

Prediksi Penyakit Jantung Berdasarkan Indikator-Indikator Kesehatan

Juan Kenny Filemon^{#1}, Wenny Franciska Senjaya^{#2}

[#]Jurusan Teknik Informatika, Universitas Kristen Maranatha
Setraduta Ciumbuleuit No.25, Kota Bandung

¹1972011@maranatha.ac.id

²wenny.fs@it.maranatha.edu

Abstract — This research aims to predict heart disease using Machine Learning. The dataset used is obtained from a website called Kaggle. The benefits of this research can help prevent someone from getting heart disease; and start a lifestyle change. The use of Machine Learning can help to predict the likelihood of a person developing heart disease. Some Machine Learning algorithms used in this research are K-Nearest Neighbor, Decision Tree, and Random Forest. The use of Machine Learning algorithms can also be supported using Hyperparameter tuning to improve the performance of Machine Learning algorithms. Hyperparameter tuning aims to find the best parameters of each algorithm that provide the most optimal performance. The results of the experiments obtained show that the most optimal algorithm is Random Forest with a value of f1-score (micro) 94%, f1-score (macro) 73%, f1-score (weighted) 94%, and balance accuracy 74%. For the best parameters from the use of Hyperparameter tuning from the Random Forest algorithm, namely `max_depth = 80`, `max_features = 4`, `min_samples_leaf = 4`, `min_samples_split = 10`, and `n_estimators = 400`. For the best parameters from the use of Hyperparameter tuning from the Decision Tree algorithm, namely `criterion = gini`, `max_depth = 100`, `max_features = sqrt`, `min_samples_leaf = 1`, and `min_samples_split = 3`, and for the best parameters from the use of Hyperparameter tuning from the K-Nearest Neighbor algorithm, namely `leaf_size = 10`, `n_neighbors = 40`, and `weights = distance`. In addition to experiments to find the best machine learning algorithm, this research also produces a web-based system that accepts input in the form of health indicators and provides prediction results.

Keywords— decision tree, heart disease, k-nearest neighbor, machine learning, random forest.

I. PENDAHULUAN

Penyebab utama kematian di seluruh dunia yang paling fatal adalah penyakit jantung, tercatat pada WHO pada tahun 2006 sejumlah 17,5 juta orang terkena penyakit jantung kardiovaskular. Banyaknya kejadian serangan jantung tiba-tiba dan tanpa ada gejala apapun dapat menyebabkan kematian. Dengan banyaknya kematian akibat serangan jantung secara tiba-tiba maka diperlukan sistem untuk dapat memprediksi penyakit jantung agar dapat membantu dalam pencegahan, diagnosis dini, dan pengelolaan penyakit jantung. Prediksi yang akurat sangat penting untuk memungkinkan pencegahan dini dan perubahan pola hidup [1].

Berbagai algoritma *machine learning* dapat digunakan untuk memprediksi penyakit jantung, seperti *Support Vector Machine*, *Naïve Bayes*, *Random Forest*, *K-Nearest Neighbor*, dan *Decision Tree* [2]. Dalam penelitian ini, algoritma *machine learning* yang digunakan adalah *K-Nearest Neighbor*, *Decision Tree*, dan *Random Forest*.

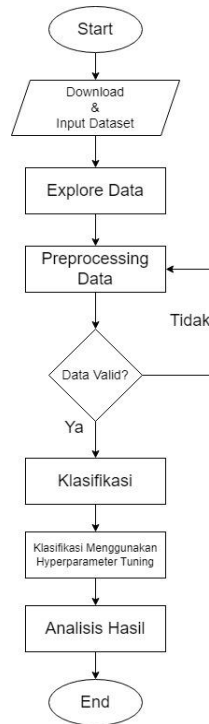
Algoritma *K-Nearest Neighbor* memprediksi penyakit jantung berdasarkan kemiripan data pasien baru dengan data pasien yang telah terdiagnosis [3]. Untuk menangani data dengan berbagai jenis atau data yang tidak lengkap adalah cara kerja untuk algoritma *Decision Tree* [4]. Cara kerja algoritma *Random Forest* adalah dengan menghasilkan simpul anak untuk setiap node secara acak bertujuan untuk meningkatkan akurasi hasil. Algoritma *Random Forest* merupakan pengembangan dari algoritma *Decision Tree* yang menghasilkan kumpulan pohon Keputusan dengan tingkat akurasi lebih tinggi [5].

Dalam melakukan penelitian, untuk mendapatkan hasil kinerja yang paling optimal maka pencarian *outlier* diterapkan, bertujuan untuk meningkatkan hasil kinerja model dan menangani ketidakseimbangan pada data [6]. Setelah pencarian *outlier* diterapkan, langkah berikutnya adalah penerapan *hyperparameter tuning* yang bertujuan untuk menentukan parameter yang paling tepat dan optimal. Proses ini melibatkan penggabungan parameter dari masing-masing algoritma yang digunakan. Setelah menerapkan *hyperparameter tuning* hasil akhir akan berupa sebuah *dashboard* atau situs web. Tujuan dari *dashboard* atau situs web adalah untuk memprediksi kemungkinan persentase seseorang terkena penyakit jantung.

