



Application of K-Nearest Neighbor Algorithm Using Wrapper as Preprocessing for Determination of Human Weight Information

Penerapan Algoritma K-Nearest Neighbor Menggunakan Wrapper Sebagai Preprocessing untuk Penentuan Keterangan Berat Badan Manusia

**Febrianda Putra¹, Hafsa Fulaila Tahiyat², Raja Muhammad Ihsan³,
Rahmaddeni⁴, Lusiana Efrizoni⁵**

^{1,2,3,4,5}Program Studi Teknik Informatika, STMIK Amik Riau, Indonesia

E-Mail: ¹febriandaputra11@gmail.com, ²hafsahft1@gmail.com, ³rajamuhammadihisan02@gmail.com,
⁴rahmaddeni@sar.ac.id, ⁵lusiana.efrizoni@sar.ac.id

Received Nov 12th 2023; Revised Dec 15th 2023; Accepted Jan 20th 2024
Corresponding Author: Rahmaddeni

Abstract

This research focuses on improving the accuracy of determining human weight information through the application of the K-Nearest Neighbor (K-NN) algorithm with the Wrapper method as preprocessing. Weight awareness is essential in maintaining health and well-being. Although humans can know their weight, the approaches often used tend to be subjective and lack precision. The research identifies problems in determining weight information and seeks solutions through the use of prediction models. The K-NN algorithm was selected due to its ability to handle classification problems with complex datasets. Wrapper method was used as a preprocessing step to select the most significant feature subset. The dataset involved body weight parameters and other influential factors. The model was developed and tested using cross-validation techniques to ensure consistency of performance. The research findings show that the application of K-NN algorithm with Wrapper preprocessing can improve the accuracy of determining human weight information. The application of the K-Nearest Neighbor and K-Nearest Neighbor methods with Wrapper as a preprocessing stage in determining human weight information gets the same accuracy value of 91%. This study is expected to serve as a foundation for the development of better evaluation methods and more accurate information related to human weight.

Keyword: BMI, Forward Selection, Height, K-NN, Numeric, Weight, Wrapper

Abstrak

Penelitian ini fokus pada peningkatan akurasi penentuan keterangan berat badan manusia melalui penerapan Algoritma K-Nearest Neighbor (K-NN) dengan metode Wrapper sebagai preprocessing. Kesadaran akan berat badan menjadi esensial dalam menjaga kesehatan dan kesejahteraan. Meskipun manusia dapat mengetahui berat badannya, pendekatan yang sering digunakan cenderung bersifat subjektif dan kurang presisi. Penelitian mengidentifikasi permasalahan dalam penentuan keterangan berat badan dan mencari solusi melalui penggunaan model prediksi. Algoritma K-NN terpilih karena kemampuannya dalam menangani permasalahan klasifikasi dengan dataset yang kompleks. Metode Wrapper digunakan sebagai langkah preprocessing untuk memilih subset fitur yang paling signifikan. Dataset melibatkan parameter berat badan dan faktor-faktor lain yang berpengaruh. Model dikembangkan dan diuji menggunakan teknik cross-validation untuk memastikan konsistensi kinerja. Temuan penelitian menunjukkan bahwa penerapan Algoritma K-NN dengan Wrapper preprocessing dapat meningkatkan akurasi penentuan keterangan berat badan manusia. Penerapan metode K-Nearest Neighbor dan K-Nearest Neighbor dengan Wrapper sebagai tahap preprocessing dalam menentukan keterangan berat badan manusia mendapatkan hasil nilai akurasi yang sama yaitu sebesar 91%. Studi ini diharapkan dapat menjadi landasan untuk pengembangan metode evaluasi yang lebih baik dan informasi yang lebih akurat terkait berat badan manusia.

Kata Kunci: Berat Badan, BMI, Forward Selection, K-NN, Numerik, Tinggi Badan, Wrapper

1. PENDAHULUAN

Saat ini teknologi merupakan salah satu bidang yang memegang peranan sangat penting dalam berbagai aspek kehidupan manusia. Penerapan kemajuan teknologi saat ini umumnya menjadikan sistem digital lebih

efektif dibandingkan sistem analog. Salah satu bentuk sistem digital yang sedang dikembangkan adalah di bidang medis, misalnya alat yang dapat digunakan untuk menghitung indeks massa tubuh (BMI). BMI merupakan pengukuran yang mengkorelasikan atau membandingkan berat badan dan tinggi badan dan sering digunakan untuk mengetahui kondisi tubuh dalam kategori berat badan kurang (*underweight*), kelebihan berat badan (*overweight*), obesitas, atau ideal [1].

Berat badan ideal termasuk tampilan fisik yang membuat seseorang tampil dengan percaya diri. Selain itu, menjaga berat badan tetap ideal cukup penting, karena dapat mencegah atau mengendalikan serangan penyakit dan dengan tubuh yang ideal membuat nyaman dalam beraktivitas. Jika seandainya berat badan terlalu kurus atau kelebihan berat badan (obesitas), maka akan membuat badan cepat letih atau merasa cepat lelah, sedangkan dengan tubuh yang ideal, tentunya akan membuat nyaman dalam melakukan sesuatu [2].

