

PERBANDINGAN KINERJA KNN, SVM, DAN ANN UNTUK MEMPREDIKSI LEVEL OBESITAS

Georgia Sugisandhea ¹⁾ Teny Handhayani ²⁾

¹⁾ Teknik Informatika Universitas Tarumanagara
Jl. Letjen S. Parman No. 1, Jakarta Barat 11440
email : geasandhea10@gmail.com

²⁾ Teknik Informatika Universitas Tarumanagara
Jl. Letjen S. Parman No. 1, Jakarta Barat 11440
email : tenyh@fti.untar.ac.id

ABSTRACT

This study aims to compare the performance of K-Nearest Neighbors (KNN), Artificial Neural Network (ANN), and Support Vector Machines (SVM), classification methods to find the best suited method to train a machine to classify someone to their group of obesity levels according to their eating habits and physical condition. This experiment uses the "Estimation of Obesity Levels based on Eating Habits and Physical Condition" dataset. The primary focus is on achieving high accuracy score and complex decision boundaries handling without minding the long training times, considering misclassification in the medical field might cause fatal consequences. This experiment's result shows that the SVM classification method with linear kernel provides the best overall performance for classifying obesity level, with the average accuracy of 0.944, precision of 0.944, recall of 0.942, and f1-score of 0.942. Notably, with the help of C kernel parameter of 200, the model teaches near-perfect performance evaluation scores that has the result of 0.99 score in accuracy, precision, recall, and f1-score.

Key words

ANN, classification, KNN, SVM

1. Pendahuluan

Obesitas adalah masalah gizi yang seringkali menjadi masalah dimana terjadi penimbunan lemak pada tubuh seseorang secara berlebihan [1]. Menurut data BKKP Kementerian Kesehatan Republik Indonesia, terdapat peningkatan prevalensi obesitas nasional dari 21.8% pada tahun 2018 ke 23.4% pada tahun 2023. Dimana DKI Jakarta berada pada urutan pertama provinsi dengan obesitas tertinggi sebesar 31.8%.

Obesitas seringkali disebabkan oleh gaya hidup, pola makan, dan pola aktivitas fisik yang tidak baik, seperti pola makan dengan berlebihan asupan yang dimakan, juga konsumsi banyaknya karbohidrat seperti nasi dan

lain lain secara bersamaan. Atau juga dapat disebabkan oleh makanan yang menjadi kesukaan masyarakat Indonesia seperti bakwan, tahu isi, dan banyak makanan gorengan lainnya yang dapat memicu obesitas karena kalori dan lemak jenuhnya yang cenderung tinggi [2].

Obesitas pun kemudian menjadi suatu penyakit tidak menular yang memprihatinkan karena banyaknya risiko yang muncul dari penyakit tersebut. Risiko penyakit kardiovaskular, kanker, hipertensi, dan diabetes menjadi beberapa dari banyak penyakit yang dapat disebabkan oleh obesitas [2].

Penelitian ini dikembangkan dengan harapan untuk menemukan metode *machine learning* yang mampu memprediksi dengan akurat level obesitas seseorang menggunakan data pola makan dan kondisi fisiknya. Sehingga dari keluaran yang didapatkan dari hasil pelatihan mesin tersebut dapat membantu seseorang mengetahui tingkat obesitasnya dan mengambil langkah perbaikan pola hidup dan pola makan yang sesuai untuk mencegah segala kemungkinan risiko obesitas sebelum terlambat.

2. Metode Penelitian

Penelitian eksperimen perbandingan metode klasifikasi KNN, SVM, dan ANN untuk memprediksi level obesitas untuk menemukan metode *machine learning* yang paling baik untuk memproses data dan menghasilkan mesin yang mampu memprediksi level obesitas seseorang secara akurat. Eksperimen ini dilaksanakan dengan urutan langkah pengumpulan data, pra-pemrosesan data, splitting data uji dan data latihan, pelatihan model, dan evaluasi model.

2.1 Pengumpulan Data

Dataset yang digunakan pada penelitian ini adalah *dataset* "Estimation of Obesity Levels Based On Eating Habits and Physical Condition" yang dilisensi oleh Creative Commons Attribution 4.0 International. Jumlah

data yang digunakan adalah 2111 data, dan masing masing memiliki 17 variabel fitur. Variabel yang ada dalam dataset yang digunakan dalam penelitian ini antara lain:

- Gender: jenis kelamin personil data (Female/Male)
- Age: usia personil data dalam tahun
- Height: tinggi badan personil data dalam meter
- Weight: berat badan personil data dalam kilogram
- Family_history_with_overweight: riwayat keluarga dengan masalah *overweight* (yes/no)
- FAVC = frekuensi personil data dalam mengonsumsi makanan berkalori tinggi (yes = sering/no = jarang/tidak)
- FCVC = frekuensi konsumsi sayur dalam *range* 1-3, dimana semakin tinggi semakin sering frekuensinya
- NCP = jumlah kali makan utama per hari
- CAEC = frekuensi konsumsi makanan di luar jam makan (no, Sometimes, Frequently, Always)
- SMOKE = status merokok personil data (yes/no)
- CH2O = jumlah air yang diminum per hari
- SCC = apakah personil data memonitor konsumsi kalori dirinya sendiri (yes/no)
- FAF = frekuensi aktivitas fisik personil data
- TUE = frekuensi penggunaan perangkat elektronik seperti ponsel, televisi, komputer, dan lain lain
- CALC = frekuensi konsumsi alkohol (no, Sometimes, Frequently, Always)
- MTRANS = penggunaan moda transportasi utama (Public Transportation, Automobile, Walking, Bike, Motorbike)
- NObeyesdad = kategori status berat badan personil data (Insufficient_Weight, Normal_Weight, Overweight_Level_I, Overweight_Level_II, Obesity_Type_I, Obesity_Type_II, dan Obesity_Type_III)

2.2 Pra-pemrosesan data

Dikarenakan oleh *dataset* yang digunakan masih memiliki variabel yang berisi nilai *string*, maka agar *dataset* tersebut mampu digunakan dan diproses oleh mesin sesuai dengan metode yang digunakan, maka data akan diproses dengan *Category Encoder* untuk mengkonversi masing masing nilai variabel kategoris menjadi *integer* yang bisa digunakan untuk pelatihan dan pengujian mesin [3].

