J. Ris. & Ap. Mat. Vol. 08 No. 01 (2024) pp. 51-68
Jurnal Riset dan Aplikasi Matematika (JRAM)
e-ISSN: 2581-0154
URL: journal.unesa.ac.id/index.php/jram

PENGARUH TEKNIK *OVERSAMPLING* PADA ALGORITMA *MACHINE LEARNING* DALAM KLASIFIKASI *BODY MASS INDEX* (BMI)

ISNAYNI FEBY HAWARI[1], MOHAMAD KHOIRUN NAJIB[2*], SRI NURDIATI[3], YOSEF FELIX YGGA MARPAUNG[4], NINDI KUSUMAWATI[5], MEYLIANA NURFADILA[6], KATHLEEN RABIKA SIJABAT[7], BANISSA FATHIMATUZZAHRA HERNAWAN[8]

[1,2,3,4,5,6,7,8] Departemen Matematika, FMIPA, Institut Pertanian Bogor

*Corresponding author: mkhoirun_najib@apps.ipb.ac.id

ABSTRAK

Body Mass Index (BMI) menjadi dasar klasifikasi berat badan seseorang yang dapat mengindikasikan adanya penyakit berbahaya seperti obesitas. Banyak penelitian yang melakukan klasifikasi BMI menggunakan berbagai algoritma *machine learning* dengan menerapkan berbagai teknik, salah satunya penerapan teknik *oversampling* untuk menangani ketidakseimbangan data. Penelitian ini bertujuan untuk membandingkan pengaruh ada dan tidaknya teknik *oversampling* pada algoritma K-*Nearest Neighbor* (KNN), *random forest*, dan *Support Vector Machine* (SVM). Data yang digunakan merupakan data *real* klasifikasi BMI yang mencakup informasi seperti jenis kelamin, tinggi badan, berat badan, dan indeks BMI. Tahapan yang dilakukan dalam penelitian ini meliputi data *pre-processing*, eksplorasi data, pelatihan dan pengujian model, evaluasi model, *tuning hyperparameter*, serta mengidentifikasi *feature importance*. Tahap eksplorasi data menunjukkan bahwa berat badan merupakan variabel yang memiliki korelasi paling kuat dengan indeks BMI yaitu sebesar 0.8 serta tidak ada multikolinearitas antar variabel. Hasil evaluasi model menggunakan *confusion matrix* yang didasarkan pada nilai F1-*score* menunjukkan bahwa model SVM tanpa penerapan teknik *oversampling* yang telah dilakukan *tuning hyperparameter* merupakan model terbaik pada penelitian ini dengan nilai F1-*score* lebih dari 0.95. Identifikasi *feature importance* dengan metode *Permutation Feature Importance* (PFI) pada model terbaik menunjukkan bahwa berat badan merupakan variabel yang paling mempengaruhi indeks BMI.
Kata Kunci: KNN, *random forest*, SVM, BMI, *oversampling*

ABSTRACT

Body Mass Index (BMI) is the basic of people's weight classification that can indicate serious diseases such as obesity. Many researches have been published about BMI classification using machine learning algorithms with some techniques, one of them is oversampling as a technique to handle imbalance data. The goal of this research is to compare the effect of either the existence and inexistence of oversampling in K-Nearest Neighbor (KNN), random forest, and Support Vector Machine (SVM). The dataset that is used in this research is a real BMI classification data including gender, height, weight, and BMI index. The methods of this research are data pre-processing, data exploration, training and testing model, model's evaluation, tuning hyperparameter, and also identify feature importance. The results of data exploration show that weight is the variable which has the strongest correlation with BMI index of 0.8 and there's also no multicollinearity. Model's evaluation using confusion matrix based on F1-score shows that the best model is the SVM model without oversampling after tuning hyperparameter with F1-score of more than 0.95. Feature importance's identification using Permutation Feature Importance (PFI) methods on the best model shows that weight is the most impactful variable in BMI classification.
Keywords: *KNN, random forest, SVM, BMI, oversampling*

2020 *Mathematics Subject Classification*: 62H30
Diterima: hh-bb-tt; direvisi: hh-bb-tt; diterima: hh-bb-tt

1 Pendahuluan

Body Mass Index (BMI) atau Indeks Massa Tubuh (IMT) merupakan suatu parameter yang digunakan untuk menentukan status berat badan seseorang [1]. Seseorang dapat mengetahui apakah proporsi tubuhnya tergolong ideal atau tidak, dalam hal ini bisa termasuk ke dalam *underweight* atau *overweight*, dengan mengetahui BMI. Proporsi tubuh seseorang dapat menjadi salah satu faktor penentu kesehatan seseorang. Rasyid [2] menyatakan bahwa *World Health Organization* (WHO) telah merekomendasikan BMI sebagai dasar klasifikasi berat badan seseorang yang meliputi derajat *underweight* dan *overweight* yang dapat menjadi indikasi adanya resiko penyakit berbahaya yang tak menular. Oleh karena itu, nilai BMI dapat memprediksi tingkat kematian dan morbiditas di masa yang akan datang. Menurut Sugondo [3], klasifikasi BMI didasarkan pada kriteria standar Asia Pasifik seperti pada Tabel 1.

Tabel 1. Klasifikasi *Body Mass Index* (BMI) berdasarkan kriteria standar Asia Pasifik

Klasifikasi	**BMI** (kg/m^2)
Underweight	< 18.5
Normal	18.5 – 22.9
Overweight	≥ 23
Beresiko	23 – 24.9
Obesitas I	25 – 29.9
Obesitas II	≥ 30

Selain klasifikasi BMI pada Tabel 1, ada banyak jenis klasifikasi BMI lain yang telah dipublikasikan oleh para ahli seperti klasifikasi BMI menurut WHO [4]. Klasifikasi BMI pada Tabel 1 ini hanya salah satu contoh klasifikasi BMI yang digunakan masyarakat secara umum. Tabel 1 menunjukkan bahwa seseorang dengan proporsi tubuh yang ideal memiliki nilai BMI pada kisaran 18.5 hingga 22.9 $\mathrm{kg/m^2}$ berdasarkan kriteria standar Asia Pasifik. Seseorang dengan nilai BMI di bawah 18.5 $\mathrm{kg/m^2}$ diklasifikasikan sebagai *underweight*. Kondisi tersebut dapat meningkatkan adanya resiko penyakit seperti kekurangan gizi [5]. Sementara itu, seseorang dengan nilai BMI di atas 22.9 $\mathrm{kg/m^2}$ diklasifikasikan sebagai *overweight* dengan tingkatan beresiko, obesitas I dan obesitas II. Kondisi tersebut dapat memperbesar resiko penyakit penyempitan pembuluh darah atau kardiovaskular karena berhubungan dengan sindrom metabolik yang ditandai oleh beberapa kondisi seperti hipertensi [6]. Oleh karena itu, BMI dapat membantu seseorang untuk mengetahui proporsi tubuh mereka sehingga dapat mencegah peningkatan resiko timbulnya penyakit. Sistem klasifikasi BMI ini membuat seseorang lebih mudah memahami proporsi tubuh mereka dibandingkan hanya menggunakan nilai BMI saja sehingga sistem klasifikasi ini penting untuk diketahui.

Pada era digitalisasi ini, sistem klasifikasi BMI tidak hanya dilakukan secara manual tetapi juga banyak yang menggunakan teknik komputasi untuk mempermudah proses pengklasifikasian. Teknik komputasi yang banyak digunakan untuk mengklasifikasikan BMI adalah *machine learning*. Menurut Chahal dan Gulia [7], *machine learning* merupakan bagian dari *Artificial Intelligence* (AI) yang melatih komputer untuk belajar. *Machine learning* didasarkan pada pemikiran bahwa sistem komputer dapat belajar dari data, kemudian mengidentifikasi pola dalam data dan membuat keputusan dengan sedikit keterlibatan dari manusia. *Machine learning* dapat dikelompokkan ke dalam dua kelompok besar yaitu *unsupervised learning* dan *supervised learning*. *Unsupervised learning* merupakan teknik *machine learning* yang membuat model keputusan menggunakan data *input* tanpa label sedangkan *supervised learning* membuat model keputusan dengan mengidentifikasi korelasi antara data *input* dengan target [8]. Salah satu teknik dalam *supervised learning* yaitu teknik klasifikasi yang digunakan dalam kasus pengklasifikasian BMI.