Kesadaran akan pentingnya pemahaman yang mendalam terhadap berat badan manusia sebagai indikator kesehatan yang krusial. Dalam konteks masyarakat, variasi berat badan manusia menjadi perhatian utama karena dapat mencerminkan gaya hidup, pola makan, dan aktivitas fisik seseorang. Berbagai faktor seperti usia, tinggi badan, komposisi tubuh, dan berat organ dalam turut memengaruhi variasi berat badan. Oleh karena itu, pengukuran berat badan tidak dapat hanya didasarkan pada angka absolut, melainkan memerlukan pendekatan yang lebih holistik. Berdasarkan pernyataan tersebut di atas, maka dilakukan penelitian yang bertujuan untuk mengembangkan alat ukur berat badan dan tinggi badan secara digital untuk menghitung BMI lansia. Alat tersebut menggunakan keyboard sebagai input untuk memasukkan usia Anda, menggunakan sensor load cell untuk mengukur berat badan Anda, dan menggunakan sensor ultrasonik untuk mengukur tinggi lutut Anda [1].

Penelitian sebelumnya dengan judul “Alat Pengukur Tinggi Badan dan Berat Badan Untuk Informasi Berat Ideal Manusia Berbasis Arduino” membuat alat pengukur untuk mendapat informasi tubuh ideal seseorang yaitu dengan BMI/ IMT, Sedangkan dalam penelitian ini akan mengklasifikasikan nilai dari yang didapat dari pengukuran BMI menggunakan metode K-Nearest Neighbor (K-NN) apakah akan termasuk dalam kategori ideal atau tidak [3].

Dalam penelitian ini, penggunaan metode analisis data menjadi krusial untuk menghasilkan informasi yang akurat dan berguna. Terdapat banyak metode klasifikasi dalam supervised learning pada machine learning, diantaranya adalah K-Nearest Neighbor (K-NN), Naive Bayes Classifier (NBC), Support Vector Machine (SVM), Neural Network (NN), Random Forest Classifier (RFC), Ada Boost Classifier (ABC), serta Quadratic Discriminant Analysis (QDA). Masing-masing metode ini mempunyai kelebihan dan kekurangan. Salah satu kelebihan metode klasifikasi ini terlihat dari caranya menangani objek kumpulan data. K-Nearest Neighbors (K-NN) merupakan algoritma yang mengklasifikasikan data berdasarkan K tetangga terdekat (*nearest neighbours*) [4].

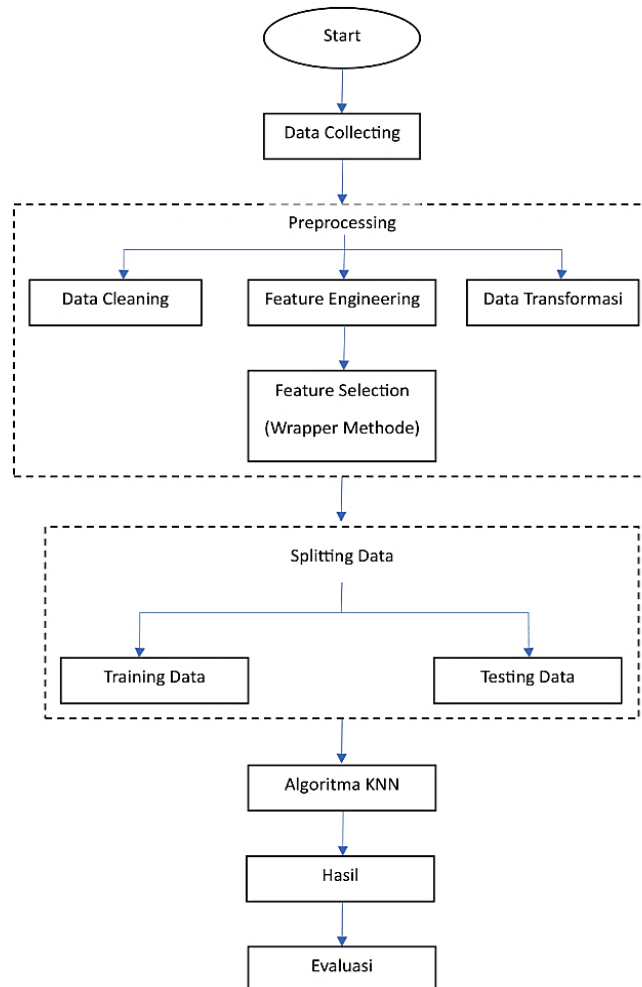
K-NN memiliki keunggulan dalam menghasilkan data yang kuat atau bersih dan efektif untuk kumpulan data yang besar. K-Nearest Neighbor juga punya kekurangan. Artinya diperlukan nilai k, jarak ke data eksperimen tidak jelas tergantung pada jenis jarak yang digunakan, dan untuk hasil terbaik sebaiknya gunakan semua atribut atau salah satu atribut tertentu saja [5]. Namun, meskipun K-NN telah memberikan hasil yang baik dalam berbagai kasus, keberhasilannya sangat tergantung pada pemilihan fitur atau atribut yang relevan. Oleh karena itu, perlu adanya tahap preprocessing yang akurat untuk meningkatkan performa K-NN. Salah satu pendekatan yang dapat digunakan adalah metode Wrapper, di mana seleksi fitur dilakukan secara iteratif dengan memanfaatkan hasil dari model yang sedang dievaluasi [6]. Metode wrapper adalah metode sederhana tetapi mungkin ada risiko overfitting model. Beberapa contoh metode filter adalah chi-square, perolehan informasi, pemilihan fitur berbasis korelasi dan bantuan [7]. Dengan demikian, algoritma pemilihan fitur berbasis pembungkus dengan algoritma genetika (GA) dan K nearest neighbor (K-NN) telah dikembangkan untuk menentukan peringkat pentingnya fitur. Dalam makalah ini, kita membahas bagaimana kebisingan dan fitur yang tidak relevan dalam kumpulan data dapat diidentifikasi. Selain itu, masalah pemilihan diskriminatif minimum juga dianalisis dan ditentukan [8]. Keakuratan pendekatan Wrapper biasanya tinggi; Namun, generalitas hasilnya terbatas, dan kompleksitas komputasinya tinggi [9]. Dengan adanya fitur seleksi wrapper dapat membantu meningkat akurasi dari metode K-NN,

