dapat memberikan pemahaman lebih dalam akan faktor risiko serta potensi serangan jantung menjadi sangat mendesak. Dengan memanfaatkan teknologi dan data medis yang tersedia, sistem analisis dan prediksi dapat menjadi alat yang berharga dalam mendukung upaya pencegahan dan pengelolaan serangan jantung secara lebih efektif.

BAB II

DASAR TEORI

2.1 Heart Attack

Jantung, yang terletak dalam rongga dada, merupakan salah satu organ utama dalam tubuh manusia yang memiliki empat ruang: dua atrium dan dua ventrikel, dengan ventrikel berada di bawah atrium. Peran penting katup-katup pada jantung adalah mencegah pencampuran darah. Jantung merupakan bagian integral dari sistem kardiovaskular, seperti sirkulasi paru-paru, namun rentan terhadap berbagai penyakit [4]. Serangan jantung, atau infark miokard akut (IMA), terjadi saat aliran darah *arteri coroner* terhenti, menyebabkan kekurangan oksigen pada otot jantung. Kondisi ini merupakan keadaan darurat yang memerlukan penanganan cepat guna meminimalkan kerusakan jantung yang mungkin terjadi [5]. Faktor risiko terkena serangan jantung di antaranya usia, jenis kelamin, merokok, kolesterol tinggi, diabetes, obesitas, riwayat keluarga, stres, dan kurang aktivitas fisik. Beberapa faktor yang meningkatkan risiko serangan jantung termasuk usia, jenis kelamin, kebiasaan merokok, kadar kolesterol yang tinggi, diabetes, kelebihan berat badan, riwayat keluarga, tingkat stres, dan kurangnya aktivitas fisik [2].

2.2 K-Nearest Neighbors

K-Nearest Neighbors merupakan metode klasifikasi terhadap objek baru berdasarkan data training yang memiliki jarak tetangga terdekat (*nearest neighbor*) dengan objek baru tersebut. Dekat atau jauhnya neighbor biasanya dihitung berdasarkan jarak *Euclidean* [6]. Tujuan dari Algoritma KNN untuk mengklasifikasi objek baru berdasarkan atribut, dimana hasil dari sampel uji yang baru diklasifikasikan berdasarkan mayoritas dari kategori K-Nearest Neighbors. Algoritma K-NN menggunakan klasifikasi ketetanggaan sebagai nilai prediksi dari sampel uji yang baru [7]. Perhitungan jarak pada KNN menggunakan rumus Euclidean Distance dapat dicari menggunakan persamaan 1 yaitu [8]:

$$D_{xy} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad \dots(1)$$

Dimana:

D = Jarak *Euclidean* yang akan dicari

x_i = Data training

y_i = Data testing

n = Jumlah data

2.3 Confusion matrix

Pengamatan dan analisis hasil dari algoritma merupakan langkah evaluasi yang penting untuk memastikan keakuratan pengujian. Evaluasi bertujuan untuk menentukan hasil uji yang terbaik dengan fokus pada tingkat ketepatan antara prediksi dan nilai aktual. Pengukuran akurasi model menggunakan *Confusion Matrix* sebagai metode evaluasi utama. *Confusion matrix* adalah alat evaluasi model yang menampilkan informasi tentang perbandingan antara prediksi dan realitas. Matrix ini terdiri dari empat kombinasi nilai prediksi dan aktual yang digambarkan sebagai matriks. Hal ini dapat dilihat pada tabel 1 [9].

Table 1. Confusion Matrix

(Sumber: A. Suresh)

Classification	Group	
	+	-
+	<i>True Positives</i>	<i>False Positives</i>
-	<i>False Negatives</i>	<i>True Negatives</i>

Keterangan:

- *True Positives* (TP): Nilai target aktual dan nilai target yang diprediksi sama-sama positif.
- *True Negatives* (TN): Nilai target aktual dan nilai target yang diprediksi sama-sama negatif.
- *False Positives* (FP): Nilai target aktual negatif, tetapi nilai target yang diprediksi positif.
Juga dikenal sebagai kesalahan Tipe 1
- *False Negatives* (FN): Nilai target aktual positif, tetapi nilai target yang diprediksi negatif.
Juga dikenal sebagai kesalahan Tipe 2

Pada *confusion matrix* evaluasi performa melibatkan analisis akurasi, presisi, *recall*, dan *f-measure* berdasarkan parameter nilai K yang telah digunakan. Secara umum, model yang baik adalah model yang memiliki nilai akurasi, *recall*, precision, dan f-score yang tinggi.

