

Perbandingan Multi Algoritma Klasifikasi dan Tuning Parameter untuk Prediksi Ketergantungan Skincare Berbasis Streamlit

Aneira Vicentiya Kuncoro*, Laila Maulin Ni'mah, Edi Faisal

Fakultas Ilmu Komputer, Program Studi Teknik Informatika, Universitas Dian Nuswantoro, Semarang, Indonesia

Email: ¹*111202214226@mhs.dinus.ac.id, ²111202214424@mhs.dinus.ac.id, ³faisal@dsn.dinus.ac.id

Email Penulis Korespondensi: 111202214226@mhs.dinus.ac.id

Submitted: 07/07/2025; Accepted: 01/09/2025; Published: 02/09/2025

Abstrak—Penggunaan produk skincare di Indonesia mengalami peningkatan signifikan seiring dengan meningkatnya kesadaran masyarakat akan pentingnya perawatan kulit, namun hal ini juga memunculkan indikasi perilaku ketergantungan yang perlu diantisipasi, terutama pada kelompok usia muda. Penelitian ini bertujuan untuk membangun sistem prediksi tingkat ketergantungan terhadap skincare berdasarkan atribut demografis, psikologis, dan perilaku yang dikumpulkan melalui survei daring. Selain itu, dilakukan perbandingan lima algoritma klasifikasi, Decision Tree, Random Forest, Support Vector Machine, K-Nearest Neighbors, dan Logistic Regression untuk menentukan model terbaik yang paling akurat dan efisien dalam memprediksi kecenderungan ketergantungan tersebut. Data yang diperoleh diproses melalui normalisasi dan transformasi fitur kategorikal dengan One-Hot Encoding, lalu dievaluasi menggunakan metrik akurasi, precision, recall, dan F1-score. Hasil penelitian menunjukkan bahwa algoritma Decision Tree memberikan performa terbaik dengan akurasi mencapai 87% dan keunggulan pada interpretabilitas model. Model ini kemudian diimplementasikan dalam bentuk aplikasi web interaktif berbasis Streamlit yang memungkinkan pengguna melakukan prediksi secara mandiri dan real-time. Kontribusi dari penelitian ini adalah tersedianya sistem prediksi yang mendukung edukasi dan pengambilan keputusan lebih bijak dalam penggunaan skincare, sekaligus membuka peluang pemanfaatan teknologi machine learning untuk isu-isu kesehatan dan kecantikan berbasis data.

Kata Kunci: Skincare; Ketergantungan; Machine Learning; Decision Tree; Streamlit; Klasifikasi; Prediksi

Abstract—The use of skincare products in Indonesia has increased significantly along with the increasing public awareness of the importance of skincare, but this also raises indications of dependence behaviour that needs to be anticipated, especially in young age groups. This research aims to build a skincare dependency prediction system based on demographic, psychological and behavioural attributes collected through an online survey. In addition, a comparison of five classification algorithms—Decision Tree, Random Forest, Support Vector Machine, K-Nearest Neighbors, and Logistic Regression—was conducted to determine the best model that is most accurate and efficient in predicting the dependency tendency. The data obtained was processed through normalisation and categorical feature transformation with One-Hot Encoding, then evaluated using accuracy, precision, recall, and F1-score metrics. The results showed that the Decision Tree algorithm provided the best performance with accuracy reaching 87% and excellence in model interpretability. The model was then implemented in the form of an interactive web application based on Streamlit that allows users to make predictions independently and in real-time. The contribution of this research is the availability of a prediction system that supports education and wiser decision-making in the use of skincare, as well as opening up opportunities for the utilisation of machine learning technology for other issues.

Keywords: Skincare; Dependency; Machine Learning; Decision Tree; Streamlit; Classification; Prediction

1. PENDAHULUAN

Industri skincare di Indonesia mengalami pertumbuhan yang signifikan dalam beberapa tahun terakhir. Perkembangan ini didorong oleh meningkatnya kesadaran masyarakat terhadap pentingnya menjaga kesehatan kulit. Selain itu, pengaruh media sosial dan promosi oleh para influencer turut membentuk persepsi estetika dan gaya hidup yang mendorong konsumsi produk perawatan kulit secara masif [1].

Dalam rangka menjaga kualitas dan keamanan produk perawatan kulit yang beredar, pemerintah telah memberlakukan regulasi kosmetik nasional. Regulasi ini mengarahkan pelaku usaha untuk memastikan bahwa produk yang dipasarkan sesuai dengan ketentuan mengenai keamanan bahan, efektivitas penggunaan, serta kualitas kemasan dan label. Di samping itu, aspek edukatif juga menjadi bagian penting dalam regulasi, guna membangun kesadaran masyarakat untuk menggunakan kosmetik secara bijak dan bertanggung jawab [2].