Fitur yang menjaga kinerja model tetap tinggi dipilih dengan pendekatan wrapper yang terdiri dari algoritma Chaotic Particle Swarm Optimization (CPSO) dan K-NN untuk menjaga kompleksitas komputasi model seminimal mungkin dan untuk mengatasi kekacauan dalam gambar fundus [10]. Dalam karya ini, kami menggunakan tiga algoritma model wrapper yang diterapkan pada pengklasifikasi K-NN yang dijelaskan di bagian sebelumnya: backward, forward, and branch and bound. Dalam forward technique, pencarian dimulai dengan serangkaian fitur kosong dan kemudian secara progresif menggabungkan fitur ke dalam subset yang lebih besar [11]. Metode wrapper terdiri dari pengklasifikasi dan algoritma pencarian. Subset yang paling sukses dipilih dengan evaluasi akurasi klasifikasi. Tujuannya adalah mengoptimalkan akurasi klasifikasi atau persamaan yang terkait dengan akurasi untuk subset yang dipilih. Untuk dataset dimensi tinggi dengan jumlah fitur N , evaluasi subset ($2^N - 1$) akan menghasilkan beban komputasi yang cukup besar. Dengan demikian, algoritma heuristik atau evolusioner lebih disukai untuk pemilihan subset. Metode wrapper mempertimbangkan ketergantungan dan hubungan antar gen. Meskipun kompleksitas komputasinya lebih

tinggi daripada metode filter, akurasi metode pembungkus jauh lebih baik [12]. Tujuan dari penelitian ini adalah menganalisis tingkat akurasi hasil prediksi berat badan menggunakan algoritma K-NN tanpa dan dengan wrapper preprocessing.

2. METODOLOGI PENELITIAN

Penelitian ini akan menggunakan pendekatan pengembangan untuk mengevaluasi efektivitas Metode K-Nearest Neighbor (K-NN) dengan Wrapper sebagai tahap preprocessing dalam menentukan keterangan berat manusia. Berikut adalah tahapan metodologi penelitiannya:



Gambar 1. Metodologi Penelitian

2.1. Pengumpulan Data

Aktivitas ini merupakan langkah awal dalam melakukan penelitian atau analisis data untuk mendapatkan informasi yang relevan. Pengumpulan data adalah tahap kritis dalam siklus penelitian dan sering kali memerlukan perencanaan yang matang untuk memastikan data yang diperoleh berkualitas. Untuk kasus ini, kami mendapatkan dataset pada website <https://www.kaggle.com>.

2.2. Pemrosesan Data

Serangkaian langkah atau proses yang dilakukan pada data sebelum data tersebut digunakan dalam analisis atau pemodelan. Tujuan dari preprocessing adalah untuk membersihkan, menormalkan, dan mempersiapkan data sehingga dapat diolah lebih efektif oleh algoritma analisis atau model machine learning. Dataset ini terlebih dahulu dilakukan pembersihan, perbaikan, dan transformasi tanpa merubah validitas isi yang terkandung dalam dataset[13]. Berikut adalah beberapa tahap umum dalam preprocessing data:

1. Pembersihan Data

Tahap pertama dari preprocessing adalah cleaning[14]. Data yang noise, tidak konsisten, atau tidak relevan dihilangkan melalui pembersihan data. [15]. Untuk kasus saat ini, kami membuang fitur yang tidak kami pakai yaitu BMI.

2. Transformasi Data

Setelah kesalahan-kesalahan pada data dihilangkan, pada tahap transformasi data ditransformasikan sesuai tipe datanya dan tipe data tersebut diklasifikasikan sebagai data kategorikal. [16]. Transformasi data merujuk pada serangkaian teknik atau metode untuk mengubah data dari bentuk atau distribusi awalnya menjadi bentuk atau distribusi yang sesuai dengan kebutuhan analisis atau model. Transformasi data sering kali diperlukan untuk memenuhi asumsi statistik tertentu atau untuk meningkatkan kinerja model machine learning. Transformasi pada penelitian ini yaitu mengubah data pada fitur BmiClass kedalam bentuk numerik, yang nantinya akan dilakukan seleksi fitur.