1. Akurasi (Accuracy): Rasio antara jumlah total prediksi yang benar dengan jumlah total prediksi. Rumus untuk menghitung nilai akurasi dapat dilihat pada persamaan 2 [9].

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

2. Recall (Recall): Rasio antara jumlah nilai target positif yang diprediksi benar dengan jumlah total nilai target positif. Rumus untuk menghitung nilai recall dapat dilihat pada persamaan 3 [9].

$$Recall = \frac{TP}{(TP + FN)} \quad \dots(3)$$

3. Presisi (Precision): Rasio antara jumlah nilai target positif yang diprediksi benar dengan jumlah total nilai target yang diprediksi positif. Rumus untuk menghitung nilai presisi dapat dilihat pada persamaan 4 [9].

$$Akurasi = \frac{TP}{(TP + FP)} \quad \dots(4)$$

4. F-measure / F1-Score: Kombinasi antara recall dan precision. Rumus untuk menghitung nilai f-measure dapat dilihat pada persamaan 5 [9].

$$F1 - Score = 2 * \frac{(\text{recall} * \text{precision})}{(\text{recall} + \text{precision})} \quad \dots(5)$$

2.4 Flask Framework

Flask adalah kerangka kerja aplikasi web yang ringan dan fleksibel untuk Python. Ini menyediakan alat, perpustakaan, dan teknologi untuk membangun aplikasi web. Flask dikenal karena kesederhanaannya dan kemudahan penggunaannya, menjadikannya pilihan populer untuk mengembangkan aplikasi web, termasuk sistem manajemen basis data. Ini memungkinkan pengembang untuk membuat aplikasi web dengan cepat dan efisien, dan cocok untuk membangun aplikasi berukuran kecil hingga menengah. Flask juga memiliki komunitas yang kuat dan dokumentasi yang luas, sehingga memudahkan pengembang untuk menemukan dukungan dan sumber daya [10].

BAB III

ANALISIS DAN DESAIN

3.1 Pengumpulan Data

Pengumpulan data untuk prediksi serangan jantung merupakan tahap krusial dalam mengembangkan sistem prediksi serangan jantung menggunakan *K-Nearest Neighbors*. Penelitian ini memanfaatkan dataset prediksi serangan jantung yang diperoleh dari platform *Kaggle*. *Kaggle*, sebagai platform terkemuka dalam bidang data, memberikan akses ke beragam dataset yang relevan dalam ranah medis dan memiliki label yang dapat dipercaya. Pengambilan dataset ini kami ambil dari *Kaggle* dengan link sebagai berikut: [Heart Attack Analysis & Prediction Dataset \(kaggle.com\)](https://www.kaggle.com/datasets/heart-attack-analysis/prediction-dataset). Data-data ini akan menjadi landasan untuk melakukan prediksi potensial terkait risiko serangan jantung. Dataset ini berisi faktor-faktor yang berpotensi mempengaruhi risiko serangan jantung pada pasien yang terdiri dari 304 data dengan 14 atribut yaitu:

1. age: Usia pasien
2. sex: Jenis kelamin pasien (0 = perempuan, 1 = laki — laki)
3. cp: Chest Pain type/tipe nyeri dada terdapat 4 value yaitu:
 - Value 1: typical angina
 - Value 2: atypical angina
 - Value 3: non-anginal pain
 - Value 4: asymptomatic
4. trtbps: resting blood pressure/ tekanan darah (dalam mm Hg)
5. chol: Kolesterol dalam mg/dl yang diukur melalui sensor BMI.
6. fbs: (fasting blood sugar > 120 mg/dl) Jika gula darah saat puasa lebih dari 120 mg/dl maka nilainya 1 (true), jika tidak 0 (false)
7. restecg: hasil elektrokardiografi saat istirahat (nilai 0,1,2)
 - Value 0: normal,
 - Value 1: memiliki kelainan gelombang ST-T (inversi gelombang T dan / atau elevasi atau depresi ST> 0,05 mV),
 - Value 2: menunjukkan kemungkinan atau pasti hipertrofi ventrikel kiri menurut kriteria Estes.
8. thalachh: denyut jantung maksimum yang dicapai.

