KLASIFIKASI TINGKAT OBESITAS BERDASARKAN BODY MASS INDEX MENGGUNAKAN METODE RANDOM FOREST CLASSIFIER

Disusun oleh:

1. Difa Farhani Hakim (1306620040)
2. Huffaz Muhammad Abdurrofi Baith (1306620075)
3. Muhammad Rizky Anugrah (1306620089)

LAPORAN PROJEK MATAKULIAH PENGATAR PEMBELAJARAN MESIN

PRODI FISIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS NEGERI JAKARTA
MARET 2023

1 Pendahuluan

1.1 Latar Belakang Masalah

1.1.1 Isu/ Masalah

Pasien obesitas berisiko tinggi mengalami penyakit komorbid yang dapat memengaruhi kehidupan sehari-hari secara signifikan hingga meningkatkan resiko kematian. [1] Langkah awal dalam pengidentifikasian pasien bergejala obesitas yaitu dengan menghitung jumlah lemak yang berlebih hingga beresiko terhadap kesehatan. Salah satu estimasi perhitungan lemak dapat dilakukan dengan menghitung nilai rasio berat badan terhadap tinggi badan yang dikuadratkan atau disebut dengan body mass index (BMI). Riset menunjukkan bahwa BMI memiliki korelasi yang kuat atas perhitungan lemak dalam tubuh dan merupakan salah satu cara yang paling mudah untuk mendiagnosa pasien yang memiliki berat badan berlebih. BMI memberikan gambaran yang baik pada setiap tingkat populasi seperti jenis kelamin dan usia. Tetapi, BMI bukanlah perhitungan lemak dalam tubuh yang sempurna dikarenakan adanya faktor berat massa otot dan tulang yang dapat lebih berat dan padat. [2, 3] Maka dari itu, pengidentifikasian gejala awal berdasarkan nilai BMI sangatlah diperlukan untuk mengatasi terjadinya obesitas.

1.1.2 Solusi yang ada

Penelitian sebelumnya telah melakukan klasifikasi tingkat obesitas menggunakan berbagai metode klasifikasi machine learning seperti Decision Tree, Random Forest, k-nearest neighbor (k-NN), Support Vector Machine (SVM) dan lain sebagainya. Adapun penelitian sebelumnya yang pernah dilakukan dalam penyelesaian masalah adalah sebagai berikut.

Penelitian oleh Quiroz (2022) dari Informatics in Medicine Unlocked yang berjudul "Estimation of obesity levels based on dietary habits and condition physical using computational intelligence" bertujuan membuat model klasifikasi tingkat obesitas berdasarkan 16 faktor kebiasaan makan dan kondisi fisik dengan tingkat obesitas sebagai target. Data yang digunakan berasal dari dataset obesitas UCI Machine Learning Repository yang terdiri dari 2.111 orang yang berasal dari negara Kolombia, Meksiko dan Peru dengan rentang usia 14 dan 61 tahun. Algoritma machine learning yang digunakan adalah Light GBM, XGBoost, Decision Tree, Random Forest, Extremely Random Trees, dan Logistic Regression. Hasil terbaik yang diperoleh pada penelitian ini adalah algoritma Light GBM dengan nilai akurasi 97,45%. [4]

Penelitian oleh Ferdowski et al (2021) dari Current Research in Behavioral Sciences yang berjudul "A Machine Learning Approach for Obesity Risk Prediction" bertujuan untuk untuk memprediksi tingkat obesitas dan risiko obesitas menggunakan algoritma pembelajaran mesin. Data yang digunakan terdiri dari 1100 subjek dan berisi 27 fitur berupa informasi kondisi fisika dan gaya hidup. Algoritma machine learning yang digunakan adalah k-nearest neighbor (k-NN), random forest, logistic regression, multilayer perceptron (MLP), support vector machine (SVM), naïve Bayes, adaptive boosting (ADA boosting), decision tree, dan gradient boosting classifier. Hasil terbaik yang diperoleh pada penelitian ini adalah algoritma logistic regression dengan nilai akurasi 97,09%. [5]

Penelitian oleh Rossman et al (2021) dari THE JOURNAL OF PEDIATRICS yang berjudul "Prediction of Childhood Obesity from Nationwide Health Records" bertujuan untuk mengevaluasi pola akselerasi indeks massa tubuh (BMI) pada anak-anak dan untuk mengembangkan model prediksi ditargetkan untuk mengiden-

tifikasi anak-anak yang berisiko tinggi mengalami obesitas sebelum jendela waktu kritis di mana peningkatan terbesar dalam persentil BMI terjadi. Dataset yang digunakan adalah 132.262 rekam medis elektronik kesehatan anak-anak Israel dari tahun 2002 hingga 2018. Algoritma Gradient Boosting dilatih dengan sebagian data dan hasil penelitian menunjukkan teknik gradient boostring tree memperoleh auROC sebesar 0,803 dan auPR sebesar 0,312.[6]

Penelitian oleh Pang et al (2021) dari International Journal of Medical Informatics yang berjudul "Prediction of early childhood obesity with machine learning and electronic health record data" bertujuan untuk membandingkan tujuh model pembelajaran mesin yang dikembangkan untuk memprediksi obesitas anak dari usia lebih dari 2 tahun sampai kurang dari sama dengan 7 tahun. Tujuh teknik pembelajaran mesin dilatih: XG Boost, Decision Tree (DT), Support Vector Machine (SVM), Regresi Logistik (LR), Neural Networks (NN), Gaussian Naive Bayes (GNB) dan Bernoulli Naive Bayes (BNB), dan hasilnya algoritma XGBoost mendapatkan hasil yang terbaik dengan akurasi 64%. [7]

Penelitian oleh Kim & Youm (2020) dari Sustainabilty yang berjudul "Development of a Web Application Based on Human Body Obesity Index and Self-Obesity Diagnosis Model Using the Data Mining Methodology" bertujuan untuk menyajikan sebuah website menggunakan model yang secara akurat mengklasifikasikan obesitas perut atau obesitas otot yang tidak dapat didiagnosa oleh BMI. Data yang digunakan terdiri dari 6413 subjek dan 148 fitur berupa kondisi fisik. Algoritma machine learning yang digunakan adalah k Random Forest, Multiclass Neural Network, Multiclass decision jungle, dan Multiclass logistic regression. Hasil terbaik yang diperoleh pada penelitian ini adalah algoritma Random Forest dan Multiclass decision jungle dengan nilai akurasi 99%. [8]

Penelitian oleh Correa et al (2019) dari Journal of Computer Science yang berjudul "Obesity Level Estimation Software based on Decision Trees" bertujuan untuk membuat software untuk prediksi indeks masa otot dan level obesitas berdasarkan input pengguna. Data yang digunakan berasal dari dataset obesitas UCI Machine Learning Repository. Algoritma machine learning yang digunakan adalah Naïve Bayes, Decision Tree, dan Simple Logistic. Hasil terbaik yang diperoleh pada penelitian ini adalah algoritma Decision Tree dengan nilai akurasi 97,4%.[9]

Penelitian oleh Lee et al (2019) dari Journal of Pediatric Nursing yang berjudul "Risk Factors for Obesity Among Children Aged 24 to 80 months in Korea: A Decision Tree Analysis" bertujuan untuk menguji beberapa faktor risiko obesitas antar generasi di antara anak usia 24 sampai 80 bulan. Database yang digunakan didapat dari Korean National Health Insurance dengan subjek 1.001.775 keluarga. Algoritma Decision Tree digunakan pada penelitian ini dan deskriptif statistik diterapkan untuk memprediksi prevalensi faktor yang terkait dengan obesitas dalam keluarga. Faktor resiko obesitas terbaik yang diprediksi oleh Decision Tree adalah sebagai berikut: ibu gemuk sebelum pembuahan, ayah gemuk, penerima pertolongan non-medis, dan ibu dengan hipertensi selama kehamilan. [10]

1.1.3 Solusi yang terbaik

Dari hasil penelitian sebelumnya diperoleh 3 hasil terbaik sebagai berikut. Dari hasil penelitian yang dilakukan oleh Quiroz di tahun 2022 yang berjudul "Estimation of obesity levels based on dietary habits and condition physical using computational intelligence" memperoleh akurasi yang terbaik menggunakan metode Light Gradient Boost Machine (LGBM) dengan nilai akurasi 97,45%. Selain itu, dari hasil penelitian yang dilakukan

oleh Ferdowski et al di tahun 2021 yang berjudul "A Machine Learning Approach for Obesity Risk Prediction" memperoleh akurasi yang terbaik menggunakan metode logistic regression dengan nilai akurasi 97,09% . Dan, dari hasil penelitian yang dilakukan oleh Kim & Youm di tahun 20220 yang berjudul "Development of a Web Application Based on Human Body Obesity Index and Self-Obesity Diagnosis Model Using the Data Mining Methodology" memperoleh akurasi yang terbaik menggunakan metode Random Forest dan Multiclass logistic regression dengan nilai akurasi 99%.

1.1.4 Solusi yang ditawarkan

Adapun solusi yang ditawarkan dalam penelitian ini dengan menggunakan algoritma Random Forest Classifier dengan Hyperparameter Tuning. Algoritma ini merupakan algoritma yang sangat populer dan kuat dalam pengenalan pola dan pembelajaran mesin untuk klasifikasi dimensi tinggi dan masalah gradient. Selain itu, secara empiris terbukti memiliki keuntungan seperti dapat menghasilkan akurasi yang tinggi untuk banyak kumpulan data, dapat memproses data dengan banyak fitur di mana setiap fitur lemah, yaitu membawa sedikit informasi, mempunyai relatif yang kuat untuk tipe variabel campuran, data yang hilang, outlier dan data noise, dan dapat membangun model relatif cepat. [11] Serta dalam peningkatan kinerja model diperlukan adanya pencarian hyperparameter yang optimal. Beberapa metode hyperparameter tuning diantara yang dapat digunakan pada Random Forest Classifier adalah Grid Search dan Random Search. [12, 13]

1.2 Permasalahan

Adapun permasalahan pada penilitian ini adalah sebagai berikut.