2.3 Splitting Data Uji dan Data Latih

Dataset yang digunakan dipisahkan menjadi data latih dan data uji. Dimana pemisahan data tersebut dilakukan dengan menggunakan metode *k-fold cross-validation* dengan $k = 5$.

2.3.1 Data Latih

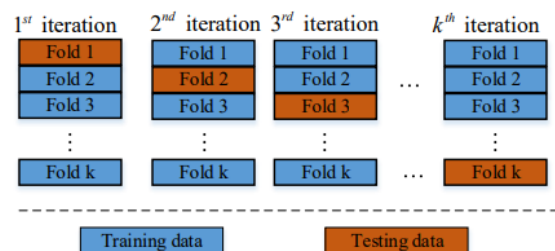
Dataset yang sudah disisihkan menjadi data latih adalah data yang digunakan untuk mengajarkan mesin pola pola data, sehingga mesin mampu mengetahui cara untuk mengklasifikasi data menjadi kelas Normal_Weight, Overweight_Level_I, Overweight_Level_II, Obesity_Type_I, Obesity_Type_II, Obesity_Type_III, dan Insufficient_Weight [4].

2.3.2 Data Uji

Dataset yang sudah disisihkan menjadi data uji kemudian dipakai untuk menguji dan mengevaluasi kinerja dari mesin yang sudah dilatih sebelumnya. Pengujian dilaksanakan menggunakan data uji untuk melihat kinerja mesin dalam mengklasifikasikan data uji ke kategori yang benar [4].

2.3.1 K-Fold Cross-Validation

Dalam metode *k-fold cross-validation*, *dataset* dibagi menjadi lipatan sebanyak k . Sebuah lipatan digunakan menjadi data uji dalam satu putaran, dimana sisanya menjadi data latih pada putaran tersebut, seperti yang ditampilkan pada Gambar 1. Dan proses tersebut dilaksanakan berulang kali sampai masing masing lipatan sudah menjadi data uji [5].



Gambar 1. K-Fold Cross Validation [5]

Kelebihan dari penggunaan *k-fold cross validation* adalah mampu mencapai hasil rata rata kesalahan dari iterasi berjumlah k yang lebih rendah daripada cara pembagian data yang hanya membagi data menjadi data training dan data testing saja dan juga risiko dari overfitting menjadi minimal karena kombinasi pelatihan data yang berbeda beda [6][7][8]

2.4 Pelatihan Model

Eksperimen ini menggunakan 3 metode, yaitu KNN, ANN, dan SVM. Kemudian ketiga metode tersebut akan di evaluasi untuk mendapatkan metode terbaik untuk memprediksi level obesitas seseorang berdasarkan *dataset* yang digunakan.

2.4.1 K-Nearest Neighbor (KNN)

Metode klasifikasi nearest neighbor merupakan metode untuk mencari label target data yang kemudian

akan didapatkan dari informasi label pola data lain yang berada di dekat target tersebut [9]. Dimana angka K yang dipilih adalah jumlah pola data terdekat yang akan diambil kemudian dilihat mayoritas labelnya untuk menentukan label target yang dicari seperti yang ditampilkan pada Gambar 2.

Metode KNN memiliki langkah:

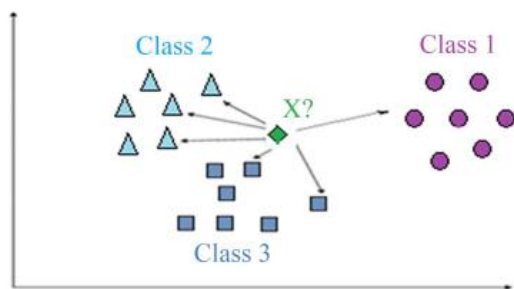
- Menentukan nilai dari K yang adalah nilai bilangan bulat positif
- Menghitung jarak antara data training dan data target dengan rumus:

$$D = \sqrt{(x1 - y1)^2 + (x2 - y2)^2} \dots(1)[10]$$

Dimana:

- x : data sampel
- y : data target
- D : jangkauan

- Menentukan urutan jangkauan paling dekat
- Menggunakan label tetangga terdekat sejumlah K untuk memprediksi label data target [10]



Gambar 2. Penentuan Kelas K-Nearest Neighbor [11]

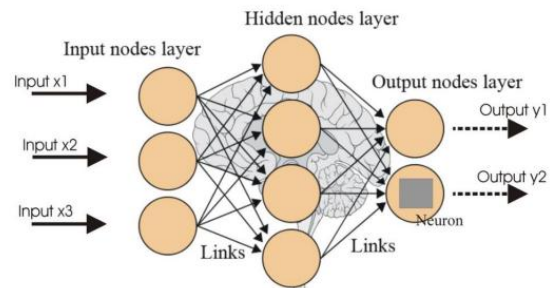
Kelebihan dari algoritma KNN adalah bahwa metode ini efektif untuk menghasilkan data yang akurat dari data yang jumlah banyak. Namun metode ini memiliki kekurangan berupa biaya komputasi yang tinggi dalam perhitungan jarak dan penentuan nilai k yang optimal [10].

2.4.2 Artificial Neural Network (ANN)

Metode ANN adalah suatu metode untuk memproses informasi dimana prosesnya menyerupai konsep biologi dari konsep saraf manusia dalam memproses informasi. Sehingga ANN menjadi model komputasi berbentuk struktur saraf yang terdiri dari ratusan neuron yang saling berhubungan untuk menyelesaikan masalah yang kompleks [12].