II. METODOLOGI PENELITIAN

A. Flowchart

Berikut Gambar 1 yaitu *flowchart* untuk prediksi penyakit jantung bertujuan untuk mempermudah pemahaman alur penelitian yang akan dikerjakan:



Gambar 1. *Flowchart* untuk prediksi penyakit jantung

Tahapan pertama dalam melakukan penelitian adalah melakukan *download* dan *input dataset* yang akan digunakan untuk memprediksi penyakit jantung. Setelah melakukan tahapan *input dataset*, maka tahapan berikutnya adalah melakukan eksplorasi data. Bertujuan untuk mengetahui dan memahami *dataset* yang akan digunakan, merupakan tahapan penting untuk *machine learning* sebelum melakukan analisis. Setelah melakukan tahapan eksplorasi data, tahapan berikutnya adalah melakukan *preprocessing data*. Tujuan dari melakukan *preprocessing data* adalah untuk *machine learning* dapat mengolah data dengan lancar. Dengan melakukan *preprocessing data* maka dapat menyaring dan memilah data dengan tujuan analisis pada prediksi penyakit jantung.

Tahapan berikutnya adalah setelah data *valid* maka akan melakukan tahapan klasifikasi, namun jika data tidak *valid* maka akan melakukan tahapan *preprocessing* kembali. Dalam tahapan klasifikasi adalah pengujian setiap algoritma yang digunakan agar menghasilkan kinerja yang optimal. Setelah melakukan tahapan klasifikasi maka melakukan tahapan klasifikasi menggunakan *hyperparameter tuning*, bertujuan untuk menentukan parameter yang paling optimal dan menghasilkan kinerja yang optimal. Proses *hyperparameter tuning* melibatkan penggabungan parameter yang tersedia untuk masing-masing algoritma.

Setelah tahapan klasifikasi menggunakan *hyperparameter tuning* diterapkan, maka tahapan berikutnya merupakan analisis hasil. Analisis hasil bertujuan untuk melakukan analisis hasil klasifikasi dan melakukan perbandingan hasil dari algoritma yang digunakan seperti *K-Nearest Neighbor*, *Decision Tree*, dan *Random Forest*. Setelah melakukan perbandingan hasil maka akan mendapatkan hasil kinerja yang paling optimal, bertujuan untuk digunakan dalam memprediksi penyakit jantung.

B. Penyakit Jantung

Sistem peredaran darah yang berperan penting dalam tubuh manusia adalah jantung, ketika kondisi jantung tidak dapat melaksanakan tugasnya dengan baik maka terindikasi sebagai penyakit jantung. Sebanyak 7,3juta penduduk dunia meninggal dikarenakan penyakit jantung berdasarkan data dari *WHO*. Penyakit jantung tidak dapat menular namun merupakan penyakit yang paling mematikan nomor satu di dunia.

Penyakit jantung dapat juga disebut sebagai penyakit jantung koroner, kerusakan berat pada jantung disebabkan oleh darah yang mengalir ke otot jantung terhenti/tersumbat sehingga penyakit jantung dapat terjadi. Penggunaan alkohol, fisik

yang tidak aktif, diet yang tidak sehat, penggunaan tembakau menjadi penyebab utama terkena penyakit jantung. Kemudian resiko terkena penyakit jantung akan bertambah dengan bertambahnya umur, memiliki kolesterol yang tinggi, kelebihan berat badan, dan tekanan darah yang tinggi [7].

C. Machine Learning

Pengembangan sistem yang berfokus untuk kemampuan belajar secara mandiri tanpa harus berulang kali diprogram oleh manusia yaitu *Machine Learning* yang merupakan cabang dari *Artificial Intelligence*. Ketika *machine learning* belajar maka membutuhkan data yang *valid* untuk digunakan ketika proses *training* dan ditujukan untuk *testing* agar dapat menghasilkan *output* yang optimal [8]. Kemudian untuk algoritma *machine learning* terbagi menjadi beberapa algoritma yaitu salah satunya *supervised machine learning*, artinya adalah untuk mengetahui hubungan antara atribut *input* dengan atribut target. Salah satu teknik yang digunakan oleh *supervised machine learning* adalah klasifikasi [9].

D. K-Nearest Neighbor

K-Nearest Neighbor (KNN) adalah suatu algoritma yang menggunakan algoritma *supervised* dimana hasil dari *query instance* yang baru diklasifikasikan berdasarkan mayoritas dari kategori pada KNN. Tujuan dari algoritma KNN adalah untuk mengklasifikasi objek baru berdasarkan atribut dan *training samples*. Dimana hasil dari sampel uji yang baru diklasifikasikan berdasarkan mayoritas dari kategori pada KNN. Langkah-langkah pada algoritma KNN:

1. Tentukan jumlah tetangga (K) yang akan digunakan untuk pertimbangan penentuan kelas.
2. Hitung jarak dari data baru ke masing-masing poin data di *dataset*.
3. Ambil sejumlah K data dengan jarak terdekat, kemudian tentukan kelas dari data baru tersebut.

