

Evaluasi Algoritma Decision Tree dan Random Forest serta Efektivitas Feature Selection dalam Memprediksi Kesehatan Mental

Veronica Marcella Angela Simalango^{#1} Wenny Franciska Senjaya^{*2}

[#]Program Studi SI Teknik Informatika, Universitas Kristen Maranatha
Jl. Surya Sumantri No. 65, Bandung, Indonesia

¹2172042@maranatha.ac.id

²wenny.fs@it.maranatha.edu

Abstract — Stress has become a significant health issue in modern society, with substantial impacts such as sleep disorders and cardiovascular diseases. This study utilizes wearable devices to monitor real-time physiological data, such as heart rate and sleep patterns, to predict stress risks using machine learning algorithms, namely Decision Tree and Random Forest. The results indicate that Random Forest excels in overall accuracy for high-dimensional data, while Decision Tree demonstrates a better balance in identifying minority classes with strong Precision, Recall, and F1-Score performance. Feature selection does not significantly enhance performance but aids in computational efficiency and model interpretability. This research contributes to developing accurate and efficient stress prediction systems based on wearable devices.

Keywords — Feature Selection, Machine Learning, Mental Health, Wearable Devices

I. PENDAHULUAN

Stres merupakan masalah kesehatan yang semakin umum akibat tekanan pekerjaan, tuntutan sosial, dan gaya hidup yang serba cepat [1]. Dampaknya bisa berupa gangguan tidur, kecemasan, hingga penyakit kardiovaskular, sehingga deteksi dini menjadi sangat penting [2]. Wearable devices seperti smartwatch dan fitness tracker dapat membantu pemantauan kesehatan secara real-time dengan mengumpulkan data fisiologis.

Data dari wearable devices, seperti detak jantung, pola tidur, dan aktivitas fisik, dapat menjadi indikator potensial stress [3]. Dengan bantuan algoritma machine learning, pola dalam data fisiologis dapat dianalisis untuk memprediksi risiko stres secara lebih akurat [4]. Namun, tantangan dalam penerapannya meliputi variabilitas data antar individu dan pengaruh faktor eksternal lainnya [5].

Feature selection menjadi aspek penting dalam meningkatkan akurasi model prediksi stres dengan memilih fitur yang paling relevan. Proses ini tidak hanya mengurangi kompleksitas komputasi tetapi juga memastikan bahwa hanya data signifikan yang digunakan. Dengan pemilihan fitur yang tepat, korelasi antara indikator fisiologis dan tingkat stres dapat diidentifikasi secara lebih baik [6].

Penelitian ini bertujuan untuk menerapkan algoritma machine learning dalam prediksi stres berdasarkan data dari wearable devices. Sistem yang akurat diharapkan dapat memberikan peringatan dini kepada individu terkait tingkat stres mereka. Dengan demikian, intervensi dapat dilakukan lebih cepat untuk mencegah dampak negatif stres yang lebih serius.

II. KAJIAN TEORI

A. Wearable Devices

Setiap Wearable devices merupakan perangkat elektronik yang dapat dikenakan pada tubuh manusia untuk mengumpulkan, memproses, dan mengirimkan data fisiologis dan perilaku secara real-time [7]. Kemampuan wearable devices dalam mengumpulkan data fisiologis secara kontinu dan non-invasif membuka peluang baru dalam pemantauan kesehatan dan deteksi dini penyakit. Data yang dikumpulkan dari wearable devices dapat diintegrasikan dengan machine learning untuk mengidentifikasi pola dan memprediksi risiko kesehatan secara personal [8].

B. Decision Tree

Decision Tree merupakan algoritma machine learning yang populer karena kemampuannya dalam memodelkan hubungan kompleks antar variabel dan menghasilkan model yang mudah diinterpretasi. Algoritma ini membangun struktur pohon yang terdiri dari node, branch, dan leaf untuk merepresentasikan atribut, aturan keputusan, dan hasil klasifikasi atau prediksi [9].

C. Random Forest

Random Forest merupakan algoritma machine learning yang powerful dan serbaguna, dikenal karena kemampuannya dalam menghasilkan prediksi yang akurat dan robust. Algoritma ini menggabungkan konsep ensemble learning dengan membangun banyak Decision Tree dan menggabungkan prediksi dari setiap pohon untuk mendapatkan hasil akhir [10].

D. Hyperparameter Tuning

Hyperparameter adalah variabel optimisasi yang digunakan selama fase pelatihan untuk mendapatkan nilai rata-rata yang dioptimalkan setelah beberapa proses trial-and-error. Untuk mengatasi masalah overfitting dengan pencarian Grid biasa, diterapkan stratified cross-validation di mana sampel dibagi menjadi K-fold secara acak. Model GridSearchCV dari Scikit learn digunakan untuk mendapatkan parameter terbaik [11].

E. Selesksi Fitur

Seleksi fitur merupakan tahapan penting dalam *machine learning* yang bertujuan untuk memilih fitur-fitur yang paling relevan dan informatif dari sekumpulan fitur yang tersedia. Proses ini bertujuan untuk meningkatkan akurasi model, mengurangi kompleksitas model, dan mempercepat waktu pelatihan model.

F. Evaluasi

Biasanya, pengukuran kinerja standar untuk masalah klasifikasi adalah akurasi. Akan tetapi, dalam kasus dengan distribusi kelas yang tidak seimbang, diperlukan pengukuran tambahan untuk memberikan penilaian yang lebih menyeluruh [12]. True Positive (TP) adalah prediksi benar untuk kelas tertentu. False Positive (FP) terjadi ketika model salah memprediksi data sebagai kelas tertentu. False Negative (FN) adalah data yang seharusnya termasuk dalam kelas tertentu tetapi salah diklasifikasikan. True Negative (TN) adalah data yang benar-benar bukan bagian dari kelas tertentu dan diprediksi dengan benar sebagai kelas lain.

1) *Confusion Matrix*: Matriks Kebingungan (Confusion Matrix) adalah alat evaluasi visual yang digunakan dalam pembelajaran mesin. Kolom-kolom dalam Matriks Kebingungan mewakili hasil kelas prediksi, dan baris-baris mewakili hasil kelas actual [13].

2) *Akurasi*: Akurasi merupakan proporsi dari total benar yang diprediksi oleh sistem dari semua data yang ada [14].

$$\text{Akurasi} = \frac{TP+TN}{TP+FP+FN+TN} \quad (1)$$

3) *Precision*: Precision menggambarkan tingkat akurasi model dalam memprediksi kelas positif.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2)$$

4) *Recall*: Recall merupakan nilai kemampuan model dalam memprediksi hasil positif, yang juga dikenal sebagai sensitivitas model.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

5) *F1-Score*: F1-score adalah metrik yang merupakan rata-rata harmonik dari precision dan recall

$$\text{F1 Score} = 2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \quad (4)$$

III. ANALISIS DAN RANCANGAN SISTEM

A. Dataset

Dataset WESAD-chest-combined-classification-hrv.csv berisi data fisiologis yang dikumpulkan melalui wearable device yang dikenakan di dada oleh 15 partisipan. Data dikumpulkan berdasarkan tiga kondisi atau kelas berbeda, yang masing-masing merepresentasikan keadaan emosional peserta penelitian.