9. exng: exercise induced angina/nyeri dada (1=iya, 0, tidak).
10. oldpeak: Previous peak
11. slp: Slope
12. caa: number of major vessels/ Penyempitan pembuluh darah (0-3)
13. thall: Thal rate/ jenis thalasemia yang dimiliki pasien
14. output: kemungkinan kecil terkena serangan jantung (0), kemungkinan besar terkena serangan jantung (1).

3.2 Preprocessing Data

Setelah mendapatkan data prediksi serangan jantung, langkah selanjutnya adalah melakukan tahap preprocessing data. Preprocessing data menjadi tahap krusial dalam persiapan data untuk model prediksi penyakit jantung menggunakan algoritma K-Nearest Neighbors (KNN). Mengingat data medis seringkali tidak sempurna dengan kemungkinan kehilangan nilai, ketidakkonsistenan, dan kebisingan, langkah-langkah seperti pembersihan data (data cleaning) dan pembagian data (data splitting) sangat penting dalam mempersiapkan dataset heart.csv.

3.2.1. Data Cleaning

Data cleaning adalah tahapan yang krusial dalam preprocessing data yang bertujuan untuk menyaring data dari berbagai masalah umum seperti nilai yang hilang, ketidakkonsistenan, kebisingan, atau kesalahan dalam dataset. Dalam penelitian ini, tujuan utama dari data cleaning adalah memastikan bahwa data yang akan digunakan untuk analisis atau pembuatan model memiliki kualitas yang prima. Hal ini memastikan bahwa data tidak terpengaruh oleh kecacatan dan siap untuk digunakan oleh algoritma pembelajaran mesin seperti K-Nearest Neighbors (K-NN) agar dapat memberikan prediksi yang akurat dan dapat diandalkan.

Adanya data duplikat dalam dataset heart.csv adalah situasi umum yang bisa mengganggu kualitas analisis dan keakuratan hasil yang diinginkan. Langkah penting dalam menyiapkan data adalah melakukan proses data cleaning untuk mengenali dan menghapus entri yang memiliki nilai yang sama di seluruh kolom atau pada bagian tertentu. Membersihkan dataset dari data duplikat akan menghasilkan dataset yang lebih terpercaya, konsisten, dan siap digunakan untuk analisis atau pemodelan lebih lanjut. Ini merupakan langkah krusial yang memastikan integritas data dan akurasi analisis yang akan dilakukan. Penghapusan data duplikat dapat dilihat pada gambar 1.

```
dups_data = df_heart.duplicated()
print("Jumlah baris yang ada duplikat: %d" % dups_data.sum())

Jumlah baris yang ada duplikat: 1

duplicate_rows = df_heart[df_heart.duplicated()]
print("Baris data yang merupakan duplikat:")
print(duplicate_rows)

Baris data yang merupakan duplikat:
   age  sex  cp  trtbps  chol  fbs  restecg  thalachh  exng  oldpeak  slp  \
164   38    1   2    138   175     0         1     173     0       0.0    2

   caa  thall  output
164    4      2       1

#drop baris yang ada duplikat
print('Jumlah baris sebelum membuang duplikat = %d' % df_heart.shape[0])
data_heart = df_heart.drop_duplicates()
print('Jumlah baris setelah membuang duplikat = %d' % data_heart.shape[0])

Jumlah baris sebelum membuang duplikat = 303
Jumlah baris setelah membuang duplikat = 302
```

Gambar 1. Data Cleaning

3.2.2. Data Splitting

Data splitting adalah proses membagi dataset menjadi dua subset yang berbeda: satu subset digunakan untuk melatih model KNN (training set), sedangkan subset lainnya digunakan untuk menguji kinerja model atau mengevaluasi kinerja model pada data yang tidak terlibat dalam proses pelatihan (test set). Pembagian yang dilakukan secara proporsional dan acak pada dataset jantung memastikan bahwa model dapat mempelajari sebagian data dengan baik dan diujikan pada data yang independen. Tujuannya adalah untuk mengukur seberapa baik model yang telah dilatih mampu melakukan prediksi pada data baru yang tidak pernah diakses sebelumnya. Gambar 2 merupakan pembagian data untuk penelitian. Pada penelitian ini, Data yang telah dibersihkan kemudian dibagi menjadi dua set, yaitu set pelatihan (80%) dan set pengujian (20%).

```

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

# Misalkan df_scaled adalah data yang telah Anda skalakan sebelumnya

# Bagi data menjadi data latih dan data uji
X_train, X_test, y_train, y_test = train_test_split(df_scaled, data_heart['output'], test_size=0.20, random_state=45)

# Inisialisasi objek StandardScaler
std_scaler = StandardScaler()

# Skalakan data latih
X_train = std_scaler.fit_transform(X_train)

# Skalakan data uji dengan menggunakan scaler yang sama dari data latih
X_test = std_scaler.transform(X_test)

print(X_train.shape)
print(X_test.shape)