Terdapat banyak algoritma *machine learning* yang dapat digunakan dalam teknik klasifikasi seperti *nearest neighbor*, *decision tree*, *Support Vector Machine* (SVM), *naïve bayes*, *random forest classification* dan sebagainya [7]. Algoritma-algoritma tersebut menggunakan data *input* dengan label sebagai target *output* yang diharapkan dan kemudian membangun model dengan cara mengidentifikasi pola dalam data untuk memprediksi hasil *output.* Model tersebut akan dilatih terlebih dahulu untuk selanjutnya akan dilakukan tes terhadap model sehingga dapat diketahui apakah model dapat memprediksi *output* dengan tepat. Performa model dalam memprediksi *output* dapat diukur menggunakan *confusion matrix*. *Confusion matrix* menunjukkan perbandingan *output* hasil prediksi model dengan *output* yang sebenarnya [9]. Umumnya, kolom-kolom dalam *confusion matrix* merepresentasikan kelas prediksi dari data dan baris-barisnya merepresentasikan kelas aktual dari data. Menurut Sammut dan Webb [10], umumnya *confusion matrix* berbentuk matriks segi 2×2 dengan setiap selnya berisi nilai *True Positive* (TP), *True Negative* (TN), *False Positive* (FP) dan *False Negative* (FN), tetapi ada juga *confusion matrix* dengan ukuran lainnya. Nilai TP menunjukkan banyak data kelas positif yang diprediksi dengan tepat dan FP sebaliknya. Sedangkan nilai TN menunjukkan banyak data kelas negatif yang diprediksi dengan tepat dan FN sebaliknya. Nilai-nilai tersebut dapat digunakan untuk menghitung metrik evaluasi bagi model yang terdiri dari *accuracy*, *precision*, *recall* dan F1-*score* [11].

Penerapan metode *machine learning* dalam klasifikasi BMI telah diterapkan pada penelitian-penelitian sebelumnya. Oktoriansah [12] melakukan klasifikasi BMI dari data 500 orang pasien menggunakan algoritma *logistic regression* dan memperoleh akurasi sebesar 78%. Amani *et al.* [13] menggunakan algoritma *random forest, gaussian naïve bayes, decision tree, support vector machine*, *multi-layer perceptron*, k-*nearest neighbor* dan *logistic regression* untuk mengklasifikasikan BMI dari total 1316 orang yang dipilih secara acak di kota Ardabil, Iran. Rodriguez *et al*. [14] melakukan prediksi terhadap orang yang mengalami *overweight* dan obesitas menggunakan algoritma *decision tree*, SVM, KNN, *gaussian naïve bayes*, *multilayer perceptron, random forest, gradient boosting* dan *extreme gradient boosting* berdasarkan data kondisi fisik dan kebiasaan makan seseorang. Model *random forest* menghasilkan performa terbaik dengan nilai F1-*score* sebesar 78% pada penelitian tersebut.

Berdasarkan penelitian [12]- [14], pemakaian algoritma yang sama dapat menghasilkan akurasi yang berbeda karena menggunakan data yang berbeda. Namun, selain pengaruh data yang digunakan, pengolahan data sebelum digunakan oleh algoritma *machine learning* juga dapat mempengaruhi hasil akurasi model, salah satunya yaitu adanya penerapan teknik *oversampling* sebagai salah satu teknik penanganan *imbalance data*. Teknik *oversampling* dapat diterapkan menggunakan berbagai metode, salah satunya yaitu *Synthetic Minority Oversampling Technique* (SMOTE) yang banyak digunakan dalam berbagai jenis permasalahan seperti pada penelitian Sari *et al.* [15] mengenai deteksi penyakit diabetes dan penelitian Thamrin *et al.* [16] mengenai obesitas di Indonesia. Keunggulan SMOTE ialah mampu mengatasi ketidakseimbangan data dengan menambahkan data buatan pada kelas data yang minoritas. Oleh karena itu, penelitian ini berfokus untuk membandingkan pengaruh ada dan tidaknya teknik *oversampling* menggunakan metode SMOTE pada tiga algoritma *machine learning* yaitu algoritma K-*Nearest Neighbors* (KNN), *random forest classification* dan *Support Vector Machine* (SVM) dalam proses pengklasifikasian BMI dengan variabel prediktor berupa gender, tinggi badan dan berat badan. Model yang dihasilkan oleh ketiga algoritma tersebut akan dievaluasi menggunakan nilai F1-*score* dan *confusion matrix* untuk kemudian dipilih model terbaik dengan akurasi tertinggi. Penelitian ini juga melakukan analisis *feature importance* untuk menentukan variabel prediktor yang paling berpengaruh terhadap variabel respon menggunakan metode *Permutation Feature Importance* (PFI) seperti penelitian Mi *et al.* [17] yang menggunakan PFI untuk menentukan indikator yang paling berpengaruh terhadap penyakit kanker ginjal dari berbagai jenis model *machine learning*.

2 Tinjauan Pustaka

2.1 Teknik *Oversampling*

Oversampling didefinisikan sebagai salah satu metode yang digunakan untuk mengatasi *imbalance data* dengan melakukan pemerataan jumlah data minoritas agar sama dengan jumlah data mayoritas [18]. Ada beberapa jenis metode *oversampling* yang dapat digunakan yaitu *Random Over Sampling* (ROS), *Synthetic Minority Oversampling Technique* (SMOTE), *Adaptive Synthetic Sampling* (ADASYN) dan *borderline*-SMOTE [19]. Penelitian ini menggunakan metode *oversampling* SMOTE untuk menangani *imbalance data*. SMOTE bekerja dengan mengambil tetangga terdekat secara acak sebanyak k dari setiap *instance* dalam kelas minoritas dan membuat *instance* baru antara *instance* tersebut dengan tetangga terdekat yang telah dipilih sebelumnya. Metode ini bekerja lebih baik dibandingkan metode ROS sehingga diharapkan dapat mengurangi kerentanan terhadap *overfitting.*

2.2 K-*Nearest Neighbor* (KNN)

Algoritma K-*Nearest Neighbor* (KNN) merupakan salah satu algoritma *machine learning* yang banyak digunakan dalam proses klasifikasi. Algoritma ini memasukkan setiap objek baru ke dalam kelas klasifikasi yang memiliki jarak terdekat dengan objek baru tersebut [20]. Jumlah tetangga terdekat (*nearest neighbor*) dinyatakan oleh k. Langkah pertama dalam metode ini yaitu menentukan $k \in \mathbb{N}$. Kemudian, menghitung jarak antar data menggunakan metrik tertentu. Algoritma KNN memiliki tiga jenis metrik, yaitu jarak *Minkowski*, jarak *Manhattan*, dan jarak *Euclidean*. Penelitian ini menggunakan jarak *Minkowski* sebagai metrik untuk menghitung jarak antar data. Jarak *Minkowski* merupakan generalisasi dari jarak *Euclidean* dan jarak *Manhattan* [21]. Menurut Agustin *et al.* [21], jarak *Minkowski* direpresentasikan dalam persamaan sebagai berikut:

$$d(x,y) = \left| \sum_{i=1}^{n} (x_i - y_i)^p \right|^{\frac{1}{p}}$$

dengan x dan y merupakan titik yang akan dihitung besar jaraknya, n merupakan jumlah fitur dan p merupakan konstanta. Persamaan tersebut akan menjadi persamaan untuk jarak *Manhattan* jika p bernilai 1 dan menjadi persamaan untuk jarak *Euclidean* jika p bernilai 2 [22]. Menurut Cholil *et al.* [23], salah satu kelebihan algoritma KNN yaitu tangguh terhadap data dalam jumlah besar dan data yang memiliki banyak *noise* sedangkan kelemahannya yaitu perlu menentukan jumlah tetangga terdekat (k) dari target data serta membutuhkan biaya komputasi yang tinggi untuk perhitungan jarak dari setiap *query instance* pada keseluruhan data *train*.

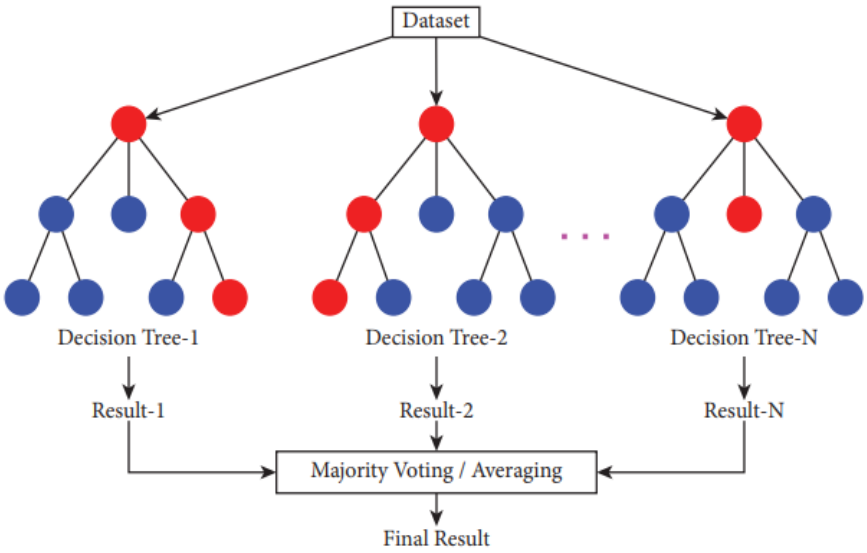
2.3 *Random Forest Classification*

Random forest merupakan algoritma *machine learning* yang terkenal dapat digunakan untuk memecahkan cakupan yang luas dari masalah klasifikasi. *Random forest* menggunakan ensemble dari banyak *decision tre*e untuk mengurangi efek *overfitting* dengan *decision tree* dihasilkan dari data yang diambil secara acak. Kelas yang dihasilkan dari proses klasifikasi pada semua *decision tree* tersebut adalah kelas yang paling banyak muncul [24]. Klasifikasi dengan *random forest* menerapkan metode *bagging* dan *random feature selection*. *Random forest* dikembangkan dari metode *bagging* dengan melakukan pemilihan variabel prediktor secara acak untuk mengurangi korelasi antar pohon yang terbentuk [25]. Berikut adalah tahapan

klasifikasi dalam menggunakan *random forest* pada gugus data yang terdiri atas n amatan dan variabel prediktor sebanyak p [26]:

1. Tahapan *bootstrap*. *Bootstrap* adalah pengambilan contoh yang disertai dengan pengembalian. Pada tahap ini, ambil sebanyak n contoh acak dari data *train*.
2. Tahapan *random feature selection*. Susun *decision tree* berdasarkan data hasil *bootstrap* sebelumnya. Pada setiap proses pemisahan, pilih beberapa variabel prediktor secara acak dari variabel prediktor yang berjumlah p. Selanjutnya, lakukan pemisahan terbaik.
3. Ulangi langkah 1-2 sebanyak k kali, sehingga diperoleh *decision tree* sebanyak k.

Lakukan penggabungan terhadap seluruh hasil prediksi *decision tree* sebanyak k dan gunakan *majority vote* yaitu mengambil *vote* terbanyak untuk menentukan hasil prediksi akhir. Ilustrasi konstruksi dari algoritma *random forest* ditunjukkan pada Gambar 1.



Gambar 1. Ilustrasi konstruksi algoritma *random forest*

2.4 *Support Vector Machine* (SVM)

Support Vector Machine (SVM) merupakan suatu teknik *machine learning* untuk melakukan klasifikasi maupun regresi. SVM berada dalam kelas *supervised learning* yaitu metode yang memerlukan data *train* untuk pembuatan model sebelum model dapat melakukan klasifikasi pada data *test* [27]. Klasifikasi SVM akan mengelompokkan data ke dalam dua atau lebih kelas menggunakan *hyperplane* dalam bentuk fungsi linear pada ruang fitur berdimensi tinggi. Pada ruang berdimensi tinggi tersebut, akan dicari *hyperplane* yang dapat memaksimumkan jarak antar kelas data. Menurut Octaviani *et al.* [28], fungsi *hyperplane* klasifikasi linear pada algoritma SVM direpresentasikan dalam persamaan sebagai berikut:

$$f(x) = \begin{cases} [\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b] \geq 1, & y_i = +1 \\ [\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b] \leq -1, & y_i = -1 \end{cases}$$

dengan $\boldsymbol{x}_i$ merupakan data *train* ke-i dan y_i merupakan label kelas dari data $\boldsymbol{x}_i$. Parapat *et al.* [29] menyatakan bahwa b merupakan koordinat garis relative terhadap titik koordinat dan $\boldsymbol{w}$ merupakan nilai bobot *support vector* yang posisinya tegak lurus antara titik pusat koordinat dengan *hyperplane*. Pencarian *hyperplane* terbaik dapat menggunakan beberapa metode, salah satunya metode *Quadratic Programming* (QP) dengan solusi berupa fungsi *Lagrange* [28].

Berdasarkan solusi akhir metode QP, kelas klasifikasi dari data dapat ditentukan menggunakan persamaan sebagai berikut:

$$f(x_t) = \sum_{s=1}^{ns} \alpha_s y_s \boldsymbol{x}_s . \boldsymbol{x}_t + b$$

dengan $\boldsymbol{x}_t$ merupakan data *test* ke-t, $\boldsymbol{x}_s$ merupakan data *support vector* ke-s $(s = 1, 2, \dots, ns)$, dan α_s merupakan pengganda fungsi *Lagrange* yang digunakan ketika menentukan solusi akhir.

SVM dapat bekerja pada data non-linear dengan menggunakan pendekatan kernel pada fitur data awal himpunan data [28]. Fungsi kernel digunakan untuk memetakan dimensi awal himpunan data ke dimensi baru. Terdapat beberapa fungsi kernel yang biasa digunakan pada data non-linear seperti kernel *Radial Basis Function* (RBF) dan kernel polinomial. Prasetyo [30] menyatakan bahwa kernel RBF direpresentasikan dalam persamaan sebagai berikut:

$$\boldsymbol{K}(x_i, x_j) = exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$

dan kernel polinomial direpresentasikan dalam persamaan sebagai berikut:

$$\boldsymbol{K}(x_i, x_j) = \left((x_i . x_j) + c\right)^d$$

dengan x_i dan x_j merupakan pasangan dua data *train* dan parameter σ, c, d merupakan konstanta bernilai lebih besar dari 0.

2.5 *Confusion Matrix*

Confusion matrix merupakan suatu metode untuk memberikan informasi hasil dari klasifikasi yang dilakukan oleh sistem yang berguna untuk menganalisis seberapa baik *classifier* mengenali *tuple* dari kelas yang berbeda. Tabel 2 menunjukkan *confusion matrix* untuk kasus klasifikasi 6 kelas pada penelitian ini. Menurut Nurdiati *et al.* [31], interpretasi dari nilai $C_{1,1}$ yaitu jumlah data yang merupakan kelas 1 secara aktual dan model memprediksinya sebagai kelas 1 juga. Sementara interpretasi untuk nilai $C_{1,2}$ yaitu jumlah data yang merupakan kelas 1 secara aktual namun model memprediksinya sebagai kelas 2. Interpretasi tersebut berlaku untuk seluruh nilai pada Tabel 2.

Tabel 2. Ilustrasi *confusion matrix* untuk 6 kelas.

		Kelas Prediksi					
		Kelas 0	***Kelas 1***	***Kelas 2***	***Kelas 3***	***Kelas 4***	***Kelas 5***
Kelas aktual	Kelas 0	$C_{0,0}$	$C_{0,1}$	$C_{0,2}$	$C_{0,3}$	$C_{0,4}$	$C_{0,5}$
	Kelas 1	$C_{1,0}$	$C_{1,1}$	$C_{1,2}$	$C_{1,3}$	$C_{1,4}$	$C_{1,5}$
	Kelas 2	$C_{2,0}$	$C_{2,1}$	$C_{2,2}$	$C_{2,3}$	$C_{2,4}$	$C_{2,5}$
	Kelas 3	$C_{3,0}$	$C_{3,1}$	$C_{3,2}$	$C_{3,3}$	$C_{3,4}$	$C_{3,5}$
	Kelas 4	$C_{4,0}$	$C_{4,1}$	$C_{4,2}$	$C_{4,3}$	$C_{4,4}$	$C_{4,5}$
	Kelas 5	$C_{5,0}$	$C_{5,1}$	$C_{5,2}$	$C_{5,3}$	$C_{5,4}$	$C_{5,5}$

Ukuran evaluasi model yang terdiri dari akurasi, presisi, *recall*, dan F1-*score* dapat dihitung berdasarkan nilai dalam *confusion matrix* pada Tabel 2. Masing-masing dari keempat ukuran tersebut direpresentasikan dalam persamaan sebagai berikut:

a) Akurasi

$$Accuracy = \frac{\sum_{i=1}^{n} C_{i,i}}{\sum_{j=1}^{n}\sum_{i=1}^{n} C_{i,j}}$$

b) Presisi

$$Precision(C_i) = \frac{C_{i,i}}{\sum_{j=1}^{n} C_{i,j}}$$

c) *Recall*

$$Recall(C_i) = \frac{C_{i,i}}{\sum_{j=1}^{n} C_{j,i}}$$

d) F1-*score*

$$F_1(C_i) = \frac{2 \times Precision(C_i) \times Recall(C_i)}{Precision(C_i) + Recall(C_i)}$$

2.6 *Tuning Hyperparameter*

Sebagian besar algoritma *machine learning* modern memiliki parameter yang perlu ditentukan sebelum menjalankannya. Parameter-parameter tersebut disebut sebagai *hyperparameter*. *Hyperparameter* tersebut dapat dicari nilai terbaiknya untuk memaksimalkan hasil kerja algoritma *machine learning*. Proses tersebut dikenal dengan sebutan *tuning hyperparameter*. Pemilihan konfigurasi *hyperparameter* yang sesuai untuk dataset tertentu dapat dilakukan menggunakan nilai *default hyperparameter* yang ditentukan dalam paket perangkat lunak implementasi atau mengkonfigurasinya secara manual. Misalnya, berdasarkan rekomendasi dari literatur, pengalaman atau metode uji coba dan kesalahan [32]. Sebagai alternatif, dapat digunakan strategi penyesuaian nilai *hyperparameter* yang bergantung pada data. Strategi pencarian ini bervariasi mulai dari pencarian grid atau acak yang sederhana hingga prosedur iteratif yang lebih kompleks seperti optimisasi *bayesian* atau *iterated f-racing* [33].