Namun, di balik pertumbuhan tersebut, muncul kekhawatiran mengenai kecenderungan penggunaan skincare yang berlebihan, khususnya di kalangan remaja dan dewasa muda. Gejala seperti pengeluaran berlebih, ketergantungan emosional terhadap produk, serta tekanan psikologis yang berkaitan dengan pencapaian standar kecantikan menunjukkan adanya potensi ketidakseimbangan dalam perilaku konsumtif [3], [4].

Situasi ini menegaskan perlunya pendekatan berbasis data yang mampu mengidentifikasi kecenderungan tersebut secara lebih sistematis dan objektif. Salah satu pendekatan yang relevan untuk digunakan adalah machine learning, yang telah digunakan luas dalam analisis perilaku dan klasifikasi preferensi pengguna berdasarkan data historis.

Studi-studi sebelumnya telah menerapkan teknologi ini, misalnya dalam pengembangan sistem rekomendasi produk skincare berbasis *deep learning* yang menyesuaikan dengan kondisi kulit dan kandungan produk [5]. Berbagai metode klasifikasi juga diterapkan untuk mengelompokkan konsumen seperti Beberapa perbandingan algoritma berdasarkan karakteristik demografis dan kebiasaan pembelian mereka [6].

Untuk menyajikan sistem prediksi yang mudah diakses dan dapat dimanfaatkan langsung oleh pengguna, dibutuhkan antarmuka web interaktif. *Streamlit* merupakan salah satu platform yang banyak digunakan untuk

mengembangkan antarmuka semacam ini secara praktis, serta mendukung visualisasi dan eksplorasi data secara *real-time* [7].

Selain itu, peningkatan performa model klasifikasi sangat bergantung pada kemampuan dalam menyesuaikan parameter algoritma melalui proses *hyperparameter tuning*. Strategi ini memungkinkan model bekerja lebih optimal terhadap karakteristik dataset tertentu, sebagaimana dibuktikan dalam sejumlah studi terkait segmentasi pasar dan deteksi risiko kredit [8].

Kendati begitu, kajian yang secara khusus memodelkan kecenderungan terhadap ketergantungan skincare menggunakan pendekatan machine learning masih sangat terbatas. Belum ditemukan penelitian yang secara bersamaan membandingkan lima algoritma klasifikasi populer seperti *Decision Tree*, *Random Forest*, *Support Vector Machine (SVM)*, *K-Nearest Neighbors (KNN)*, dan *Logistic Regression*, dengan penerapan penyetelan parameter yang optimal, serta implementasi langsung dalam bentuk aplikasi web interaktif.

Penelitian ini bertujuan untuk mengembangkan sistem prediksi tingkat ketergantungan terhadap produk skincare berbasis survei daring. Survei ini mengumpulkan data terkait demografi (jenis kelamin dan usia), kondisi psikologis (tingkat stres dan persepsi terhadap pentingnya skincare), serta perilaku konsumen (frekuensi penggunaan, durasi penggunaan media sosial, dan pengeluaran bulanan). Kelima algoritma pembelajaran mesin akan diuji dan dibandingkan menggunakan metrik evaluasi seperti akurasi, *precision*, *recall*, dan *F1-score*. Penyesuaian parameter akan dilakukan melalui proses tuning guna memperoleh performa optimal dari masing-masing model.

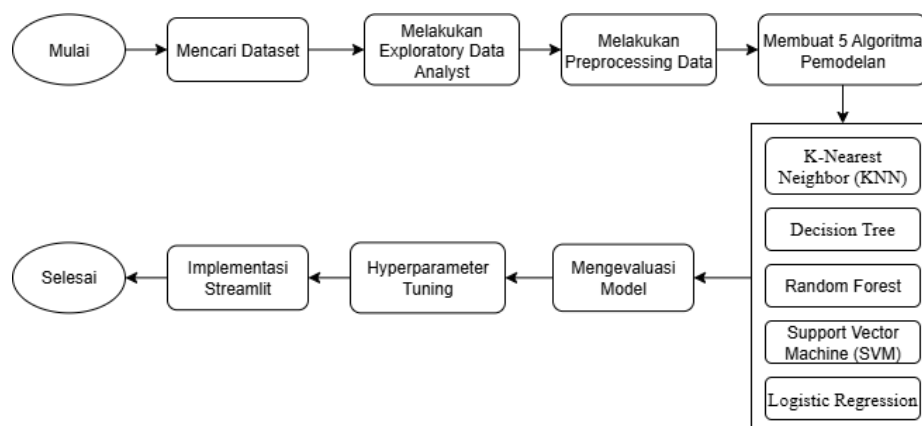
Penelitian ini memberikan kontribusi dalam bentuk evaluasi komparatif dari lima algoritma klasifikasi dalam konteks prediksi ketergantungan skincare, serta implementasi model terbaik ke dalam platform aplikasi web yang interaktif. Hasil yang diinginkan dari penelitian ini adalah diperolehnya algoritma dengan performa paling optimal dari sisi akurasi dan efisiensi, yang dapat diimplementasikan secara nyata dalam aplikasi prediktif berbasis web. Hasilnya tidak hanya berguna dalam konteks akademik sebagai studi pembandingan performa algoritma, tetapi juga menghasilkan aplikasi fungsional yang dapat digunakan secara langsung oleh masyarakat.