TABEL I
KELAS PADA DATASET

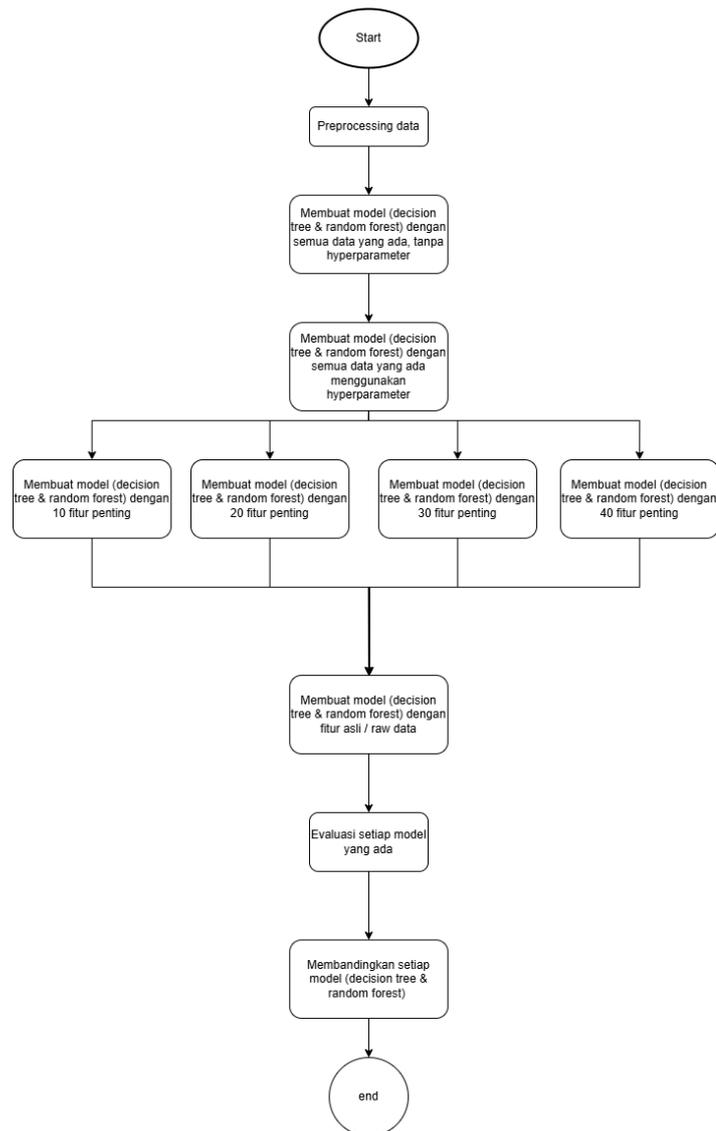
Kelas	Label	Deskripsi
Kelas 0	Baseline	Menggambarkan kondisi peserta penelitian saat tidak berada dalam keadaan stres maupun senang.
Kelas 1	Amusement	Menunjukkan peserta penelitian berada dalam keadaan senang, seperti saat menonton video lucu.
Kelas 2	Stress	Menggambarkan peserta penelitian dalam keadaan stres yang diinduksi melalui metode Trier Social Stress Test (TSST).

Dataset ini memiliki 67 fitur yang mengukur berbagai parameter fisiologis, seperti MEAN_RR, SDRR, RMSSD, dan SDDSD yang berkaitan dengan interval RR dan variabilitas detak jantung, serta HR, pNN25, pNN50, SD1, SD2, KURT, dan SKEW yang mencerminkan distribusi data dan keseimbangan sistem saraf simpatis serta parasimpatis. Selain itu, terdapat fitur berbasis frekuensi seperti VLF, LF, HF, serta rasio LF_HF, dan fitur transformasi logaritmik seperti MEAN_RR_LOG dan LF_BOXCOX untuk analisis yang lebih mendalam. Dataset ini terdiri dari 135.650 sampel, dengan distribusi kelas Baseline sebanyak 71.640 sampel, Amusement 23.064 sampel, dan Stress 40.946 sampel.

Analisis data bertujuan untuk memahami karakteristik dataset, mengidentifikasi pola dan tren terkait kondisi stres, serta mendeteksi anomali atau outlier yang dapat memengaruhi hasil pemodelan. Informasi yang diperoleh dari analisis ini akan menjadi dasar dalam tahap preprocessing data dan seleksi fitur agar model machine learning dapat bekerja dengan optimal. Dengan memahami struktur dataset dan potensi permasalahan di dalamnya, dapat ditentukan strategi preprocessing yang tepat dan metode seleksi fitur yang sesuai. Hal ini akan mendukung pengembangan model prediksi stres yang lebih akurat dan andal menggunakan data dari wearable devices.

B. Alur Penelitian

Alur penelitian ini, seperti pada gambar dibawah, diawali dengan tahap preprocessing data untuk memastikan data dalam kondisi bersih dan siap digunakan dalam proses pemodelan. Setelah itu, dilakukan pembangunan model decision tree dan random forest menggunakan seluruh data tanpa pengaturan hyperparameter sebagai baseline. Selanjutnya, model dikembangkan dengan mengatur hyperparameter untuk meningkatkan performa. Tahapan berikutnya adalah pemilihan fitur berdasarkan tingkat kepentingannya. Model dibangun secara bertahap menggunakan 10, 20, 30, dan 40 fitur penting yang diidentifikasi. Selain itu, model juga dibuat dengan menggunakan seluruh fitur asli (raw data) tanpa proses pemilihan fitur. Setelah semua model dibuat, tahap berikutnya adalah evaluasi performa model berdasarkan metrik seperti akurasi, precision, recall, dan F1-score. Sebagai langkah akhir, semua model yang telah dibuat dibandingkan untuk menentukan model terbaik berdasarkan hasil evaluasi performa yang dilakukan



Gambar 1. Alur Penelitian

IV. IMPLEMENTASI

A. Preprocessing

Tahap preprocessing dimulai dengan menghapus kolom yang tidak relevan (condition, SSSQ class, subject id, dan condition label). Fitur disimpan dalam variabel X, sementara label (condition label) disimpan dalam y. Data dibagi menjadi 80% latih dan 20% uji menggunakan `train_test_split` dengan `random_state=42`. Selanjutnya, fitur dinormalisasi menggunakan `StandardScaler` agar memiliki rata-rata 0 dan standar deviasi 1, meningkatkan performa model machine learning.

B. Decision Tree

Setelah preprocessing, model Decision Tree Classifier dibuat dan dilatih. Model diinisialisasi dengan `max_depth=5`, `min_samples_split=5`, dan `min_samples_leaf=5` untuk menghindari overfitting. `random_state=42` digunakan agar hasil konsisten. Model dilatih dengan `X_train_scaled` dan `y_train` menggunakan metode `fit()`, sehingga dapat mempelajari pola hubungan fitur dan target dengan keseimbangan antara kompleksitas dan generalisasi.

C. Random Forests

Model Random Forest Classifier dikembangkan dengan $n_estimators=100$ (100 pohon keputusan), $max_depth=5$, $min_samples_leaf=5$, dan $min_samples_split=5$ untuk menjaga keseimbangan model. $random_state=42$ diterapkan untuk hasil yang konsisten. Model dilatih menggunakan X_train dan y_train dengan $fit()$, di mana kombinasi prediksi dari banyak pohon meningkatkan akurasi dan stabilitas model.

D. Hyperparameter Tuning

Hyperparameter Tuning dilakukan menggunakan GridSearchCV pada Decision Tree dan Random Forest Classifier untuk mencari kombinasi parameter terbaik berdasarkan akurasi. Parameter yang diuji mencakup kedalaman pohon (max_depth), jumlah sampel minimum untuk pemisahan ($min_samples_split$), jumlah sampel minimum di setiap leaf node ($min_samples_leaf$), dan metode pemisahan node ($criterion$: gini atau entropy). Proses tuning menggunakan cross-validation untuk evaluasi menyeluruh, dengan hasil terbaik ditentukan berdasarkan $best_params_$ dan $best_score_$.

Pada Decision Tree Classifier, GridSearchCV diuji dengan 5-fold cross-validation dan berbagai kombinasi parameter. Setelah pencarian selesai, model terbaik dipilih menggunakan $best_estimator_$, yang kemudian digunakan untuk memprediksi data uji (X_test) dengan parameter optimal. Pendekatan ini memastikan model tidak hanya sesuai untuk data latih tetapi juga memiliki kemampuan generalisasi yang baik.

Sedangkan pada Random Forest Classifier, GridSearchCV diterapkan dengan 3-fold cross-validation dan berbagai kombinasi parameter seperti $n_estimators$, max_depth , $max_features$, serta metode pemisahan node. Proses tuning memanfaatkan semua core prosesor ($n_jobs=-1$) untuk efisiensi. Model terbaik yang diperoleh dari $best_estimator_$ digunakan untuk prediksi data uji, sehingga menghasilkan model yang optimal dan lebih stabil dibandingkan pendekatan individual pohon keputusan.