2.7 *Feature Importance*

Feature importance merupakan teknik yang digunakan untuk menghitung tingkat pengaruh variabel prediktor terhadap variabel respon. Penelitian ini menggunakan metode *Permutation Feature Importance* (PFI) untuk mengetahui tingkat pengaruh masing-masing variabel prediktor dalam memprediksi variabel respon untuk setiap algoritma *machine learning* yang digunakan. Menurut Kaneko [34], algoritma perhitungan PFI untuk iterasi sebanyak J dilakukan dengan tahapan sebagai berikut:

1. Konstruksi model menggunakan data *train.*
2. Hitung skor referensi (rs) untuk model pada data validasi. Skor referensi merupakan nilai akurasi dari model klasifikasi atau koefisien determinasi untuk model regresi.
3. Untuk setiap variabel prediktor i atau kolom ke-i pada data validasi dan untuk setiap pengulangan j, lakukan pengacakan untuk setiap variabel prediktor i dengan tujuan membangkitkan *Corrupted Validation Data* (CVD) dan menghitung skor $s_{i,j}$ pada model $\mathrm{CVD}_{i,j}$.
4. Hitung PFI_i untuk variabel prediktor ke-i menggunakan persamaan sebagai berikut:

$$\mathrm{PFI}_i = rs - \frac{1}{J}\sum_{j=1}^{J} s_{i,j}$$

dengan rs merupakan skor referensi untuk model pada data validasi, J merupakan jumlah total iterasi yang digunakan, dan $s_{i,j}$ merupakan nilai skor untuk model CVD dengan variabel prediktor i dan iterasi ke-j.

3 Metode Penelitian

3.1 Sumber Data

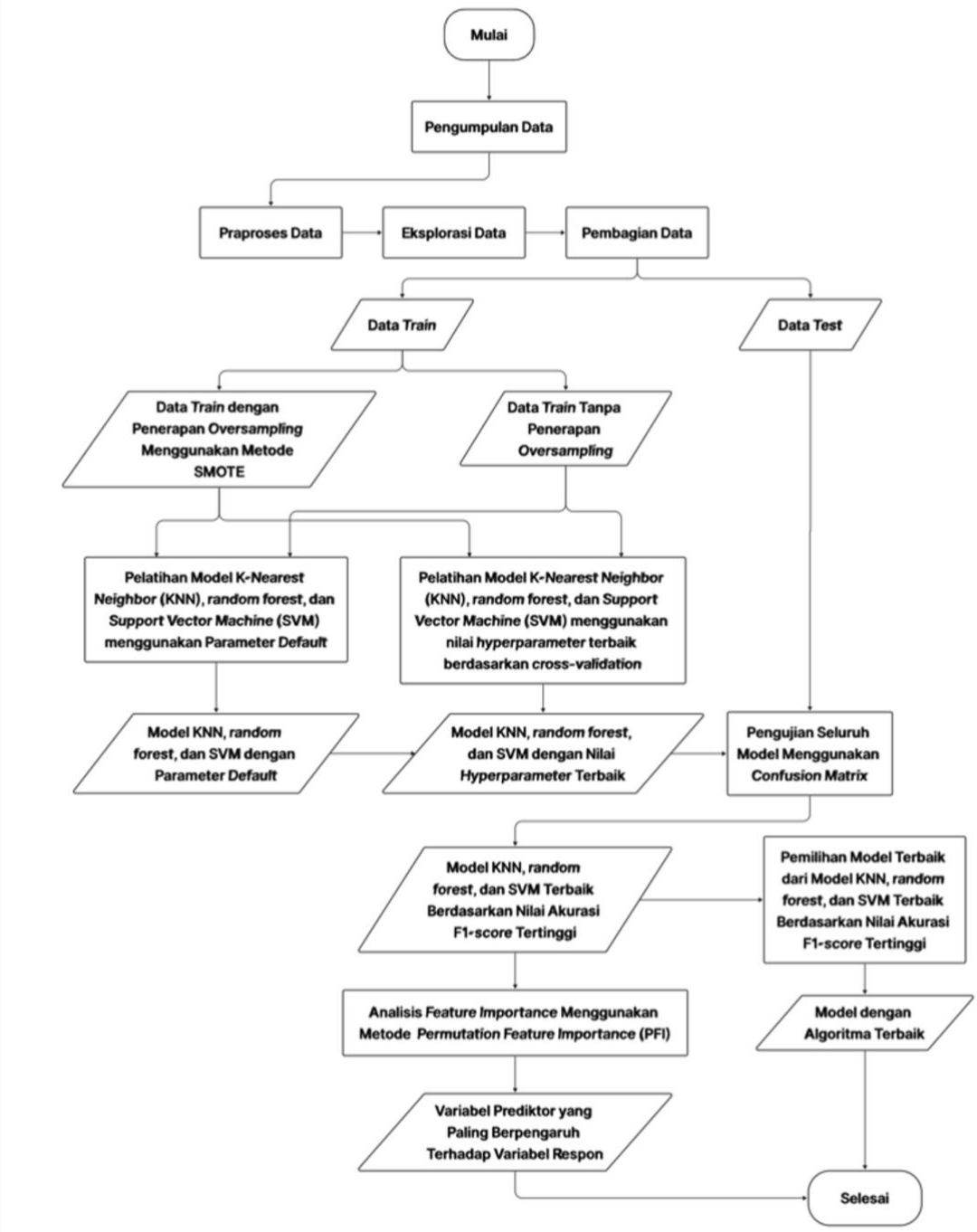
Penelitian ini menggunakan data *real* klasifikasi *Body Mass Index* (BMI) yang diperoleh dari situs Kaggle dalam tautan https://www.kaggle.com/datasets/sjagkoo7/bmi-body-mass-index . Dataset tersebut berisi 400 baris dan 4 kolom yang terdiri dari kolom jenis kelamin individu, tinggi individu dalam satuan cm, berat individu dalam satuan kg, dan indeks BMI individu yang dikategorikan menjadi 6 kelas. Kelas tersebut terdiri atas kelas 0 yang diartikan sangat lemah, kelas 1 diartikan lemah, kelas 2 diartikan normal, kelas 3 diartikan kelebihan berat badan, kelas 4 diartikan obesitas, dan kelas 5 diartikan sangat obesitas. Berdasarkan hasil proses eksplorasi data, data ini tidak memiliki data kosong dan memiliki data duplikasi sebanyak 8 data. Setelah dilakukan penanganan data duplikasi berupa penghapusan data, maka data yang dipakai dalam penelitian ini terdiri dari 392 baris dan 4 kolom.

3.2 Tahapan Penelitian

Tahapan penelitian yang dilakukan dalam penelitian ini yaitu sebagai berikut:

1. Melakukan studi literatur dari berbagai sumber terpercaya mengenai teknik *oversampling*, sistem klasifikasi BMI serta algoritma KNN, *random forest classification* dan SVM sebagai algoritma yang digunakan.
2. Mengumpulkan data yang terdiri dari gender, tinggi badan, berat badan dan indeks BMI.
3. Melakukan praproses data dan eksplorasi data yang terdiri dari:
   a) Mendeteksi data kosong dan data duplikasi serta melakukan penanganannya.
   b) Membuat *heatmap* korelasi untuk melihat korelasi antara variabel prediktor dan respon serta mendeteksi adanya multikolinearitas.
   c) Mendeteksi ketidakseimbangan data dengan membuat data *copy* untuk menerapkan teknik *oversampling* menggunakan metode SMOTE pada salah satu data, sedangkan data lainnya dibiarkan tanpa penerapan teknik *oversampling*.
   d) Membagi data menjadi data *train* sebesar 75% dan data *test* sebesar 25% secara acak.
4. Melakukan pelatihan dan pengujian tiga jenis model menggunakan algoritma KNN, *random forest*, dan SVM dengan membangun model parameter *default* terlebih dahulu.
5. Melakukan *tuning hyperparameter* terhadap beberapa jenis *hyperparameter* dengan menggunakan *cross validation* sebanyak 4 bagian dengan tujuan meningkatkan akurasi model dan menangani *overfitting*. Model hasil *tuning hyperparameter* akan dibandingkan dengan model parameter *default* untuk menentukan model terbaik pada setiap algoritma.
6. Melakukan evaluasi terhadap seluruh model terbaik berdasarkan *confussion matrix* dan *classification report* yaitu nilai *precision*, *recall*, dan F1-*score*.
7. Melakukan analisis *feature importance* menggunakan metode PFI dengan tujuan menentukan variabel yang paling berpengaruh dalam mengklasifikasikan BMI pada setiap model terbaik.