3. Seleksi Fitur

Seleksi fitur adalah salah satu teknik klasifikasi prapemrosesan yang umum digunakan dalam pembelajaran mesin dan statistik untuk meningkatkan kinerja pembelajaran dan memecahkan masalah dengan data berdimensi tinggi.[17]. Proses pemilihan subset fitur atau variabel dari dataset yang akan digunakan untuk membangun model atau melakukan analisis. Tujuannya adalah untuk meningkatkan kinerja model dengan menggunakan hanya fitur-fitur yang paling relevan atau signifikan, sambil mengurangi kompleksitas dan overhead komputasi yang terkait dengan menggunakan seluruh set fitur. Pada kasus ini kami menggunakan Metode Wrapper sebagai seleksi fitur:

Metode Wrapper

Metode ini menilai kinerja model dengan menggunakan subset fitur tertentu. Sejumlah model dibangun dengan subset fitur yang berbeda, dan performa model dievaluasi. Contoh metode wrapper termasuk Recursive Feature Elimination (RFE), Backward Selection dan Forward Selection. Dengan pengklasifikasi yang telah ditentukan, model pembungkus tipikal akan melakukan langkah-langkah berikut[18]:

- a. Langkah 1: mencari subset fitur,
- b. Langkah 2: mengevaluasi subset fitur yang dipilih berdasarkan kinerja pengklasifikasi,
- c. Langkah 3: mengulangi Langkah 1 dan Langkah 2 hingga kualitas yang diinginkan tercapai.

2.3. Pembagian Data

Pembagian data adalah proses membagi dataset menjadi dua atau lebih subset yang berbeda, biasanya untuk keperluan pelatihan dan pengujian model machine learning. Tujuan utama dari pembagian data adalah untuk mengukur kinerja model pada data yang belum pernah dilihat selama proses pelatihan. Ada beberapa jenis pembagian data yang umum digunakan:

1. Pelatihan Data

Data training digunakan untuk pengembangan model dan melatih algoritma[19]. Subset dari dataset yang digunakan untuk melatih model machine learning. Data pelatihan ini berisi contoh-contoh yang diberi label atau output yang diinginkan, dan model akan menggunakan data ini untuk belajar dan membuat prediksi pada data baru yang belum pernah dilihat sebelumnya.

2. Pengujian Data

Ketika dihadapkan pada data baru yang tidak teridentifikasi, pengujian dilakukan untuk mengamati performa algoritma yang telah dilatih sebelumnya. [19]. Subset dari dataset yang digunakan untuk menguji kinerja model machine learning setelah model dilatih dengan data pelatihan. Testing data berfungsi untuk mengukur sejauh mana model mampu melakukan generalisasi pada data yang belum pernah dilihat sebelumnya, yang disebut sebagai data baru atau data yang belum terlihat.

2.4. Algoritma K-Nearest Neighbor (K-NN)

K Klasifikasi adalah metode membangun model klasifikasi dari sampel data pelatihan. Klasifikasi melibatkan analisis data masukan dan membangun model dengan mendeskripsikan kelas data algoritma[20]. K-NN merupakan metode yang menggunakan algoritma supervised[21]. K-NN adalah teknik untuk mengklasifikasikan dataset yang telah diklasifikasi sebelumnya. Keakuratan algoritma K-NN sangat dipengaruhi oleh fitur yang berbeda ketika nilainya tidak sesuai dengan nilai perkiraan. Beberapa studi yang menggunakan algoritma K-NN menangani hampir secara eksklusif pemilihan fitur dan pembobotan untuk meningkatkan efisiensi algoritma dalam klasifikasi[22]. Algoritma K-NN memiliki keuntungan dalam menghasilkan data yang kuat atau tidak ambigu dan efektif bila digunakan pada data yang cukup besar. [15].

Tujuan dari algoritma K-NN adalah untuk mengklasifikasikan objek baru berdasarkan atribut dan data pelatihannya. K-NN menggunakan klasifikasi pada tetangga-tetangga sebagai nilai prediksi dari query instance yang baru[23]. Proses metode K-NN dilakukan dengan mencari kelompok k objek pada data latih yang paling dekat dengan objek pada data baru atau data uji. Langkah-langkah untuk menghitung metode K-Nearest Neighbor antara lain[24]:

1. Menentukan parameter K.
2. Menghitung jarak antara data training dan data testing.
Perhitungan jarak yang paling umum dipakai pada perhitungan pada algoritma K-NN adalah menggunakan perhitungan jarak Euclidean. Rumusnya adalah sebagai berikut:

$$euc = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad (1)$$

Dimana:

- p_i = sample data / data training
- q_i = data uji / data testing
- i = variable data
- n = dimensi data

3. Mengurutkan jarak yang terbentuk.
4. Menentukan jarak terdekat sampai urutan K.
5. Memasangkan kelas yang bersesuaian.
6. Mencari jumlah kelas dari tetangga yang terdekat dan tetapkan kelas tersebut sebagai kelas data yang akan dievaluasi.