E. Seleksi Fitur

Implementasi seleksi fitur dengan feature importance bertujuan untuk memilih fitur paling signifikan terhadap prediksi model. Dalam penelitian ini, fitur dipilih berdasarkan pemeringkatan kontribusinya, dengan variasi 10, 20, 30, dan 40 fitur terpenting. Tujuan utama adalah mengoptimalkan kinerja model sekaligus mempercepat penelitian.

Proses seleksi diawali dengan menghitung nilai feature importance dari model Random Forest yang telah dilatih. Fitur diurutkan berdasarkan tingkat kepentingannya, lalu data latih dan uji difilter sesuai fitur terpilih. Selanjutnya, model Decision Tree dan Random Forest dilatih ulang menggunakan parameter terbaik yang diperoleh dari GridSearchCV.

V. PENGUJIAN

A. Pengujian Menggunakan Data Mentah / Asli

Fitur asli pada penelitian ini adalah data mentah yang diambil langsung dari perangkat pengukuran (seperti detak jantung atau sinyal dari sensor), dan tidak melalui perhitungan matematis atau analisis spektral. Berdasarkan hasil analisis performa Decision Tree dan Random Forest tanpa Hyperparameter Tuning pada data mentah yang tersedia pada Tabel II.

TABEL II
PERBANDINGAN HASIL RAW DATA TANPA HYPERPARAMETER TUNING

Metrik	Decision Tree (DT)	Random Forest (RF)
Akurasi	0.84	0.82
Precision Kelas 0	0.87	0.78
Recall Kelas 0	0.89	0.97
F1-Score Kelas 0	0.88	0.87
Precision Kelas 1	0.71	0.99
Recall Kelas 1	0.58	0.23
F1-Score Kelas 1	0.64	0.37
Precision Kelas 2	0.83	0.87
Recall Kelas 2	0.88	0.87
F1-Score Kelas 2	0.86	0.87
Rata-rata F1-Score	0.79	0.70

Pada Tabel II juga dapat terlihat kelas minoritas atau kelas 1, Decision Tree memiliki F1-Score 0.64, jauh lebih baik dibandingkan Random Forest yang hanya mencapai 0.37. Hal ini disebabkan oleh kombinasi Precision dan Recall yang lebih seimbang pada Decision Tree (Precision: 0.71, Recall: 0.58) dibandingkan Random Forest yang sangat bias terhadap Precision (0.99) namun memiliki Recall yang rendah (0.23). Sementara itu, untuk kelas 2, kedua model menunjukkan performa yang

hampir setara, dengan F1-Score masing-masing 0.86 untuk Decision Tree dan 0.87 untuk Random Forest. Secara keseluruhan, rata-rata F1-Score Decision Tree (0.79) lebih tinggi dibandingkan Random Forest (0.70), mengindikasikan bahwa Decision Tree lebih mampu menangani ketidakseimbangan data.

Berdasarkan Tabel III, terlihat bahwa model memiliki kemampuan yang cukup baik dalam mengklasifikasikan data pada Kelas 0, dengan akurasi 89,12%. Namun, performa model pada Kelas 1 dan Kelas 2 relatif kurang optimal, dengan tingkat akurasi masing-masing 58,17% dan 70,04%. Kesalahan klasifikasi yang terjadi, terutama pada prediksi Kelas 1 dan Kelas 2, menunjukkan bahwa model Decision Tree tanpa tuning belum cukup optimal untuk data ini.

TABEL III
CONFUSION MATRIX DECISION TREE TANPA HYPERPARAMETER TUNING PADA RAW DATA

Prediksi \ Aktual	Kelas 0 (%)	Kelas 1 (%)	Kelas 2 (%)
Kelas 0	89.12	16.87	9.35
Kelas 1	8.64	58.17	8.12
Kelas 2	4.92	7.04	87.71

TABEL IV
CONFUSION MATRIX RANDOM FOREST TANPA HYPERPARAMETER TUNING PADA RAW DATA

Prediksi \ Aktual	Kelas 0 (%)	Kelas 1 (%)	Kelas 2 (%)
Kelas 0	97.20	0.11	4.73
Kelas 1	20.09	22.99	7.83
Kelas 2	7.30	0.09	87.46

Sementara itu, berdasarkan Tabel IV model Random Forest menunjukkan hasil klasifikasi yang lebih baik dibandingkan Decision Tree. Akurasi untuk Kelas 0 mencapai 97,20%, jauh lebih tinggi dibandingkan model sebelumnya. Namun, akurasi untuk Kelas 1 dan Kelas 2 masih menunjukkan kesalahan yang cukup signifikan, yaitu masing-masing 22,99% dan 87,46%.

Melalui analisis kedua Confusion Matrix, dapat disimpulkan bahwa Random Forest cenderung bias terhadap kelas mayoritas atau kelas 0, sebagaimana terlihat dari jumlah prediksi benar yang sangat tinggi pada kelas ini namun mengorbankan akurasi pada kelas minoritas kelas. Sebaliknya, Decision Tree menunjukkan distribusi prediksi yang lebih seimbang, dengan Recall pada kelas minoritas. Pada kelas 2, kedua model memiliki performa yang baik dengan Recall tinggi. Hasil ini menunjukkan bahwa Decision Tree lebih unggul dalam mengenali kelas minoritas, membuatnya lebih cocok untuk dataset yang tidak seimbang seperti ini.

Pada Tabel V terlihat perbandingan kinerja antara model *Decision Tree* dan *Random Forest* menggunakan hyperparameter tuning, berdasarkan beberapa metrik, seperti akurasi, *Precision*, *Recall*, dan F1-score untuk setiap kelas (Kelas 0, 1, dan 2). Keduanya mencapai akurasi yang tinggi sebesar 97%, menandakan bahwa keduanya dapat memprediksi dengan sangat baik pada dataset tersebut. Namun, ketika melihat lebih detail, *Decision Tree* dan *Random Forest* menunjukkan pola yang sedikit berbeda dalam hal *Precision*, *Recall*, dan F1-score untuk setiap kelas.

TABEL V
PERBANDINGAN HASIL RAW DATA DENGAN HYPERPARAMETER TUNING

Metrik	<i>Decision Tree</i> (DT)	<i>Random Forest</i> (RF)
Akurasi	0.97	0.97
<i>Precision</i> Kelas 0	0.98	0.96
<i>Recall</i> Kelas 0	0.98	0.99
F1-Score Kelas 0	0.98	0.98
<i>Precision</i> Kelas 1	0.94	0.98
<i>Recall</i> Kelas 1	0.94	0.90
F1-Score Kelas 1	0.94	0.94
<i>Precision</i> Kelas 2	0.96	0.97
<i>Recall</i> Kelas 2	0.97	0.97
F1-Score Kelas 2	0.96	0.97
Rata-rata F1-Score	0.96	0.96

Decision Tree cenderung memiliki performa yang sangat baik untuk kelas mayoritas (Kelas 0 dan Kelas 2) dengan nilai *Precision*, *Recall*, dan F1-score yang hampir sempurna (sekitar 0.98). Sementara itu, *Random Forest* sedikit lebih unggul dalam menangani kelas minoritas (Kelas 1), dengan *Precision* tinggi sebesar 0.98 dan F1-score yang tetap tinggi yaitu 0.94. Rata-rata F1-score kedua model menunjukkan bahwa *Decision Tree* sedikit unggul dengan nilai 0.96 dibandingkan *Random*

Forest yang mencapai 0.96. Secara keseluruhan, meskipun kedua model menunjukkan performa yang sangat baik, *Random Forest* memiliki keunggulan dalam menangani ketidakseimbangan data dengan baik, terutama pada kelas minoritas.

Pada Tabel VI, terlihat bahwa performa model meningkat secara signifikan dibandingkan sebelumnya. Akurasi prediksi untuk Kelas 0 mencapai 98,00%, sedangkan untuk Kelas 1 dan Kelas 2 masing-masing mencapai 94,07% dan 97,41%. Peningkatan ini menunjukkan bahwa tuning parameter pada Decision Tree berhasil memperbaiki akurasi model dalam mengklasifikasikan data ke kelas yang tepat.