1. Bagaimana hasil preprocessing dan EDA pada data yang digunakan?
2. Bagaimana performa klasifikasi tingkat obesitas berdasarkan body mass index dengan menggunakan metode Random Forest Calssifier?
3. Bagaimana pengaruh variasi parameter pada metode Random Forest Classifier terhadap akurasi klasifikasi tingkat obesitas berdasarkan body mass index?
4. Bagaimana interpretasi hasil klasifikasi tingkat obesitas berdasarkan body mass index dengan menggunakan metode Random Forest Classifier untuk memberikan insight yang lebih baik mengenai kondisi obesitas pada populasi yang diteliti?

1.3 Tujuan Penelitian

Adapun tujuan penelitian adalah sebagai berikut.

1. Membuat model klasifikasi tingkat obesitas pasien berdasarkan body mass index dengan menggunakan metode Random Forest Classifier
2. Menganalisis dataset tingkat obesitas menggunakan pendekatan Exploratory Data Analysis
3. Menganalisis hasil model klasifikasi tingkat obesitas yang telah di hyperparameter tuning

2 Pekerjaan Terkait

Penelitian oleh Quiroz (2022) dari Informatics in Medicine Unlocked yang berjudul "Estimation of obesity levels based on dietary habits and condition physical using computational intelligence" bertujuan membuat model klasifikasi tingkat obesitas berdasarkan 16 faktor kebiasaan makan dan kondisi fisik. Data yang digunakan pada penilitian ini terdiri dari 2.111 orang dari negara Kolombia, Meksiko dan Peru, berusia antara 14 dan 61 tahun. Dataset yang digunakan pada penelitian ini menggunakan dataset UCI repository dengan judul Estimation of obesity levels based on eating habits and physical condition [14] yang terdapat 7 tingkat obesitas. Adapun pada penelitian ini, digunakan enam algoritma untuk melakukan klasifikasi yaitu: Light GBM, XGBoost, Decision Tree, Random Forest, Extremely Random Trees, dan Logistic Regression. Dari enam algoritma tersebut, LigthGBM menjadi algoritma terbaik dengan nilai akurasi 97,45%. Angka ini melampaui nilai akurasi penelitian sebelumnya [9] di mana Decision Tree menjadi algoritma terbaik dengan akurasi 97,4% sedangkan pada penelitian ini, akurasi Decision Tree hanya 85%. [4]

Penelitian oleh Ferdowski et al (2021) dari Current Research in Behavioral Sciences yang berjudul "A Machine Learning Approach for Obesity Risk Prediction" bertujuan untuk untuk memprediksi tingkat obesitas dan risiko obesitas menggunakan algoritma pembelajaran mesin. Penelitian ini menjelaskan tingkat obesitas, resiko pasien obesitas serta penyebab yang melatarbelakangi terjadinya obesitas. Dataset yang digunakan terdiri dari 1100 subjek yang dibagi menjadi 3 kelas obesitas yaitu: High, Medium, dan Low. Dataset berisi 27 fitur yang terdiri dari informasi mengenai tinggi badan, berat badan, umur, beberapa informasi tentang riwayat penyakit, informasi asupan makanan, dan gaya hidup. Dalam penyelesaiannya, penelitian ini menggunakan beberapa algoritma di antaranya: k-nearest neighbor (k-NN), random forest, logistic regression, multilayer perceptron (MLP), support vector machine (SVM), naïve Bayes, adaptive boosting (ADA boosting), decision tree, dan gradient boosting classifier. Setelah menjalani serangkaian uji coba, ratio terbaik untuk data training-data testing yaitu di angka 80%-20%. Untuk evaluasi perfoma, digunakan nilai accuracy, sensitivity, precision, recall, dan F1-score. Hasil penelitian ini menunjukan bahwa algoritma logistic regression mempunyai akurasi tertinggi dengan nilai 97,09% dan algoritma dengan akurasi terendah yaitu gradient boosting dengan akurasi 64,08%. [5]

Pekerjaan oleh Rossman et al (2021) dari THE JOURNAL OF PEDIATRICS yang berjudul "Prediction of Childhood Obesity from Nationwide Health Records" bertujuan untuk mengevaluasi pola akselerasi indeks massa tubuh (BMI) pada anak-anak dan untuk mengembangkan model prediksi ditargetkan untuk mengidentifikasi anak-anak yang berisiko tinggi mengalami obesitas sebelum jendela waktu kritis di mana peningkatan terbesar dalam persentil BMI terjadi. Dataset yang digunakan adalah 132.262 record medis elektronik kesehatan anak-anak Israel dari tahun 2002 hingga 2018. Data tersebut mencakup diagnosis, obat yang diresepkan, tes laboratorium anak dan keluarga, dan data demografis. Model dilatih dengan data dari 2 tahun pertama kehidupan dan risiko obesitas diperkirakan pada 5 dan 6 tahun. Algoritma Gradient Boosting dilatih dengan sebagian data, kualitas model dievaluasi dengan menghitung luas di bawah kurva ROC dan area di bawah kurva PR. Hasil penelitian menunjukkan teknik gradient boostring tree memperoleh auROC sebesar 0,803 dan auPR sebesar 0,312. [6]

Penelitian oleh Pang et al (2021) dari International Journal of Medical Informatics yang berjudul "Prediction of early childhood obesity with machine learning and electronic health record data" bertujuan untuk membandingkan tujuh model pembelajaran mesin yang dikembangkan untuk memprediksi obesitas anak dari

usia lebih dari 2 tahun sampai kurang dari sama dengan 7 tahun. Dataset yang digunakan didapat dari the Pediatric Big Data (PBD) repository, berasal dari EHR di Rumah Sakit Anak Philadel phia (CHOP). Sebanyak 860.510 anak dengan 11.194.579 pertemuan layanan kesehatan. Tujuh teknik pembelajaran mesin dilatih: XG Boost, Decision Tree (DT), Support Vector Machine (SVM), Regresi Logistik (LR), Neural Networks (NN), Gaussian Naive Bayes (GNB) dan Bernoulli Naive Bayes (BNB); untuk memprediksi kejadian obesitas seperti yang didefinisikan oleh Pusat Pengendalian dan Pencegahan Penyakit Philadelphia. Kinerja model dievaluasi menggunakan beberapa metrik klasifikasi: AUC, presisi, skor F1, akurasi, dan spesifisitas, serta perbedaannya antara tujuh model dibandingkan menggunakan uji Q Cochran dan uji berpasangan post-hoc. Dari hasil penelitian diketahui bahwa algoritma XGBoost mendapatkan hasil yang terbaik dengan akurasi 64%. [7]

Penelitian oleh Kim & Youm (2020) dari Sustainabilty yang berjudul "Development of a Web Application Based on Human Body Obesity Index and Self-Obesity Diagnosis Model Using the Data Mining Methodology" bertujuan untuk menyajikan sebuah website menggunakan model yang secara akurat mengklasifikasikan obesitas perut atau obesitas otot yang tidak dapat didiagnosa oleh BMI. Menggunakan model tersebut kalkulator berbasis web yang dibuat dapat memberikan informasi tentang obesitas dengan memprediksi rentang sehat dan tingkat BMI seperti kurus atapun kelebihan berat badan. Data massa muskuloskeletal dan komposisi tubuh diperoleh dari Korean Size 2015. Dataset dibagi menjadi empat kelompok tingkat obesitas dan enam nilai lingkar tubuh digunakan untuk mengklasifikasikan tingkat dan jenis obesitas. Dataset terdiri dari 6413 subjek dengan rentang usia 20-60 tahun dan 148 fitur berupa kondisi fisik. Penelitian ini membagi bagian tubuh menjadi menjadi enam bagian (yaitu, dada, pinggang, pinggul, paha, betis, dan lengan); mengukur panjang dan lebar masing-masing bagian tubuh; menggunakan variabel seperti berat badan, tinggi badan, massa otot rangka, dan massa lemak tubuh untuk memeriksa dan memeriksa apakah prediksi yang dihasilkan kemungkinan besar/ tidak mungkin obesitas, berdasarkan yang dipelajari data, yang mengarah pada evaluasi yang lebih akurat dibandingkan dengan metode BMI tradisional. Pada penelitian ini, digunakan empat algoritma untuk melakukan klasifikasi yaitu: Random Forest, Multiclass Neural Network, Multiclass decision jungle, dan Multiclass logistic regression. Dari empat algoritma tersebut, Random Forest dan Multiclass decision jungle menjadi algoritma terbaik dengan nilai akurasi 99%. [8]

Penelitian oleh Correa et al (2019) dari Journal of Computer Science yang berjudul "Obesity Level Estimation Software based on Decision Trees" bertujuan untuk membuat software untuk prediksi indeks masa otot dan level obesitas berdasarkan input pengguna. Penelitian ini menggunakan dataset obesitas dari UCI Machine Learning Repository [14]. Dataset dibagi menjadi 7 kelas level obesitas dan berisi informasi mengenai umur, gender, tinggi badan, berat badan, serta beberapa informasi tentang asupan makanan dan gaya hidup. Pada penelitian ini, digunakan metodologi SEMMA (Sample, Explore, Modify, Model and Assess) pada pembuatan program serta menggunakan tiga algoritma untuk melakukan klasifikasi yaitu: Naïve Bayes, Decision Tree, dan Simple Logistic. Dari tiga algoritma tersebut diperoleh bahwa Decision Tree menjadi algoritma terbaik dengan nilai akurasi 97,4%. [9]

Penelitian oleh Lee et al (2019) dari Journal of Pediatric Nursing yang berjudul "Risk Factors for Obesity Among Children Aged 24 to 80 months in Korea: A Decision Tree Analysis" bertujuan untuk menguji beberapa faktor risiko obesitas antar generasi di antara anak usia 24 sampai 80 bulan. Database yang digunakan didapat dari Korean National Health Insurance dengan subjek sebanyak 1.001.775 keluarga. Terdapat empat dataset dari database tersebut yaitu qualification, NICHC, NAHC, claims. Data qualification berisi semua informasi tentang orang yang diasuransikan dan tanggungannya. Dataset National Infants and Children Health Checkup

(NICHC) menetapkan target anak usia 4 sampai 80 bulan, dan catatan pemeriksaan kesehatan orang tua mereka diperoleh dari dataset National Adult Health Checkup (NAHC). Data claims berisi kode penyakit menggunakan Korean Standard Classification of Diseases (KCD). Algoritma Decision Tree digunakan pada penelitian ini dan deskriptif statistik diterapkan untuk memprediksi prevalensi faktor yang terkait dengan obesitas dalam keluarga. Hasilnya menunjukkan bahwa prevalensi obesitas adalah 6,57% dan kelebihan berat badan adalah 11,31% di seluruh populasi penelitian Faktor resiko obesitas terbaik yang diprediksi oleh Decision Tree adalah sebagai berikut: ibu gemuk sebelum hamil, ayah gemuk, penerima pertolongan non-medis, dan ibu dengan hipertensi selama kehamilan. [10]

3 Metode yang digunakan

3.1 Dataset

Data yang digunakan merupakan studi penyebab obesitas pada warga di negara Colombia, Mexico dan Peru pada rentang umur 14 hingga 61 tahun. Data tersebut diambil dari UC Irvine Machine Learning Repository. [14] Data tersebut berukuran 2.111 record dengan 17 fitur dan tingkatan obesitas (BMI) sebagai target.