Melalui sistem ini, diharapkan pengguna dapat lebih memahami kebiasaan penggunaan skincare mereka secara mandiri. Aplikasi juga ditujukan untuk membantu pengguna dalam mengambil keputusan yang lebih bijak, mencegah potensi ketergantungan, serta mendorong perilaku perawatan kulit yang lebih sehat dan bertanggung jawab. Sistem ini dapat menjadi pijakan awal bagi pengembangan inovasi digital lainnya di bidang kecantikan dan kesehatan berbasis data.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Penelitian ini menggunakan pendekatan kuantitatif dengan metode eksperimen, yang bertujuan untuk membandingkan performa lima algoritma klasifikasi dalam memprediksi tingkat ketergantungan pengguna terhadap produk skincare. Alur lengkap dari tahapan-tahapan tersebut dapat dilihat pada Gambar 1. Diagram tersebut menjelaskan langkah-langkah inti yang dilakukan, seperti eksplorasi data, preprocessing, pembuatan model klasifikasi, evaluasi performa, proses pelatihan ulang jika hasil belum optimal, hingga tahap *hyperparameter tuning* dan implementasi model terbaik menggunakan Streamlit. dan mengacu pada penelitian terdahulu dalam pengembangan sistem berbasis web dan prediksi data [9], [10].



Gambar 1. Tahapan Penelitian

Gambar 1 menunjukkan proses dimulai dari pencarian dataset yang relevan, kemudian dilanjutkan dengan tahap eksplorasi data (*exploratory data analysis*) dan *preprocessing* untuk mempersiapkan data yang bersih dan siap digunakan. Setelah itu, dilakukan pelatihan model menggunakan lima algoritma berbeda, lalu dievaluasi untuk menentukan model dengan akurasi tertinggi. Jika hasilnya belum optimal, maka proses pelatihan diulang hingga mendapatkan akurasi yang diharapkan. Apabila model sudah memenuhi kriteria, maka dilakukan hyperparameter

tuning untuk mengoptimalkan performa model. Setelah model terbaik diperoleh, langkah terakhir adalah mengimplementasikannya ke dalam aplikasi berbasis Streamlit untuk keperluan visualisasi dan interaktifasi pengguna. Proses ini diakhiri ketika sistem berhasil dijalankan secara efektif.

2.2 Persiapan Dataset

Dataset yang digunakan dalam penelitian ini diperoleh dari platform *Kaggle*, yang berisi data terkait kebiasaan penggunaan skincare oleh pengguna. Dataset ini terdiri dari 500 data pengguna dengan 6 atribut utama, mencakup atribut-atribut penting seperti usia, jenis kelamin, frekuensi penggunaan skincare, waktu penggunaan perangkat digital, pengeluaran bulanan, dan persepsi pentingnya skincare. Dataset yang digunakan dalam penelitian ini dibagi menjadi dua bagian dengan rasio 80:20, di mana 80% data digunakan sebagai data latih dan 20% sebagai data uji [11]. Pembagian dilakukan secara acak menggunakan fungsi `train_test_split` dari library `scikit-learn`, dengan pengaturan `random_state` agar hasil dapat direproduksi. Penggunaan dataset dari *Kaggle* memungkinkan peneliti untuk bekerja dengan data nyata dalam skala yang lebih besar. Referensi pendekatan serupa juga digunakan dalam penelitian klasifikasi kulit wajah [11] dan prediksi kesehatan [12]. Pada Tabel 1 berikut menyajikan deskripsi ringkas atribut-atribut yang digunakan dalam dataset penelitian ini.