TABEL VI
CONFUSION MATRIX DECISION TREE DENGAN HYPERPARAMETER TUNING PADA RAW DATA

Prediksi \ Aktual	Kelas 0 (%)	Kelas 1 (%)	Kelas 2 (%)
Kelas 0	98.00	1.14	0.86
Kelas 1	3.43	94.07	2.80
Kelas 2	2.10	0.49	97.41

TABEL VII
CONFUSION MATRIX RANDOM FOREST DENGAN HYPERPARAMETER TUNING PADA RAW DATA

Prediksi \ Aktual	Kelas 0 (%)	Kelas 1 (%)	Kelas 2 (%)
Kelas 0	99.00	0.17	0.58
Kelas 1	7.17	90.00	2.98
Kelas 2	2.62	0.54	97.84

Model Random Forest pada Tabel VII juga menunjukkan hasil yang sangat baik. Akurasi untuk Kelas 0, Kelas 1, dan Kelas 2 masing-masing adalah 99,00%, 90,00%, dan 97,84%. Meskipun akurasi Kelas 1 sedikit lebih rendah dibandingkan Decision Tree, Random Forest secara keseluruhan tetap menunjukkan performa yang sangat baik, dengan keunggulan dalam pengklasifikasian Kelas 0.

Berdasarkan Confusion Matrix pada Decision Tree dan Random Forest (RF) menunjukkan perbedaan kinerja dalam mengklasifikasikan data ke dalam kelas-kelas yang tersedia. Hal ini menunjukkan bahwa Decision Tree lebih unggul dalam menangani kelas minoritas Kelas 1 setelah dilakukan tuning. Meskipun Random Forest umumnya memberikan hasil yang lebih stabil pada data, dalam kasus ini, tuning pada Decision Tree mampu meningkatkan sensitivitas terhadap Kelas 1. Oleh karena itu, untuk skenario di mana klasifikasi kelas minoritas menjadi prioritas, Decision Tree dengan tuning dapat menjadi pilihan yang lebih efektif dibandingkan Random Forest.

B. Pengujian Menggunakan Semua Data

Pada penelitian ini, dilakukan pelatihan model dengan memanfaatkan seluruh fitur yang terdapat dalam data untuk mengeksplorasi potensi penuh data tanpa melakukan seleksi fitur atau reduksi dimensi. Pada pengujian ini terlihat dalam pada Tabel VIII bahwa secara keseluruhan akurasi Decision Tree lebih tinggi, yaitu 86%, dibandingkan Random Forest yang mencapai 82%. Precision dan recall untuk Kelas 0 pada Decision Tree masing-masing adalah 0,84 dan 0,94, menunjukkan performa yang baik dalam mengidentifikasi data dari Kelas 0. Namun, precision dan recall untuk Kelas 1 cukup rendah, masing-masing 0,48 dan 0,62, menunjukkan kesulitan model dalam menangani kelas minoritas. Untuk Kelas 2, precision dan recall juga berada pada tingkat yang moderat, masing-masing 0,70 dan 0,92.

TABEL VIII
PERBANDINGAN HASIL SEMUA DATA TANPA HYPERPARAMETER TUNING

	<i>Decision Tree</i>	<i>Random Forest</i>
Akurasi	0.86	0.82
<i>Precision</i> Kelas 0	0.84	0.78
<i>Recall</i> Kelas 0	0.94	0.97
F1-Score Kelas 0	0.89	0.87
<i>Precision</i> Kelas 1	0.88	0.99
<i>Recall</i> Kelas 1	0.48	0.23
F1-Score Kelas 1	0.62	0.37
<i>Precision</i> Kelas 2	0.89	0.89
<i>Recall</i> Kelas 2	0.92	0.89
F1-Score Kelas 2	0.90	0.89
Rata-rata F1-Score	0.80	0.71

Sementara itu, Random Forest menunjukkan hasil yang sedikit lebih rendah dibandingkan Decision Tree, dengan precision untuk Kelas 0 sebesar 0,78 dan recall sebesar 0,97. Namun, pada Kelas 1, Random Forest memiliki precision yang sedikit lebih tinggi dibandingkan Decision Tree, yaitu 0,52, meskipun recall tetap rendah pada 0,23. Untuk Kelas 2, performa Random Forest hampir setara dengan Decision Tree, dengan precision dan recall masing-masing sebesar 0,89 dan 0,89.

Berdasarkan rata-rata F1-Score, Decision Tree lebih unggul dengan skor 0,80 dibandingkan Random Forest yang mencapai 0,71. Performa yang lebih baik pada Decision Tree menunjukkan bahwa model ini lebih efektif dalam menangani data secara keseluruhan, khususnya pada Kelas 0 dan Kelas 2. Namun, kedua model masih memiliki kelemahan dalam menangani Kelas 1, sehingga diperlukan strategi tambahan, seperti oversampling atau pemberian bobot lebih pada kelas minoritas, untuk meningkatkan kemampuan dalam mengklasifikasikan kelas tersebut. Berikut ditampilkan pula perbandingan Confusion Matrix dalam bentuk persentase untuk masing-masing model. Persentase dihitung berdasarkan jumlah sampel dalam setiap kelas.

TABEL IX
CONFUSION MATRIX DECISION TREE TANPA HYPERPARAMETER TUNING

Prediksi \ Aktual	Kelas 0 (%)	Kelas 1 (%)	Kelas 2 (%)
Kelas 0	94.19	1.99	3.81
Kelas 1	42.50	47.91	9.59
Kelas 2	8.05	0.12	91.83

Pada tabel IX, algoritma Decision Tree menunjukkan performa yang cukup baik dalam mengklasifikasikan kelas 0, dengan tingkat akurasi sebesar 94.19%, dan juga untuk kelas 2 dengan akurasi sebesar 91.83%. Namun, model kesulitan dalam mengklasifikasikan kelas 1, yang merupakan kelas minoritas, dengan hanya 47.91% data yang terklasifikasi dengan benar. Sebagian besar data kelas 1 salah diklasifikasikan sebagai kelas 0 (42.50%), yang menunjukkan adanya ketidakseimbangan dalam kemampuan model untuk mengenali kelas tersebut.

TABEL X
CONFUSION MATRIX RANDOM FOREST TANPA HYPERPARAMETER TUNING

Prediksi \ Aktual	Kelas 0 (%)	Kelas 1 (%)	Kelas 2 (%)
Kelas 0	97.25	0.01	2.74
Kelas 1	66.30	22.83	10.86
Kelas 2	10.50	0.11	89.39

Pada tabel X, Random Forest juga menunjukkan hasil yang sangat baik untuk kelas 0, dengan akurasi mencapai 97.25%, dan performa yang cukup baik untuk kelas 2 dengan akurasi sebesar 89.39%. Namun, performa pada kelas 1 masih rendah, dengan hanya 22.83% data yang terklasifikasi dengan benar. Sebagian besar data kelas 1 salah diklasifikasikan ke kelas 0 (66.30%), yang mengindikasikan bahwa model masih kesulitan dalam menangani kelas minoritas meskipun secara keseluruhan Random Forest cenderung lebih stabil dibandingkan Decision Tree. Berdasarkan analisis Confusion Matrix terdapat perbedaan kinerja yang mencolok antara Decision Tree dan Random Forest (RF) pada dataset yang tidak seimbang.

Pada kelas mayoritas Random Forest menunjukkan kemampuan klasifikasi yang lebih baik, Namun, kesalahan klasifikasi pada kelas mayoritas cenderung lebih kecil pada Decision Tree dibandingkan Random Forest. Untuk kelas minoritas (kelas 1), Decision Tree memiliki keunggulan signifikan dibandingkan Random Forest. Random Forest juga lebih sering salah mengklasifikasikan kelas 1 sebagai kelas mayoritas. Pada kelas 2, kedua model memiliki performa yang hampir seimbang. Namun, Random Forest lebih sering salah mengklasifikasikan kelas 2 ke kelas mayoritas dibandingkan Decision Tree. Secara keseluruhan, Decision Tree menunjukkan performa yang lebih baik dalam menangani data tidak seimbang dibandingkan Random Forest, terutama pada kelas minoritas (kelas 1).