Neuron yang ada di dalam struktur ANN tersusun menjadi dua lapisan atau lebih, dimana masing masing berinteraksi dengan satu sama lain dengan koneksi yang bebannya telah ditimbang. Lapisan lapisan pada struktur terdiri dari lapisan *input*, *output*, dan *hidden* seperti yang ditunjukkan pada Gambar 3 [13]. Dimana lapisan input data akan dimasukkan, dan lapisan *output* memberikan hasil pemrosesan data dikeluarkan. Diantara kedua

lapisan tersebut, terdapat *hidden layers* yang menjadi lapisan perantara lapisan *input* dan *output*, dimana lapisan ini akan memproses data lebih dalam sesuai kebutuhan [14]. Dimana masing masing layer yang saling berhubungan hanya akan menerima informasi yang diberikan oleh lapisan sebelumnya [15].



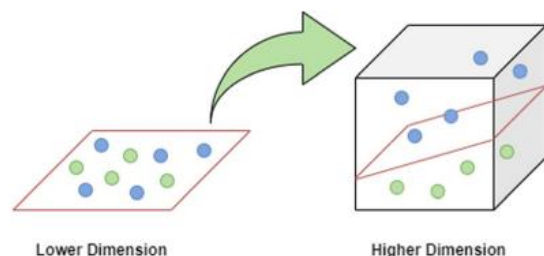
Gambar 3. Struktur Lapisan ANN [13]

Kelebihan dari algoritma ANN adalah mampu untuk memecahkan permasalahan yang kompleks, dikarenakan variabel yang banyak dan data yang besar dan non linier [16].

2.4.3 Support Vector Machine (SVM)

Support Vector Machine merupakan algoritma klasifikasi dan regresi dengan metode mencari *hyperplane* yang memiliki margin tertinggi yang mampu untuk membagi masing masing kelas secara linear [17]. Metode ini membuat garis batas terbaik yang tidak terbatas pada ruang 2 dimensi, tapi mampu mengisolasi ruang n-dimensi menjadi kelas kelas yang mudah untuk mengklasifikasi titik informasi baru, seperti yang ditampilkan pada Gambar 4 [18].

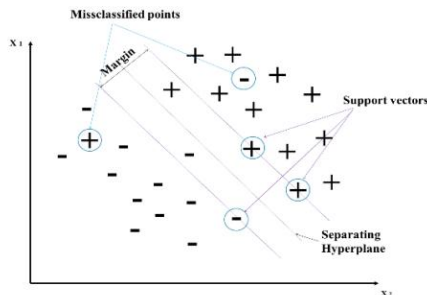
Metode SVM terbagi menjadi dua jenis, yaitu linier dan non-linier. SVM linier digunakan pada *dataset* dapat terbagi menjadi dua kelas yang dibatasi oleh satu garis lurus [18], yaitu metode SVM dengan kernel linear. Sedangkan SVM non-linier digunakan pada *dataset* yang tidak dapat dipisahkan oleh satu garis lurus secara langsung [18], yaitu metode SVM dengan kernel RBF dan dengan kernel polynomial [19].



Gambar 4. Pembagian Data dari Dimensi Rendah ke Dimensi Tinggi untuk SVM Non-Linier [18]

Penggunaan SVM dalam mengklasifikasikan data non-linier berarti data tersebut tidak dapat dipisahkan secara linier. Dalam kasus tersebut, beberapa titik data

dapat melintasi margin pembagian data linier sehingga terjadi kesalahan klasifikasi seperti ditunjukkan pada Gambar 5. Penggunaan parameter regularisasi, C, digunakan untuk menyelesaikan masalah tersebut. Jumlah titik data yang boleh melanggar margin batas dibatasi dengan parameter C tersebut [20].



Gambar 5. Titik Data Melintasi Margin Pembagian [20]

Kelebihan dari algoritma SVM adalah mampu memodelkan hubungan antara masing masing data dengan baik karena kemampuan SVM dalam memetakan data dalam ruang dimensi yang lebih tinggi [21].

2.5 Evaluasi Model

Evaluasi model pada masing masing metode dilaksanakan untuk mengetahui seberapa baik metode tersebut dalam pengujian setelah dilatih dan menemukan metode terbaik dalam memprediksi level obesitas seseorang. Evaluasi model dilaksanakan dengan library *sklearn.metrics classification_report*, dimana hasil nilai yang dikonsiderasi adalah *accuracy, precision, recall*, dan *f1-score*.

Accuracy atau akurasi adalah nilai seberapa baik model dapat mengklasifikasi data dengan benar, atau bisa dikatakan dengan persentase keberhasilan model dalam mengklasifikasi kelas data uji [4].

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \dots\dots\dots(2) [22]$$

Dimana:

- TP (True Positive) = Jumlah data yang diprediksi positif yang memang positif
- TN (True Negative) = Jumlah data yang diprediksi negative yang memang negatif
- FP (False Positive) = Jumlah data diprediksi positif yang sebenarnya negatif
- FN (False Negative) = Jumlah data diprediksi negatif yang sebenarnya positif

Precision atau presisi adalah nilai seberapa nilai positif yang dihasilkan oleh algoritma memang benar benar positif [18]

$$Precision = \frac{TP}{TP+FP} \dots\dots\dots(3)[22]$$

Dimana:

- TP (True Positive) = Jumlah data diprediksi positif yang memang positif
- FP (False Positive) = Jumlah data diprediksi positif yang sebenarnya tidak positif

Recall adalah nilai seberapa banyak nilai data uji positif yang berhasil diprediksi algoritma dengan benar [19].

$$Recall = \frac{TP}{TP+FN} \dots\dots\dots(4)[22]$$

Dimana:

- TP (True Positive) = Jumlah data yang diprediksi positif yang memang positif
- FN (False Negative) = Jumlah data diprediksi negatif yang sebenarnya positif

F1-score adalah nilai yang mempertimbangkan nilai *precision* dan nilai *recall* untuk menghitung kinerja dari model tersebut [22].