Untuk mencari dekat atau jauhnya jarak antar titik pada kelas k biasanya dihitung menggunakan jarak *Euclidean*. Jarak *Euclidean* adalah formula untuk mencari jarak antara 2 titik dalam ruang dua dimensi [10].

E. Decision Tree

Decision Tree merupakan salah satu algoritma yang dapat digunakan untuk memprediksi dari data yang sebelumnya sudah tersedia. Berikut langkah-langkah dalam membuat pohon keputusan:

- a. Memilih atribut untuk dijadikan sebagai akar yang berdasarkan dari nilai *gain* tertinggi dari atribut yang sudah ada
- b. Membuat cabang pada tiap-tiap nilai
- c. Membuat setiap kasus pada cabang
- d. Membagi kasus pada cabang
- e. Mengulangi setiap tahapan supaya mendapatkan kelas yang sama dari setiap cabang maupun setiap kasus [11].

F. Random Forest

Random Forest adalah algoritma regresi dan klasifikasi yang menjadi bagian dari kelompok *ensemble learning*. Pengembangan dari *decision tree* merupakan algoritma *random forest* dimana setiap *decision tree* telah melakukan proses pelatihan menggunakan sampel individu. *Random forest* yang akan dihasilkan memiliki banyak *tree* dan setiap *tree* ditanam dengan cara yang sama dan dengan bertambahnya *dataset*, maka *tree* akan ikut berkembang.

Pemilihan atribut pada sebuah *node* akan dipecah dan diambil secara acak, setiap *tree* diberi sampel data pelatihan dengan menggunakan metode *bagging*, dan setiap *tree* dibangun menggunakan metode yang sama bertujuan untuk membangun klasifikasi dan pohon regresi [2].

G. Hyperparameter Tuning

Sebagian besar algoritma *machine learning* memiliki sejumlah parameter yang harus ditentukan sebelum dijalankan, yang disebut sebagai *hyperparameter*. Nilai optimal dari *hyperparameter* ini dapat dicari untuk memaksimalkan kinerja algoritma *machine learning*, melalui proses yang dikenal sebagai *hyperparameter tuning*.

Untuk memilih konfigurasi *hyperparameter* yang sesuai untuk *dataset* tertentu, bisa digunakan nilai *default* yang disediakan oleh paket perangkat lunak atau dengan mengatur secara manual berdasarkan rekomendasi dari literatur, pengalaman atau metode uji coba dan kesalahan [12].

H. Outlier

Dalam proses pengumpulan data ada istilah yang dikenal sebagai *outlier*. Merupakan sesuatu yang terpisah dibandingkan dengan yang lainnya dalam kumpulan data. Hasil dari kesalahan ketika proses koleksi data dapat disebut sebagai *outlier*. Beberapa metode yang dapat digunakan untuk mendeteksi *outlier* yaitu:

- Menggunakan metode *IQR-Score*, yaitu mengukur dispersi statistik yang sama dengan perbedaan antara persentil ke-75 dan ke-25 atau antara kuartil atas dan bawah.
- Menemukan *outlier* dengan cara *box plot* dengan memvisualisasikan data [13].

I. Dashboard Untuk Prediksi Penyakit Jantung

Pembuatan *dashboard* untuk dapat memprediksi penyakit jantung dengan cara pemindahan model kedalam situs web yang bernama *flask*. Bertujuan untuk memprediksi penyakit jantung berdasarkan indikator-indikator kesehatan dan penggunaan algoritma menggunakan hasil kinerja yang paling optimal. Berikut Gambar 2. Tampilan awal untuk *dashboard* memprediksi penyakit jantung:



Gambar 2. Tampilan awal untuk *dashboard* memprediksi penyakit jantung

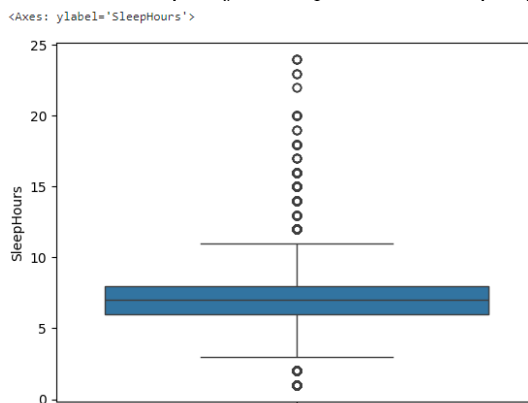
Pada Gambar 2. Tampilan awal untuk *dashboard* memprediksi penyakit jantung merupakan tampilan awal dari *dashboard* yang akan digunakan untuk memprediksi penyakit jantung, berisi indikator-indikator kesehatan seperti jenis kelamin dari responden, memiliki penyakit, dan tombol yang dapat memprediksi persentase terkena penyakit jantung.