Setelah menggunakan pengaturan *Hyperparameter* terbaik, terlihat pada Tabel XI performa Decision Tree dan Random Forest menunjukkan peningkatan yang signifikan. Decision Tree mencapai akurasi 97.74%, sementara Random Forest unggul dengan akurasi 99.22%. Perbedaan akurasi ini mencerminkan kemampuan Random Forest yang lebih baik dalam memanfaatkan pengaturan optimal untuk menghasilkan prediksi yang lebih akurat secara keseluruhan.

TABEL XI
PERBANDINGAN HASIL SEMUA DATA DENGAN HYPERPARAMETER TUNING

Metrik	Decision Tree (DT)	Random Forest (RF)
Akurasi	0.9774	0.9922
Precision Kelas 0	0.99	0.99
Recall Kelas 0	0.98	0.99
F1-Score Kelas 0	0.99	0.99

Precision Kelas 1	0.95	1.00
Recall Kelas 1	0.96	0.97
F1-Score Kelas 1	0.96	0.99
Precision Kelas 2	0.98	0.99
Recall Kelas 2	0.99	1.00
F1-Score Kelas 2	0.98	0.99
Rata-rata F1-Score	0.977	0.990

Perbedaan performa yang lebih mencolok terlihat pada kelas minoritas (kelas 1). Random Forest memiliki keunggulan yang signifikan, dengan F1-Score sebesar 0.99, sementara Decision Tree berada di angka 0.96. Keunggulan ini terutama disebabkan oleh precision yang lebih tinggi pada Random Forest (1.00 dibandingkan 0.95 untuk Decision Tree), yang berarti Random Forest mampu memprediksi kelas 1 dengan lebih akurat tanpa terlalu banyak menghasilkan false positives. Meskipun recall untuk kelas 1 pada Decision Tree sedikit lebih tinggi (0.96 dibandingkan 0.97 pada Random Forest), keunggulan dalam precision memberikan Random Forest performa keseluruhan yang lebih baik dalam menangani kelas minoritas.

Secara keseluruhan, Random Forest menunjukkan kinerja yang lebih unggul dengan rata-rata F1-Score sebesar 0.990, dibandingkan dengan Decision Tree yang mencapai rata-rata F1-Score 0.977. Keunggulan Random Forest dalam memanfaatkan pengaturan hyperparameter optimal menjadikannya model yang lebih andal untuk klasifikasi dengan distribusi data yang melibatkan kelas minoritas. Meski demikian, Decision Tree tetap memberikan hasil yang kompetitif dan menunjukkan peningkatan signifikan dibandingkan sebelum tuning. Peningkatan ini menunjukkan pentingnya langkah hyperparameter tuning untuk mengoptimalkan performa model, terutama pada kasus klasifikasi multikelas dengan distribusi data yang tidak seimbang.

Berikut adalah perbandingan Confusion Matrix dalam bentuk persentase untuk masing-masing model. Persentase dihitung berdasarkan jumlah sampel dalam setiap kelas.

TABEL XII
CONFUSION MATRIX DECISION TREE DENGAN HYPERPARAMETER TUNING

Prediksi \ Aktual	Kelas 0 (%)	Kelas 1 (%)	Kelas 2 (%)
Kelas 0	98.21	3.22	1.30
Kelas 1	0.79	95.99	0.85
Kelas 2	0.20	0.79	98.54

Berdasarkan Tabel XII, algoritma Decision Tree dengan hyperparameter tuning menunjukkan peningkatan signifikan dalam mengklasifikasikan semua kelas. Untuk kelas 0, akurasi prediksi mencapai 98.21%, dengan sebagian kecil data kelas 0 salah diklasifikasikan sebagai kelas 1 (3.22%) dan kelas 2 (1.30%). Pada kelas 1, Decision Tree berhasil mengklasifikasikan 95.99% data dengan benar, sementara sebagian kecil salah diklasifikasikan ke kelas 0 (0.79%) dan kelas 2 (0.85%). Untuk kelas 2, akurasi mencapai 98.54%, dengan kesalahan klasifikasi yang minimal, yaitu ke kelas 0 sebesar 0.20% dan kelas 1 sebesar 0.79%. Hal ini menunjukkan performa yang konsisten untuk ketiga kelas setelah tuning.

TABEL XIII
CONFUSION MATRIX RANDOM FOREST DENGAN HYPERPARAMETER TUNING

Prediksi \ Aktual	Kelas 0 (%)	Kelas 1 (%)	Kelas 2 (%)
Kelas 0	99.49	0.02	0.51
Kelas 1	2.39	97.19	0.42
Kelas 2	0.06	0.02	99.87

Sementara itu, Tabel XIII menunjukkan performa Random Forest yang lebih unggul dalam beberapa aspek. Akurasi prediksi untuk kelas 0 sangat tinggi, yaitu 99.49%, dengan hanya 0.02% data yang salah diklasifikasikan ke kelas 1 dan 0.51% ke kelas 2. Untuk kelas 1, Random Forest mencapai akurasi 97.19%, lebih tinggi dibandingkan Decision Tree, dengan sebagian kecil data salah diklasifikasikan ke kelas 0 (2.39%) dan kelas 2 (0.42%). Pada kelas 2, Random Forest menunjukkan akurasi luar biasa sebesar 99.87%, dengan kesalahan klasifikasi yang hampir tidak ada. Hal ini mengindikasikan kemampuan Random Forest untuk menghasilkan prediksi yang lebih presisi dan akurat.

Secara umum, performa Random Forest lebih baik dibandingkan Decision Tree, terutama pada kelas 1 yang merupakan kelas minoritas. Random Forest mampu menangani distribusi data yang tidak seimbang dengan lebih efektif, terlihat dari akurasi dan tingkat kesalahan klasifikasinya yang lebih rendah. Namun, kedua model menunjukkan peningkatan yang signifikan setelah hyperparameter tuning, yang membuktikan bahwa langkah tuning dapat secara substansial meningkatkan performa model dalam klasifikasi multikelas. Untuk aplikasi lebih lanjut, Random Forest tampak lebih andal dalam menghasilkan prediksi yang lebih akurat pada semua kelas.

C. Pengujian Menggunakan Seleksi Fitur

Pada penelitian ini juga dilakukan pemilihan fitur menggunakan metode feature importance untuk menentukan fitur-fitur yang paling relevan dalam membangun model. Pemilihan fitur dilakukan dengan berbagai jumlah fitur, yaitu 10, 20, 30, dan 40 fitur terbaik. Selain itu, proses pelatihan model dilakukan dengan menggunakan pengaturan Hyperparameter optimal yang diperoleh melalui GridSearch pada kedua model.

Hasil model Decision Tree menggunakan seleksi fitur terlihat pada Tabel XIV. Decision Tree dengan 10 fitur menghasilkan akurasi sebesar 0.97. Meskipun akurasi tinggi, performa model menunjukkan beberapa tantangan, terutama pada kelas minoritas, yaitu Kelas 1, yang memiliki Recall sebesar 95% dan f1-score sebesar 0.95. Kelas 0 dan Kelas 2 memiliki kinerja yang sangat baik dengan nilai Recall dan f1-score yang hampir sempurna, masing-masing mencapai 98%.

Penambahan fitur menjadi 20 menghasilkan akurasi yang meningkat menjadi 0.98. Model ini menunjukkan kinerja yang lebih konsisten pada semua kelas dengan nilai Recall dan f1-score yang stabil. Kelas 0 dan Kelas 2 masih memiliki performa terbaik, masing-masing dengan Recall sebesar 99% dan f1-score 0.99. Kelas 1 juga menunjukkan peningkatan, meskipun sedikit penurunan pada Precision menjadi 94%.