```

(241, 13)
(61, 13)

Gambar 2. Data Splitting

3.3 Modelling

Modeling dalam penelitian ini mencakup pembangunan sebuah model komputasi yang memanfaatkan algoritma K-Nearest Neighbors (K-NN) untuk menganalisis data terkait serangan jantung. Model ini difokuskan pada prediksi serangan jantung dengan memanfaatkan informasi dari rekam medis pasien yang telah mengalami serangan jantung sebelumnya atau memiliki faktor risiko tertentu yang terkait dengan kondisi tersebut. Ada 2 tahap dalam penerapan dan pembuatan model menggunakan algoritma K-Nearest Neighbors (K-NN):

1. Pemilihan nilai K

Pada penelitian ini, nilai K yang merupakan jumlah tetangga terdekat yang digunakan dalam model K-NN dipilih dengan dua metode. Pertama, menggunakan metode **“RandomizedSearchCV”** yang melakukan pencarian acak dari 1 hingga 50 dengan 20 iterasi dan validasi silang sebanyak 5 kali untuk mencari nilai optimal dari parameter `n_neighbors`. Hasil yang didapat yaitu 6. Selanjutnya, dengan menggunakan fungsi **“find_k_values”** yang memplot error rate terhadap nilai K dari 1 hingga 40, penelitian ini berhasil menemukan nilai K baru yang optimal, yaitu $K = 35$, sebagai jumlah tetangga terdekat yang digunakan untuk melakukan prediksi dalam model K-NN.

Cari nilai `n_neighbors` terbaik

```
from sklearn.model_selection import RandomizedSearchCV

knn = KNeighborsClassifier()
parameter_knn = {'n_neighbors': np.arange(1, 50)}

# Mendefinisikan jumlah iterasi (n_iter) untuk pencarian acak
n_iter = 20

# Mencari k-values terbaik dengan RandomizedSearchCV
knn_rand_model = RandomizedSearchCV(knn, parameter_knn, n_iter=n_iter, cv=5, verbose=3).fit(X_train, y_train)
n_neig_rand = knn_rand_model.best_params_['n_neighbors']

print('K value terbaik berdasarkan RandomizedSearchCV dengan', n_iter, 'iterasi adalah:', n_neig_rand)
```