2.5. Evaluasi

Evaluasi memiliki makna yang berbeda dengan penilaian dan pengukuran. Evaluasi didahului dengan penilaian (assessment), sedangkan penilaian didahului dengan pengukuran[25]. Terdapat beberapa cara untuk mengevaluasi model K-nearest neighbors (K-NN), diantaranya adalah :

1. *Precision*
Presisi dapat diartikan sebagai kecocokan antara permintaan informasi dengan jawaban terhadap permintaan tersebut.
2. *Recall*
Recall didefinisikan sebagai rasio dari item relevan yang dipilih terhadap total jumlah item relevan yang tersedia.
3. *F1-Score*
Measure adalah harmonic mean antara nilai presisi dan recall, F-measure juga kadang disebut dengan nama *F1-Score*.

$$AKURASI = \frac{(TP+TN)}{(TP+TN+FT+FN)} \quad (2)$$

$$PRECISION = \frac{TP}{TP+FP} \quad (3)$$

$$RECALL = \frac{TP}{TP+FN} \quad (4)$$

$$F - MEASURE = \frac{2(PRECISION \times RECALL)}{PRECISION + RECALL} \quad (5)$$

4. Akurasi
Akurasi adalah inti dari setiap proses dan produk yang sukses. dalam kasus proses business intelligence, akurasi mengacu pada keakuratan input data dan output yang dihasilkan. Menghitung akurasi total dari model, ini juga memberikan gambaran umum tentang Tingkat akurasi model berkinerja pada seluruh dataset.
5. Interpretasi Model
Memahami faktor-faktor apa yang mempengaruhi prediksi model. Wrapper akan mempertimbangkan apakah fitur-fitur yang dipilih secara efektif meningkatkan kinerja model.

3. HASIL DAN ANALISIS

Berdasarkan dari metodologi penelitian, maka proses Penerapan Algoritma K-Nearest Neighbor Menggunakan Wrapper Sebagai Preprocessing untuk Penentuan Keterangan Berat Badan Manusia menghasilkan beberapa hasil.

3.1 Pengumpulan Data

Data yang dikumpulkan atau diperoleh berupa dataset yang berasal dari data Mahasiswa tahap akhir di website Kaggle. Dengan 741 data yang telah didapatkan. Berikut data yang didapatkan dari sumber <https://www.kaggle.com/datasets/rukenmissonnier/age-weight-height-bmi-analysis>. Setelah data didapatkan, langkah berikutnya yaitu melakukan preprocessing data untuk mempersiapkan data yang meliputi beberapa tahap seperti pembersihan data, perubahan data, dan pengintegrasian data pada algoritma machine learning. Berikut adalah hasil dari preprocessing data.

3.2 Pemrosesan Data

Proses pemrosesan data dimulai dengan membuang fitur Bmi dikarenakan fitur yang digunakan hanya 3 (*Age*, *Height*, dan *Weight*) yang nantinya akan diseleksi dengan menggunakan *Forward Selection*. Forward selection dilakukan menggunakan metode Wrapper yang akan memilih fitur-fitur apa saja yang berpengaruh dalam menentukan fitur BmiClassnya. Selanjutnya mengubah label BmiClass menjadi nilai numerik. Lalu proses yang terakhir yaitu mengambil fitur-fitur yang nantinya akan diuji dengan metode K-Nearest Neighbor.

3.3 Algoritma K-Nearest Neighbor (K-NN)

Berikut adalah kode program Splitting Data menggunakan perbandingan 80:20, 70:30, 60:40.

Kode Program 1

```
split_ratios = [0.2,0.3,0.4]

for ratio in split_ratios:
    print('\nData testing : ',ratio)
    print('')
    X = dataset[['Age','Height','Weight']]
    y = dataset['BmiClass']

    X_train, X_test, y_train, y_test=
    train_test_split(X,y,test_size=(ratio),random_state=42)
    y_pred_lr = regressor.predict(X_test)

    df = pd.DataFrame({'Actual':y_test,'Predicted':y_pred_lr})
    print(df)

    lr_sfs_1 = LinearRegression()

    sfs_1 = SFS(lr_sfs_1,
                k_features=1,
                cv=3,
                floating=False,
                scoring='neg_mean_squared_error',
                n_jobs=-1)

    sfs_1 = sfs_1.fit(X_train, y_train)
    X_train_sfs_1 = sfs_1.transform(X_train)
    X_test_sfs_1 = sfs_1.transform(X_test)

    lr_sfs_1.fit(X_train_sfs_1, y_train)
    y_pred_sfs_1 = lr_sfs_1.predict(X_test_sfs_1)
```

Kemudian, berikut adalah kode program Splitting Data 70:30.