TABEL XIV
PERBANDINGAN HASIL SELEKSI FITUR DECISION TREE

Metrik	Decision Tree (10 Fitur)	Decision Tree (20 Fitur)	Decision Tree (30 Fitur)	Decision Tree (40 Fitur)
Akurasi	0.97	0.98	0.98	0.98
Precision Kelas 0	0.98	0.99	0.99	0.99
Recall Kelas 0	0.98	0.98	0.98	0.98
F1-Score Kelas 0	0.98	0.99	0.98	0.99
Precision Kelas 1	0.95	0.94	0.95	0.95
Recall Kelas 1	0.95	0.97	0.96	0.96
F1-Score Kelas 1	0.95	0.95	0.96	0.96
Precision Kelas 2	0.98	0.98	0.98	0.98
Recall Kelas 2	0.98	0.99	0.99	0.99
F1-Score Kelas 2	0.98	0.98	0.98	0.98
Rata-rata F1	0.96	0.97	0.97	0.97

Decision Tree dengan 30 fitur menghasilkan akurasi yang tetap tinggi, yaitu 0.98. Performa model hampir identik dengan penggunaan 20 fitur, dengan Recall dan f1-score yang tinggi di semua kelas. Kelas 0 dan Kelas 2 mempertahankan nilai Recall dan f1-score yang sempurna, sedangkan Kelas 1 tetap memiliki nilai yang solid, meskipun sedikit penurunan pada Precision menjadi 95%. Dengan tambahan fitur hingga 40, Decision Tree menunjukkan akurasi tetap sebesar 0.98. Kinerja model hampir tidak berubah, dengan Recall dan f1-score yang sangat baik di semua kelas. Kelas 0 dan Kelas 2 menunjukkan hasil yang hampir sempurna, sedangkan Kelas 1 tetap menunjukkan Precision yang solid di angka 95%.

Berdasarkan analisis akurasi dan f1-score, Decision Tree dengan 20 hingga 40 fitur memberikan keseimbangan terbaik antara kompleksitas model dan performa. Penambahan fitur hingga 20 menunjukkan sedikit peningkatan dalam Recall dan f1-score, namun hasil tetap konsisten hingga penggunaan 40 fitur tanpa menurunkan akurasi signifikan. Oleh karena itu, penggunaan 20 hingga 40 fitur memberikan hasil optimal untuk mengoptimalkan model Decision Tree dalam klasifikasi multidimensi. Berikut adalah tabel Confusion Matrix untuk Decision Tree dengan berbagai jumlah fitur, disertai dengan persentase tiap kelas:

Berdasarkan Tabel XV menggunakan 20 fitur pada model Decision Tree dapat menjadi solusi yang baik karena performanya hanya sedikit berbeda dibandingkan dengan model yang menggunakan lebih banyak fitur. Selain itu, penggunaan 20 fitur juga dapat membantu mengurangi waktu pelatihan model. Dengan jumlah fitur yang lebih sedikit, model dapat bekerja lebih efisien, dan mempercepat proses pelatihan tanpa mengorbankan performa yang signifikan. Hal ini sangat bermanfaat terutama pada dataset besar atau dalam kasus-kasus di mana waktu pelatihan menjadi factor

TABEL XV
CONFUSION MATRIX SELEKSI FITUR DECISION TREE

Jumlah Fitur	Prediksi \Aktual	Kelas 0 (%)	Kelas 1 (%)	Kelas 2 (%)
10 Fitur	Kelas 0	97.80	1.31	0.90
	Kelas 1	5.13	94.58	0.29
	Kelas 2	1.15	0.35	98.50
20 Fitur	Kelas 0	98.55	1.58	0.57
	Kelas 1	1.45	97.03	1.58
	Kelas 2	0.10	0.79	98.89

30 Fitur	Kelas 0	98.86	1.13	0.62
	Kelas 1	2.24	95.73	1.87
	Kelas 2	0.40	0.78	98.52
40 Fitur	Kelas 0	98.85	1.04	0.76
	Kelas 1	2.48	96.00	1.51
	Kelas 2	0.37	0.92	98.50

Tabel XVI memuat daftar nama-nama fitur atau nama-nama kolom yang terdapat dalam dataset yang terpilih sebagai 20 fitur penting pada model Decision Tree. Fitur-fitur ini telah dipilih berdasarkan tingkat kepentingannya yang dihitung melalui metode analisis model Decision Tree, yang menentukan seberapa besar kontribusi masing-masing fitur dalam memprediksi hasil dari model tersebut. Setiap fitur mencerminkan berbagai aspek yang berkaitan dengan variabilitas detak jantung dan analisis sinyal fisiologis, seperti Heart Rate Variability (HRV), yang digunakan untuk mengukur dan menganalisis kesehatan jantung serta faktor-faktor terkait lainnya.

TABEL XVI
20 FITUR TERATAS DECISION TREE

No.	Feature	Importance
1	HR_SQRT	299.126
2	SDRR_REL_RR	131.172
3	SD1_BOXCOX	115.446
4	HR	108.044
5	SSSQ Label	87.050
6	MEAN_RR	45.320
7	HF_BOXCOX	38.878
8	MEDIAN_REL_RR	37.013
9	LF_BOXCOX	33.438
10	pNN50	31.379
11	MEDIAN_RR	28.944
12	LF_PCT	11.945
13	LF	10.315
14	HF_VLF	10.104
15	HR_HF	6.810
16	TP_SQRT	3.135
17	SKEW_REL_RR	1.880
18	KURT_YEO_JONSON	0
19	SDRR_RMSSD_LOG	0
20	RMSSD_LOG	0

Model random forest dengan seleksi fitur juga menunjukkan hasil yang positif hal ini bisa terlihat pada Tabel XVII dibawah. Pada model *Random Forest* dengan 10 fitur, akurasi mencapai 0.99 dengan konsistensi yang tinggi dalam metrik lainnya, seperti *Precision*, *Recall*, dan *F1-Score*. *Precision* untuk Kelas 0, Kelas 1, dan Kelas 2 semuanya di atas 0.98, menunjukkan kemampuan yang sangat baik dalam mengidentifikasi kelas-kelas tersebut. *Recall* untuk Kelas 0 mencapai 0.99, menunjukkan bahwa model ini sangat mampu mendeteksi kasus-kasus positif pada Kelas 0. *F1-Score* secara keseluruhan juga tinggi di semua kelas, dengan rata-rata *F1* sebesar 0.99, menegaskan performa yang sangat baik secara keseluruhan.

Model *Random Forest* dengan 20 fitur menunjukkan akurasi yang sama tinggi, yaitu 0.99. *Precision* untuk Kelas 0 tetap stabil di angka 1.00, sementara *Recall* untuk Kelas 1 meningkat menjadi 0.98, menunjukkan deteksi yang lebih baik pada kasus positif. *F1-Score* untuk Kelas 1 juga meningkat menjadi 0.99, memperkuat performa pada kelas tersebut. Secara keseluruhan, meskipun jumlah fitur meningkat, model tetap mempertahankan kinerja yang sangat baik dengan rata-rata *F1-Score* sebesar 0.99.

TABEL XVII
PERBANDINGAN HASIL SELEKSI FITUR RANDOM FOREST

Metrik	Random Forest (10 Fitur)	Random Forest (20 Fitur)	Random Forest (30 Fitur)	Random Forest (40 Fitur)
Akurasi	0.99	0.99	0.99	0.99
<i>Precision</i> Kelas 0	0.99	1.00	1.00	1.00
<i>Recall</i> Kelas 0	0.99	0.99	0.99	0.99
<i>F1-Score</i> Kelas 0	0.99	0.99	0.99	0.99

Metrik	Random Forest (10 Fitur)	Random Forest (20 Fitur)	Random Forest (30 Fitur)	Random Forest (40 Fitur)
Precision Kelas 1	0.99	1.00	1.00	1.00
Recall Kelas 1	0.96	0.98	0.97	0.97
F1-Score Kelas 1	0.98	0.99	0.98	0.98
Precision Kelas 2	0.98	0.98	0.98	0.98
Recall Kelas 2	0.99	1.00	1.00	1.00
F1-Score Kelas 2	0.99	0.99	0.99	0.99
Rata-rata F1	0.99	0.99	0.99	0.99

Model *Random Forest* dengan 30 fitur menunjukkan stabilitas yang sama dengan akurasi mencapai 0.99. *Precision*, *Recall*, dan F1-Score tetap tinggi di semua kelas, dengan nilai rata-rata F1 tetap di sekitar 0.99. Meskipun jumlah fitur bertambah, model tetap mampu menjaga performa yang tinggi dan dapat menangani kompleksitas data dengan baik, mempertahankan hasil yang konsisten di semua aspek.