Gambar 3. Code penemuan nilai K

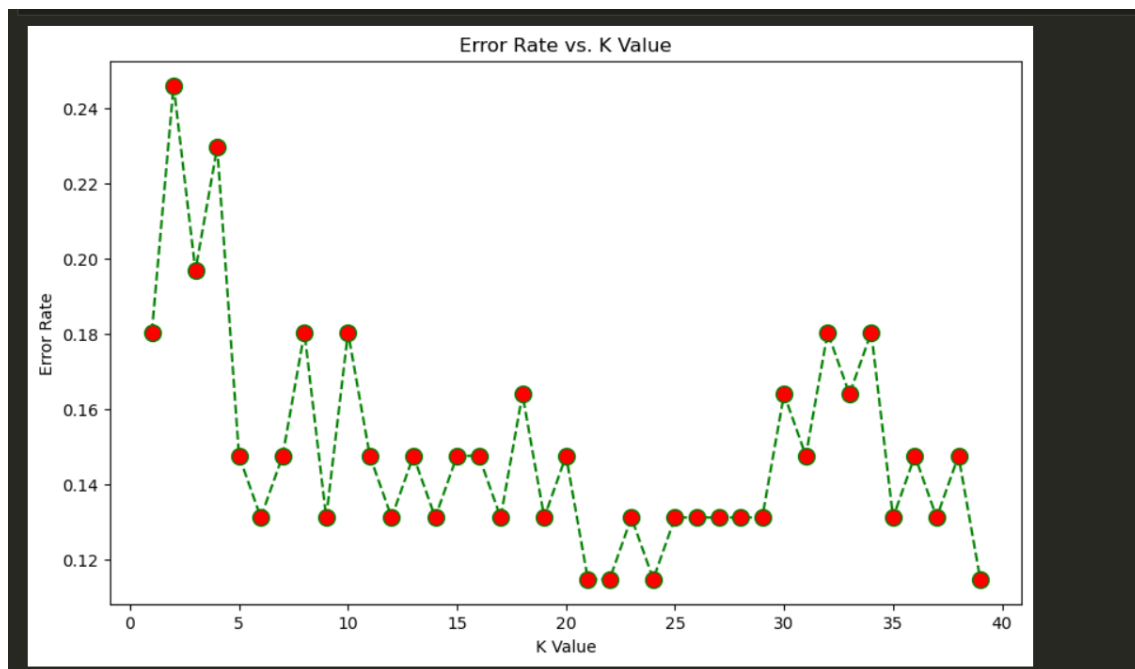
```
Fitting 5 folds for each of 20 candidates, totalling 100 fits
[CV 1/5] END .....n_neighbors=43;; score=0.816 total time= 0.0s
[CV 2/5] END .....n_neighbors=43;; score=0.792 total time= 0.0s
[CV 3/5] END .....n_neighbors=43;; score=0.812 total time= 0.0s
[CV 4/5] END .....n_neighbors=43;; score=0.792 total time= 0.0s
[CV 5/5] END .....n_neighbors=43;; score=0.771 total time= 0.0s
[CV 1/5] END .....n_neighbors=20;; score=0.837 total time= 0.0s
[CV 2/5] END .....n_neighbors=20;; score=0.792 total time= 0.0s
[CV 3/5] END .....n_neighbors=20;; score=0.833 total time= 0.0s
[CV 4/5] END .....n_neighbors=20;; score=0.792 total time= 0.0s
[CV 5/5] END .....n_neighbors=20;; score=0.833 total time= 0.0s
[CV 1/5] END .....n_neighbors=47;; score=0.837 total time= 0.0s
[CV 2/5] END .....n_neighbors=47;; score=0.792 total time= 0.0s
[CV 3/5] END .....n_neighbors=47;; score=0.792 total time= 0.0s
[CV 4/5] END .....n_neighbors=47;; score=0.771 total time= 0.0s
[CV 5/5] END .....n_neighbors=47;; score=0.771 total time= 0.0s
[CV 1/5] END .....n_neighbors=15;; score=0.878 total time= 0.0s
[CV 2/5] END .....n_neighbors=15;; score=0.771 total time= 0.0s
[CV 3/5] END .....n_neighbors=15;; score=0.812 total time= 0.0s
[CV 4/5] END .....n_neighbors=15;; score=0.792 total time= 0.0s
[CV 5/5] END .....n_neighbors=15;; score=0.833 total time= 0.0s
[CV 1/5] END .....n_neighbors=40;; score=0.837 total time= 0.0s
[CV 2/5] END .....n_neighbors=40;; score=0.792 total time= 0.0s
[CV 3/5] END .....n_neighbors=40;; score=0.812 total time= 0.0s
[CV 4/5] END .....n_neighbors=40;; score=0.812 total time= 0.0s
...
[CV 3/5] END .....n_neighbors=17;; score=0.812 total time= 0.0s
[CV 4/5] END .....n_neighbors=17;; score=0.771 total time= 0.0s
[CV 5/5] END .....n_neighbors=17;; score=0.854 total time= 0.0s
K value terbaik berdasarkan RandomizedSearchCV dengan 20 iterasi adalah: 6
```

Gambar 4. Hasil pencarian nilai tetangga terbaik

```
def find_k_values(X_train, y_train):
    error_rate = []
    for i in range(1,40):
        knn = KNeighborsClassifier(n_neighbors=i)
        knn.fit(X_train,y_train)
        pred_i = knn.predict(X_test)
        error_rate.append(np.mean(pred_i != y_test))

    plt.figure(figsize=(10,6))
    plt.plot(range(1,40),error_rate,color='green', linestyle='dashed', marker='o',
             markerfacecolor='red', markersize=10)
    plt.title('Error Rate vs. K Value')
    plt.xlabel('K Value')
    plt.ylabel('Error Rate')
    plt.show()
```