Tabel 1. Prediksi Atribut

Atribut	Jenis Data	Deskripsi
Usia	Numerik	Umur responden
Jenis Kelamin	Kategorikal	Laki-laki / Perempuan
Frekuensi Pemakaian	Numerik	Kali pemakaian skincare per minggu
Waktu Layar Harian	Numerik	Jam penggunaan gadget per hari
Pengeluaran Skincare	Numerik	Estimasi pengeluaran bulanan (Rp)
Persepsi Penting Skincare	Skala Likert	Penilaian 1–5 terhadap pentingnya skincare

Berdasarkan Tabel 1, dataset memiliki 6 atribut utama, yaitu *age* (usia), *gender* (jenis kelamin), *routine_frequency* (frekuensi penggunaan skincare), *social_media_hours* (waktu penggunaan perangkat digital), *monthly_expense* (pengeluaran bulanan), dan *skincare_importance* (persepsi pentingnya skincare). Atribut *age*, *social_media_hours*, *monthly_expense*, dan *skincare_importance* bertipe data numerik, sedangkan *gender* dan *routine_frequency* bertipe data kategori/teks.

2.3 Exploratory Data Analyst

Langkah awal dalam proses ini dilakukan dengan pendekatan eksploratif yang bertujuan untuk mengamati keseluruhan data secara komprehensif dan sistematis. Fokus utamanya adalah memahami struktur, isi, serta pola-pola tersembunyi dalam dataset skincare, termasuk hubungan antar variabel yang terkandung di dalamnya. Dalam penelitian ini, proses EDA dilaksanakan melalui beberapa tahapan utama berikut :

a. Statistik Deskriptif

Tahap ini dilakukan dengan menampilkan ringkasan data numerik yang bertujuan memberikan gambaran awal terhadap karakteristik populasi dalam dataset. Variabel yang dianalisis mencakup usia, durasi tidur, pengeluaran bulanan untuk skincare, serta tingkat stres. Informasi numerik seperti rerata, nilai minimum dan maksimum, serta sebaran data secara umum dihitung untuk memberikan pemahaman awal yang *representatif*.

b. Analisis Korelasi

Langkah berikutnya fokus pada pengukuran hubungan antara dua variabel numerik dalam dataset. Misalnya, dapat dianalisis hubungan antara besar pengeluaran dengan tingkat kepedulian terhadap skincare, atau hubungan antara waktu tidur dan tingkat stres pengguna. Kekuatan dan arah hubungan tersebut dihitung dengan teknik korelasi Pearson dan divisualisasikan dalam bentuk peta panas (*heatmap*) untuk memudahkan dalam memahami pola interaksi antar variabel.

c. Visualisasi Data

Visualisasi berfungsi sebagai alat bantu untuk mempermudah interpretasi data serta menampilkan pola atau anomali yang sulit dikenali dari tampilan angka biasa. Berbagai jenis grafik seperti *histogram*, *boxplot*, dan *bar chart* digunakan untuk menggambarkan distribusi data serta membandingkan variabel kategori. Sebagai contoh, grafik dapat digunakan untuk memperlihatkan perbedaan tingkat pemakaian skincare antara kelompok jenis kelamin, atau variasi pengeluaran berdasarkan tipe kulit. Penyajian visual ini menjadikan hasil eksplorasi data lebih informatif dan mudah dipahami oleh berbagai pihak, baik akademisi maupun praktisi [13].

2.4 Pre-Processing Data

Langkah pra-pemrosesan bertujuan untuk mempersiapkan data agar layak digunakan dalam tahap analisis dan pembangunan model. Tahapan-tahapan yang dilakukan meliputi:

- Proses dimulai dengan pemeriksaan terhadap setiap atribut dalam dataset untuk memastikan tidak terdapat nilai kosong atau duplikat yang dapat mengganggu integritas data. Validasi ini penting untuk memastikan bahwa data bersih dan dapat diandalkan dalam proses analisis berikutnya.



- b. Selanjutnya dilakukan peninjauan terhadap struktur dan tipe data pada seluruh atribut, baik yang bersifat numerik maupun kategorikal. Pemeriksaan ini dilakukan untuk memastikan bahwa format data sudah sesuai dan tidak memerlukan penyesuaian tambahan sebelum pemrosesan lebih lanjut.
- c. Data numerik seperti usia dan pengeluaran bulanan dianalisis untuk melihat distribusi nilai, mendeteksi keberadaan nilai ekstrem, serta memahami karakteristik statistik seperti rata-rata dan simpangan baku. Sementara itu, data kategorikal seperti jenis kelamin, tipe kulit, dan persepsi pentingnya skincare diperiksa berdasarkan jumlah dan keragaman kategorinya.
- d. Untuk memberikan pemahaman visual terhadap pola data, dilakukan visualisasi menggunakan grafik. *Histogram* digunakan untuk melihat sebaran data numerik, sedangkan diagram batang digunakan untuk menggambarkan jumlah setiap kategori dalam fitur kategorikal.
- e. Seluruh fitur non-numerik dikonversi ke bentuk numerik agar dapat dikenali oleh algoritma pembelajaran mesin. Konversi ini memastikan semua variabel dapat diproses secara konsisten dalam pemodelan, tanpa kehilangan makna data aslinya.
- f. Terakhir, dataset dibagi menjadi dua bagian yaitu data latih dan data uji dengan proporsi tertentu. Pembagian ini bertujuan agar model dapat diuji menggunakan data yang belum dikenalnya, sehingga performa model dapat dievaluasi secara objektif dan tidak bias terhadap data pelatihan [12], [14], [15].