Status obesitas responden pada dataset ini diukur dengan menggunakan indeks massa tubuh (BMI) yang dihitung dari tinggi dan berat badan responden. Responden yang memiliki BMI di atas 30 dianggap mengalami obesitas, sedangkan responden dengan BMI di antara 25 dan 30 dianggap mengalami kelebihan berat badan.

Attribute	Deskripsi
Gender	Jenis kelamin
Age	Umur
Height	Tinggi badan
Weight	Berat badan
FHWO	Riwayat eluarga dengan berat badan berlebih
FAVC	Frekuensi mengonsumsi makanan dengan kalori tinggi
FCVC	Frekuensi memakan sayauran
NCP	Frekuensi jumlah makan besar dalam sehari
CAEC	Mengonsumi makanan di sela waktu makan besar
SMOKE	Frekuensi merokok
CH20	Jumlah konsumsi air harian (Liter)
SCC	Pemantauan Kalori harian
FAF	Frekuensi olahraga per minggu
TUE	Waktu penggunaan perangkat elektronik per hari
CALC	Frekuensi meminum alkohol
MTRANS	Frekuensi menggunakan transportasi
BMI	Body mass index, tingkatan obesitas

Table 1: Deskripsi atribut dataset obesitas

3.2 Algoritma Random Forest Classifier

Random Forest adalah algoritma pembelajaran statistik atau mesin untuk prediksi. Algoritma ini diperkenalkan oleh Leo Breiman pada tahun 2001. Algoritma ini merupakan salah satu jenis algoritma ensemble learning yang terdiri dari banyak pohon keputusan (decision tree) yang dihasilkan dari sampel bootstrap . [15] Random Forest dapat digunakan untuk berbagai tujuan analisis, seperti prediksi default kartu kredit, prediksi jumlah saham artikel online, prediksi konsentrasi serbuk sari ragweed berdasarkan data historis dan 85 variabel predictor, atau analisis driver yang mengidentifikasi variabel-variabel yang berpengaruh terhadap kepuasan pelanggan. [16] Untuk menggunakan metode Random Forest, kita perlu menentukan ukuran hutan (jumlah pohon) dan ukuran sampel X (jumlah variabel yang dipilih secara acak untuk setiap pohon). Random Forest juga dapat digunakan untuk menentukan variabel penting dalam suatu model. Variabel penting adalah variabel yang memiliki pengaruh besar terhadap hasil prediksi. [[17]

Random Forest Classifier merupakan salah satu algoritma yang paling sukses dalam teknik pembelajaran mesin yang telah terbukti kebenarannya. Algoritma ini merupakan algoritma yang sangat populer dan kuat dalam pengenalan pola dan pembelajaran mesin untuk klasifikasi dimensi tinggi dan masalah gradient. Algortima Random Forest juga secara empiris terbukti memiliki keuntungan seperti dapat menghasilkan akurasi

yang tinggi untuk banyak kumpulan data, dapat memproses data dengan banyak fitur di mana setiap fitur lemah, yaitu membawa sedikit informasi, mempunyai relatif yang kuat untuk tipe variabel campuran, data yang hilang, outlier dan data noise, dan dapat membangun model relatif cepat (lebih cepat daripada mengantongi dan meningkatkan). [11] Akan tetapi kelemahan yang terkait dengan pohon pengklasifikasi adalah variansnya yang tinggi. Dalam praktiknya, tidak jarang terjadi perubahan kecil pada kumpulan data pelatihan pohon yang sangat berbeda. Alasannya terletak pada sifat hierarki dari pengklasifikasi pohon. Kesalahan yang terjadi di node dekat dengan akar pohon merambat sampai ke daun. Untuk membuat klasifikasi pohon lebih stabil, Metodologi decision tree telah ditemukan. Metodologinya diusulkan oleh Ho [18], Amit dan Geman [19] dan kemudian oleh Breiman [11], dalam bentuk terintegrasi (sebagai "Random Forest"). Decision Tree adalah ensemble pohon keputusan yang bisa dilihat sebagai salah satu classifier yang berisi beberapa metode klasifikasi atau satu metode tetapi berbagai parameter kerja.

3.3 Hyperparameter Tuning

Hyperparameter tuning pada Random Forest Classifier adalah proses untuk memilih kumpulan hyperparameter yang sesuai untuk algoritma pembelajaran. Hyperparameter adalah nilai untuk parameter yang digunakan untuk mempengaruhi proses pembelajaran. [13] Tujuannya adalah untuk meningkatkan kinerja model dengan menemukan kombinasi hyperparameter yang optimal. [12]

Ada beberapa metode yang dapat digunakan untuk melakukan hyperparameter tuning pada Random Forest Classifier. Beberapa di antaranya adalah:

1. Grid Search: Metode ini mencoba semua kombinasi dari nilai-nilai hyperparameter yang telah ditentukan sebelumnya dan memilih kombinasi yang menghasilkan performa terbaik
2. Random Search: Metode ini mencoba kombinasi acak dari nilai-nilai hyperparameter yang telah ditentukan sebelumnya dan memilih kombinasi yang menghasilkan performa terbaik

Pada modul library scikit-learn versi 1.2.2 terdapat hyperparameter yang pada Random Forest Classifier sebagai berikut. [20]

1. n_estimators: Jumlah pohon dalam hutan.
2. criterion: Fungsi untuk mengukur kualitas split.
3. max_depth: Kedalaman maksimum pohon.
4. min_samples_split: Jumlah sampel minimum yang diperlukan untuk membagi node internal.
5. min_samples_leaf: Jumlah sampel minimum yang diperlukan untuk menjadi node daun.
6. min_weight_fraction_leaf: Fraksi bobot minimum dari jumlah total bobot (dari semua sampel input) yang diperlukan untuk menjadi node daun.
7. max_features: Jumlah fitur yang dipertimbangkan saat mencari split terbaik.
8. bootstrap: Apakah sampel diambil dengan penggantian atau tidak saat membangun setiap pohon.
9. oob_score: Apakah menggunakan out-of-bag samples untuk memperkirakan akurasi generalisasi.

10. n_jobs: Jumlah pekerjaan yang dijalankan secara paralel untuk fit dan prediksi.
11. random_state: Seed yang digunakan oleh generator bilangan acak.
12. verbose: Tingkat kebisingan saat fit dan prediksi.
13. warm_start: Apakah menggunakan solusi sebelumnya untuk menyesuaikan dan menambahkan pohon dalam hutan.
14. class_weight: Bobot yang dikaitkan dengan kelas.
15. ccp_alpha: Parameter kompleksitas untuk pruning minimal biaya-kompleksitas.
16. max_samples: Jumlah sampel yang diambil dari X untuk melatih setiap pohon dasar.

4 Eksperimen

4.1 Data Preparation

Preprocessing yang dilakukan adalah pengecekkan data, penampilan statistik data, dan transformasi data kategorik menjadi numerik. Sebelum melakukan preprocessing, data ditampilkan terlebih dahulu melalui fungsi data.head() dan data.info()

Figure 1: Top 5 dataset

	Gender	Age	Height	Weight	FHWO	FAVC	FCVC	NCP	CAEC	SMOKE	CH2O	SCC	FAF	TUE	CALC	MTRANS	BMI
0	Female	21.0	1.62	64.0	yes	no	2.0	3.0	Sometimes	no	2.0	no	0.0	1.0	no	Public_Transportation	Normal_Weight
1	Female	21.0	1.52	56.0	yes	no	3.0	3.0	Sometimes	yes	3.0	yes	3.0	0.0	Sometimes	Public_Transportation	Normal_Weight
2	Male	23.0	1.80	77.0	yes	no	2.0	3.0	Sometimes	no	2.0	no	2.0	1.0	Frequently	Public_Transportation	Normal_Weight
3	Male	27.0	1.80	87.0	no	no	3.0	3.0	Sometimes	no	2.0	no	2.0	0.0	Frequently	Walking	Overweight_Level_I
4	Male	22.0	1.78	89.8	no	no	2.0	1.0	Sometimes	no	2.0	no	0.0	0.0	Sometimes	Public_Transportation	Overweight_Level_II