Seluruh tahapan tersebut dapat diilustrasikan dalam diagram alur seperti pada Gambar 2.



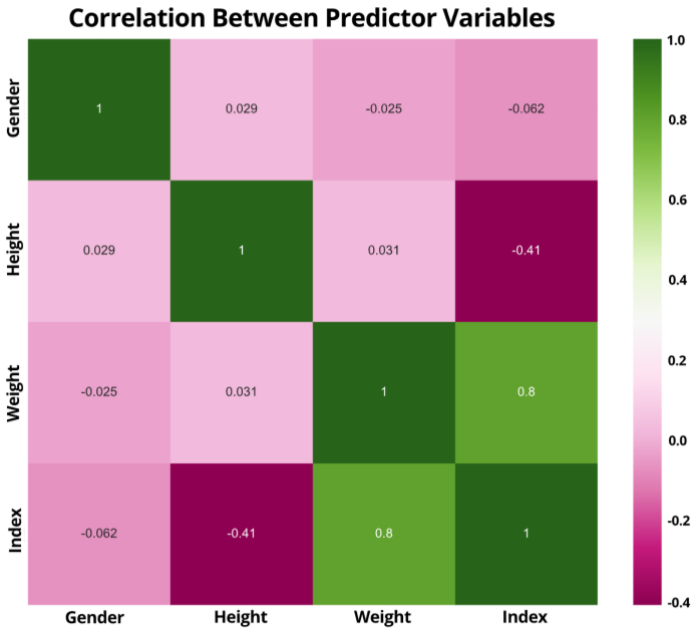
Gambar 2. *Diagram alur tahapan penelitian*

4 Hasil dan Pembahasan

4.1 Eksplorasi Data

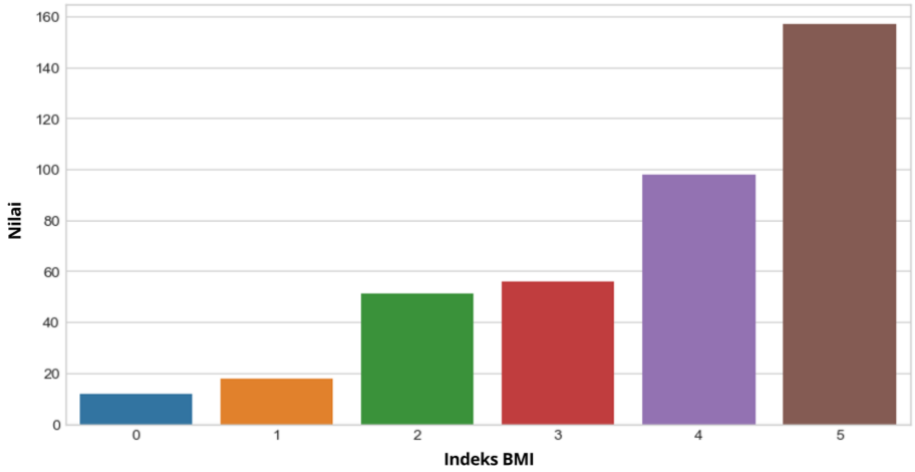
Proses eksplorasi data pada tahap praproses data menghasilkan *heatmap* yang menunjukkan nilai korelasi antara variabel prediktor yaitu gender, tinggi badan, dan berat badan dengan variabel respon yaitu indeks BMI. Korelasi didefinisikan sebagai cara untuk menunjukkan keeratan hubungan antara dua atau lebih variabel yang direpresentasikan dalam koefisien korelasi [35]. Koefisien korelasi dapat bernilai positif pada rentang 0 hingga 1 dan dapat juga bernilai negatif pada rentang 0 hingga -1. Nilai korelasi antara variabel prediktor

dengan variabel respon pada *heatmap* yang ditunjukkan oleh Gambar 3 merupakan nilai korelasi Pearson.



Gambar 3. *Heatmap nilai korelasi antara variabel prediktor dengan variabel respon*

Gambar 3 menunjukkan bahwa ada variabel prediktor yang berkorelasi positif dengan variabel respon yaitu berat badan dan ada juga yang berkorelasi negatif yaitu gender dan tinggi badan. Nilai korelasi tertinggi dihasilkan oleh variabel prediktor berat badan yang berkorelasi linear positif dengan indeks BMI sebesar 0.8 sedangkan gender menghasilkan nilai korelasi terendah dengan indeks BMI yaitu sebesar -0.062. Sementara itu, tinggi badan berkorelasi negatif dengan indeks BMI sebesar -0.41. Selain korelasi antara variabel prediktor dengan variabel respon, adanya multikolinearitas juga dapat diidentifikasi berdasarkan nilai korelasi antar variabel prediktor pada Gambar 3. Multikolinearitas didefinisikan sebagai kondisi yang ditandai dengan adanya korelasi yang cukup besar antar variabel prediktor atau dengan kata lain, variabel prediktor tidak bersifat saling bebas [36]. Berdasarkan Gambar 3, tidak ada koefisien korelasi antar variabel prediktor yang melebihi 0.1 atau -0.1 sehingga dapat disimpulkan bahwa tidak terjadi multikolinearitas pada data yang artinya gender, tinggi badan, dan berat badan seseorang tidak saling mempengaruhi satu sama lain.



Gambar 4. Proporsi masing-masing kelas dalam variabel respon

Selain nilai korelasi antar variabel, proses eksplorasi data juga menghasilkan grafik batang yang menunjukkan proporsi masing-masing kelas dalam variabel respon seperti ditunjukkan pada Gambar 4. Terlihat bahwa proporsi kelas 0, 1, 2, dan 3 memiliki perbedaan yang sangat jauh jika dibandingkan dengan proporsi kelas 5. Hal serupa juga terjadi pada proporsi kelas 4 yang berbeda jauh dengan kelas 5 meskipun tidak sejauh kelas lainnya. Perbedaan yang jauh tersebut mengindikasikan adanya data tak seimbang atau *imbalance data*. *Imbalance data* akan menyebabkan performa model yang dihasilkan algoritma *machine learning* menurun karena model kurang mempelajari data dengan proporsi kelas yang lebih kecil. Akibatnya, akurasi model dalam memprediksi kelas tersebut akan buruk. Sebaliknya, model akan lebih banyak mempelajari data dengan proporsi kelas yang lebih besar sehingga akurasi model dalam memprediksi kelas tersebut akan jauh lebih baik. Oleh karena itu, perlu dilakukan penanganan *imbalance data*, salah satunya yaitu teknik *oversampling*. Namun, penggunaan teknik *oversampling* akan lebih cenderung membuat model menjadi *overfitting*, khususnya ketika teknik *oversampling* yang digunakan berlebihan [37]. Oleh karena itu, dalam penelitian ini akan dilakukan perbandingan model dengan penerapan teknik *oversampling* dan model tanpa penerapan teknik *oversampling*. Setelah diterapkan teknik *oversampling* dengan metode SMOTE pada data *train*, proporsi kelas dalam variabel respon menjadi seimbang yaitu sebanyak 120 data untuk masing-masing kelas.

4.2 Pelatihan dan Pengujian Model

4.2.1 K-*Nearest Neighbor*

Penelitian ini melakukan beberapa percobaan dalam membangun model KNN untuk memperoleh model KNN dengan akurasi terbaik. Percobaan pertama yaitu membangun model KNN dengan menggunakan parameter *default* atau tanpa mengubah parameter apapun. Hasil yang diperoleh ditunjukkan pada Tabel 3 berikut.

Tabel 3. Hasil akurasi model KNN menggunakan parameter *default*

	Data *Train*	**Data *Test***
Tanpa *oversampling*	0.91837	0.88776
Dengan *oversampling*	0.95798	0.88776

Tabel 3 menunjukkan bahwa terjadi *overfitting* pada model KNN dengan penerapan teknik *oversampling* yang ditandai dengan adanya perbedaan yang cukup jauh antara nilai akurasi pada data *train* dengan nilai akurasi pada data *test*. Sementara itu, model KNN tanpa penerapan teknik *oversampling* tidak menunjukkan adanya *overfitting* karena akurasi yang diperoleh pada data *train* tidak jauh berbeda dengan akurasi pada data *test*. Selain terjadi *overfitting*, akurasi data *train* yang diperoleh model KNN dengan *oversampling* lebih tinggi dibandingkan akurasi data *train* pada model KNN tanpa *oversampling* sedangkan akurasi pada data *test* tidak mengalami perbedaan. Selanjutnya, dilakukan *tuning hyperparameter* untuk meningkatkan akurasi model. Selain itu, *tuning hyperparameter* juga dilakukan sebagai upaya untuk menangani *overfitting*. *Tuning hyperparameter* dilakukan dengan menggunakan *cross validation* sebanyak 4 bagian dan dilakukan terhadap parameter metrik dan jumlah *neighbors* pada kedua model KNN yang telah dibangun sebelumnya. Penelitian ini melakukan *tuning hyperparameter* pada setiap jenis metrik untuk mencari nilai terbaik bagi parameter jumlah *neighbors*. Hasil akurasi model KNN setelah dilakukan *tuning hyperparameter* ditunjukkan dalam Tabel 4.