2.5 K-Nearest Neighbor (K-NN)

Algoritma *K-Nearest Neighbor* (K-NN) merupakan metode klasifikasi yang bekerja dengan mencari sejumlah tetangga terdekat (k) dari data baru berdasarkan jarak (biasanya *Euclidean*), lalu mengklasifikasikan data tersebut berdasarkan mayoritas label dari tetangganya. K-NN bersifat non-parametrik dan termasuk dalam *supervised learning* [16]. Kelebihan dari algoritma ini adalah sederhana dan efektif dalam klasifikasi data yang memiliki batasan yang jelas. Namun, kelemahannya yaitu sensitif terhadap skala data dan performanya dapat menurun jika jumlah data sangat besar karena perhitungan jarak yang intensif.

Prediksi kelas ditentukan berdasarkan mayoritas kelas dari k tetangga terdekat :

$$d(x, x_i) = \left(\sum_{j=1}^n |x_j - x_{ij}|^p \right)^{1/p} \quad (1)$$

Model KNN menggunakan parameter `n_neighbors` sebanyak 5 untuk menentukan jumlah tetangga terdekat yang akan dipertimbangkan dalam proses klasifikasi. Metode pengukuran jarak yang digunakan adalah *minkowski* dengan parameter $p=2$, yang setara dengan penggunaan jarak *Euclidean*. Parameter ini diuji dalam berbagai konfigurasi untuk melihat sensitivitas model terhadap perubahan jumlah tetangga.

2.6 Decision Tree

Metode klasifikasi berbasis struktur pohon yang memetakan kondisi dan keputusan dari data ke dalam bentuk hierarki. Proses pembentukan pohon dilakukan dengan memilih atribut terbaik berdasarkan pengukuran seperti *information gain* atau *Gini index* [17]. Kelebihan dari algoritma ini yaitu interpretasi yang mudah dan hasil model yang dapat divisualisasikan secara intuitif. Namun, kekurangannya adalah mudah mengalami *overfitting* jika tidak dilakukan *pruning* atau pengendalian kedalaman pohon.

Impurity node dihitung dengan Gini Index :

$$Gini(D) = 1 - \sum_{i=1}^k p_i^2 \quad (2)$$

atau dengan Entropy :

$$Entropy(D) = - \sum_{i=1}^k p_i \log_2 p_i \quad (3)$$

Model *Decision Tree* menggunakan parameter dasar seperti `criterion` yang diatur pada nilai 'gini' untuk mengukur tingkat impuritas, serta `max_depth` untuk membatasi kedalaman pohon agar tidak *overfitting*. Nilai `min_samples_split` juga ditentukan untuk mengatur jumlah minimal sampel yang dibutuhkan agar suatu node dapat dibagi. Selama proses *tuning*, dilakukan eksplorasi nilai kedalaman pohon antara 3 hingga 10 untuk memperoleh hasil terbaik

2.7 Random Forest

Random Forest merupakan algoritma ensemble learning yang membentuk banyak pohon keputusan (*Decision Trees*) dari subset data yang diacak (*bootstrapping*), lalu menggabungkan hasil prediksi dari masing-masing pohon melalui voting. Pendekatan ini membantu mengatasi *overfitting* dan meningkatkan akurasi [18]. Keunggulan algoritma ini terletak pada kemampuannya menangani data dengan jumlah fitur besar dan mengurangi risiko *overfitting*. Sementara itu, kelemahannya terletak pada interpretasi model yang kompleks dan waktu komputasi yang lebih lama dibanding satu pohon. Prediksi akhir diambil dengan voting mayoritas :

$$\hat{y} = mode(h_1(x), h_2(x), \dots, h_n(x)) \quad (4)$$



Model Random Forest dikonfigurasi dengan jumlah estimator ($n_{estimators}$) sebanyak 100 pohon keputusan, yang masing-masing dilatih dengan teknik bootstrapping. Parameter $max_features$ diatur ke 'auto' untuk secara otomatis memilih subset fitur saat membentuk split di setiap pohon. Model ini memanfaatkan prinsip voting dari banyak pohon untuk menghasilkan prediksi yang lebih stabil dan tahan terhadap *overfitting*.