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \dots\dots\dots(5)[22]$$

3. Hasil dan Pembahasan

Evaluasi model dilaksanakan dengan melihat *accuracy, precision, recall*, dan *F1-Score*, dari hasil pelatihan dan pengujian *dataset* yang digunakan menggunakan metode *K-Nearest Neighbor, Artificial Neural Network*, dan *Support Vector Machine*, dimana data dibagi dengan metode *K-Fold Cross Validation*. Evaluasi masing masing model tersebut juga mengkonsiderasi waktu pelatihan dari masing masing model.

3.1 K-Nearest Neighbor (KNN)

Hasil pengujian semua *fold* dalam *dataset* menggunakan metode K-Nearest Neighbor beserta rata ratanya tampak pada tabel 1.

Tabel 1. Hasil Pengujian KNN

Pengujian fold ke-	Accuracy	Precision	Recall	F1-Score
1	0.83	0.82	0.82	0.81
2	0.87	0.87	0.87	0.86
3	0.89	0.90	0.89	0.89
4	0.88	0.88	0.87	0.87
5	0.91	0.91	0.90	0.89
Rata-rata	0.876	0.876	0.87	0.864

Dan waktu yang digunakan dalam pelatihan masing masing *fold* dalam *dataset* beserta rata ratanya tampak pada tabel 2.

Tabel 2. Waktu Pelatihan KNN

Pengujian fold ke-	Waktu Pelatihan
--------------------	-----------------

1	0.016129
2	0.006134
3	0.011994
4	0.006219
5	0.010106
Rata-rata	0.0101164

Melalui hasil evaluasi pengujian dan pelatihan dataset menggunakan metode K-Nearest Neighbor yang terdapat pada tabel 1, dapat dilihat bahwa hasil *accuracy*, *precision*, *recall*, dan *f1-score* dari masing masing *fold* tidak saling berdekatan. *Accuracy* minimum terletak di angka 0.83 pada *fold* 1, dan angka maksimum bernilai 0.91 pada *fold* 5. *Precision* minimum juga terletak pada *fold* 1 dengan angka 0.82, dan yang paling maksimum ada pada nilai 0.91 di *fold* 5. Sedangkan untuk nilai *recall*, nilai minimum berada pada *fold* 1 dengan nilai 0.82, dan nilai maksimum berada pada *fold* 5 dengan angka 0.90. Dan untuk nilai *f1-score*, nilai minimumnya adalah 0.81 yang terletak pada hasil pengujian *fold* 1, dan nilai maksimumnya adalah 0.89 yang terletak pada hasil pengujian *fold* 3 dan 5.

Rata-rata dari keseluruhan pengujian berada pada angka yang cukup berdekatan, yaitu *accuracy* pada angka 0.867, *precision* di nilai 0.876, *recall* pada nilai 0.87, dan nilai *f1-score* sebesar 0.864. Angka yang saling berdekatan ini menunjukkan bahwa model stabil dan cukup baik dalam memprediksi semua kelas yang ada, tanpa bias kepada salah satu kelas. Hasil masing masing nilai pun berarti hasil sudah baik untuk memprediksi tingkat obesitas seseorang, namun nilai tersebut masih bisa ditingkatkan untuk digunakan dalam prediksi kelas dalam lingkup kesehatan.

Waktu maksimum yang digunakan oleh model dalam melaksanakan pelatihan ada pada saat melatih *fold* 1, yaitu 0.016129 detik. Sedangkan waktu tercepat yang digunakan dalam pelatihan ada saat melatih *fold* 2 yaitu 0.006134 detik. Rata-rata dari waktu pelatihan model KNN ini pun cenderung sangat singkat, berada pada 0.0101164 detik. Artinya, waktu yang dibutuhkan untuk pelatihan model tersebut tidak begitu banyak.

3.2 Artificial Neural Network

Hasil pengujian semua *fold* dalam dataset menggunakan metode Artificial Neural Network beserta rata ratanya tampak pada tabel 3.

Tabel 3. Hasil Pengujian ANN

Pengujian fold ke-	Accuracy	Precision	Recall	F1-Score
1	0.71	0.71	0.71	0.70
2	0.82	0.82	0.81	0.81
3	0.83	0.83	0.83	0.82
4	0.86	0.87	0.86	0.86
5	0.85	0.85	0.85	0.85
Rata-rata	0.814	0.816	0.812	0.808

Dan waktu yang digunakan dalam pelatihan masing masing *fold* dalam dataset beserta rata ratanya tampak pada tabel 4.

Tabel 4. Waktu Pelatihan ANN

Pengujian fold ke-	Waktu Pelatihan
1	2.027073
2	5.938043
3	5.457058
4	8.201298
5	4.447272
Rata-rata	5.2141488

Dengan melihat hasil evaluasi pelatihan dan pengujian dataset dengan metode Artificial Neural Network pada tabel 3, dapat dilihat bahwa hasil *accuracy*, *precision*, *recall*, dan *f1-score* oleh masing masing pengujian *fold* tidak terlalu mirip. Nilai minimum dari *accuracy*, *precision*, *recall*, dan *f1-score* semuanya terletak pada *fold* 1, yaitu 0.71, 0.71, 0.71, dan 0.70. Sedangkan nilai maksimum dari keempat nilai tersebut, semuanya terletak pada *fold* 4, yaitu 0.86, 0.87, 0.86, dan 0.86.