Gambar 5. Pencarian nilai K optimal untuk KNN



Gambar 6. Hasil Error Rate

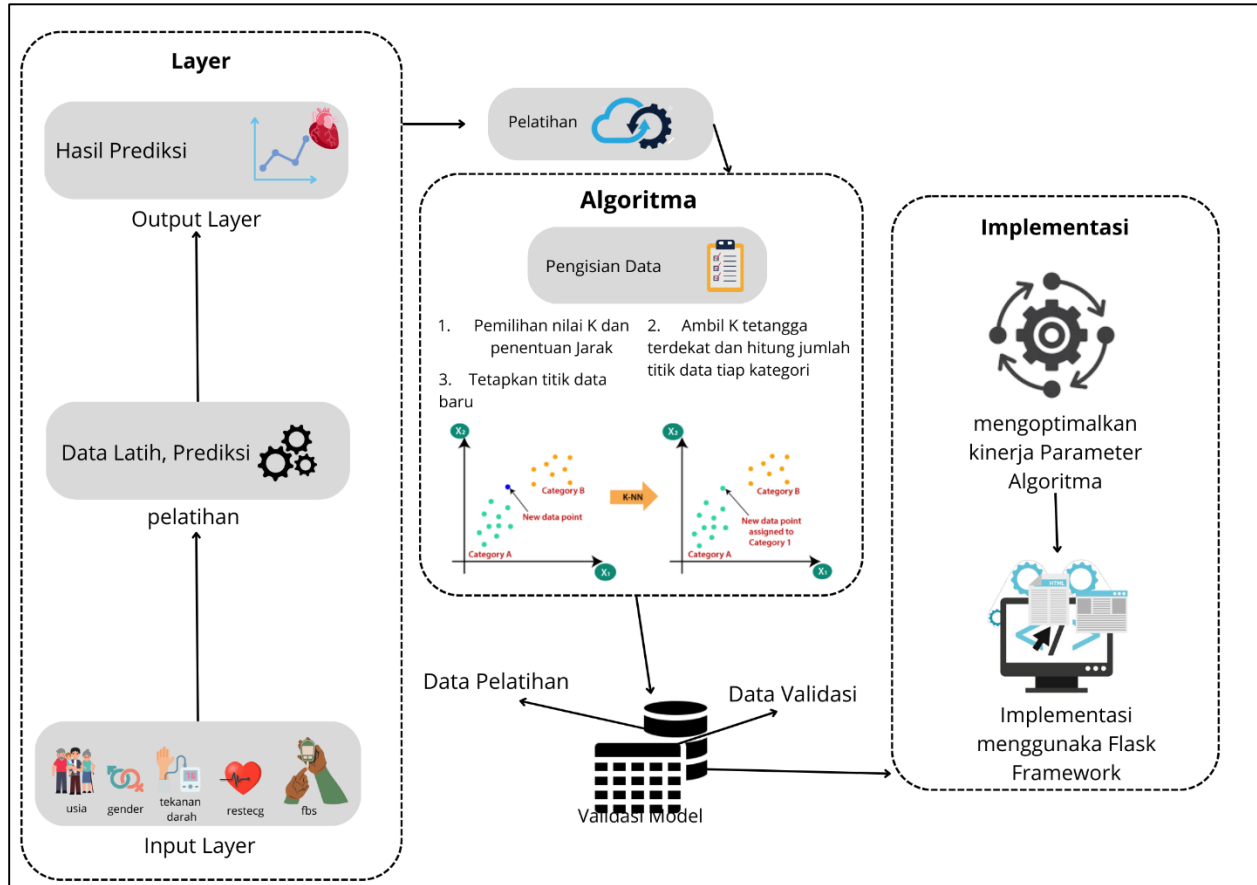
2. Pelatihan model

Pada tahap pelatihan, model KNN akan mempelajari hubungan antara fitur-fitur x dan label y. Hubungan tersebut dipelajari dengan cara menghitung jarak antara data baru (x) dengan semua data pelatihan. Data baru kemudian diklasifikasikan ke dalam kelas yang sama dengan tetangga terdekatnya. Pada penelitian ini, **x_train** yang dipakai yaitu: age, sex, cp, trtbps, chol, thalachh, oldpeak, slp, caa, thall, fbs, restecg, dan exng. Sedangkan untuk **y_trainnya** yaitu output.