III. IMPLEMENTASI

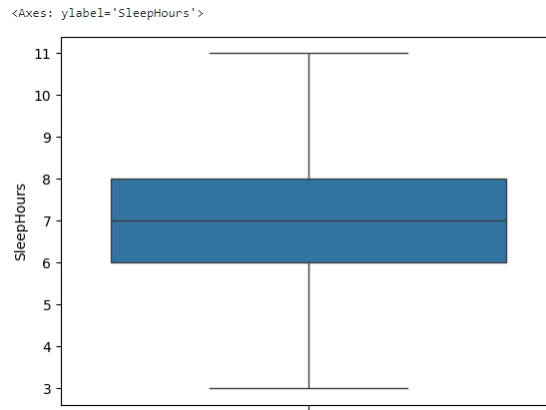
Melakukan *encoding dataset* dan bertujuan untuk *dataset* dapat diproses oleh algoritma *machine learning*. *Dataset* dibagi kedalam 80% untuk *training* dan untuk *testing* sebesar 20%. Melakukan penghapusan untuk *outlier* agar meningkatkan kinerja algoritma pada fitur *SleepHours*, *HeightInMeters*, *WeightInKilograms*, dan *BMI* dengan menggunakan rumus *IQR-Score*.

Berikut merupakan

Gambar 3. Outlier pada fitur *SleepHours* dan tahapan penghapusan *outlier*:

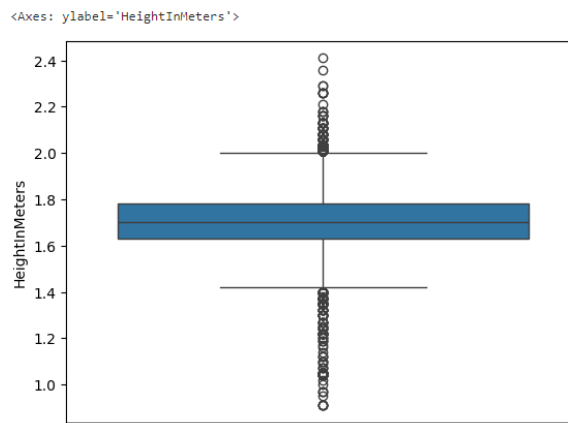


Gambar 3. Outlier pada fitur *SleepHours*

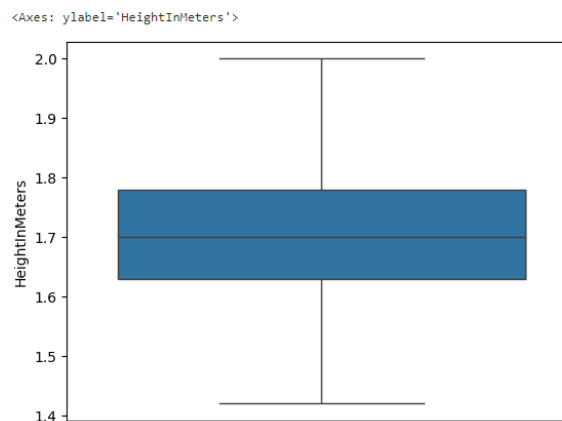


Gambar 4. *Outlier* yang terhapus pada fitur *SleepHours*

Berdasarkan Gambar 3 dan Gambar 4 terdapat *outlier* yang telah di-visualisasikan menggunakan *boxplot* seperti titik yang melebihi batas maksimal dan batas minimum. Setelah melakukan visualisasi untuk mendapatkan *outlier* maka melakukan penghapusan dengan rumus *IQR-Score*. Pada Gambar 4 merupakan tampilan *outlier* yang terhapus. Tahapan berikutnya merupakan penghapusan pada fitur *HeightInMeters*, berikut merupakan Gambar 5. *Outlier* pada fitur *HeightInMeters*:

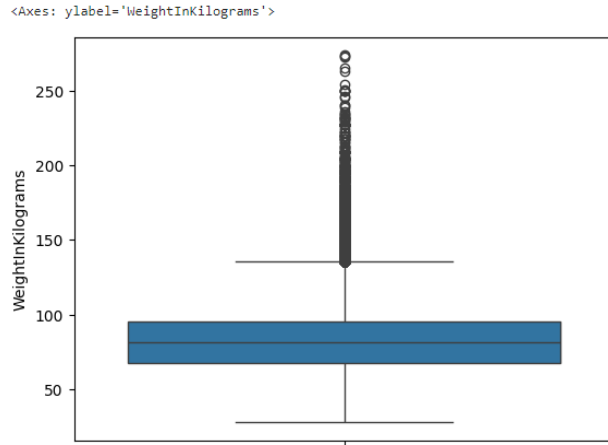


Gambar 5. *Outlier* pada fitur *HeightInMeters*

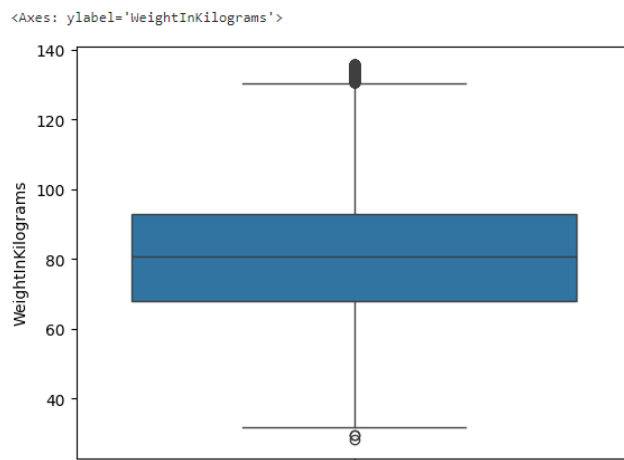


Gambar 6. *Outlier* yang terhapus pada fitur *HeightInMeters*

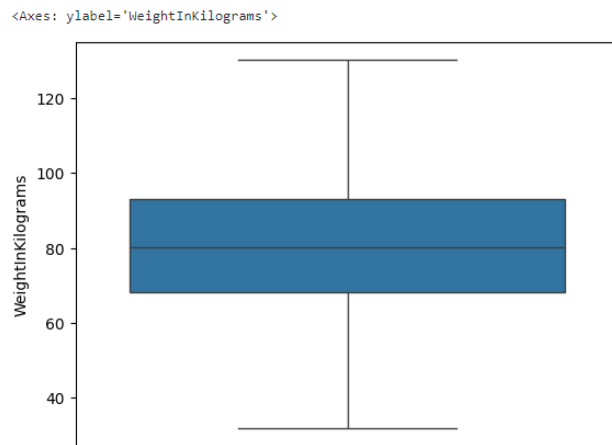
Berdasarkan Gambar 5 dan Gambar 6 terdapat *outlier* yang telah di-visualisasikan menggunakan *boxplot* seperti titik yang melebihi batas maksimal dan batas minimum. Setelah melakukan visualisasi untuk mendapatkan *outlier* maka melakukan penghapusan dengan rumus *IQR-Score*. Pada Gambar 6 merupakan tampilan *outlier* yang terhapus.



Gambar 7. Outlier pada fitur *WeightInKilograms*

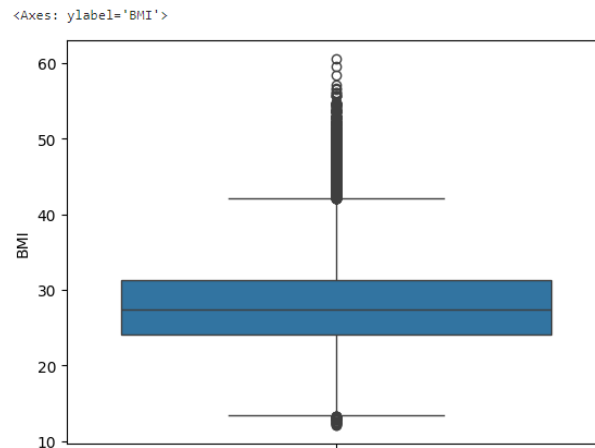


Gambar 8. Tahapan pertama untuk outlier yang terhapus pada fitur *WeightInKilograms*

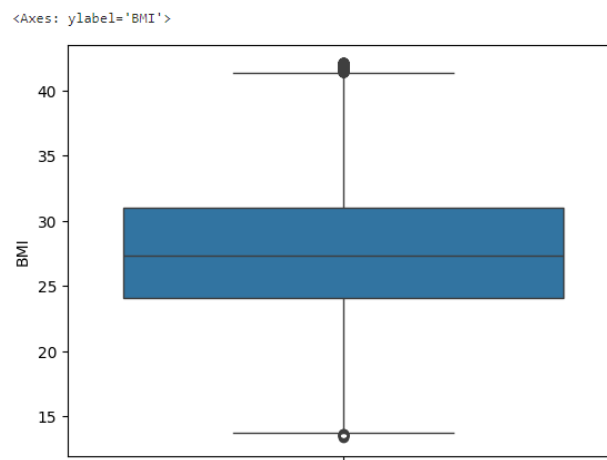


Gambar 9. Tahapan kedua untuk outlier yang terhapus pada fitur *WeightInKilograms*

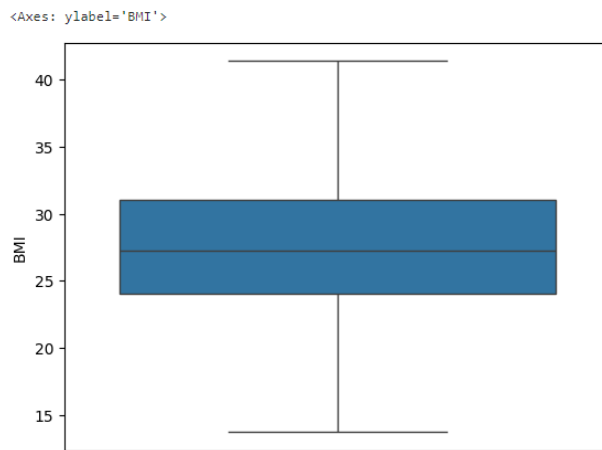
Berdasarkan Gambar 7, Gambar 8, dan Gambar 9 merupakan tahapan penghapusan outlier dengan melakukan visualisasi menggunakan *boxplot*. Untuk Gambar 8 merupakan tahapan pertama penghapusan outlier dan masih terdapat beberapa titik/outlier diatas maksimal dan minimum yang terdeteksi oleh *boxplot*. Setelah tahapan pertama penghapusan outlier, Gambar 9 merupakan tahapan kedua untuk penghapusan outlier pada fitur *WeightInKilograms*.