2.8 Support Vector Machine (SVM)

Algoritma SVM merupakan salah satu jenis klasifikasi pada penambahan data yang memanfaatkan *hyperplane* sebagai garis pemisah dari beberapa kelas data. Algoritma ini akan mencari *hyperplane* (garis pemisah) optimal terhadap margin (jarak) [6], [19]. Keunggulan dari teknik ini adalah efektif serta mampu menangani data berdimensi tinggi dalam menentukan margin yang tepat pada pengelompokan data. Kekurangan dari algoritma ini yaitu waktu proses yang diperlukan lebih lama dan performa kurang optimal bahkan menurun jika terdapat banyak *noise* pada data.

SVM berusaha mencari hyperplane terbaik :

$$f(x) = w^T x + b \tag{5}$$

dengan syarat margin maksimum dan :

$$\min \frac{1}{2} \|w\|^2 \text{ dengan } y_i(w^T x_i + b) \geq 1 \tag{6}$$

Untuk kernel RBF :

$$K(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right) \tag{7}$$

Pada algoritma SVM, digunakan kernel rbf (*Radial Basis Function*) yang cocok untuk data non-linear. Parameter C disetel pada 1.0 sebagai penalti kesalahan klasifikasi dan gamma diatur ke 'scale' agar nilai default disesuaikan dengan jumlah fitur. Penggunaan kernel ini memungkinkan pemisahan kelas dengan margin maksimum di ruang berdimensi tinggi.

2.9 Logistic Regression

Logistic Regression adalah metode klasifikasi yang memodelkan hubungan antara input dan output dalam bentuk probabilitas dengan menggunakan fungsi *sigmoid*. Algoritma ini sering digunakan untuk klasifikasi biner dan menghasilkan keluaran dalam bentuk nilai probabilitas yang kemudian dikonversi menjadi label kelas [6]. Keunggulan algoritma ini adalah interpretasi model yang sederhana dan efisien untuk data linier. Namun, kekurangannya adalah kurang efektif untuk data yang memiliki hubungan non-linear kecuali dilakukan transformasi fitur sebelumnya. Logistic Regression memodelkan probabilitas kelas :

$$P(y = 1|x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \dots + \beta_n x_n)}} \tag{8}$$

Logistic Regression dikonfigurasi dengan solver='liblinear' karena cocok untuk dataset kecil hingga menengah dan mendukung regularisasi L1 maupun L2. Regularisasi L2 dipilih untuk menghindari *overfitting*, dengan parameter C=1.0 sebagai pengontrol kekuatan regularisasi. Iterasi maksimum disetel ke 1000 untuk memastikan model dapat mencapai konvergensi dalam proses pelatihan.

2.10 Evaluasi Model

Pendekatan ini menggunakan sebuah matriks seperti yang ditunjukkan pada Tabel 2 dengan menampilkan evaluasi performa model klasifikasi dapat dilakukan secara lebih *komprehensif*. Visualisasi ini memberikan gambaran yang lebih jelas mengenai jumlah prediksi yang benar dan salah dalam masing-masing kelas. di mana jika dataset terdiri dari dua kelas, salah satu kelas dianggap sebagai kelas positif, sedangkan kelas lainnya diasumsikan sebagai kelas negatif [20].

Tabel 2. Model Confusion Matrix

Correct Classification	Classified	
	Positive	Negative
Positive	True Positive (TP)	False Negative (FN)
Negative	False Positive (FP)	True Negative (TN)

Berdasarkan informasi yang ditampilkan pada Tabel 2, evaluasi model dapat dilakukan menggunakan berbagai metrik seperti *Accuracy*, *Precision*, *Recall*, dan *F1-Score*. Berikut penjelasan dari masing-masing metrik :

a. *Accuracy*

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \tag{9}$$

Akurasi mengukur seberapa sering model memprediksi dengan benar, baik untuk kelas positif maupun negatif. Nilai ini dihitung dari jumlah prediksi benar dibagi total seluruh prediksi. Semakin tinggi nilai akurasi, semakin baik performa umum model dalam mengklasifikasikan ulasan skincare secara keseluruhan.

b. *Precision*

$$Precision = \frac{TP}{TP+FP} \quad (10)$$

Precision menunjukkan proporsi prediksi kelas positif yang benar-benar positif. Dalam konteks ulasan skincare, *precision* yang tinggi mengindikasikan bahwa model mampu menghindari kesalahan dalam mengklasifikasikan ulasan yang sebenarnya negatif sebagai positif.