Figure 2: Info dataset

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2111 entries, 0 to 2110
Data columns (total 17 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Gender  2111 non-null   object
 1   Age     2111 non-null   float64
 2   Height  2111 non-null   float64
 3   Weight  2111 non-null   float64
 4   FHWO    2111 non-null   object
 5   FAVC    2111 non-null   object
 6   FCVC    2111 non-null   float64
 7   NCP     2111 non-null   float64
 8   CAEC    2111 non-null   object
 9   SMOKE   2111 non-null   object
 10  CH2O    2111 non-null   float64
 11  SCC     2111 non-null   object
 12  FAF     2111 non-null   float64
 13  TUE     2111 non-null   float64
 14  CALC    2111 non-null   object
 15  MTRANS  2111 non-null   object
 16  BMI     2111 non-null   object
dtypes: float64(8), object(9)
memory usage: 280.5+ KB
```

Dari data tersebut diperoleh bahwa setiap atribut memiliki ukuran data yang sama yaitu 2111 dengan tidak ada missing value. Data tersebut terdiri dari data numerik berjumlah 8 atribut dan data kategorik berjumlah 9 atribut. Data numerik antara lain: Age, Height, Weight, FCVC, NCP, CH20, FAF, dan TUE. Sedangkan, data kategorik antara lain: Gender, FHWO, FAVC, CAEC, SMOKE, SCC, CALC, MTRANS, dan BMI.

Figure 3: Nilai statistik dari dataset

	Age	Height	Weight	FCVC	NCP	CH2O	FAF	TUE
count	2111.000000	2111.000000	2111.000000	2111.000000	2111.000000	2111.000000	2111.000000	2111.000000
mean	24.312600	1.701677	86.586058	2.419043	2.685628	2.008011	1.010298	0.657866
std	6.345968	0.093305	26.191172	0.533927	0.778039	0.612953	0.850592	0.608927
min	14.000000	1.450000	39.000000	1.000000	1.000000	1.000000	0.000000	0.000000
25%	19.947192	1.630000	65.473343	2.000000	2.658738	1.584812	0.124505	0.000000
50%	22.777890	1.700499	83.000000	2.385502	3.000000	2.000000	1.000000	0.625350
75%	26.000000	1.768464	107.430682	3.000000	3.000000	2.477420	1.666678	1.000000
max	61.000000	1.980000	173.000000	3.000000	4.000000	3.000000	3.000000	2.000000

Selanjutnya penampilan nilai-nilai statistik dapat dilakukan melalui fungsi data.describe(). Diperoleh bahwa

tidak terdapat data numerik yang inkonsisten.

Langkah berikutnya, dilakukan pengecekan unique value pada setiap data kategorik. Diperoleh bahwa tidak terdapat adanya inkonsisten pengisian data. Pada target diperoleh bahwa kategori tertinggi ada pada obesitas tipe 1 berjumlah 351 orang, sedangkan kategori terendah ada pada kekurangan berat badan berjumlah 272 orang.

```
0 Value Unique Gender:
 Male      1068
Female     1043
Name: Gender, dtype: int64

1 Value Unique FHWO:
 yes     1726
no       385
Name: FHWO, dtype: int64

2 Value Unique FAVC:
 yes     1866
no       245
Name: FAVC, dtype: int64

3 Value Unique CAEC:
 Sometimes     1765
Frequently      242
Always           53
no               51
Name: CAEC, dtype: int64

4 Value Unique SMOKE:
 no      2067
yes       44
Name: SMOKE, dtype: int64
```

```
5 Value Unique SCC:
 no      2015
yes       96
Name: SCC, dtype: int64

6 Value Unique CALC:
 Sometimes     1401
no              639
Frequently       70
Always            1
Name: CALC, dtype: int64

7 Value Unique MTRANS:
 Public_Transportation    1580
Automobile                457
Walking                    56
Motorbike                  11
Bike                        7
Name: MTRANS, dtype: int64