Rata rata dari keseluruhan pengujian berada pada tingkat yang cukup stabil, karena nilai dari *accuracy*, *precision*, *recall*, dan *f1-score*, yang saling berdekatan. Rata rata dari nilai *accuracy* berada pada nilai 0.814, *precision* pada nilai 0.816, *recall* di nilai 0.812, dan *f1-score* di nilai 0.808. Nilai nilai ini menunjukkan bahwa model sudah cukup stabil dan cukup baik dalam memprediksi tingkat obesitas seseorang, dan hasil prediksinya cukup rata tanpa bias pada kelas tertentu. Namun, hasil ini masih bisa dikembangkan untuk menjadi model yang lebih layak untuk digunakan dalam dunia kesehatan.

Waktu yang digunakan dalam pelatihan model ini memiliki rata rata sebesar 5.2141488 detik. Waktu maksimum yang digunakan dalam pelatihan berada pada *fold* 4 sebesar 8.201298 detik, sedangkan waktu tersingkat yang digunakan berada pada *fold* 1 sebesar 2.027073 detik. Maka artinya waktu yang digunakan dalam pelatihan ANN cenderung cukup singkat.

3.3 Support Vector Machine Kernel RBF

Hasil pengujian semua *fold* dalam dataset menggunakan metode Support Vector Machine kernel RBF beserta rata ratanya tampak pada tabel 5.

Tabel 5. Hasil Pengujian SVM Kernel RBF

Pengujian fold ke-	C	Accuracy	Precision	Recall	F1-Score
1	5	0.61	0.62	0.61	0.60
2	50	0.78	0.78	0.78	0.78
3	70	0.77	0.78	0.77	0.76
4	170	0.85	0.85	0.85	0.85
5	200	0.86	0.87	0.86	0.85

Rata-rata	0.774	0.78	0.774	0.768
-----------	-------	------	-------	-------

Dan waktu yang digunakan dalam pelatihan masing masing *fold* dalam dataset beserta rata ratanya tampak pada tabel 6.

Tabel 6. Waktu Pelatihan SVM Kernel RBF

Pengujian fold ke-	Waktu Pelatihan
1	0.068859
2	0.198321
3	0.141629
4	0.306008
5	0.096715
Rata-rata	0.1623064

Seperti dapat dilihat pada tabel 5, bahwa penggunaan nilai variabel C sangat berpengaruh kepada hasil akhir dari pengujian secara keseluruhan. Penggunaan variabel C terendah yaitu 5 menghasilkan hasil *accuracy*, *precision*, *recall*, dan *f1-score* yang sangat rendah dan tidak layak untuk digunakan dalam dunia kesehatan, yaitu 0.61, 0.62, 0.61, dan 0.60. Seiring naiknya variable C, nilai dari *accuracy*, *precision*, *recall*, dan *f1-score* pun ikut meningkat secara perlahan. Namun kenaikan tersebut sempat mengalami penurunan pada saat menggunakan variabel C sebesar 70, walaupun hanya sedikit, yaitu penurunan sebesar 0.01 pada nilai *accuracy* dan *recall*, dan penurunan sebesar 0.02 pada nilai *f1-score*. Penggunaan variabel C tertinggi yaitu 200 menghasilkan hasil *accuracy*, *precision*, *recall*, dan *f1-score* yang baik, yaitu 0.86, 0.87, 0.86, dan 0.85. Nilai nilai ini mengartikan bahwa penggunaan variabel C sebesar 200 pada pelatihan dataset menggunakan model SVM dengan kernel RBF sudah cukup baik untuk digunakan dalam memprediksi tingkat obesitas dari seseorang. Namun rata rata dari hasil pelatihan dan pengujian dataset menggunakan SVM kernel RBF menunjukkan angka yang cenderung rendah, yaitu *accuracy* sebesar 0.774, *precision* sebesar 0.78, *recall* sebesar 0.774, dan *f1-score* sebesar 0.768. Nilai tersebut menjadi nilai yang rendah dan tidak layak, bahkan berbahaya, untuk digunakan dalam dunia kesehatan karena masih mampu memberikan kesalahan yang cukup banyak dalam mengklasifikasi seseorang.

Waktu yang digunakan model dalam pelatihan cenderung sedikit, seperti yang ditunjukkan dalam tabel 6. Waktu tersingkat yang digunakan oleh model terdapat pada pelatihan *fold* 5, yaitu 0.096715 detik, sedangkan waktu terpanjang yang digunakan oleh model digunakan pada saat pelatihan *fold* 4, yaitu 0.306008 detik. Secara rata rata, pelatihan menggunakan 0.1623064 detik, yang berarti pelatihan menggunakan SVM kernel RBF tidak membutuhkan waktu yang begitu banyak.

3.4 Support Vector Machine Kernel Linear

Hasil pengujian semua *fold* dalam dataset menggunakan metode Support Vector Machine kernel Linear beserta rata ratanya tampak pada tabel 7.

Tabel 7. Hasil Pengujian SVM Kernel Linear

Pengujian fold ke-	C	Accuracy	Precision	Recall	F1-Score
1	5	0.81	0.81	0.80	0.81
2	50	0.97	0.97	0.97	0.97
3	70	0.97	0.97	0.97	0.97
4	170	0.98	0.98	0.98	0.97
5	200	0.99	0.99	0.99	0.99
Rata-rata		0.944	0.944	0.942	0.942

Dan waktu yang digunakan dalam pelatihan masing masing *fold* dalam dataset beserta rata ratanya tampak pada tabel 8.