Kode Program 2

```
X = dataset[['Age','Height','Weight']]
y = dataset['BmiClass']

X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.3,random_state=42)

K-NN = KNeighborsClassifier (n_neighbors=5)

y_pred = K-NN.predict (X_test)

print('\nConfusion Matrix')
cm = confusion_matrix(y_test,y_pred)
```

```
print(cm)

print('\nClassification report dengan data training dan data testing = 80:20')
cr = classification_report(y_test,y_pred)

print(cr)
```

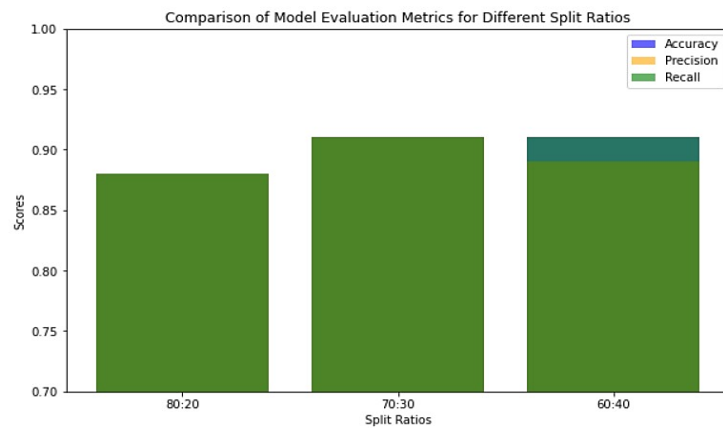
Pada kode program 1, terdapat *source code feature selection* menggunakan metode *wrapper*. Dimulai dengan mencari *splitting* data terbaik dari 80 data training dan 20 data uji, 70 data training dan 30 data uji, 60 data training dan 40 data testing. Yang nantinya akan menghasilkan fitur-fitur yang berpengaruh terhadap fitur *BmiClas*. Pada kode program 2, Terdapat *source code* untuk mencari akurasi menggunakan *K-Nearest Neighbor*. Penelitian ini dilakukan berdasarkan metode *K-Nearest Neighbor* dengan *splitting* data 80:20, 70:30, 60:40, dengan nilai $k=5$, $random_state = 42$. Berdasarkan hasil pengujian yang dilakukan, menghasilkan Tingkat presisi, recall, f1-score dan support. Yang nantinya akan diambil salah satu *splitting* data yang terbaik.

3.4 Evaluasi

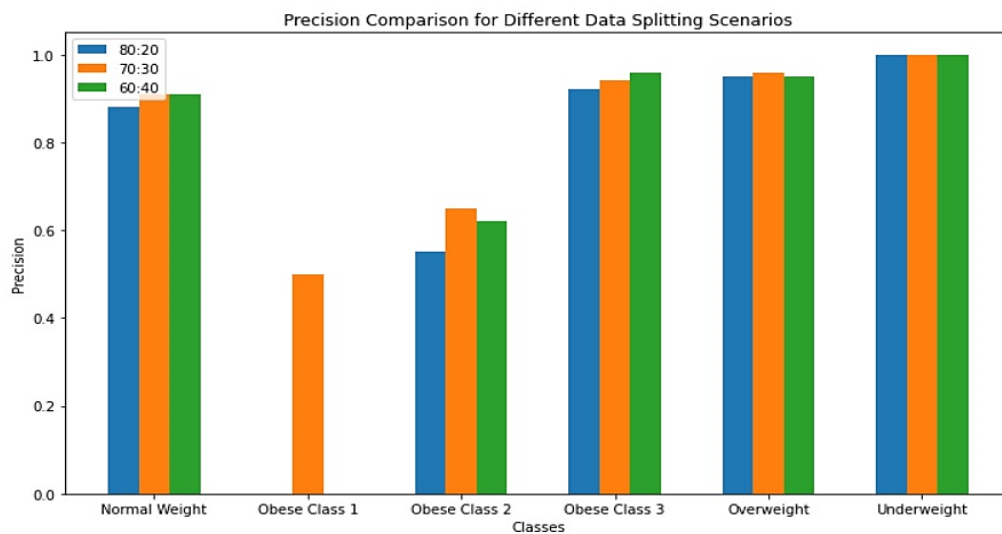
Tahapan evaluasi pada algoritma K-NN dapat dilihat pada tabel 1, gambar 4 dan gambar 5.

Tabel 1. Classification Report

| Data Training (%) | Data Testing (%) | Accuracy (%) | Precision (%) | Recall (%) | F1-score | Support |
|-------------------|------------------|--------------|---------------|------------|----------|---------|
| 60 | 40 | 0.91 | 0.89 | 0.91 | 0.89 | 297 |
| 70 | 30 | 0.91 | 0.91 | 0.91 | 0.90 | 223 |
| 80 | 20 | 0.88 | 0.88 | 0.88 | 0.88 | 149 |



Gambar 4. Perbandingan nilai Evaluasi Model dari Setiap *Splitting* Data



Gambar 5. Perbandingan nilai presisi dari setiap *splitting* data.

Pada tabel 1 dan pada gambar 4, terdapat data laporan klasifikasi dan visualisasi dari setiap *splitting* data yang diuji. Dengan *splitting* data 80:20; 70:30; 60:40, didapatkan nilai keseluruhan yang tertinggi yaitu 70:30 sebesar 91%. Pada gambar 5, Terdapat visualisasi nilai presisi dari beberapa *splitting* data pada fitur *BmiClas*. Dengan *splitting* data 70:30, didapatkan hasil presisi yang lebih unggul dibanding *splitting* data lainnya. Pada *BmiClas* ‘*Obese Class 1*’, *splitting* data 70:30 mendapatkan nilai presisi yang jauh lebih baik dibanding kawan-kawannya, yaitu sebesar 50%.