c. *Recall*

$$Recall = \frac{TP}{TP+FN} \quad (11)$$

Recall menunjukkan seberapa baik model dalam menemukan seluruh data yang termasuk dalam kelas positif. Jika model memiliki nilai *recall* tinggi, berarti sebagian besar ulasan positif berhasil diidentifikasi dengan benar.

d. *F1-Score*

$$F1 - Score = \frac{2 \cdot Precision \cdot Recall}{Precision+Recall} \quad (12)$$

F1-Score merupakan metrik harmonik antara *precision* dan *recall*, sehingga cocok digunakan saat dibutuhkan keseimbangan antara kedua metrik tersebut. Dalam konteks ini, *F1-Score* merepresentasikan akurasi model dalam menangani ulasan skincare yang bersifat positif secara adil dan seimbang.

2.11 Hyperparameter Tuning

Setelah model terbaik dipilih berdasarkan hasil evaluasi awal (seperti akurasi, *precision*, *recall*, dan *F1-score*), proses dilanjutkan dengan hyperparameter tuning untuk memaksimalkan performa model tersebut [21]. *Hyperparameter tuning* dilakukan guna menemukan kombinasi parameter optimal yang tidak dipelajari langsung dari data, namun sangat mempengaruhi hasil akhir prediksi [22]. Proses ini biasanya menggunakan pendekatan seperti *Grid Search* atau *Random Search*, yang menguji berbagai nilai parameter untuk menentukan konfigurasi terbaik berdasarkan skor evaluasi tertinggi. Dengan tahapan ini, model yang telah dipilih akan bekerja lebih efektif dan akurat pada data ulasan skincare.

Dalam penelitian ini, proses tuning dilakukan menggunakan pendekatan *Grid Search*, yaitu metode pencarian sistematis dengan mengevaluasi kombinasi dari sejumlah nilai parameter tertentu. Misalnya, pada model *Decision Tree*, dilakukan eksplorasi terhadap parameter *max_depth* (kedalaman maksimum pohon), *min_samples_split* (jumlah minimum data pada node internal), dan *criterion* (fungsi pengukuran seperti 'gini' dan 'entropy'). Kombinasi-kombinasi tersebut dievaluasi berdasarkan skor rata-rata hasil validasi silang (*cross-validation*) dengan tujuan untuk memilih parameter yang menghasilkan performa tertinggi.

Pendekatan ini memungkinkan peneliti mengeksplorasi seluruh ruang parameter yang relevan secara menyeluruh, meskipun membutuhkan waktu komputasi yang lebih besar. Hasil tuning kemudian digunakan untuk melatih ulang model terbaik dengan parameter terpilih, sehingga model dapat beradaptasi lebih baik terhadap pola dalam data, sekaligus meminimalkan kesalahan prediksi pada data uji. Dengan tahapan ini, model yang telah dikembangkan akan bekerja lebih akurat dan efisien dalam memprediksi tingkat ketergantungan terhadap skincare berdasarkan data pengguna.

2.12 Implementasi Streamlit

Aplikasi dibangun dengan memanfaatkan *Streamlit*, sebuah *framework* yang mendukung pengembangan tampilan antarmuka yang interaktif dan mudah digunakan oleh pengguna [10], [14]. Model klasifikasi yang telah dilatih sebelumnya akan disematkan ke dalam aplikasi untuk mengidentifikasi sentimen dari ulasan skincare yang dimasukkan oleh pengguna. Pengguna cukup mengetikkan ulasan pada kolom input yang tersedia, kemudian sistem akan menampilkan hasil klasifikasi apakah ulasan tersebut termasuk dalam kategori positif atau negatif. Untuk mendukung fungsionalitas ini, aplikasi dibangun menggunakan *framework* *Streamlit* yang memungkinkan integrasi model machine learning ke dalam antarmuka berbasis web secara ringan dan interaktif.

3. HASIL DAN PEMBAHASAN

3.1 Exploratory Data Analyst

Tahap ini bertujuan untuk memahami struktur dan karakteristik data, seperti distribusi, nilai kosong, dan proporsi kelas, sebagai dasar langkah preprocessing selanjutnya.

a. Menampilkan Informasi Umum Dataset

Langkah awal dalam proses eksplorasi data adalah memahami struktur dan karakteristik dasar dari dataset yang digunakan. Hal ini mencakup identifikasi jumlah fitur, jumlah sampel, tipe data setiap atribut, serta variabel target yang menjadi fokus analisis bisa dilihat pada Gambar 2.

```

Data columns (total 11 columns):
#   Column              Non-Null Count  Dtype
---  -
0   respondent_id       100 non-null    int64
1   gender              100 non-null    object
2   age                 100 non-null    int64
3   skin_type           100 non-null    object
4   stress_level        100 non-null    int64
5   sleep_hours         100 non-null    float64
6   routine_frequency   100 non-null    object
7   monthly_expense     100 non-null    float64
8   social_media_hours  100 non-null    float64
9   skincare_importance 100 non-null    int64
10  dependency_label    100 non-null    object
dtypes: float64(3), int64(4), object(4)
    
```