Tabel 8. Waktu Pelatihan SVM Kernel Linear

Pengujian fold ke-	Waktu Pelatihan
1	0.600944
2	10.903727
3	10.889620
4	9.908126
5	17.938797
Rata-rata	10.0482428

Pada tabel 6, terlihat bahwa perubahan nilai C dari nilai 5 ke nilai 50 memberikan perubahan nilai yang cukup signifikan pada hasil pengujian. Namun ketika nilai C berada pada nilai 50 sampai 200, hasil pengujiannya tidak memberikan perubahan yang tidak terlalu signifikan. Namun, pada nilai C tersebut, memang hasil dari pengujiannya sudah mendekati sempurna, dimana nilainya berkisar dari 0.97 sampai 0.99. Nilai C sebesar 200 pada *fold* 5 menghasilkan nilai *accuracy*, *precision*, *recall*, dan *f1-score* yang semuanya 0.99. Hasil rata rata dari keseluruhan pengujian pun memberikan hasil yang sangat baik dan layak untuk digunakan dalam bidang kesehatan, yaitu *accuracy* sebesar 0.944, *precision* sebesar 0.944, *recall* sebesar 0.942, dan *f1-score* sebesar 0.942.

Waktu yang digunakan SVM kernel Linear dalam pelatihan cukup singkat, yaitu dengan rata-rata 10.0482428 detik. Waktu terlama yang digunakan pada saat pelatihan berada pada *fold* 5, yaitu 17.938797 detik, dan waktu tersingkat yang digunakan ada pada *fold* 1, yaitu 0.600944. Maka artinya penggunaan SVM kernel Linear dalam melatih data hanya membutuhkan waktu yang cukup singkat.

3.5 Support Vector Machine Kernel Polynomial

Hasil pengujian semua *fold* dalam dataset menggunakan metode Support Vector Machine kernel Polynomial beserta rata ratanya tampak pada tabel 9.

Tabel 9. Hasil Pengujian SVM Kernel Polynomial

Peng-ujian fold ke-	C	Accuracy	Precision	Recall	F1-Score
1	5	0.66	0.66	0.66	0.65
2	50	0.79	0.79	0.79	0.79
3	100	0.80	0.80	0.80	0.79
4	200	0.83	0.83	0.82	0.82
5	250	0.86	0.87	0.85	0.85
Rata-rata		0.788	0.79	0.784	0.78

Dan waktu yang digunakan dalam pelatihan masing *fold* dalam dataset beserta rata ratanya tampak pada tabel 10.

Tabel 10. Waktu Pelatihan SVM Kernel Polynomial

Pengujian fold ke-	Waktu Pelatihan
1	0.261904
2	0.124999
3	0.072325
4	0.384906
5	0.084686
Rata-rata	0.185764

Dilihat dari hasil pengujian yang terdapat pada tabel 9, perubahan penggunaan nilai C memberikan perubahan nilai hasil pengujian searah, tapi tidak memberikan perubahan yang terlalu signifikan, Penggunaan C terendah sebesar 5 memiliki hasil pengujian yang tidak bagus, yaitu *accuracy* sebesar 0.66, *precision* sebesar 0.66, *recall* sebesar 0.66, dan *f1-score* sebesar 0.65. Hasil ini sangat tidak layak untuk digunakan dalam bidang kesehatan, karena angka tersebut menunjukkan bahwa model masih terlalu sering membuat kesalahan dalam melakukan prediksi kelas obesitas dari seseorang. Angka terbaik yang dihasilkan oleh model ini adalah pada pengujian *fold* ke-5 dengan nilai C sebesar 250, yaitu *accuracy* sebesar 0.86, *precision* sebesar 0.87, *recall* sebesar 0.85, dan *f1-score* sebesar 0.85. Nilai ini mengatakan bahwa model ini sudah baik untuk digunakan dalam memprediksi kelas obesitas dari seseorang, karena tingkat kesalahannya tidak cukup besar, namun nilai tersebut bisa lebih di tingkatkan, agar dapat menjadi lebih aman ketika digunakan dalam bidang dunia kesehatan, dimana kesalahan prediksi bisa menjadi kesalahan yang kritis.

Waktu yang digunakan oleh model dalam melaksanakan pelatihan cenderung sangat singkat, karena semua waktu pelatihan dari semua *fold* tidak ada yang melewati 0.5 detik. Waktu tersingkat yang digunakan oleh model adalah saat melaksanakan pelatihan *fold* 3, yaitu 0.072325 detik, dan waktu terpanjang yang digunakan oleh model adalah saat melaksanakan pelatihan pada *fold*

1, yaitu 0.261904. Rata rata waktu dari keseluruhan pelatihan adalah 0.185764 detik per *fold*, dan nilai tersebut menunjukkan bahwa pelatihan model menggunakan SVM kernel polynomial tidak membutuhkan waktu yang terlalu lama.

3.6 Perbandingan Semua Model

Rata rata dari hasil pengujian dari model KNN, ANN, dan SVM dikompilasi dan dapat dilihat serta dibandingkan pada tabel 11, dan rata rata waktu pengujian dari masing masing model dapat dilihat pada tabel 12.

Tabel 11. Perbandingan Rata Rata Semua Model Klasifikasi

Model	Accuracy	Precision	Recall	F1-Score
KNN	0.876	0.876	0.87	0.864
ANN	0.814	0.816	0.812	0.808
SVM RBF	0.774	0.78	0.774	0.768
SVM Linear	0.944	0.944	0.942	0.942
SVM Polynomial	0.788	0.79	0.784	0.78

Tabel 12. Perbandingan Rata Rata Waktu Semua Model Klasifikasi

Pengujian fold ke-	Waktu Pelatihan
KNN	0.0101164
ANN	5.2141488
SVM RBF	0.1623064
SVM Linear	10.0482428
SVM Polynomial	0.185764

Melalui tabel 11, dapat dilihat bahwa model yang memiliki nilai keseluruhan yang paling tinggi adalah SVM dengan kernel Linear, dimana nilai *accuracy*, *precision*, *recall*, dan *f1-score* semuanya berada di atas 0.94, sedangkan model model lain tidak ada yang melewati nilai 0.9 pada nilai evaluasi apapun. Maka, dalam tujuan untuk memprediksi kelas obesitas seseorang, model klasifikasi SVM linear menjadi model yang paling baik untuk memenuhi tujuan tersebut. Dan model yang memiliki nilai terendah dibandingkan yang lain adalah SVM dengan kernel RBF, karena nilai *accuracy*, *precision*, *recall*, dan *f1-score* dari model tersebut semuanya berada di angka 0.78 atau lebih rendah. Namun model model yang lain semuanya berada pada nilai 0.78 atau lebih tinggi, sehingga SVM dengan kernel RBF menjadi model yang paling tidak cocok untuk memprediksi kelas obesitas seseorang.