Gambar 10. *Outlier* pada fitur *BMI*



Gambar 11. Tahapan pertama untuk *outlier* yang terhapus pada fitur *BMI*



Gambar 12. Tahapan kedua untuk *outlier* yang terhapus pada fitur *BMI*

Berdasarkan Gambar 10, Gambar 11, dan Gambar 12 merupakan tahapan penghapusan *outlier* dengan melakukan visualisasi menggunakan *boxplot*. Untuk Gambar 11 merupakan tahapan pertama penghapusan *outlier* dan masih terdapat beberapa titik/*outlier* diatas maksimal dan minimum yang terdeteksi oleh *boxplot*. Setelah tahapan pertama penghapusan *outlier*, Gambar 12 merupakan tahapan kedua untuk penghapusan *outlier* pada fitur *BMI*.

Setelah melakukan penghapusan pada *outlier* maka tahapan berikutnya adalah melakukan *oversampling*. Bertujuan untuk menyeimbangi data yang bersifat *imbalanced*. Penggunaan *oversampling* yaitu dengan cara menambahkan jumlah data pada kelas minoritas sehingga dapat mengimbangi atau mendekati jumlah data pada kelas mayoritas [14]. Untuk fitur yang di-*oversampling* merupakan fitur target yaitu *HadHeartAttack* dan menggunakan metode *RandomOverSampler*.

```
#Data sebelum oversampling
HadHeartAttack
0          173436
1           9703

from imblearn.over_sampling import RandomOverSampler
over = RandomOverSampler()
X_resampled, y_resampled = over.fit_resample(X_train, y_train)

#Data setelah oversampling
0          173436
1          173436
```

Berdasarkan pada kode diatas dapat disimpulkan bahwa data sebelum melakukan *oversampling* untuk kelas minoritas pada fitur target sejumlah 9703, dan setelah melakukan *oversampling* data pada kelas minoritas berjumlah 173436.

Setelah melakukan tahapan *oversampling* maka berikutnya merupakan tahapan penggunaan *hyperparameter tuning*. Penggunaan *hyperparameter tuning* digunakan pada setiap algoritma agar dapat mencari hasil kinerja yang paling optimal dan parameter yang terbaik. Metode yang digunakan untuk *hyperparameter tuning* adalah *GridSearchCV*. Berikut merupakan contoh penggunaan *hyperparameter tuning* pada algoritma *random forest*:

```
parameters = {'n_estimators': [200,400],
              'max_features': [2,4],
              'max_depth': [80,100],
              'min_samples_split': [10,12],
              'min_samples_leaf': [4,8]
             }
grid_obj = GridSearchCV(rf, parameters)
grid_obj = grid_obj.fit(X_resampled, y_resampled)
# Melakukan kombinasi yang terbaik untuk parameter
rf = grid_obj.best_estimator_
# Melakukan pelatihan model
rf.fit(X_resampled,y_resampled)
y_pred = rf.predict(X_test)
```

Berdasarkan penggunaan *hyperparameter tuning*, berikut merupakan parameter terbaik untuk algoritma *machine learning* yang digunakan:

TABEL I
PARAMETER TERBAIK UNTUK ALGORITMA *MACHINE LEARNING* DENGAN MENGGUNAKAN *HYPERPARAMETER TUNING*

<i>K-Nearest Neighbor</i>	<i>Decision Tree</i>	<i>Random Forest</i>
<i>leaf_size = 10,</i> <i>n_neighbors = 40,</i> <i>weights = distance</i>	<i>criterion = gini,</i> <i>max_depth = 100,</i> <i>max_features = sqrt,</i> <i>min_samples_leaf = 1,</i> <i>min_samples_split = 3</i>	<i>max_depth = 80,</i> <i>max_features = 4,</i> <i>min_samples_leaf = 4,</i> <i>min_samples_split = 10,</i> <i>n_estimators = 400</i>

Berdasarkan hasil penelitian, algoritma *random forest* dengan melakukan *oversampling* menghasilkan performa yang paling optimal dalam memprediksi penyakit jantung. Hal ini ditunjukkan dengan nilai *balanced_accuracy* dan *f1-score(micro, macro, dan weighted)* yang paling optimal dibandingkan dengan algoritma yang lainnya. Berikut **Error! Reference source not found.**

TABEL III
PERBANDINGAN HASIL ALGORITMA *MACHINE LEARNING* MENGGUNAKAN *OVERSAMPLING* DAN TIDAK MENGGUNAKAN *OVERSAMPLING*

Algoritma	Oversampling	balanced_accuracy	f1-score(micro)	f1-score(macro)	f1-score(weighted)
<i>K-Nearest Neighbor</i>	Ya	0.69	0.74	0.53	0.81
<i>K-Nearest Neighbor</i>	Tidak	0.50	0.94	0.48	0.92
<i>Decision Tree</i>	Ya	0.60	0.92	0.61	0.92
<i>Decision Tree</i>	Tidak	0.57	0.94	0.60	0.93
<i>Random Forest</i>	Ya	0.74	0.94	0.73	0.94
<i>Random Forest</i>	Tidak	0.54	0.95	0.56	0.93

Berdasarkan **Error! Reference source not found.** dapat disimpulkan bahwa:

- Algoritma *Random Forest* dengan melakukan *oversampling* merupakan hasil kinerja yang paling optimal untuk memprediksi penyakit jantung.
- Tanpa melakukan *oversampling*, algoritma *decision tree* dan *random forest* menghasilkan performa yang setara pada *f1-score(weighted)*.
- Dengan melakukan *oversampling* maka membantu meningkatkan performa prediksi pada *dataset* yangimbang.