Gambar 6. User Interface

Pada gambar 6, merupakan tampilan *user interface* (pengguna antar muka), jika dibuat kedalam bentuk aplikasi.

4. KESIMPULAN

Penerapan metode K-Nearest Neighbor dan K-Nearest Neighbor dengan Wrapper sebagai tahap preprocessing dalam menentukan keterangan berat manusia mendapatkan hasil nilai akurasi yang sama yaitu sebesar 91%. Ini menandakan bahwa penelitian ini dapat membuka peluang baru dalam menentukan keterangan berat manusia dengan akurat. Dari penelitian yang telah dilakukan, pendekatan ini menggabungkan kelebihan K-NN dalam menangani masalah klasifikasi dengan kemampuan Wrapper sebagai seleksi fitur, menciptakan model yang bijak dan efisien. Dengan implementasi yang cermat, metode ini dapat menjadi kontribusi berharga dalam penelitian kesehatan dan kebugaran, membantu dalam memberikan rekomendasi yang lebih personal dan akurat untuk individu.

REFERENSI

- [1] Brian Ikhsana, M. A. Riyadi, and S. Sudjadi, "Perancangan alat pengukur berat badan dan tinggi badan digital untuk mengetahui body mass index (BMI) pada lansia," *Transient J. Ilm. Tek. Elektro*, vol. 4, no. 2, pp. 244–252, 2015.
- [2] R. B. Wicaksana, B. Hidayat, and S. Aulia, "Citra Telapak Kaki Dengan Metode Gabor Wavelet Dan K-NN Berbasis Android Classification Height and Weight of Human Footprint Using Gabor," *e-proceeding Eng.*, vol. 3, no. 2, pp. 1695–1701, 2016.
- [3] D. M. K. N. K-nn, T. Annuqayah, and G. Sumenep, "CLASSIFICATION OF THE HUMAN BODY MASS INDEX OR THE," pp. 1–7.
- [4] A. M. Argina, "Penerapan Metode Klasifikasi K-Nearest Neighbor pada Dataset Penderita Penyakit Diabetes," *Indones. J. Data Sci.*, vol. 1, no. 2, pp. 29–33, 2020, doi: 10.33096/ijodas.v1i2.11.
- [5] F. Yunita, "Sistem Klasifikasi Penyakit Diabetes Mellitus Menggunakan Metode K-Nearest Neighbor (K-Nn)," *Bappeda*, vol. 2, no. 1, pp. 223–230, 2016.
- [6] P. Cunningham and S. J. Delany, "K-Nearest Neighbour Classifiers-A Tutorial," *ACM Comput. Surv.*, vol. 54, no. 6, 2021, doi: 10.1145/3459665.
- [7] Y. B. Wah, N. Ibrahim, H. A. Hamid, S. Abdul-Rahman, and S. Fong, "Feature selection methods: Case of filter and wrapper approaches for maximising classification accuracy," *Pertanika J. Sci. Technol.*, vol. 26, no. 1, pp. 329–340, 2018.
- [8] J. Leng, C. Valli, and L. Armstrong, "A Wrapper-Based Feature Selection for Analysis of Large Data Sets," *Proc. 2010 3rd Int. Conf. Comput. Electr. Eng. (ICCEE 2010)*, no. 2010, pp. 167–170, 2010.
- [9] N. M. Hewahi and E. A. Alashqar, "Wrapper feature selection based on genetic algorithm for