Dalam membandingkan rata rata waktu pelatihan semua model klasifikasi yang digunakan dalam penelitian ini, dapat dilihat pada tabel 12 bahwa model pelatihan yang memakan waktu paling banyak adalah model SVM linear dengan waktu sebanyak 10.0482428 detik, walaupun waktu tersebut masih tergolong dengan waktu pelatihan yang cukup singkat. Sedangkan model

klasifikasi yang menggunakan waktu yang paling singkat dalam pelatihannya adalah KNN dengan waktu sebesar 0.0101164 detik. Namun SVM dengan kernel RBF dan SVM dengan kernel polynomial pun tidak memakan waktu yang jauh berbeda dari KNN, yaitu pada waktu 0.1623064 detik dan 0.185764 detik. Maka secara keseluruhan, semua, model yang digunakan dalam penelitian ini tidak menggunakan waktu yang begitu lama dalam melaksanakan pelatihannya.

4. Kesimpulan

Berdasarkan hasil pelatihan, pengujian, serta evaluasi yang telah dilaksanakan, maka kesimpulan yang dapat ditarik adalah bahwa metode yang paling baik untuk digunakan dalam memprediksi tingkat obesitas seseorang dengan fokus utama adalah kebaikan dan kestabilan model tersebut dalam mengklasifikasi masing masing data adalah penggunaan metode SVM dengan kernel linear, yang menggunakan nilai parameter kernel C sebesar 200. Metode dan nilai parameter tersebut berhasil memberikan hasil nilai *accuracy*, *precision*, *recall*, dan *f1-score* yang hampir sempurna, yaitu 0.99 pada semua nilainya, seperti yang dapat dilihat pada tabel 7.

Selain itu, walaupun penggunaan nilai parameter kernel C sebesar 200 pada metode SVM dengan kernel linear memberikan hasil kinerja yang terbaik, rata rata nilai *accuracy*, *precision*, *recall*, dan *f1-score* dari hasil pengujian model SVM dengan kernel linear juga secara konsisten lebih tinggi daripada metode yang lainnya. Seperti yang ditampilkan pada tabel 11, metode selain SVM dengan kernel linear tidak ada yang memiliki nilai *accuracy*, *precision*, *recall*, dan *f1-score* yang menyentuh angka 0.9, sedangkan metode SVM dengan kernel linear memiliki nilai rata rata *accuracy*, *precision*, *recall*, dan *f1-score* yang semuanya berada di atas nilai 0.9, yaitu 0.944, 0.944, 0.942, dan 0.942. Hal ini menunjukkan bahwa model memiliki keunggulan yang tidak terbatas hanya pada satu konfigurasi parameter saja, melainkan memang secara umum memiliki kinerja yang stabil dan unggul.

Namun demikian, dapat dilihat pada tabel 12 bahwa penggunaan model SVM dengan kernel linear yang memiliki hasil kinerja dengan nilai tertinggi, juga memiliki rata-rata penggunaan waktu pelatihan yang tertinggi juga dibandingkan dengan metode lainnya, yaitu dengan rata rata penggunaan waktu pelatihan sebesar 10.0482428 detik. Sedangkan metode dengan rata-rata penggunaan waktu pelatihan yang tersingkat adalah metode KNN dengan rata rata waktu pelatihan sebanyak 0.0101164 detik. Hasil evaluasi dari penggunaan metode KNN adalah kedua paling tertinggi setelah metode SVM dengan kernel linear. Hasil evaluasi dari penggunaan metode KNN memiliki nilai *accuracy* sebesar 0.876, *precision* sebesar 0.876, *recall* sebesar 0.87, dan *f1-score* sebesar 0.864. Hasil tersebut cukup tinggi dibandingkan dengan metode lainnya selain metode SVM dengan kernel linear, dimana metode lainnya memiliki nilai rata rata *accuracy*, *precision*, *recall*, dan *f1-score* sebesar 0.816 atau lebih rendah.

Penggunaan waktu pelatihan yang paling singkat dan nilai evaluasi yang cukup baik oleh KNN berarti metode tersebut menjadi model yang cukup ideal dalam memprediksi kelas dengan efisiensi waktu pelatihan yang tinggi. Namun, dalam konteks penerapan di bidang kesehatan yang menuntut kestabilan dan kinerja model yang optimal – karena kesalahan klasifikasi bisa berdampak fatal – model KNN masih kurang ideal untuk diimplementasikan berdasarkan hasil penelitian ini.

Di sisi lain, walau metode SVM dengan kernel linear memiliki rata rata penggunaan waktu pelatihan yang paling tinggi dibandingkan model lain, waktunya masih tergolong cukup singkat dan tetap layak untuk dipertimbangkan, terutama karena hasil evaluasinya yang sangat baik. Oleh karena itu, SVM dengan kernel linear menjadi pilihan yang tepat untuk memprediksi level obesitas dari seseorang dengan andal dan akurat.