Tabel 4. Hasil akurasi model KNN untuk setiap metrik setelah dilakukan *tuning hyperparameter*

		Tanpa *oversampling*	**Dengan *oversampling***
Metrik *Euclidean*	Data *Train*	1.0	1.0
	Data *Test*	0.86735	0.86735
	Jumlah *Neighbors*	1	1
Metrik *Manhattan*	Data *Train*	0.91497	1.0
	Data *Test*	0.88776	0.89796
	Jumlah *Neighbors*	4	1
Metrik *Minkowski*	Data *Train*	1.0	1.0
	Data *Test*	0.86735	0.86735
	Jumlah *Neighbors*	1	1

Tabel 4 menunjukkan bahwa model KNN dengan *oversampling* masih mengalami *overfitting* pada ketiga metrik yang digunakan. Selain itu, model KNN tanpa *oversampling* juga mengalami *overfitting* pada metrik *Euclidean* dan metrik *Minkowski* sedangkan pada metrik *Manhattan* tidak mengalami *overfitting* dan hanya sedikit mengalami perbedaan pada akurasi data *train* dengan model KNN tanpa *oversampling* sebelum dilakukan *tuning hyperparameter*. Hal tersebut menunjukkan bahwa *tuning hyperparameter* tidak dapat menangani *overfitting* pada model KNN dengan parameter *default* meskipun telah menggunakan teknik *cross validation.* Oleh karena itu, algoritma KNN kurang sesuai untuk digunakan dalam mengklasifikasikan data BMI pada penelitian ini. Model KNN terbaik yang diperoleh merupakan model KNN yang menggunakan parameter *default* tanpa penerapan teknik *oversampling* dengan akurasi data *train* sebesar 0.91837 dan akurasi pada data *test* sebesar 0.88776.

Tabel 5. Hasil evaluasi model KNN terbaik pada setiap kelas variabel respon dengan *confusion matrix*

		Kelas Prediksi					
		Kelas 0	**Kelas 1**	**Kelas 2**	**Kelas 3**	**Kelas 4**	**Kelas 5**
Kelas Aktual	Kelas 0	4	1	0	0	0	0
	Kelas 1	0	6	1	0	0	0
	Kelas 2	0	1	10	4	0	0
	Kelas 3	0	0	0	10	0	0
	Kelas 4	0	0	0	1	20	2
	Kelas 5	0	0	0	0	1	37

Evaluasi model KNN terbaik dilakukan menggunakan *confusion matrix* yang ditunjukkan pada Tabel 5. Terlihat bahwa model KNN terbaik mampu memprediksi kelas 0 dengan tepat sebanyak 4 data, kelas 1 sebanyak 6 data, kelas 2 sebanyak 10 data, kelas 3 sebanyak 10 data, kelas 4 sebanyak 20 data, dan kelas 5 sebanyak 37 data. Model KNN terbaik paling banyak melakukan kesalahan prediksi ketika memprediksi kelas 2 dan kelas 3 berdasarkan Tabel 5. Kemudian dilakukan perhitungan nilai akurasi model KNN terbaik menggunakan *classification report* dengan hasilnya ditunjukkan oleh Tabel 6. Terlihat bahwa nilai akurasi F1-*score* terbesar dihasilkan oleh kelas 5 sebesar 0.96 dan nilai akurasi F1-*score* terkecil dihasilkan oleh kelas 2 sebesar 0.77. Hal tersebut menunjukkan bahwa model KNN terbaik menghasilkan performa terbaiknya ketika memprediksi kelas 5 dan performa terburuknya ketika memprediksi kelas 0.

Tabel 6. Hasil evaluasi model KNN terbaik dengan *classification report*

	Kelas 0	**Kelas 1**	**Kelas 2**	**Kelas 3**	**Kelas 4**	**Kelas 5**
Precision	1.00	0.75	0.91	0.67	0.95	0.95
Recall	0.80	0.86	0.67	1.00	0.87	0.97
F1-*Score*	0.89	0.80	0.77	0.80	0.91	0.96

4.2.2 *Random Forest*

Model pertama yang dibangun menggunakan algoritma ini juga merupakan model dengan parameter *default*. Kemudian model tersebut akan dibandingkan dengan model kedua yang merupakan model hasil *tuning hyperparameter* seperti pada algoritma sebelumnya. Perbandingan hasil akurasi kedua model ditunjukkan oleh Tabel 7. Terlihat bahwa model *random forest classifier* dengan parameter *default* atau sebelum *tuning hyperparameter* mengalami *overfitting*, baik pada model yang menerapkan teknik *oversampling* maupun yang tidak. *Tuning hyperparameter* dilakukan menggunakan *cross validation* sebanyak 4 bagian dan dilakukan terhadap parameter jumlah pohon, kedalaman maksimum pohon dan jumlah maksimum sampel yang digunakan oleh suatu pohon. Namun demikian, setelah dilakukan *tuning hyperparameter*, model tetap mengalami *overfitting*. Oleh karena itu, algoritma *random forest classifier* juga kurang sesuai digunakan untuk mengklasifikasikan data BMI pada penelitian ini. Model yang dianggap sebagai model terbaik dibandingkan keempat model *random forest classifier* lainnya adalah model tanpa *oversampling* sebelum *tuning hyperparameter* dengan akurasi data *train* sebesar 1.0 dan akurasi pada data *test* sebesar 0.82653 berdasarkan Tabel 7. Hal tersebut didasarkan pada nilai akurasi pada data *test* yang diperoleh model tanpa *oversampling* lebih baik dibandingkan dengan *oversampling* dan *overfitting* lebih kecil sebelum dilakukan *tuning hyperparameter* pada model tersebut.

Tabel 7. Hasil akurasi model *random forest* sebelum dan setelah dilakukan *tuning hyperparameter*

		Tanpa *oversampling*	**Dengan *oversampling***
Sebelum *Tuning Hypeparameter*	Data *Train*	1.0	1.0
	Data *Test*	0.82653	0.77551
	Jumlah Pohon	100	100
	Kedalaman Maksimum	None	None
	Jumlah Sampel Maksimum	None	None
Setelah *Tuning Hyperparameter*	Data *Train*	1.0	1.0
	Data *Test*	0.81633	0.77551
	Jumlah Pohon	400	300
	Kedalaman Maksimum	None	None
	Jumlah Sampel Maksimum	None	None

Model *random forest classifier* terbaik dievaluasi menggunakan *confusion matrix* seperti pada algoritma sebelumnya dan hasil evaluasi ditunjukkan oleh Tabel 8. Terlihat bahwa hasil evaluasi model *random forest classifier* terbaik tidak jauh berbeda dengan model KNN terbaik karena model *random forest classifier* terbaik juga menghasilkan performa terbaiknya ketika memprediksi kelas 5 dan menghasilkan performa terburuknya ketika memprediksi kelas 0. Namun, model ini banyak melakukan kesalahan prediksi ketika memprediksi kelas 2, 3, dan 4 berdasarkan Tabel 8. Hal tersebut menunjukkan bahwa performa model *random forest classifier* lebih buruk dibandingkan dengan model KNN terbaik. Selanjutnya dilakukan perhitungan nilai akurasi untuk model *random forest classifier* terbaik menggunakan

classification report dan hasilnya ditunjukkan oleh Tabel 9. Terlihat bahwa nilai akurasi F1-*score* terbesar dihasilkan oleh kelas 5 sebesar 0.93 dan nilai akurasi F1-*score* terkecil dihasilkan oleh kelas 3 sebesar 0.67.

Tabel 8. Hasil evaluasi model *random forest* terbaik dengan *confusion matrix*

		Kelas Prediksi					
		Kelas 0	**Kelas 1**	**Kelas 2**	**Kelas 3**	**Kelas 4**	**Kelas 5**
Kelas Aktual	Kelas 0	3	2	0	0	0	0
	Kelas 1	0	6	1	0	0	0
	Kelas 2	0	0	11	4	0	0
	Kelas 3	0	0	0	8	2	0
	Kelas 4	0	0	0	2	15	6
	Kelas 5	0	0	0	0	0	38

Tabel 9. Hasil evaluasi model *random forest* terbaik dengan *classification report*

	Kelas 0	**Kelas 1**	**Kelas 2**	**Kelas 3**	**Kelas 4**	**Kelas 5**
Precision	1.00	0.75	0.92	0.57	0.88	0.86
Recall	0.60	0.86	0.73	0.80	0.65	1.00
F1-*Score*	0.75	0.80	0.81	0.67	0.75	0.93

4.2.3 *Support Vector Machine* (SVM)

Perbandingan antara hasil akurasi model dengan parameter *default* atau sebelum *tuning hyperparameter* dengan model setelah dilakukan *tuning hyperparameter* juga dilakukan pada algoritma ini. Hasil akurasi keempat model SVM ditunjukkan pada Tabel 10.