8 Value Unique BMI:
 Obesity_Type_I         351
Obesity_Type_III        324
Obesity_Type_II         297
Overweight_Level_I      290
Overweight_Level_II     290
Normal_Weight           287
Insufficient_Weight     272
Name: BMI, dtype: int64
```

Figure 4: Unique value dari atribut category

4.2 Data Transformation

Sebelum melakukan analisis data, diperlukan transformasi data kategorik menjadi data numerik menggunakan fungsi OrdinalEncoder() dan encoder secara manual pada target menggunakan fungsi .replace().

Pada target dilakukan transformasi berupa Insufficient_Weight : 0, Normal_Weight: 1, Overweight_Level_I: 2, Overweight_Level_II: 3, Obesity_Type_I: 4, Obesity_Type_II: 5,Obesity_Type_III: 6.

Figure 5: Hasil transformasi data category

	Gender	Age	Height	Weight	FHWO	FAVC	FCVC	NCP	CAEC	SMOKE	CH2O	SCC	FAF	TUE	CALC	MTRANS	BMI
0	0.0	21.000000	1.620000	64.000000	1.0	0.0	2.0	3.0	2.0	0.0	2.000000	0.0	0.000000	1.000000	3.0	3.0	1
1	0.0	21.000000	1.520000	56.000000	1.0	0.0	3.0	3.0	2.0	1.0	3.000000	1.0	3.000000	0.000000	2.0	3.0	1
2	1.0	23.000000	1.800000	77.000000	1.0	0.0	2.0	3.0	2.0	0.0	2.000000	0.0	2.000000	1.000000	1.0	3.0	1
3	1.0	27.000000	1.800000	87.000000	0.0	0.0	3.0	3.0	2.0	0.0	2.000000	0.0	2.000000	0.000000	1.0	4.0	2
4	1.0	22.000000	1.780000	89.800000	0.0	0.0	2.0	1.0	2.0	0.0	2.000000	0.0	0.000000	0.000000	2.0	3.0	3
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2106	0.0	20.976842	1.710730	131.408528	1.0	1.0	3.0	3.0	2.0	0.0	1.728139	0.0	1.676269	0.906247	2.0	3.0	6
2107	0.0	21.982942	1.748584	133.742943	1.0	1.0	3.0	3.0	2.0	0.0	2.005130	0.0	1.341390	0.599270	2.0	3.0	6
2108	0.0	22.524036	1.752206	133.689352	1.0	1.0	3.0	3.0	2.0	0.0	2.054193	0.0	1.414209	0.646288	2.0	3.0	6
2109	0.0	24.361936	1.739450	133.346641	1.0	1.0	3.0	3.0	2.0	0.0	2.852339	0.0	1.139107	0.586035	2.0	3.0	6
2110	0.0	23.664709	1.738836	133.472641	1.0	1.0	3.0	3.0	2.0	0.0	2.863513	0.0	1.026452	0.714137	2.0	3.0	6

2111 rows × 17 columns

Selanjutnya dilakukan, plot distribusi data BMI melalui bar plot dan pie chart diperoleh bahwa data BMI tidak balance pada setiap kategorinya. Data tertinggi ada pada tingkat obesitas tipe 1 diikuti dengan tipe 3 dan sisanya berjumlah tergolong sama dengan pasien kekurangan berat badan terendah. Sehingga, diperlukan data balancing terlebih dahulu menyesuaikan dengan data yang terendah. Hasil ukuran data yang diperoleh sebesar 1904 record.

Figure 6: Bar Distribusi Data BMI

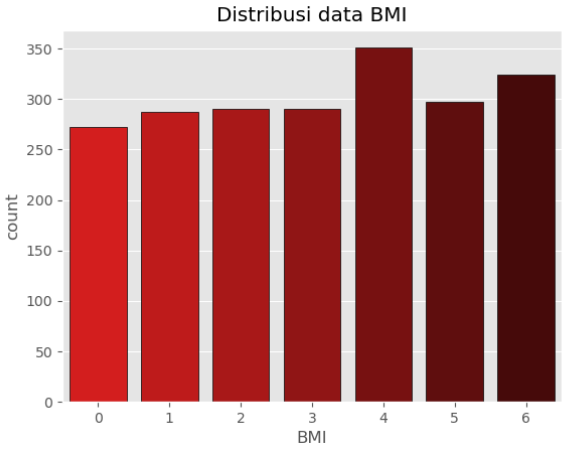


Figure 7: Pie Distribusi Data BMI

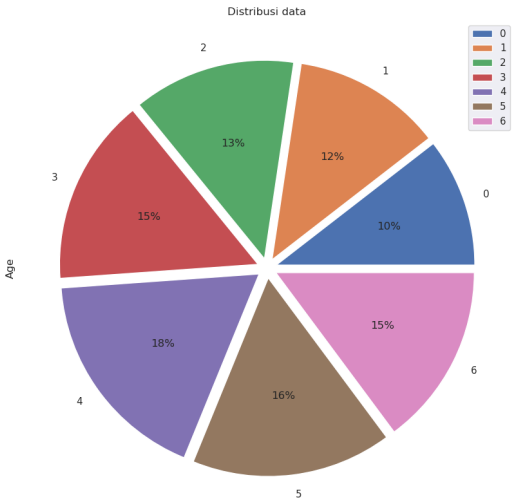
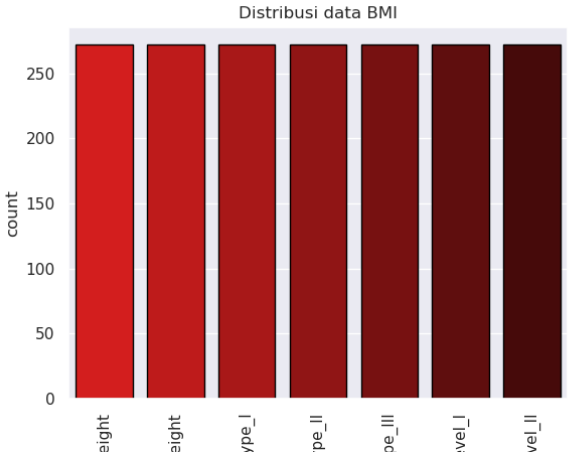


Figure 8: Bar Distribusi data BMI Balance



Terakhir setelah dilakukan balancing data. Data kategorik diubah menjadi numerik menggunakan fungsi pandas get_dummies().

Figure 9: Data Kategorik Dummy

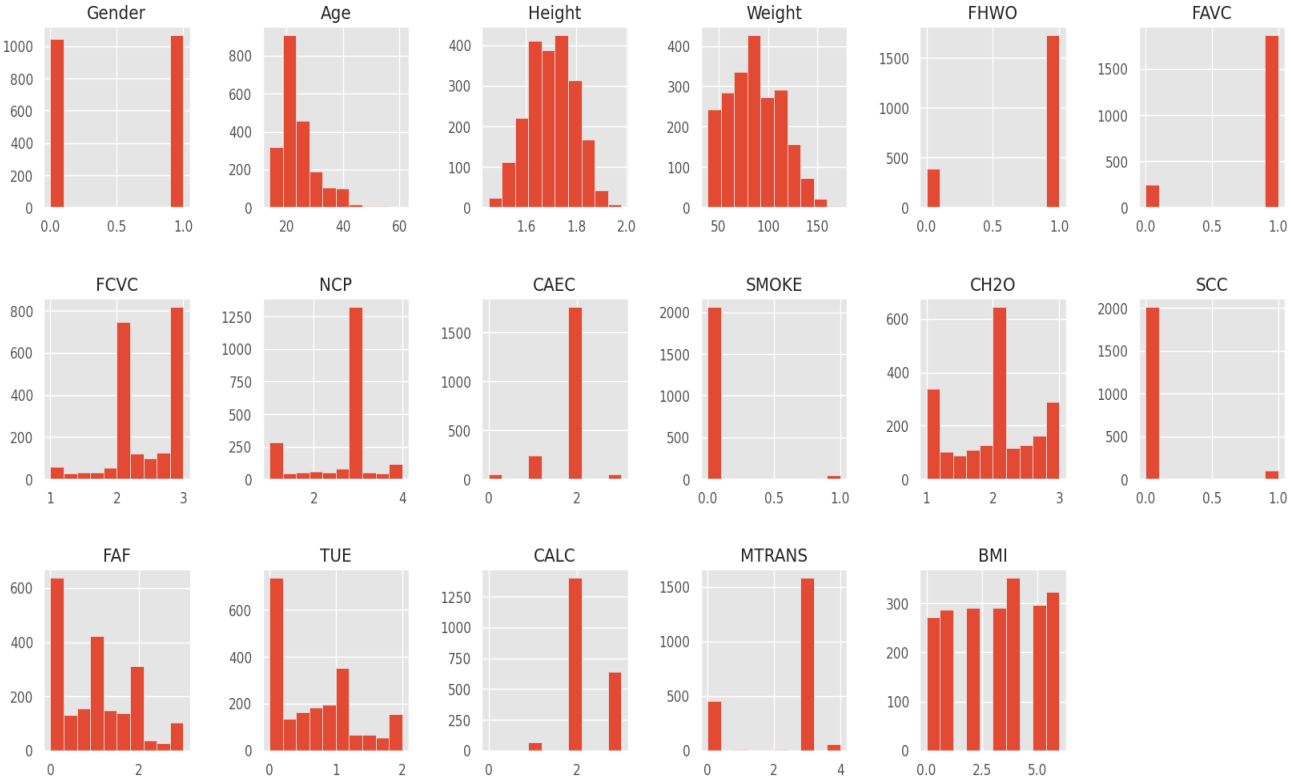
	Age	Height	Weight	FCVC	CH2O	FAF	TUE	BMI	FHWO_no	FHWO_yes	FAVC_no	FAVC_yes
0	21.310907	1.720640	50.000000	2.919584	1.147121	0.993058	0.089220	0	0	1	0	1
1	23.000000	1.717601	51.073918	2.000000	1.426874	2.202080	2.000000	0	0	1	0	1
2	17.067130	1.896734	59.895052	2.842102	2.000000	2.511157	0.560351	0	0	1	0	1
3	22.000000	1.670000	50.000000	3.000000	3.000000	2.000000	1.000000	0	0	1	1	0
4	17.210933	1.819557	58.325122	2.559600	2.000000	2.000000	0.331483	0	0	1	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...
267	36.310292	1.701397	83.000000	2.000000	2.000000	1.472172	0.000000	3	0	1	0	1
268	24.068940	1.706912	85.743727	2.000000	2.747302	0.000000	0.799756	3	0	1	0	1
269	20.000000	1.650000	75.000000	3.000000	2.000000	1.000000	1.000000	3	0	1	0	1
270	37.000000	1.680000	83.000000	2.000000	2.000000	0.000000	0.000000	3	0	1	0	1
271	33.000000	1.850000	93.000000	2.000000	1.000000	1.000000	1.000000	3	0	1	0	1

1904 rows × 22 columns

4.3 Exploratory Data Analysis (EDA)

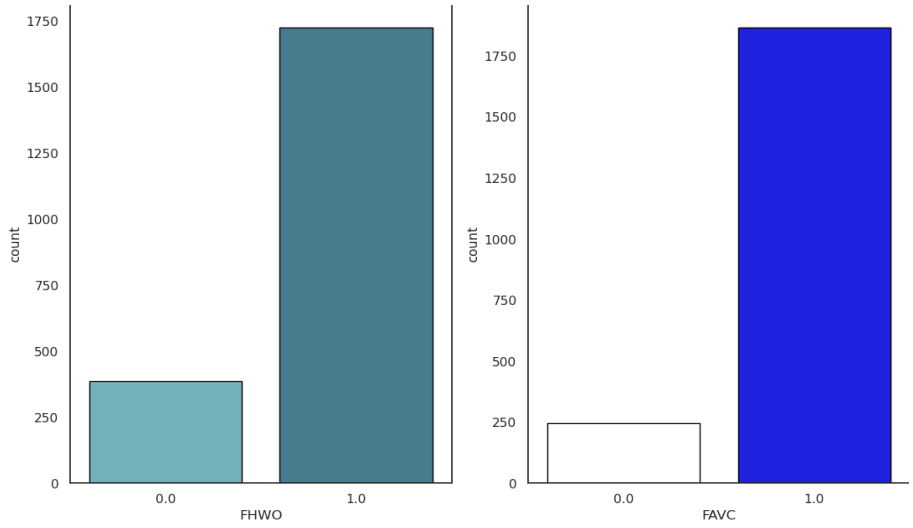
4.3.1 Univariate Analysis

Figure 10: Frekuensi atribut data



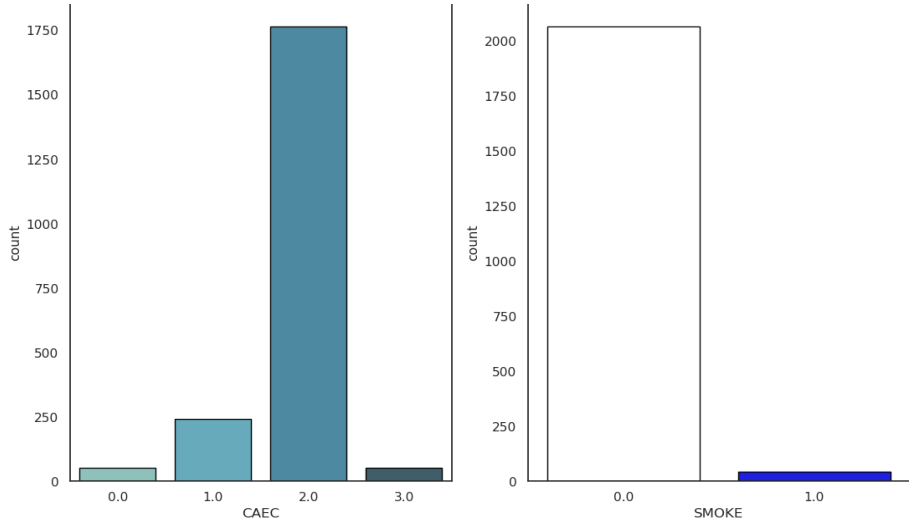
Berikut adalah analisis frekuensi dari atribut FHWO, FAVC, CAEC, SMOKE, SCC, CALC, dan MTRANS. Pada plot frekuensi FHWO diperoleh bahwa nilai pasien yang memiliki keluarga dengan berat badan berlebih sangat signfikan 1726 banding 385. Pada plot frekuensi FAVC, Nilai pasien yang mengonsumsi makanan dengan kalori tinggi juga sangat signifikan 1866 banding 245.

Figure 11: Frekuensi atribut FHWO dan FAVC



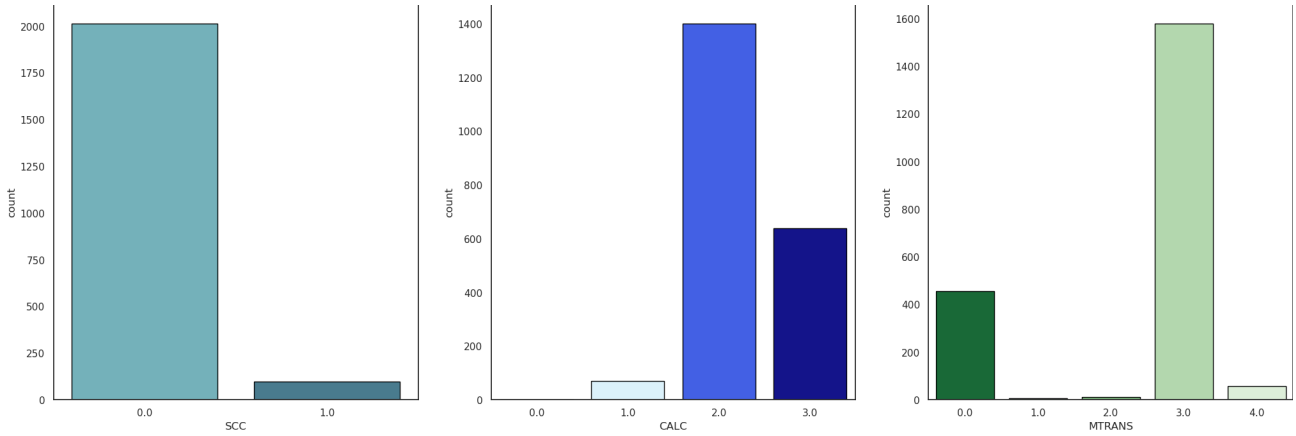
Pada plot frekuensi CAEC, nilai pasien yang makan di sela waktu makan besar dengan kategori kadang-kadang (1765 pasien) sangatlah signfikan diikuti dengan sering (242 pasien), selalu (53 pasien) dan tidak (51 pasien). Pada plot frekuensi SMOKE, pasien banyak yang tidak merokok dibanding yang merokok.