Setelah melakukan perbandingan hasil algoritma *machine learning* menggunakan *oversampling* dan tidak menggunakan *oversampling*. Tahapan berikutnya adalah tampilan *dashboard* yang dapat memprediksi penyakit jantung berdasarkan indikator-indikator kesehatan. Berikut merupakan tampilan *dashboard*:

Heart Disease Prediction

Predict the probability of Heart Disease

What is your Gender?

Overall, how is your health condition?

In the last 30 days, how many days were there when you felt your physical health was not good due to sickness or injury?

In the last 30 days, how many days were there when you felt your mental health was not good due to stress, depression or problem with emotions?

How long has it been since you last visited a doctor for a routine checkup?

Besides your regular work, did you engage in any exercise routines like running, bodyweight workouts, golf, gardening, or walking in the last month?

On average, how many hours of sleep do you get in a 24-hour period?

How many of your permanent teeth have been removed due to tooth decay or gum disease(Not including teeth lost for injury or orthodontics)?

Gambar 13. *Dashboard* prediksi penyakit jantung berdasarkan indikator kesehatan

Do you have a history of Angina?

Do you have a history of Stroke?

Do you have a history of Asthma?

Do you have a history of Skin Cancer?

Do you have a history of COPD(Chronic Obstructive Pulmonary Disease)?

Do you have a history of Depressive Disorder?

Do you have a history of Kidney Disease?

Do you have a history of Arthritis?

Do you ever had Diabetes?

Gambar 14. *Dashboard* prediksi penyakit jantung berdasarkan indikator kesehatan

Pada Gambar 13 dan Gambar 14 merupakan *dashboard* untuk memprediksi penyakit jantung berdasarkan indikator kesehatan yang telah terisi untuk fitur masing-masing. Fitur-fitur yang disediakan seperti *gender* responden, kondisi kesehatan responden, dan riwayat penyakit yang dimiliki oleh responden.

Do you have a serious difficulty of hearing or deaf?

Do you have a serious difficulty of seeing, even when wearing glasses?

Do you have a serious difficulty of concentrating, remembering, or making decisions because of mental condition?

Do you have a serious difficulty of walking or climbing stairs?

Do you have a difficulty of dressing or bathing?

Do you have a difficulty doing errands alone such as visiting a doctor office or shopping because of your physical health?

Do you smoke?

Have you ever used e-cigarettes or vapes?

Have you ever had a CT or CAT scan of your chest area?

Gambar 15. *Dashboard* prediksi penyakit jantung berdasarkan indikator kesehatan

How old are you?
Age 65 to 69 ▾

How tall are you (In Meters)? 1.65

How much do you weight (In Kilograms)? 78

Based on your weight and height, your estimated Body Mass Index(BMI) is? 28.7

Do you ever consumed an alcohol in the past 30 days (for adults)?
Yes ▾

Do you ever been tested for HIV (for adults)?
No ▾

Considering the past year, have you had any type of flu vaccine, injected or sprayed?
No ▾

Gambar 16. Dashboard prediksi penyakit jantung berdasarkan indikator kesehatan

Have you ever had a pneumonia shot also known as a pneumococcal vaccine?
No ▾

Have you received a tetanus shot in the past 10 years? Was this Tdap, the tetanus shot that also has pertussis or whooping cough vaccine?
No, did not receive any tetanus shot in the past 10 years ▾

In the past year, have you ever injected any drug, treated for a sexually transmitted disease or received money in exchange for sex?
No ▾

Have you ever tested positive for COVID-19?
Yes ▾

Predict Probability

Gambar 17. Dashboard prediksi penyakit jantung berdasarkan indikator kesehatan

Pada Gambar 15, Gambar 16, dan Gambar 17 merupakan dashboard berisikan indikator kesehatan yang telah terisi seperti fitur umur, berat badan, tinggi badan, dan riwayat penyakit lainnya.

Predict Probability

Probability of Heart Attack is 0.78

Gambar 18 Hasil dashboard prediksi penyakit jantung berdasarkan indikator kesehatan

Pada Gambar 18 merupakan hasil dari responden yang telah mengisi indikator-indikator kesehatan, sebesar 78% responden terkena penyakit jantung.