Pada model *Random Forest* dengan 40 fitur, hasilnya hampir identik dengan yang sebelumnya, dengan akurasi tetap di angka 0.99. *Precision*, *Recall*, dan F1-Score tetap sangat tinggi di semua kelas, menunjukkan bahwa peningkatan jumlah fitur tidak berdampak signifikan pada penurunan kinerja. Rata-rata F1-Score tetap stabil di 0.99, menunjukkan kemampuan model untuk menjaga konsistensi hasil yang sangat baik.

TABEL XVIII
CONFUSION MATRIX SELEKSI FITUR RANDOM FOREST

Jumlah Fitur	Prediksi \ Aktual	Kelas 0 (%)	Kelas 1 (%)	Kelas 2 (%)
10 Fitur	Kelas 0	99.43	0.04	0.60
	Kelas 1	1.12	96.32	2.57
	Kelas 2	0.26	0.29	99.35
20 Fitur	Kelas 0	99.42	0.02	0.66
	Kelas 1	1.67	97.32	1.10
	Kelas 2	0.05	0.18	99.77
30 Fitur	Kelas 0	99.42	0.01	0.61
	Kelas 1	1.84	96.91	1.25
	Kelas 2	0.10	0.18	99.72
40 Fitur	Kelas 0	99.46	0.01	0.53
	Kelas 1	2.13	97.34	0.61
	Kelas 2	0.07	0.11	99.78

Berdasarkan hasil analisis, penggunaan 20 fitur sudah cukup memberikan performa yang optimal dalam hal akurasi dan metrik lainnya. Pada Tabel XVIII juga terlihat bahwa penggunaan 20 fitur memiliki hasil prediksi yang tinggi untuk kelas minoritas atau kelas. Sementara penggunaan 40 fitur hanya memberikan sedikit peningkatan yang tidak signifikan dalam hal performa. Oleh karena itu, penggunaan 20 fitur adalah pilihan yang efisien untuk menjaga performa tinggi sambil mempertahankan waktu pelatihan yang optimal.

TABEL XIX
20 FITUR PADA RANDOM FOREST

No.	Feature	Importance
1	HR	0.092491
2	MEAN_RR	0.084468
3	MEAN_RR_LOG	0.083686
4	HR_SQRT	0.078871
5	MEDIAN_RR	0.071372
6	MEAN_RR_SQRT	0.066333
7	SD1_BOXCOX	0.028371
8	SSSQ Label	0.020204
9	MEDIAN_REL_RR_LOG	0.018500
10	LF_BOXCOX	0.016567
11	HF	0.016282
12	SDRR_REL_RR	0.015952
13	SD1_LOG	0.015806
14	HF_LOG	0.015559
15	RMSSD_REL_RR_LOG	0.015390

16	SDSD_REL_RR_LOG	0.014951
17	MEDIAN_REL_RR	0.014832
18	LF	0.014785
19	SD1	0.014733
20	RMSSD_REL_RR	0.014698

Tabel XIX menyajikan daftar nama-nama fitur yang terpilih sebagai 20 fitur terpenting dalam model Random Forest. Fitur-fitur ini dipilih berdasarkan analisis kepentingan yang dihitung menggunakan model Random Forest, yang menggabungkan beberapa pohon keputusan untuk menentukan variabel paling relevan dalam prediksi. Fitur-fitur yang terpilih ini mencerminkan aspek penting dari variabilitas detak jantung dan analisis sinyal fisiologis lainnya, yang berhubungan dengan kesehatan jantung. Selain urutan fitur hingga peringkat ke-20, lampiran juga menyertakan penjelasan mengenai arti dari setiap fitur tersebut. Penjelasan ini bertujuan memberikan pemahaman tentang kontribusi masing-masing fitur dalam model prediksi. Dengan mengetahui peran setiap fitur, pembaca dapat lebih memahami bagaimana fitur-fitur tersebut berinteraksi untuk menghasilkan prediksi yang akurat.

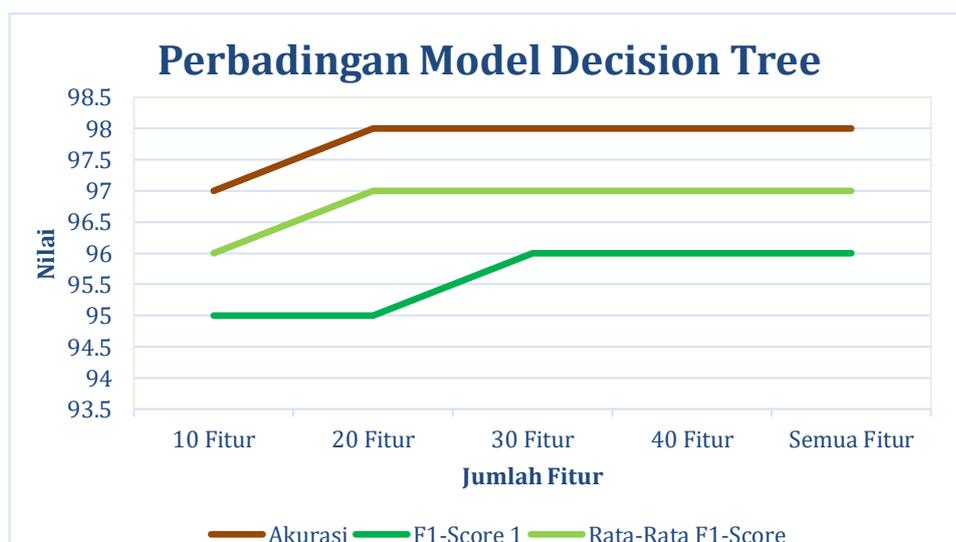
Decision Tree menawarkan interpretasi yang lebih sederhana dengan visualisasi yang mudah dipahami, serta menunjukkan akurasi tinggi antara 0.97 hingga 0.98, bahkan dengan jumlah fitur yang bervariasi. Selain itu, Precision, Recall, dan F1-Score tetap konsisten, mencerminkan kestabilan model dalam berbagai kondisi. Keunggulan utama dari Decision Tree adalah kemampuannya dalam memberikan keputusan yang transparan dan mudah diinterpretasikan.

Di sisi lain, Random Forest menunjukkan performa yang lebih unggul dengan akurasi yang mencapai 0.99 dalam semua pengujian, terlepas dari jumlah fitur yang digunakan. Metrik evaluasi seperti Precision, Recall, dan F1-Score juga tetap tinggi dan stabil di semua kelas. Keunggulan utama Random Forest terletak pada kemampuannya mengurangi overfitting dengan menggabungkan banyak pohon keputusan yang bekerja secara independen, sehingga mampu menangani kompleksitas data yang lebih besar.

Meskipun Decision Tree lebih mudah diinterpretasikan, Random Forest memberikan hasil yang lebih konsisten dan akurat. Oleh karena itu, untuk analisis yang lebih kompleks dan memerlukan kestabilan yang tinggi, Random Forest menjadi pilihan yang lebih baik karena mampu mengintegrasikan berbagai pohon keputusan untuk mencapai performa yang optimal.