BAB IV

ARSITEKTUR dan IMPLEMENTASI

4.1 Arsitektur model



Gambar 7. Arsitektur Model

Model dimulai dengan menerima informasi kesehatan individu sebagai input pada lapisan awal, di mana setiap data diwakili sebagai vektor fitur yang mencakup berbagai parameter kesehatan. Informasi ini kemudian melewati beberapa lapisan proses yang menggunakan fungsi aktivasi khusus untuk mempelajari pola kompleks antara parameter-parameter tersebut. Lapisan ini bertujuan untuk mengekstraksi fitur yang relevan dan memungkinkan model untuk memberikan prediksi yang akurat mengenai risiko serangan jantung. Setelah proses melalui lapisan ini, output dihasilkan pada lapisan terakhir, memberikan estimasi risiko serangan jantung berdasarkan kombinasi parameter kesehatan.

Pada saat yang sama, algoritma KNN dieksekusi untuk mengevaluasi seluruh data guna mengidentifikasi pola terdekat dari data kesehatan yang ada. Hasil dari algoritma KNN, yaitu data yang memiliki karakteristik serupa dengan inputan, kemudian disatukan dengan model. Model ini digunakan untuk memberikan prediksi atau penilaian pada setiap data, dan data dengan prediksi tertinggi dapat dianggap sebagai prediksi risiko serangan jantung yang paling mungkin terjadi. Proses ini memastikan bahwa prediksi risiko serangan jantung tidak hanya mempertimbangkan parameter kesehatan tetapi juga menggali kecenderungan individu, sehingga dapat memberikan solusi yang sesuai dengan kondisi kesehatan tiap individu.

4.2 Deployment Model

Model yang telah diterapkan terintegrasi dalam situs web melalui Flask. Flask bekerja sebagai server belakang yang menerima permintaan pengguna untuk memprediksi risiko serangan jantung. Ketika pengguna memasukkan data kesehatan ke dalam formulir di situs, Flask menggunakan algoritma KNN untuk menganalisis dataset yang luas, membandingkan entri baru dengan pola yang ada, dan memprediksi risiko serangan jantung berdasarkan karakteristik yang diberikan. Setelah proses analisis selesai, informasi mengenai risiko serangan jantung akan dikirim kembali kepada pengguna. Gambar 8 menampilkan *deploy* formulir pengisian yang memungkinkan pengguna untuk memasukkan data, sementara pada Gambar 9 terlihat hasil dari prediksi yang diberikan berdasarkan informasi yang dimasukkan sebelumnya.

Heart Attack Prediction

Usia:

Jenis Kelamin:

Tipe Nyeri Dada (cp):

Tekanan Darah (TRTBPS):

Kolesterol:

Gula Darah Puasa (fbs):

Elektrokardiografi Istirahat (restecg):

Denyut Jantung Maksimal (thalachh):

Latihan yang Dipicu Angina (exng):

Depresi ST Induk (oldpeak):

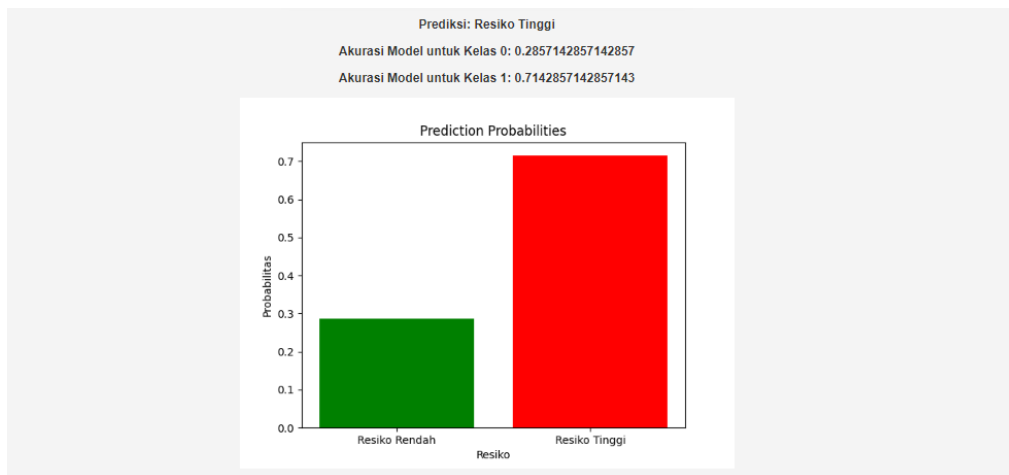
Kemiringan Segmen ST Selama Latihan (slp):

Jumlah Pembuluh Besar Berwarna (caa):

Thalassemia (thall):