Figure 12: Frekuensi atribut CAEC dan SMOKE



Pada plot frekuensi SCC, nilai pasien yang memonitori jumlah kalori setiap harinya sangatlah rendah. Pada plot frekuensi CALC, nilai pasien dengan frekuensi meminum alkohol kadang-kadang (1401 pasien) paling tinggi diikuti dengan tidak meminum alkohol (639 pasien) dan sisanya tergolong sering dan selalu yang berjumlah sangat dikit. Pada plot frekuensi MTRANS, nilai pasien tertinggi ada pada yang menggunakan transportasi umum (1580 pasien) yang diikuti dengan pasien menggunakan mobil (457 pasien) dan menggunakan sepeda, sepeda motor, dan berjalan kaki berjumlah sedikit.

Figure 13: Frekuensi atribut SCC, CALC, dan MTRANS

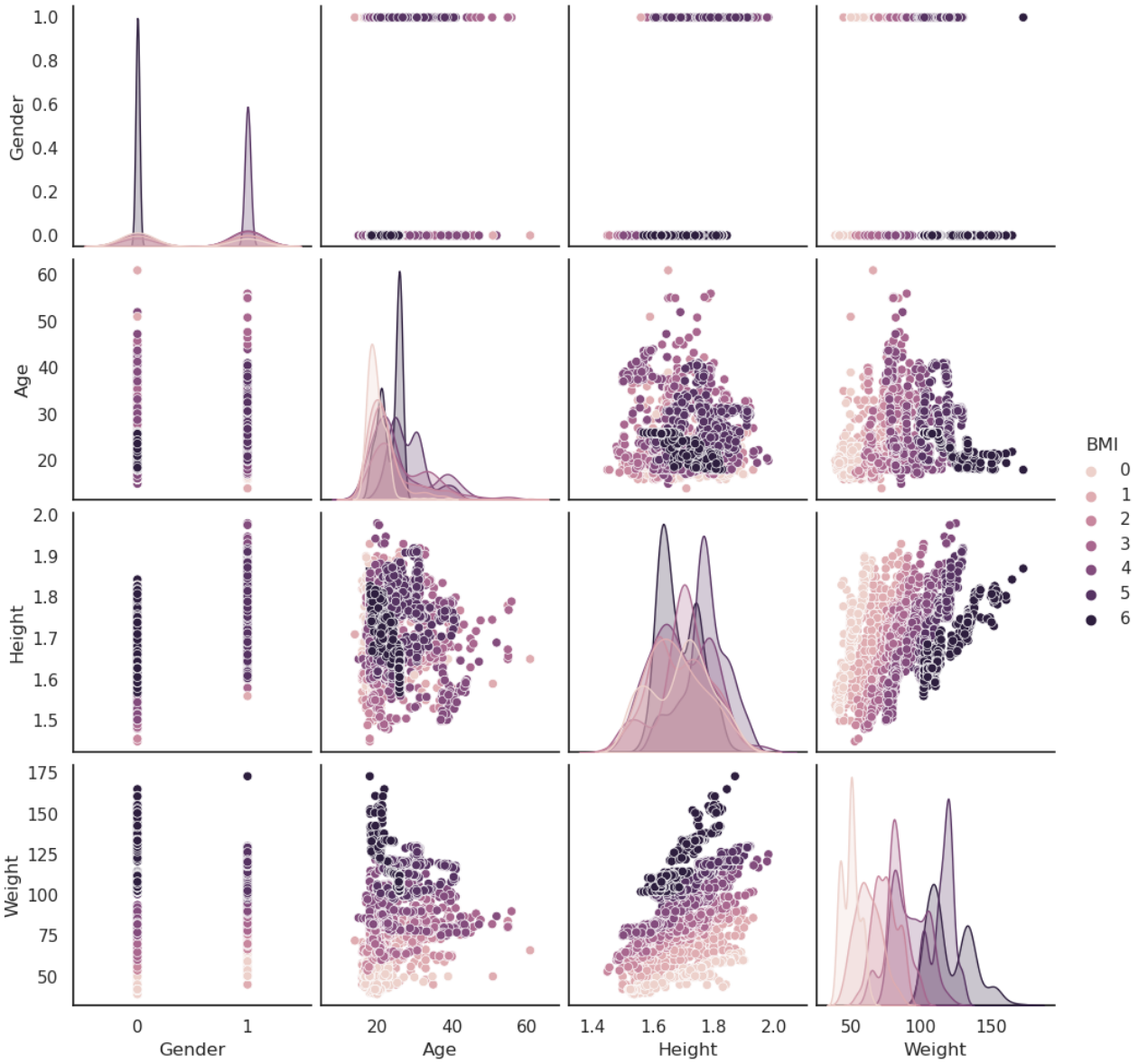


4.3.2 Bivariate Analysis

Hasil pairplot antara atribut gender, age, height, dan weight bagian count histogram diperoleh bahwa jenis kelamin perempuan memiliki jumlah BMI tingkat 6 (obesitas tipe 3) yang paling besar diantara tingkatan BMI lainnya, sedangkan pada jenis kelamin laki-laki memiliki jumlah BMI tingkat 5 yang paling besar. Pada count histogram age diperoleh bahwa pasien dengan BMI tingkat 6 memiliki puncak pada rentang 30 hingga 35 tahun. Sedangkan tingkatan lainnya ada pada rentang 20 hingga 30 tahun. Pada count histogram height, BMI tingkat 6 berada pada rentang tinggi 1.6 hingga 1.8 m sedangkan pada rentang BMI tingkat 5 berada pada rentang 1.7 hingga 1.9 m. Pada count histogram weight, BMI tingkat 6 berada pada pasien dengan berat badan rentang 100 hingga 150 ke atas dengan BMI tingkat 5 memiliki histogram yang sama.

Dilihat hubungan antara age dan weight, pasien yang berumur 20 memiliki BMI tingkat 6 yang terbanyak dengan rentang berat badan 125 - 175 kg. Sedangkan jika dilihat dari hubungan antara height dan weight, pasien yang memiliki berat badan diatas 100 cenderung tergolong pada BMI tingkat 4, 5 dan 6 dengan rentang tinggi badan dimulai dari 1.6 m.

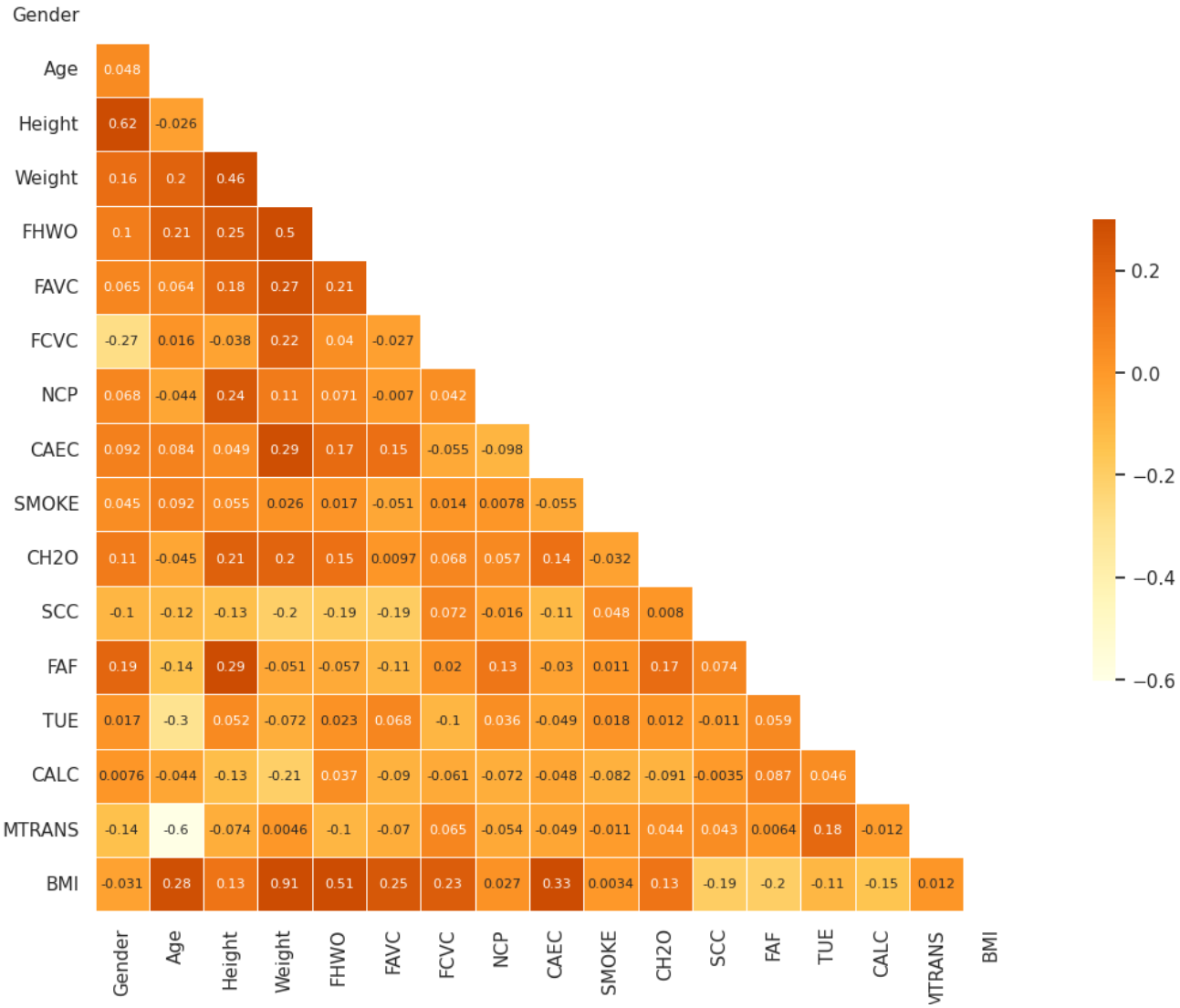
Figure 14: Pairplot Gender, Age, Height, dan Weight



4.3.3 Korelasi

Hasil plotting yang telah dibuat menampilkan sebuah heatmap untuk visualisasi korelasi antar variabel dari sebuah dataset. Korelasi menggambarkan seberapa kuat hubungan antara dua variabel, di mana nilai korelasi berkisar dari -1 hingga 1. Semakin dekat nilai korelasi ke -1 atau 1, semakin kuat hubungan antar variabel, sedangkan semakin dekat ke 0, semakin lemah atau tidak ada hubungan.

Figure 15: Korelasi atribut data BMI



Pada heatmap tersebut, nilai korelasi antara dua variabel dapat dibaca pada persimpangan antara dua variabel di heatmap. Jika dilihat dari heatmap, terdapat beberapa korelasi yang menonjol antara variabel-variabel tersebut. Misalnya, korelasi antara BMI dan Weight memiliki nilai korelasi yang tinggi (0.91) dan positif, yang berarti semakin berat seseorang, semakin besar kemungkinan bahwa mereka memiliki BMI yang lebih tinggi.

Beberapa korelasi lainnya yang menonjol antara variabel-variabel tersebut adalah antara FHWO dan Weight (0.5), Gender dan Height (0.62). Sebaliknya, beberapa korelasi yang tidak begitu signifikan antara variabel-variabel tersebut adalah antara SMOKE dan Age (0.092), CAEC dan Height (0.055), dan NCP dan DAVC (-0.007).

Selain itu, dari hasil plot korelasi antar atribut terhadap target diperoleh bahwa atribut yang berkorelasi

positif tinggi dengan BMI antara lain: Weight, FHWO, CAEC, Age, FAVC, FCVC, dan CH20. Sisanya berada pada korelasi lemah dengan SCC dan FAF terendah berada pada korelasi -0.19 dan -0.2. Selain itu atribut lainnya yang berkorelasi positif tinggi adalah Height dengan Gender (0.62), FHWO dengan Weight (0.5) dan Weight dengan Height (0.46). Serta atribut lainnya yang berkolerasi negatif lemah adalah FCVC dengan Gender (-0.27), TUE dengan Age (-0.3), dan CALC dengan Weight (-0.21). Dan terdapat juga atribut yang sangat mendekati 0 seperti Gender (-0.031), NCP (0.027), SMOKE (0.0034), dan MTRANS (0.012). Sehingga, atribut tersebut akan di drop atau tidak digunakan dalam training.

4.4 Training

Dataset dibagi untuk training dan testing dengan train_size = 0,8. Dilakukan dua percobaan pada penelitian ini yaitu menggunakan model Random Forest fitting biasa dan model Random Forest dengan tuning hyperparameter.

4.4.1 Fitting

Fitting dimulai dengan menginput dataset train ke dalam model RandomForestClassifier yang telah disediakan oleh scikit-learn dengan nilai parameter random state yang divariasikan.