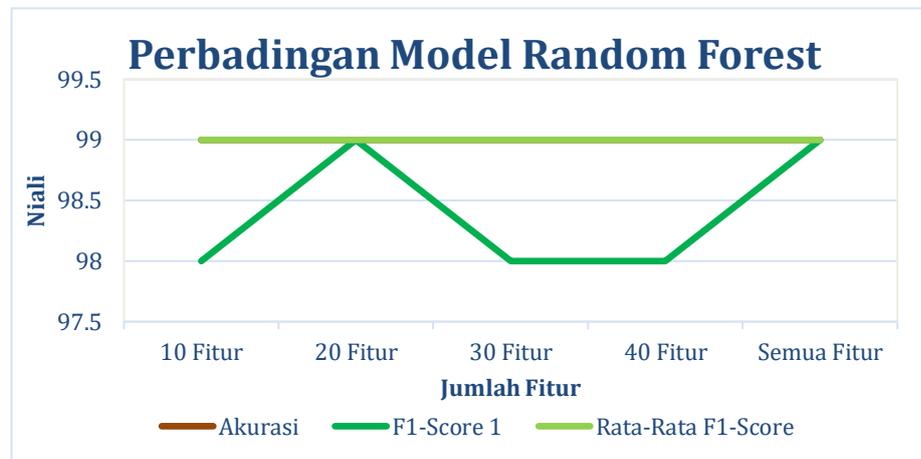
D. Analisis Perbandingan Hasil Seluruh Pengujian

Pada model Decision Tree terlihat pada gambar 5.1, pengurangan jumlah fitur menjadi 20 fitur menghasilkan kinerja yang hampir setara atau sedikit meningkat untuk sebagian besar metrik, seperti akurasi dan presisi untuk Kelas 0. Namun, ada sedikit trade-off pada presisi dan recall untuk Kelas 1, di mana penggunaan semua fitur menunjukkan stabilitas yang lebih baik. Confusion matrix menunjukkan bahwa meskipun dengan 20 fitur, model tetap mampu mempertahankan akurasi tinggi untuk Kelas 0 dan Kelas 2, tetapi sedikit penurunan pada Kelas 1. Temuan ini menunjukkan bahwa seleksi fitur dapat mengoptimalkan kompleksitas model tanpa penurunan kinerja yang signifikan, mendukung penggunaannya untuk meningkatkan efisiensi komputasi sambil mempertahankan kinerja klasifikasi yang tangguh.



Gambar 2. Grafik Perbandingan Model Decision Tree

Pada model Random Forest menurunkan jumlah fitur menjadi 20 fitur menunjukkan hasil yang sangat baik bisa terlihat dari grafik yang ada pada Gambar 5.2, dengan akurasi tetap tinggi sebesar 0.99 dan hampir sempurna untuk semua kelas. Matriks kebingungan menunjukkan bahwa model mampu mempertahankan presisi, *recall*, dan F1-score yang sangat baik untuk Kelas 0 dan Kelas 2, sementara untuk Kelas 1, meskipun sedikit penurunan pada presisi dan *recall*, kinerja secara keseluruhan tetap tinggi. Dibandingkan dengan penggunaan seluruh fitur, model dengan 20 fitur menunjukkan keseimbangan yang baik antara kompleksitas model dan akurasi, menjaga efisiensi komputasi tanpa kehilangan kinerja yang signifikan.



Gambar 3. Grafik Perbandingan Model Random Forest

Tabel XX menunjukkan bahwa seleksi fitur berhasil mengurangi waktu pelatihan pada kedua model, yaitu Decision Tree dan Random Forest. Seiring dengan penurunan jumlah fitur, waktu pelatihan juga mengalami penurunan yang signifikan. Misalnya, pada model Decision Tree, dengan hanya 10 fitur, waktu pelatihan adalah 2.25 detik, sementara pada penggunaan semua fitur, waktu pelatihan meningkat menjadi 16.45 detik. Hal yang serupa terjadi pada Random Forest, di mana pelatihan dengan 10 fitur memakan waktu 25.13 detik, dan waktu pelatihan meningkat menjadi 114.61 detik saat menggunakan semua fitur. Hal ini menunjukkan bahwa seleksi fitur, yang mengurangi jumlah fitur yang digunakan dalam pelatihan, dapat mempercepat proses pelatihan model dengan mengurangi kompleksitasnya.

TABEL XX
PERBANDINGAN WAKTU PELATIHAN DALAM DETIK (S)

	Decision Tree	Random Forest
10 Fitur	2.25	25.13
20 Fitur	3.76	36.59
30 Fitur	5.29	45.77
40 Fitur	9.25	54.63
Semua Fitur	16.45	114.61

VI. SIMPULAN DAN SARAN

A. Simpulan

Penelitian ini mengevaluasi Decision Tree dan Random Forest dalam memprediksi kesehatan mental serta efektivitas seleksi fitur. Tanpa seleksi fitur, Random Forest unggul dalam data berdimensi besar tetapi kurang baik dalam mengenali kelas minoritas, sementara Decision Tree lebih seimbang meskipun akurasinya sedikit lebih rendah. Optimasi Hyperparameter meningkatkan performa keduanya, dengan Random Forest mempertahankan akurasi tinggi dan Decision Tree unggul dalam keseimbangan metrik. Seleksi fitur membantu interpretasi, mengurangi waktu pelatihan, dan tetap menjaga performa model. Decision Tree lebih efektif untuk kelas minoritas, sedangkan Random Forest lebih baik untuk klasifikasi kelas mayoritas, sehingga pemilihan algoritma bergantung pada kebutuhan spesifik.

B. Saran

Penelitian selanjutnya disarankan menggunakan dataset yang lebih lengkap dan representatif untuk meningkatkan generalisasi model dalam prediksi kesehatan mental. Selain itu, eksplorasi model lain seperti Gradient Boosting, XGBoost, atau deep learning dapat membantu menangani data kompleks dan meningkatkan akurasi. Teknik penyeimbangan data seperti oversampling atau undersampling juga perlu dipertimbangkan untuk mengatasi ketidakseimbangan kelas secara lebih efektif.

DAFTAR PUSTAKA

- [1] A. Gjoreski, M. Luštrek dan M. Gams, "Machine learning for stress detection and stress level prediction from physiological data," IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1-8, 2017.
- [2] S. H. Seo, C. H. Choi, K. H. Kim, Y. J. Lee, H. J. Kim dan J. K. Lee, "Stress and Heart Rate Variability: A Meta-Analysis and Review of the Literature," Psychiatry investigation, vol. 15.3, p. 239, 2018.
- [3] M. A. Khan, "A survey of wearable sensors and their applications.," IEEE Communications Surveys & Tutorials, no. 22.3, 2020.
- [4] P. Melillo, A. B. Asif, A. C. Finzi dan S. L. Tofani, "Stress detection in computer users based on multimodal physiological measures," IEEE Transactions on Affective Computing, vol. 11.2, pp. 315-328, 2018.
- [5] J. S. Choi, S. H. Lee dan M. Lee, "A review of stress detection using wearable sensors and machine learning.," IEEE Access, vol. 8, 2020.
- [6] R. A. G. Vos, K. T. Z. S. Vos dan R. M. Aarts, "Generalizable machine learning for stress monitoring from wearable devices: A systematic literature review," International Journal of Medical Informatics, vol. 173, 2023.
- [7] S. L. Lau, "Sensitivity and Specificity of Wearable Device Measured Physical Activity Compared to Research-Grade Accelerometry: Systematic Review and Meta-Analysis," Journal of Medical Internet Research, vol. 24.11, 2022.
- [8] A. Mundada, "Wearable Sensors for Remote Health Monitoring in the Era of Internet of Things," 2022.
- [9] A. Puspitaningrum, "Analisis Perbandingan Algoritma C4.5 dan Naïve Bayes untuk Klasifikasi Status Gizi Balita," Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK), vol. 6.2, pp. 377-384, 2019.
- [10] S. Wibowo dan A. Puspitaningrum, "Perbandingan Algoritma Random Forest dan Support Vector Machine (SVM) untuk Klasifikasi Data Pasien Diabetes," Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK), vol. 7, no. 1, 2020.
- [11] B. H. Shekar dan G. Dagnev, "Grid Search-Based Hyperparameter Tuning and Classification of Microarray Cancer Data," 2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP), pp. 1-8, 2019.
- [12] A. Handayani, A. Jamal dan A. A. Septiandri, "Evaluasi Tiga Jenis Algoritme Berbasis Pembelajaran," Jurnal Nasional Teknik Elektro dan Teknologi Informasi, vol. 6, no. 4, pp. 394-403, 2017.
- [13] J. Xu, Y. Zhang dan D. Miao, "Three-way confusion matrix for classification: A measure driven view," Information Sciences, vol. 507, pp. 772-794, 2020.
- [14] A. N. Fauziah, "nalisis sentimen menggunakan naïve bayes classifier dan inset lexicon pada twitter (studi kasus: Mie Gacoan).," Bachelor's thesis, Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta, 2023.