Prediksi

Gambar 8. Tampilan Form Pengisian Kesehatan



Gambar 9. Hasil Prediksi

4.3 Evaluasi Model

Dalam penelitian ini, hasil evaluasi model yang telah dilakukan oleh peneliti meliputi akurasi KNN, Confusion Matrix dan Classification Report yaitu precision, recall, dan F1-score untuk setiap kelas. Untuk hasil evaluasi model dapat dilihat pada gambar 10.


```
knn_new = KNeighborsClassifier(n_neighbors = 35).fit(X_train, y_train)
y_pred = knn_new.predict(X_test)
print('Accuracy score for KNN: {}'.format(accuracy_score(y_test, y_pred)))
print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))
```

Accuracy score for KNN: 0.8688524590163934

```
[[21  6]
 [ 2 32]]
```

	precision	recall	f1-score	support
0	0.91	0.78	0.84	27
1	0.84	0.94	0.89	34
accuracy			0.87	61
macro avg	0.88	0.86	0.86	61
weighted avg	0.87	0.87	0.87	61

Gambar 10. Evaluasi Model

REFERENSI

- [1] M. H.K. and D. Ramesh, “Heart Attack Analysis and Prediction using SVM,” *Int. J. Comput. Appl.*, vol. 183, no. 27, pp. 35–39, 2021, doi: 10.5120/ijca2021921658.
- [2] P. B. Thakkar and R. H. Talwekar, “Analysis and risk estimation system for heart attack using EDENN algorithm,” *Int. J. Health Sci. (Qassim)*, vol. 6, no. April, pp. 5321–5330, 2022, doi: 10.53730/ijhs.v6ns1.6093.
- [3] Rokom, “Penyakit Jantung Penyebab Utama Kematian, Kemenkes Perkuat Layanan Primer,” *sehatNegeriku(kemkes)*, 2022. <https://sehatnegeriku.kemkes.go.id/baca/rilis-media/20220929/0541166/penyakit-jantung-penyebab-utama-kematian-kemenkes-perkuat-layanan-primer/> (accessed Nov. 24, 2023).
- [4] T. Aniamarta and A. S. Huda, “Review Artikel : Penyebab dan Pengobatan Serangan Jantung Review Article : Causes and Treatments of Heart Attack,” vol. 4, no. 1, pp. 22–31, 2022.
- [5] D. Kurniawan, K. Ibrahim, A. P. P, U. Borneo, F. Keperawatan, and U. Padjajdjaran, “Pengalaman Pasien Mengalami Serangan Jantung Pertama Kali yang dirawat di Ruang CICU Life Experiences of First-Time Heart Attack Patients who are Hospitalised in CICU,” vol. 3, pp. 67–76.
- [6] H. A. Dwi Fasnuari, H. Yuana, and M. T. Chulkamdi, “Penerapan Algoritma K-Nearest Neighbor Untuk Klasifikasi Penyakit Diabetes Melitus,” *Antivirus J. Ilm. Tek. Inform.*, vol. 16, no. 2, pp. 133–142, 2022, doi: 10.35457/antivirus.v16i2.2445.
- [7] M. D. Alkhussayid and F. Ferdiansyah, “Implementasi Algoritma K-Nearest Neighbors Pada Penentuan Jurusan Siswa,” *J. Sist. Komput. dan Inform.*, vol. 4, no. 1, p. 25, 2022, doi: 10.30865/json.v4i1.4759.
- [8] A. J. T, D. Yanosma, and K. Anggriani, “Implementasi Metode K-Nearest Neighbor (Knn) Dan Simple Additive Weighting (Saw) Dalam Pengambilan Keputusan Seleksi Penerimaan

- Anggota Paskibraka,” *Pseudocode*, vol. 3, no. 2, pp. 98–112, 2017, doi: 10.33369/pseudocode.3.2.98-112.
- [9] A. Suresh, “What is a confusion matrix?,” *Analytics Vidhya Medium*, 2020. <https://medium.com/analytics-vidhya/what-is-a-confusion-matrix-d1c0f8feda5> (accessed Nov. 28, 2023).
- [10] P. Banerjee, B. Kumar, A. Singh, R. Kumar, and R. Kumar, “Comparative performance analysis of optimized round robin scheduling(ORR) using dynamic time quantum with round robin scheduling using static time quantum in Real Time System,” *Int. J. Eng. Comput. Sci.*, vol. 8, no. 12, pp. 24890–24893, 2019, doi: 10.18535/ijecs/v8i12.4399.