REFERENSI

- [1] S. Kurnia Saraswati *et al.*, “MEDIA KESEHATAN MASYARAKAT INDONESIA Literature Review: Faktor Risiko Penyebab Obesitas,” *MEDIA KESEHATAN MASYARAKAT INDONESIA*, 2020.
- [2] A. Az-Zahra, I. Muyassar, and S. Maharani, “Pengaruh Gaya Hidup Terhadap Kejadian Obesitas di Indonesia,” 2022.
- [3] W. Zhu, R. Qiu, and Y. Fu, “Comparative Study on the Performance of Categorical Variable Encoders in Classification and Regression Tasks,” Jan. 2024, [Online]. Available: <http://arxiv.org/abs/2401.09682>
- [4] S. Rabbani, D. Safitri, N. Rahmadhani, A. A. F. Sani, and M. K. Anam, “Perbandingan Evaluasi Kernel SVM untuk Klasifikasi Sentimen dalam Analisis Kenaikan Harga BBM,” *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 3, no. 2, 2023, doi: 10.57152/malcom.v3i2.897.
- [5] I. K. Nti, O. Nyarko-Boateng, and J. Aning, “Performance of Machine Learning Algorithms with Different K Values in K-fold CrossValidation,” *International Journal of Information Technology and Computer Science*, vol. 13, no. 6, pp. 61–71, Dec. 2021, doi: 10.5815/ijitcs.2021.06.05.
- [6] X. Zhang and C. A. Liu, “Model averaging prediction by K-fold cross-validation,” *JEconom*, vol. 235, no. 1, 2023, doi: 10.1016/j.jeconom.2022.04.007.
- [7] W. Wijiyanto, A. I. Pradana, S. Sopingi, and V. Atina, “Teknik K-Fold Cross Validation untuk Mengevaluasi Kinerja Mahasiswa,” *Jurnal Algoritma*, vol. 21, no. 1, May 2024, doi: 10.33364/algoritma/v.21-1.1618.
- [8] J. White and S. D. Power, “k-Fold Cross-Validation Can Significantly Over-Estimate True Classification Accuracy in Common EEG-Based Passive BCI Experimental Designs: An Empirical

- Investigation,” *Sensors*, vol. 23, no. 13, 2023, doi: 10.3390/s23136077.
- [9] S. Zhang and J. Li, “KNN Classification With One-Step Computation,” *IEEE Trans Knowl Data Eng*, vol. 35, no. 3, 2023, doi: 10.1109/TKDE.2021.3119140.
- [10] A. Putri, C. Syaficha Hardiana, E. Novfuja, F. Try Puspa Siregar, Y. Fatma, and R. Wahyuni, “Comparison of K-NN, Naive Bayes and SVM Algorithms for Final-Year Student Graduation Prediction Komparasi Algoritma K-NN, Naive Bayes dan SVM untuk Prediksi Kelulusan Mahasiswa Tingkat Akhir,” *Institut Riset dan Publikasi Indonesia (IRPI) MALCOM: Indonesian Journal of Machine Learning and Computer Science Journal Homepage*, vol. 3, no. 1, 2023.
- [11] E. Y. Boateng, J. Otoo, and D. A. Abaye, “Basic Tenets of Classification Algorithms K-Nearest-Neighbor, Support Vector Machine, Random Forest and Neural Network: A Review,” *Journal of Data Analysis and Information Processing*, vol. 08, no. 04, 2020, doi: 10.4236/jdaip.2020.84020.
- [12] M. G. M. Abdolrasol *et al.*, “Artificial neural networks based optimization techniques: A review,” 2021, doi: 10.3390/electronics10212689.
- [13] M. M. Mijwil, “Artificial Neural Networks Advantages and Disadvantages,” *Mesopotamian Journal of Big Data*, vol. 2021, 2021, doi: 10.58496/mjbd/2021/006.
- [14] M. Uzair and N. Jamil, “Effects of Hidden Layers on the Efficiency of Neural networks,” in *Proceedings - 2020 23rd IEEE International Multi-Topic Conference, INMIC 2020*, 2020, doi: 10.1109/INMIC50486.2020.9318195.
- [15] G. R. Yang and X. J. Wang, “Artificial Neural Networks for Neuroscientists: A Primer,” 2020, doi: 10.1016/j.neuron.2020.09.005.
- [16] I. Gunawan, “Optimasi Model Artificial Neural Network untuk Klasifikasi Paket Jaringan,” *SIMETRIS*, vol. 14, no. 2, 2020, doi: 10.51901/simetris.v14i2.135.
- [17] D. M. Abdullah and A. M. Abdulazeez, “Machine Learning Applications based on SVM Classification: A Review,” *Qubahan Academic Journal*, vol. 1, no. 2, 2021, doi: 10.48161/qaj.v1n2a50.
- [18] A. Kurani, P. Doshi, A. Vakharia, and M. Shah, “A Comprehensive Comparative Study of Artificial Neural Network (ANN) and Support Vector Machines (SVM) on Stock Forecasting,” 2023, doi: 10.1007/s40745-021-00344-x.
- [19] B. Abd-ElMassieh Aiad, K. Basem Zarif, and Z. Mahmoud Gadallah, “Support Vector Machine Kernel Functions Comparison,” 2021.
- [20] M. Yalsavar, P. Karimghaee, A. Sheikh-Akbari, M. H. Khooban, J. Dehmeshki, and S. Al-Majeed, “Kernel Parameter Optimization for Support Vector Machine Based on Sliding Mode Control,” *IEEE Access*, vol. 10, 2022, doi: 10.1109/ACCESS.2022.3150001.
- [21] B. A. Maulana, M. J. Fahmi, A. M. Imran, and N. Hidayati, “Analisis Sentimen Terhadap Aplikasi Pluang Menggunakan Algoritma Naive Bayes dan Support Vector Machine (SVM),” *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 4, no. 2, pp. 375–384, Feb. 2024, doi: 10.57152/malcom.v4i2.1206.
- [22] M. Vakili, M. Ghamsari, and M. Rezaei, “Performance Analysis and Comparison of Machine and Deep Learning Algorithms for IoT Data Classification,” 2020.

Georgia Sugisandhea, saat ini sebagai Mahasiswa program studi Teknik Informatika Universitas Tarumanagara.

Teny Handhayani, dosen di Fakultas Teknologi Informasi, Universitas Tarumanagara.