Tabel 10. Hasil akurasi model SVM sebelum dan setelah dilakukan *tuning hyperparameter*

	Sebelum *tuning hyperparameter*			**Setelah *tuning hyperparameter***		
	Data train	**Data test**	**Kernel**	**Data train**	**Data test**	**Kernel**
Tanpa *oversampling*	0.73809	0.67347	RBF	0.94558	0.95918	Linear
Dengan *oversampling*	0.84874	0.80612	RBF	0.92857	0.95918	Linear

Terlihat bahwa tidak ada satu pun model yang mengalami *overfitting* pada algoritma ini, sangat berbeda dengan kedua algoritma sebelumnya. Hal tersebut menunjukkan bahwa algoritma SVM sesuai digunakan untuk mengklasifikasikan data BMI pada penelitian ini. Selain itu, Tabel 10 menunjukkan bahwa model SVM sebelum *tuning hyperparameter* dengan penerapan teknik *oversampling* menghasilkan akurasi yang lebih baik dibandingkan dengan model serupa tanpa *oversampling*. Namun, setelah dilakukan *tuning hyperparameter* dengan *cross validation* sebanyak 4 bagian terhadap parameter kernel, model SVM tanpa *oversampling* menghasilkan akurasi yang lebih baik terutama pada akurasi data *test* sebesar 0.95918 yang lebih tinggi dibandingkan akurasi pada data *train*. Oleh karena itu, model SVM tanpa *oversampling* setelah dilakukan *tuning hyperparameter* merupakan model SVM terbaik dalam penelitian ini.

Model SVM terbaik dievaluasi menggunakan *confusion matrix* dan hasilnya ditunjukkan oleh Tabel 11. Terlihat bahwa model SVM terbaik juga menghasilkan performa terbaiknya ketika memprediksi kelas 5 dan menghasilkan performa terburuknya ketika memprediksi kelas 0, sama dengan model terbaik untuk kedua algoritma sebelumnya. Namun, perbedaan terlihat

pada kecilnya jumlah kesalahan prediksi yang dilakukan model SVM terbaik dibandingkan dengan model terbaik untuk kedua algoritma sebelumnya berdasarkan Tabel 11. Hal tersebut menunjukkan bahwa performa model SVM terbaik jauh lebih baik dibandingkan dengan model terbaik untuk kedua algoritma sebelumnya. Selanjutnya juga dilakukan perhitungan nilai akurasi untuk model SVM terbaik menggunakan *classification report* seperti model terbaik untuk kedua algoritma sebelumnya. Hasil perhitungan ditunjukkan oleh Tabel 12. Terlihat bahwa F1-*score* terbesar dihasilkan oleh kelas 0 dan kelas 1 sebesar 1.00, sementara nilai F1-*score* terkecil pada model ini dihasilkan oleh kelas 3 sebesar 0.90. Nilai tersebut sangat besar jika dibandingkan dengan hasil F1-*score* pada model terbaik untuk kedua algoritma sebelumnya. Oleh karena itu, model SVM terbaik dipilih menjadi model terbaik pada penelitian ini berdasarkan hasil evaluasi model terbaik untuk ketiga algoritma yang digunakan.

Tabel 11. Hasil evaluasi model SVM terbaik dengan *confusion matrix*

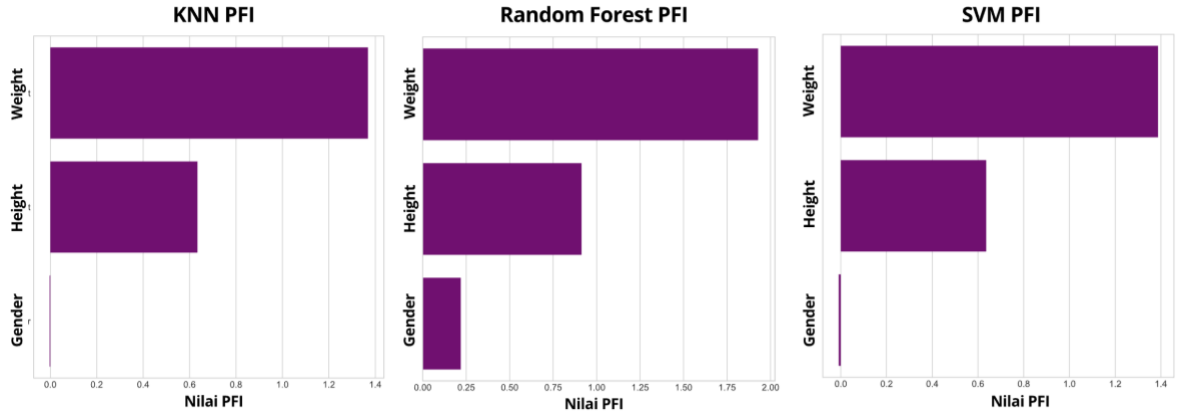
		Kelas Prediksi					
		Kelas 0	***Kelas 1***	***Kelas 2***	***Kelas 3***	***Kelas 4***	***Kelas 5***
Kelas Aktual	Kelas 0	5	0	0	0	0	0
	Kelas 1	0	7	0	0	0	0
	Kelas 2	0	0	14	1	0	0
	Kelas 3	0	0	0	9	1	0
	Kelas 4	0	0	0	0	21	2
	Kelas 5	0	0	0	0	0	38

Tabel 12. Hasil evaluasi model SVM terbaik dengan *classification report*

	Kelas 0	**Kelas 1**	**Kelas 2**	**Kelas 3**	**Kelas 4**	**Kelas 5**
Precision	1.00	1.00	1.00	0.90	0.95	0.95
Recall	1.00	1.00	0.93	0.90	0.91	1.00
F1-*Score*	1.00	1.00	0.97	0.90	0.93	0.97

4.3 *Permutation Feature Importance*

Masing-masing variabel prediktor memiliki pengaruh yang berbeda terhadap setiap model dalam mengklasifikasikan data BMI yang dapat dilihat menggunakan metode *permutation feature importance* (PFI). Ada variabel prediktor yang berpengaruh besar terhadap model dan ada juga yang kurang berpengaruh. Hasil PFI terhadap seluruh variabel prediktor dalam model terbaik untuk setiap algoritma ditunjukkan dalam Gambar 5. Terlihat bahwa variabel berat badan memiliki pengaruh paling besar terhadap model terbaik untuk setiap algoritma. Hal tersebut berarti berat badan memiliki pengaruh besar dalam penentuan BMI seseorang. Sebaliknya, variabel gender memiliki pengaruh paling kecil terhadap model terbaik untuk setiap algoritma. Selain itu, Gambar 5 menunjukkan bahwa hasil PFI untuk model KNN terbaik dan model SVM terbaik tidak memiliki perbedaan. Namun, terdapat sedikit perbedaan pada model *random forest classifier* terbaik yaitu variabel gender memiliki sedikit pengaruh terhadap model sedangkan variabel gender tidak memiliki pengaruh terhadap model terbaik untuk algoritma KNN dan SVM. Hal tersebut berarti gender seseorang tidak banyak atau sama sekali tidak menentukan BMI seseorang.



Gambar 5. Grafik *feature importance* untuk seluruh variabel prediktor pada model terbaik

5 Kesimpulan

Proses eksplorasi data menunjukkan bahwa variabel berat badan berkorelasi positif dengan indeks BMI, sedangkan variabel tinggi badan berkorelasi negatif dengan indeks BMI. Variabel berat badan memiliki korelasi paling kuat terhadap indeks BMI. Proses eksplorasi data juga menunjukkan bahwa tidak terjadi multikolinearitas antar variabel. Berdasarkan hasil evaluasi seluruh model, algoritma SVM merupakan algoritma yang paling akurat untuk memprediksi BMI berdasarkan nilai akurasi F1-*score*. Model terbaik juga diperoleh dari algoritma SVM yang telah dilakukan *tuning hyperparameter* tanpa penerapan teknik *oversampling*. Selanjutnya berdasarkan hasil metode *Permutation Feature Importance* (PFI) yang dilakukan pada model terbaik berdasarkan nilai akurasi F1-*score* tertinggi, berat badan merupakan variabel yang paling mempengaruhi indeks BMI dibandingkan variabel prediktor lainnya.

Daftar Pustaka

[1] L. A. Arini and I. K. Wijana, "Korelasi antara Body Mass Index (BMI) dengan Blood Pressure (BP) berdasarkan ukuran antropometri pada atlet," *Jurnal Kesehatan Perintis*, vol. 7, no. 1, pp. 32-40, 2020.

[2] M. F. A. Rasyid, "Pengaruh asupan kalsium terhadap Indeks Massa Tubuh (IMT)," *Jurnal Medika Hutama*, vol. 2, no. 4, pp. 1094-1097, 2021.