```
rs_range = np.arange(1,20000)
for i in rs_range:
    rfc = RandomForestClassifier(random_state = i)
    rfc.fit(train_features,train_labels)
```

Selanjutnya, didapatkan nilai parameter random_state terbaik berdasarkan score dari test.

```
test_score = rfc.score(test_features, test_labels)
```

Selanjutnya dilakukan evaluasi model dengan melihat hasil confusion matrix dan classification report.

```
rfc_pred = rfc.predict(test_features)
cm = confusion_matrix(test_labels,rfc_pred)

report_rfc = classification_report(test_labels, rfc_pred, digits=4)
print('Classification Report of Random Forest Classifier : \n', report_rfc)
```

4.4.2 Hyperparameter Tuning

Pada hyperparameter tuning, digunakan dua metode yaitu diawali dengan RandomSearchCV lalu selanjutnya dilakukan tuning kembali menggunakan metode GridSearchCV. Hal ini dilakukan karena RandomSearchCV digunakan untuk menentukan rentang nilai hyperparameter yang akan diuji dan dipilih kombinasi secara acak dari rentang tersebut. Hal Ini berguna untuk mendapatkan hasil yang cepat. Dari besarnya rentang hyperparameter di awal, kemudian rentang akan dikerucutkan dan dilakukan tuning kembali menggunakan metode GridSearchCV. Hal ini karena GridSearchCV memerlukan definisi yang jelas tentang setiap nilai hyperparameter yang akan diuji. Ini cocok digunakan ketika memiliki informasi yang pasti tentang rentang setiap nilai hyperparameter yang harus dicari.

Berikut adalah parameter-parameter yang dituning menggunakan metode RandomSearchCV:

```
'n_estimators': [int(x) for x in np.linspace(100, 300, num = 12)],
'max_features': ['sqrt', 'log2', 10],
'max_depth' : [int(x) for x in np.linspace(10, 50, num = 12)],
'min_samples_split': [2, 3, 4],
'min_samples_leaf': [1, 2],
'criterion' : ['gini', 'entropy'],
'bootstrap' : [True, False]
```

Berikut adalah parameter-parameter yang dituning menggunakan metode GridSearchCV:

```
'n_estimators': [int(x) for x in np.linspace(n_estRS - 30, n_estRS + 30, 7)],
'max_features': [best_param['max_features']],
'max_depth' : [int(x) for x in np.linspace(max_dRS - 10, max_dRS + 10, 7)],
'min_samples_split': [best_param['min_samples_split']],
'min_samples_leaf': [best_param['min_samples_leaf']],
'criterion' : [best_param['criterion']],
'bootstrap' : [best_param['bootstrap']]
```

5 Hasil dan Diskusi

Pada eksperimen ini, kami menggunakan 12 fitur utama sebagai input model. Fitur-fitur ini dipilih berdasarkan analisis korelasi dan pentingnya fitur terhadap variabel target. Dilakukan juga encoding terhadap fitur-fitur yang bersifat ketegorikal menjadi numerikal dengan menggunakan pandas.get dummies() sehingga fiturnya menjadi 21. Kami juga melakukan beberapa transformasi pada fitur-fitur seperti normalisasi. Selanjutnya, dataset dibagi untuk training dan testing dengan train size = 0,8 dimana sebanyak 80% dari total 1904 data akan menjadi data training dan 20% nya akan menjadi data testing. Dari variasi nilai random_state yang dicari pada proses split data, didapatkan bahwa nilai parameter random_state terbaik yaitu 14993.

Dilakukan dua percobaan pada penelitian ini yaitu menggunakan model RF fitting biasa dan model RF hyperparameter. Model RF dengan hyperparameter tuning diatur menggunakan metode RandomSearchCV dan GridSearchCV dimana RandomSearchCV digunakan untuk mengerucutkan rentang hyperparameter awal untuk selanjutnya dilakukan tuning menggunakan metode GridSearchCV pada hyperparameter yang sudah mengerucut. dDengan menggunakan iterasi sebanyak 10 dan 5-fold cross-validation untuk menghindari overfitting. Performa kedua model diukur menggunakan metrik classification report menggunakan library sklearn.

5.1 Accuracy

Hasil accuracy terbaik yang didapat dari dua percobaan yaitu:

	Random Forest	Hyperparameter Tuning
Training	100%	100%
Test	97.64%	98.69%

Table 2: Hasil Training Model Random Forest dan Hyperparameter Tuning

Hasil eksperimen menunjukkan bahwa Random Forest Classifier yang menggunakan hyperparameter tuning menghasilkan akurasi yang lebih baik dibandingkan dengan Random Forest Classifier biasa pada saat menggunakan data test. Pada eksperimen ini, akurasi Random Forest Classifier biasa sebesar 97.64%, sedangkan akurasi Random Forest Classifier dengan hyperparameter tuning sebesar 98.69%. Hasil ini menunjukkan bahwa hyperparameter tuning dapat meningkatkan kinerja model prediksi.

Kami juga melakukan analisis perbedaan nilai hyperparameter yang digunakan pada Random Forest Classifier dengan hyperparameter tuning. Hasil menunjukkan bahwa nilai hyperparameter yang optimal untuk eksperimen ini adalah