IV. SIMPULAN DAN SARAN

Berdasarkan penelitian yang telah dibuat dapat disimpulkan dalam pembuatan sistem untuk pemodelan memerlukan kinerja yang optimal maka memerlukan persiapan data, pemilihan algoritma yang tepat berdasarkan hasil kinerja yang paling optimal. Untuk hasil kinerja algoritma *machine learning* yang paling optimal adalah *random forest* dengan nilai *f1-score(micro)* 94%, *f1-score(macro)* 73%, dan *f1-score(weighted)* 94%. Ketika menggunakan *hyperparameter tuning* memerlukan pemahaman antara parameter satu sama lainnya, bertujuan untuk menghasilkan *output* yang maksimal dan menemukan parameter yang optimal.

Adapun beberapa saran yang dapat bermanfaat sebagai acuan dalam melakukan prediksi penyakit jantung yaitu melakukan eksplorasi data mengenai fitur yang berhubungan dengan target agar hasil dapat mendekati optimal berdasarkan fitur yang saling berhubungan dengan tepat, melakukan pembagian *dataset* yang lebih mendalam dengan menggunakan *k-fold*, melakukan eksplorasi data dan *preprocessing data* lebih mendalam dikarenakan terdapat fitur yang banyak dapat mengurangi *machine learning* bekerja dengan optimal.

V. DAFTAR PUSTAKA

- [1] W. N. Santosa and B. , "Penyakit Jantung Koroner dan Antioksidan," *Jurnal Kesehatan dan Kedokteran*, vol. 1, no. 2, p. 95, 2020.
- [2] D. Irawan, E. B. Perkasa, Yurindra, D. Wahyuningsih and E. Helmud, "Perbandingan Klasifikasi SMS Berbasis Support Vector Machine, Naive Bayes Classifier, Random Forest, dan Bagging Classifier," *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 10, no. 3, p. 435, 2021.
- [3] 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," *Jurnal Pseudocode*, vol. 3, no. 2, p. 101, 2016.
- [4] "Decision Tree: Pengertian, Manfaat, Cara Buat & Plus Minusnya," ocbc, 23 Mei 2023. [Online]. Available: <https://www.ocbc.id/article/2023/06/20/decision-tree-adalah>. [Accessed 20 Maret 2024].
- [5] S. A. I. S. Putro, D. I. J. Raharjo, M.T. and S. R. S. M. Ph.D, "Klasifikasi Bobot Sapi Berdasarkan Citra Digital Dengan Metode Fraktal Dan Random Forest," *e-Proceeding of Engineering*, vol. 8, no. 2, p. 1476, 2021.
- [6] Sugriyono and M. U. Siregar, "Prapemrosesan klasifikasi algoritme kNN menggunakan K-means dan matriks jarak untuk dataset hasil studi mahasiswa," *Jurnal Teknologi dan Sistem Komputer*, vol. 8, no. 4, p. 312, 2020.
- [7] T. A. Putri and N. Huda, "Analisis Dan Prediksi Penyakit Jantung Menggunakan Algoritma K-Means Clustering Pada Rumah Sakit Umum Daerah Prabumulih," *Bina Darma Conference on Computer Science*, vol. 2, no. 4, p. 198, 2021.
- [8] I. Cholissodin, S. A. A. Soebroto, U. Hasanah and Y. I. Febiola, *AI, Machine Learning & Deep Learning (Teori & Implementasi)*, Malang: Basic Science to High Scientific Solution for Any Problem, 2020.
- [9] N. L. P. C. Savitri, R. A. Rahman, R. Venyutzky and N. A. Rakhmawati, "Analisis Klasifikasi Sentimen Terhadap Sekolah Daring pada Twitter Menggunakan Supervised Machine Learning," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 7, no. 1, p. 49, 2021.
- [10] U. Jamari, "Penjelasan Cara Kerja Algoritma K-Nearest Neighbor (KNN)," *Laboratorium Dasar Komputasi*, 20 Maret 2022. [Online]. Available: <http://labdas.si.fti.unand.ac.id/2022/03/20/penjelasan-cara-kerja-algoritma-k-nearest-neighbor-knn/>. [Accessed 18 Maret 2024].
- [11] T. A. Q. Putri, A. Triayudi and R. T. Aldisa, "Implementasi Algoritma Decision Tree dan Naive Bayes Untuk Klasifikasi Sentimen Terhadap Kepuasan Pelanggan Starbucks," *Journal of Information System Research*, vol. 4, no. 2, p. 643, 2023.
- [12] I. F. Hawari, M. K. Najib, S. Nurdiati, Y. F. Y. Marpaung, N. Kusumawati, M. Nurfadila, K. R. Sijabat and B. F. Hernawan, "Pengaruh Teknik Oversampling Pada Algoritma Machine Learning Dalam Klasifikasi Body Mass Index (BMI)," *Jurnal Riset dan Aplikasi Matematika*, vol. 8, no. 1, p. 57, 2024.
- [13] D. Sanjaya and S. Budi, "Prediksi Pencapaian Target Kerja Menggunakan Metode Deep Learning dan Data Envelopment Analysis," *Jurnal Teknologi Informatika dan Sistem Informasi*, vol. 6, no. 2, p. 293, 2020.
- [14] S. Choirunnisa, "Metode Hibrida Oversampling Dan Undersampling Untuk Menangani Ketidakseimbangan Data Kegagalan Akademik Universitas XYZ," *Institut Teknologi Sepuluh Nopember, Surabaya*, 2019.