[3] Sugondo, Buku Ajar Penyakit Dalam, Jakarta: EGC, 2009.

[4] J. U. Lim, J. H. Lee, J. S. Kim, Y. I. Hwang, T. H. Kim, S. Y. Lim, K. H. Yoo, K. S. Jung, Y. K. Kim and C. K. Rhee, "Comparison of World Health Organization and Asia-Pacific body mass index classifications in COPD patients," *International Journal of COPD*, vol. 12, pp. 2465-2475, 2017.

[5] A. W. Kurniawan, R. Maulina and A. Fernandes, "Faktor yang berhubungan dengan berat badan kurang pada balita di Timor Leste," *Jurnal Kesehatan Vokasional*, vol. 7, no. 3, pp. 139-147, 2022.

[6] D. R. Kaparang, E. Padaunan and G. F. Kaparang, "Indeks massa tubuh dan lemak viseral mahasiswa," *AKSARA: Jurnal Ilmu Pendidikan Nonformal*, vol. 8, no. 3, pp. 1579-1586, 2022.

[7] A. Chahal and P. Gulia, "Machine learning and deep learning," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 12, pp. 4910-4914, 2019.

[8] A. A. Soofi and A. Awan, "Classification techniques in machine learning: applications and issues," *Journal of Basics & Applied Sciences*, vol. 13, pp. 459-465, 2017.

[9] W. Hidayat, M. Ardiansyah and A. Setyanto, "Pengaruh algoritma ADASYN dan SMOTE terhadap performa Support Vector Machine pada ketidakseimbangan dataset airbnb," *Edumatic: Jurnal Pendidikan Informatika,* vol. 5, no. 1, pp. 11-20, 2021.

[10] C. Sammut and G. I. Webb, Encyclopedia of Machine Learning, New York: Springer, 2010.

[11] I. W. Saputro and B. W. Sari, "Uji performa algoritma Naive Bayes untuk prediksi masa studi mahasiswa," *Citec Journal,* vol. 6, no. 1, pp. 1-11, 2019.

[12] D. F. Oktoriansah, "Klasifikasi BMI (Body Mass Index) berdasarkan tinggi dan berat badan menggunakan Logistic Regression," in *Seminar Nasional Pendidikan IPA dan Matematika 2023*, Malang, 2023.

[13] F. Amani, A. Mohammadnia, P. Amani, S. Abdollahi-Asl and M. Bahadoram, "Using machine learning method for classification body mass index of people for clinical decision," *Journal of Renal Endocrinology,* vol. 8, pp. 1-5, 2022.

[14] E. Rodriguez, E. Rodriguez, L. Nascimento, A. D. Silva and F. Marins, "Machine learning techniques to predict overweight or obesity," in *4th International Conference on Informatics & Data-Driven Medicine,* Aachen, 2021.

[15] F. Y. Sari, M. S. Kuntari, W. A. Yati and K. H, "Comparison of support vector machine performance with oversampling and outlier handling in diabetic disease detection classification," *Matrik: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer,* vol. 22, no. 3, pp. 539-552, 2023.

[16] S. A. Thamrin, D. Sidik, H. Kuswanto, A. Lawi and Ansariadi, "Exploration of obesity status of Indonesia Basic Health Research 2013 with Synthetic Minority Over-Sampling Techniques," *Indonesian Journal of Statistics and Its Applications,* vol. 5, no. 1, pp. 75-91, 2021.

[17] X. Mi, B. Zou, F. Zou and J. Hu, "Permutation-based identification of important biomarkers for complex diseases via machine learning models," *Nature Communications,* vol. 12, no. 3008, pp. 1-12, 2021.

[18] S. Diantika, "Penerapan teknik random oversampling untuk mengatasi imbalance class dalam klasifikasi website phishing menggunakan Algoritma LightGBM," *Jurnal Mahasiswa Teknik Informatika,* vol. 7, no. 1, pp. 19-25, 2023.

[19] Y. A. Sir and A. H. H. Soepranoto, "Pendekatan resampling data untuk menangani masalah ketidakseimbangan kelas," *J-ICON,* vol. 10, no. 1, pp. 31-38, 2022.

[20] L. Anshori, R. R. M. Putri and Tibyani, "Implementasi metode K-Nearest Neighbor untuk rekomendasi keminatan studi (studi kasus: jurusan teknik informatika Universitas Brawijaya)," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer,* vol. 2, no. 7, pp. 2745-2753, 2018.

[21] A. P. Agustin, A. C. Fauzan and Harliana, "Implementasi K-Nearest Neighbor dengan jarak Minkowski untuk deteksi dini COVID-19 pada citra CT-scan paru-paru," *Jurnal Ilmiah Intech: Information Technology Journal of UMUS,* vol. 4, no. 1, pp. 23-30, 2022.

[22] M. S. Fajri, N. Septian and E. Sanjaya, "Evaluasi implementasi algoritma machine learning K-Nearest Neighbor (KNN) pada data spektroskopi gamma resolusi rendah," *Al-Fiziya: Journal of Materials Science, Geophysics, Instrumentation and Theoretical Physics,* vol. 3, no. 1, pp. 9-14, 2020.

[23] S. R. Cholil, T. Handayani, R. Prathivi and T. Ardianita, "Implementasi algoritma klasifikasi K-Nearest Neighbor (KNN) untuk klasifikasi seleksi penerima beasiswa," *IJCIT: Indonesian Journal on Computer and Information Technology,* vol. 6, no. 2, pp. 118-127, 2021.

[24] G. Biau, "Analysis of a random forests model," *Journal of Machine Learning Research*, vol. 13, no. 1, pp. 1063-1095, 2012.

[25] T. Hastie, R. Tibshirani and J. Friedman, The Elements of Statistical Learning, 2nd ed., New York: Springer, 2008.

[26] B. Sartono and U. D. Syafitri, "Metode pohon gabungan: solusi pilihan untuk mengatasi kelemahan pohon regresi dan klasifikasi tunggal," *Forum Statistika Komputasi*, vol. 15, no. 1, pp. 1-7, 2010.

[27] A. S. Ritonga and E. S. Purwaningsih, "Penerapan metode Support Vector Machine (SVM) dalam klasifikasi kualitas pengelasan SMAW (Shield Metal Arc Welding)," *Jurnal Ilmiah Edutic,* vol. 5, no. 1, pp. 17-25, 2018.

[28] P. A. Octaviani, Y. Wilandari and D. Ispriyanti, "Penerapan metode klasifikasi Support Vector Machine (SVM) pada data akreditasi Sekolah Dasar (SD) di Kabupaten Magelang," *Jurnal Gaussian*, vol. 3, no. 4, pp. 811-820, 2014.

[29] I. M. Parapat, M. T. Furqon and Sutrisno, "Penerapan metode Support Vector Machine (SVM) pada klasifikasi penyimpangan tumbuh kembang anak," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer,* vol. 2, no. 10, pp. 3163-3169, 2018.

[30] E. Prasetyo, Data Mining Konsep dan Aplikasi Menggunakan MATLAB, Jakarta: Andi Publisher, 2012.

[31] S. Nurdiati, M. K. Najib, F. Bukhari, M. R. Ardhana, S. Rahmah and T. P. Blante, "Perbandingan AlexNet dan VGG untuk pengenalan ekspresi wajah pada dataset kelas komputasi lanjut," *Jurnal Techno.Com*, vol. 21, no. 3, pp. 500-510, 2022.

[32] H. J. P. Weerts, A. C. Mueller and J. Vanschoren, "Importance of Tuning Hyperparameters of Machine Learning Algorithms," 2020. [Online]. Available: http://arxiv.org/abs/2007.070588. [Accessed 15 November 2023].

[33] P. Probst and B. Bischl, "Tunability: Importance of Hyperparameters of Machine Learning Algorithms," 2019. [Online]. Available: http://jmlr.org/papers/v20/18-444.html. [Accessed 15 November 2023].

[34] H. Kaneko, "Cross-validated permutation feature importance considering correlation between features," *Analytical Science Advances*, vol. 3, no. 9, pp. 278-287, 2022.

[35] R. A. Wibowo and A. A. Kurniawan, "Analisis korelasi dalam penentuan arah antar faktor pada pelayanan angkutan umum di Kota Magelang," *Theta Omega: Journal of Electrical Engineering, Computer and Information Technology*, vol. 1, no. 2, pp. 45-50, 2020.

[36] M. Sriningsih, D. Hatidja and J. D. Prang, "Penanganan multikolinearitas dengan menggunakan analisis regresi komponen utama pada kasus impor beras di Provinsi SULUT," *Jurnal Ilmiah Sains*, vol. 18, no. 1, pp. 18-24, 2018.

[37] S. Choirunnisa, "Metode hibrida oversampling dan undersampling untuk menangani ketidakseimbangan data kegagalan akademik Universitas XYZ," Institut Teknologi Sepuluh Nopember, Surabaya, 2019.