```
n_estimators: 220,
max_features: 10,
max_depth: 36,
min_samples_leaf: 2,
min_samples_split: 4,
criterion: 'entropy',
bootstrap: False,
```

Nilai-nilai hyperparameter ini memberikan kontribusi terbesar dalam meningkatkan kinerja model. Dari eksperimen yang dilakukan, diketahui bahwa Random Forest Classifier biasa dan Random Forest Classifier

dengan hyperparameter mempunyai kelebihan dan kekurangannya sendiri. Kelebihan RF biasa yaitu lebih mudah dan cepat dalam penggunaannya, karena tidak memerlukan proses hyperparameter tuning yang rumit.

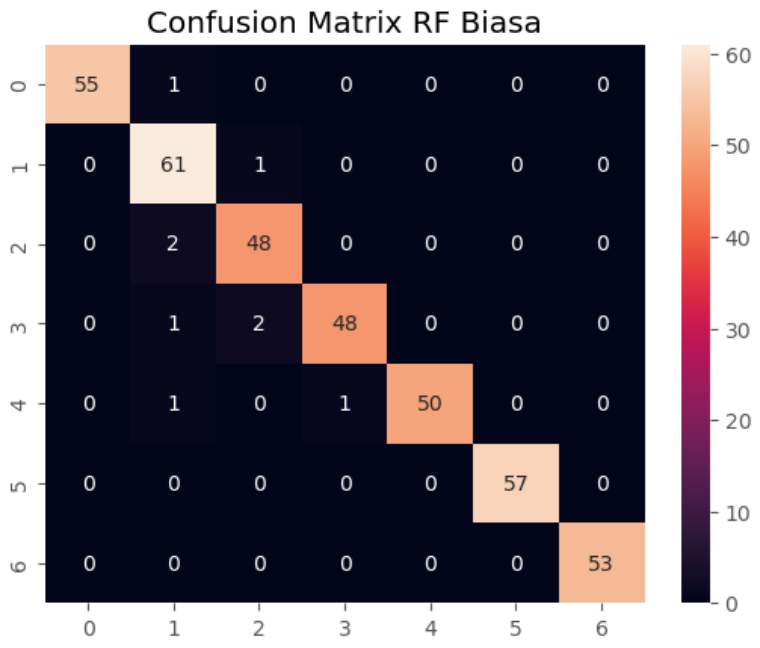
Kekurangan RF biasa yaitu karena tidak mempertimbangkan pengaruh dari setiap nilai hyperparameter terhadap kinerja model, sehingga kemungkinan besar tidak akan menghasilkan model yang optimal. Kelebihan RF hyperparameter sendiri yaitu menghasilkan model yang lebih optimal dan akurat, karena nilai hyperparameter telah diatur dengan baik. Namun memliki kekurangan yaitu memerlukan waktu dan usaha yang lebih banyak dalam proses hyperparameter tuning, terutama jika menggunakan teknik pencarian hyperparameter yang kompleks. Selain itu, diperlukan juga pemahaman yang lebih mendalam tentang algoritma dan pengaruh setiap nilai hyperparameter terhadap kinerja model.

5.2 Evaluasi

Hasil evaluasi model pada percobaan ini melalui confusion matrix dan classification report adalah sebagai berikut.

1. Model Random Forest Biasa

Figure 16: Confusion Matrix Model RF Biasa



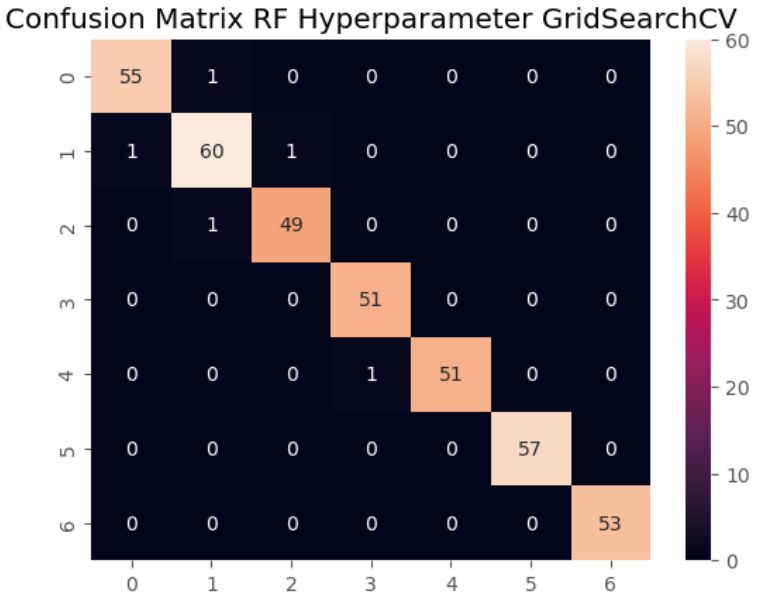
Classification Report of Random Forest Classifier

	precision	recall	f1-score	support
0	1.0000	0.9821	0.9910	56
1	0.9242	0.9839	0.9531	62
2	0.9412	0.9600	0.9505	50
3	0.9796	0.9412	0.9600	51
4	1.0000	0.9615	0.9804	52
5	1.0000	1.0000	1.0000	57
6	1.0000	1.0000	1.0000	53
accuracy			0.9764	381
macro avg	0.9779	0.9755	0.9764	381
weighted avg	0.9772	0.9764	0.9765	381

Table 3: Hasil Report Klasifikasi Random Forest Biasa

2. Model Random Forest Hyperparameter Tuning

Figure 17: Confusion Matrix Model RF Hyperparameter Tuning



Classification Report of Random Forest Classifier

	precision	recall	f1-score	support
0	0.9821	0.9821	0.9821	56
1	0.9677	0.9677	0.9677	62
2	0.9800	0.9800	0.9800	50
3	0.9808	1.0000	0.9903	51
4	1.0000	0.9808	0.9903	52
5	1.0000	1.0000	1.0000	57
6	1.0000	1.0000	1.0000	53
accuracy			0.9869	381
macro avg	0.9872	0.9872	0.9872	381
weighted avg	0.9869	0.9869	0.9869	381

Table 4: Hasil Report Klasifikasi Random Forest Hyperparameter

Dari hasil confusion matrix dan classification report tersebut dapat kita evaluasi hasil prediksi dari model Random Forest Biasa dan Random Forest yang telah di hyperparameter tuning adalah sebagai berikut.

1. Pada kategori 5 dan 6, kedua model berhasil memprediksi dengan sangat baik ditandai dengan metrik evaluasi precision, recall, dan f1-score bernilai 1.0000.

2. Pada kategori 0 - 4, kedua model berhasil memprediksi dengan baik ditandai dengan metrik evaluasi precision, recall, dan f1-score bernilai diatas 0.92

3. Model RF biasa mempunyai kekurangan dalam memprediksi nilai positive pada data kategori 1. Hal ini ditunjukkan dengan nilai presisi yang relatif rendah dibandingkan dengan kategori lain akibat terdapat banyak false positive yang dihasilkan model. Selain itu, model RF biasa juga mempunyai kekurangan lain yaitu model mendapati banyak false positive pada kategori 3, yang mengakibatkan metrik evaluasi recallnya mempunyai nilai yang relatif lebih rendah dibandingkan kategori lain.

4. Pada kategori 1, Model RF Hyperparameter mempunyai nilai presisi dan recall yang paling rendah dibandingkan kategori lainnya, hal ini disebabkan oleh nilai false positive dan false negative yang tinggi pada

kategori 1. Selain itu, model RF Hyperparameter juga menghasilkan false positive pada kategori 0 sedangkan model RF biasa tidak.

5. Dari hasil dua pendekatan pernghitungan rata-rata metrik, diketahui bahwa model RF Hyperparameter mempunyai hasil yang lebih baik dibandingkan model RF biasa untuk semua metrik evaluasi pada macro avg dan weighted avg.

6 Kesimpulan

Adapun kesimpulan pada penilitian ini adalah sebagai berikut.

1. Klasifikasi tingkat obesitas berdasarkan body mass index menggunakan metode Random Forest Classifier dan Hyperparameter tuning berhasil dilakukan.

2. Tingkat akurasi pada model Random Forest Classifier diperoleh 97.64% sedangkan model RF yang telah di optimasi diperoleh 98.69%.

3. Hyperparameter yang optimal untuk eksperimen ini adalah sebagai berikut.

```
n_estimators: 220,
max_features: 10,
max_depth: 36,
min_samples_leaf: 2,
min_samples_split: 4,
criterion: 'entropy',
bootstrap: False,
```

4. Kedua model Random Forest mampu memprediksi dengan sangat baik pada kategori 5 dan 6 atau obesitas tipe 2 dan obesitas tipe 3.

7 Kontribusi

Hasil penelitian ini berkontribusi terhadap pengetahuan dan informasi dalam explaratory data analisis dan membangun model metode pembelajaran mesin Random Forest serta pengambilan keputusan dalam mendiagnosa tingkat obesitas pasien dengan body mass index (BMI).

References

[1] S. M. Fruh, "Obesity: Risk factors, complications, and strategies for sustainable long-term weight management," *Journal of the American Association of Nurse Practitioners*, vol. 29, 2017.

[2] "Obesity and overweight." [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/obesity-and-overweight#: :text=%2Fm2).-,Adults,than%20or%20equal%20to%2030

[3] "Why use bmi?" [Online]. Available: https://www.hsph.harvard.edu/obesity-prevention-source/obesity-definition/obesity-definition-full-story/#references

[4] J. P. Santisteban Quiroz, "Estimation of obesity levels based on dietary habits and condition physical using computational intelligence," *Informatics in Medicine Unlocked*, vol. 29, p. 100901, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352914822000521

[5] F. Ferdowsy, K. S. A. Rahi, M. I. Jabiullah, and M. T. Habib, "A machine learning approach for obesity risk prediction," *Current Research in Behavioral Sciences*, vol. 2, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666518221000401

[6] H. Rossman, S. Shilo, S. Barbash-Hazan, N. S. Artzi, E. Hadar, R. D. Balicer, B. Feldman, A. Wiznitzer, and E. Segal, "Prediction of childhood obesity from nationwide health records," *J Pediatr*, 2021.

[7] X. Pang, C. B. Forrest, F. Lê-Scherban, and A. J. Masino, "Prediction of early childhood obesity with machine learning and electronic health record data," *International Journal of Medical Informatics*, vol. 150, p. 104454, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1386505621000800

[8] C. Kim and S. Youm, "Development of a web application based on human body obesity index and self-obesity diagnosis model using the data mining methodology," *Sustainability*, vol. 12, 2020.

[9] E. De-La-Hoz-Correa, F. E. Mendoza-Palechor, A. De-La-Hoz-Manotas, R. C. Morales-Ortega, and S. H. Beatriz Adriana, "Obesity level estimation software based on decision trees," *Journal of Computer Science*, vol. 15, 2019.

[10] I. Lee, K.-S. Bang, H. Moon, and J. Kim, "Risk factors for obesity among children aged 24 to 80 months in korea: A decision tree analysis," *J Pediatr Nurs*, 2019.

[11] L. Breiman, "Random forests," *Machine Learning*, 2001.

[12] "Hyperparameter tuning for machine learning models." [Online]. Available: https://towardsdatascience.com/hyperparameter-tuning-for-machine-learning-models-1b80d783b946

[13] "Hyperparameter tuning the random forest in python." [Online]. Available: https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74

[14] "Estimation of obesity levels based on eating habits and physical condition ," UCI Machine Learning Repository, 2019.

[15] M. Schonlau and R. Y. Zou, "The random forest algorithm for statistical learning," *The Stata Journal*, 2020.

[16] G. Biau, "Analysis of a random forests model," *Journal of Machine Learning Research*, 2012.

[17] N. K. Dewi, U. D. Syafitri, and S. Y. Mulyadi, "Penerapan metode random forest dalam driver analysis," *Indonesian Journal of Statistics and Its Applications*, 2011.

[18] T. K. Ho, "Random decision forests," in *Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1) - Volume 1*. IEEE Computer Society, 1995.

[19] Y. Amit and D. Geman, "Shape quantization and recognition with randomized trees," *Neural Comput.*, vol. 9, no. 7, p. 1545–1588, oct 1997. [Online]. Available: https://doi.org/10.1162/neco.1997.9.7.1545

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.