- recognizing objects from satellite imagery,” *J. Inf. Technol. Res.*, vol. 8, no. 3, pp. 1–20, 2015, doi: 10.4018/JITR.2015070101.
- [10] Y. B. Özçelik and A. Altan, “Overcoming Nonlinear Dynamics in Diabetic Retinopathy Classification: A Robust AI-Based Model with Chaotic Swarm Intelligence Optimization and Recurrent Long Short-Term Memory,” *Fractal Fract.*, vol. 7, no. 8, 2023, doi: 10.3390/fractalfract7080598.
- [11] G. M. Bellino, L. Schiaffino, M. Battisti, J. Guerrero, and A. Rosado-Muñoz, “Optimization of the K-NN supervised classification algorithm as a support tool for the implantation of deep brain stimulators in patients with Parkinson’s Disease,” *Entropy*, vol. 21, no. 4, pp. 1–14, 2019, doi: 10.3390/e21040346.
- [12] Ö. S. Sönmez, M. Dağtekin, and T. Ensari, “Gene expression data classification using genetic algorithm-based feature selection,” *Turkish J. Electr. Eng. Comput. Sci.*, vol. 29, no. 7, pp. 3165–3179, 2021, doi: 10.3906/ELK-2102-110.
- [13] M. E. N. Inayah, K. B. Sirait, N. Casie, and ..., “... Produksi Telur di Provinsi Jawa Barat dengan Menggunakan Algoritma K-Means: Grouping of Egg Production in West Java Province Using the K-Means Algorithm,” ... *Nas. Penelit. dan ...*, pp. 207–214, 2022, [Online]. Available: <https://journal.irpi.or.id/index.php/sentimas/article/view/349%0Ahttps://journal.irpi.or.id/index.php/sentimas/article/download/349/152>
- [14] D. Efriadi, R. Rahmaddeni, A. Agustin, and J. Junadhi, “Prediksi Penambahan Piutang Iuran Jaminan Sosial Ketenagakerjaan menggunakan Algoritma K-Nearest Neighbor,” *Edumatic J. Pendidik. Inform.*, vol. 6, no. 1, pp. 49–57, 2022, doi: 10.29408/edumatic.v6i1.5255.
- [15] S. J. P. Syarifah, Q. Attaqwa, and A. Pratama, “... Karyawan Menggunakan Algoritma C4. 5 dan K-nearest Neighbor: Classification Determining Employee Data Work Schedules Using C4. 5 and K-Nearest Neighbor ...,” ... *Nas. Penelit. dan ...*, pp. 215–221, 2022, [Online]. Available: <https://journal.irpi.or.id/index.php/sentimas/article/view/327%0Ahttps://journal.irpi.or.id/index.php/sentimas/article/download/327/156>
- [16] S. Cumel, David Zamri, Rahmaddeni, “Perbandingan Metode Data Mining untuk Prediksi Banjir Dengan Algoritma Naïve Bayes dan K-NN,” *SENTIMAS Semin. Nas. Penelit. dan ...*, pp. 40–48, 2022, [Online]. Available: <https://journal.irpi.or.id/index.php/sentimas/article/view/353%0Ahttps://journal.irpi.or.id/index.php/sentimas/article/download/353/132>
- [17] D. ARIYOGA, “Perbandingan Metode Seleksi Fitur Filter, Wrapper, Dan Embedded Pada Klasifikasi Data Nirs Manga Menggunakan Random Forest Dan Support Vector Machine ...,” 2022, [Online]. Available: <https://dspace.uui.ac.id/handle/123456789/38955>
- [18] C. C. Aggarwal, “Educational and software resources for data classification,” *Data Classif. Algorithms Appl.*, pp. 657–665, 2014, doi: 10.1201/b17320.
- [19] A. Rahmansyah, O. Dewi, P. Andini, T. Hastuti, P. Ningrum, and M. E. Suryana, “Membandingkan Pengaruh Feature Selection Terhadap Algoritma Naïve Bayes dan Support Vector Machine,” *Semin. Nas. Apl. Teknol. Inf.*, pp. 1907–5022, 2018.
- [20] P. Putra, A. M. H. Pardede, and S. Syahputra, “Analisis Metode K-Nearest Neighbour (K-NN) Dalam Klasifikasi Data Iris Bunga,” *J. Tek. Inform. Kaputama*, vol. 6, no. 1, pp. 297–305, 2022.
- [21] A. Saputra, R. P. Hasibuan, Renaldi, and Rahmaddeni, “Perbandingan Tingkat Kadar Minuman Beralkohol di Dunia Menggunakan Algoritma Naïve Bayes dan K-Nearest Neighbor,” *SENTIMAS Semin. Nas. Penelit. dan Pengabd. Masy.*, pp. 127–132, 2022, [Online]. Available: <https://journal.irpi.or.id/index.php/sentimas/article/view/314%0Ahttps://journal.irpi.or.id/index.php/sentimas/article/download/314/143>
- [22] A. Putri, C. Syaficha Hardiana, E. Novfuja, F. Try Puspa Siregar, Y. Fatma, and R. Wahyuni, “Comparison of K-NN, Naive Bayes and SVM Algorithms for Final-Year Student Graduation Prediction Komparasi Algoritma K-NN, Naive Bayes dan SVM untuk Prediksi Kelulusan Mahasiswa Tingkat Akhir,” *Inst. Ris. dan Publ. Indones. MALCOM Indones. J. Mach. Learn. Comput. Sci. J. Homepage*, vol. 3, no. 1, pp. 20–26, 2023.
- [23] R. Wahyudi, M. Orisa, and N. Vendyansyah, “Penerapan Algoritma K-Nearest Neighbors Pada Klasifikasi Penentuan Gizi Balita (Studi Kasus Di Posyandu Desa Bluto),” *JATI (Jurnal Mhs. Tek. Inform.*, vol. 5, no. 2, pp. 750–757, 2021, doi: 10.36040/jati.v5i2.3738.
- [24] F. Tangguh Admojo and Ahsanawati, “Indonesian Journal of Data and Science Klasifikasi Aroma Alkohol Menggunakan Metode K-NN,” *Indones. J. Data an Sci.*, vol. 1, no. 2, pp. 34–38, 2020.
- [25] Hartati, “Studi literatur: problematika evaluasi pembelajaran dalam mencapai tujuan pendidikan di era merdeka belajar,” *Konf. Ilm. Pendidik. Univ. Pekalongan 2020*, pp. 10–15, 2020, [Online]. Available: <https://proceeding.unikal.ac.id/index.php/kip>