Gambar 2. Informasi Dataset

Pada Gambar 2 Kolom *respondent_id* berfungsi sebagai identifikasi unik, sedangkan *gender*, *skin_type*, *routine_frequency*, dan *dependency_label* merupakan data kategorikal (tipe *object*). Sementara itu, *age*, *stress_level*, dan *skincare_importance* bertipe numerik integer, sedangkan *sleep_hours*, *monthly_expense*, dan *social_media_hours* bertipe *float*. Tidak terdapat nilai yang hilang (*missing values*) pada dataset ini, sehingga data sudah bersih dan siap untuk proses eksplorasi lebih lanjut maupun modeling. Kolom target yang akan diprediksi adalah *dependency_label*, yang kemungkinan mengklasifikasikan tingkat ketergantungan responden terhadap produk perawatan kulit. Data ini berpotensi digunakan dalam analisis klasifikasi atau segmentasi konsumen berdasarkan kebiasaan dan kondisi kulit mereka.

b. Statistik Deskriptif

Statistik deskriptif digunakan untuk memberikan gambaran umum mengenai sebaran nilai pada setiap fitur dalam dataset. Analisis ini mencakup nilai minimum, maksimum, rata-rata (mean), median, standar deviasi, serta kuartil yang dijelaskan pada Gambar 3.

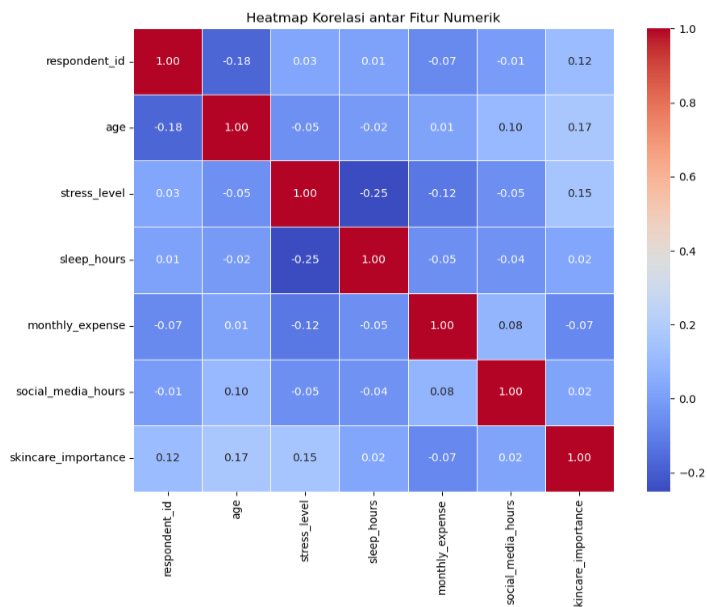
	respondent_id	age	stress_level	sleep_hours	monthly_expense	social_media_hours	skincare_importance
count	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000
mean	50.500000	21.410000	3.000000	6.485000	296.651700	3.044000	3.170000
std	29.011492	2.314186	1.428427	1.421152	165.232388	1.757369	1.400253
min	1.000000	18.000000	1.000000	4.100000	1.560000	0.000000	1.000000
25%	25.750000	19.750000	2.000000	5.300000	175.295000	1.400000	2.000000
50%	50.500000	21.000000	3.000000	6.550000	308.065000	3.200000	3.000000
75%	75.250000	23.000000	4.000000	7.425000	435.712500	4.425000	4.000000
max	100.000000	25.000000	5.000000	8.900000	591.170000	6.000000	5.000000

Gambar 3. Statistik Deskriptif

Gambar 3 menampilkan statistik deskriptif dari 100 responden yang meliputi variabel usia, tingkat stres, durasi tidur, pengeluaran bulanan, durasi penggunaan media sosial, dan tingkat pentingnya perawatan kulit. Rata-rata usia responden adalah 21,41 tahun, dengan tingkat stres rata-rata sebesar 3 (dalam skala 1–5) dan rata-rata waktu tidur 6,48 jam per hari. Pengeluaran bulanan menunjukkan nilai rata-rata sebesar 296,65 ribu, namun dengan standar deviasi sebesar 165,23 ribu yang mengindikasikan adanya variasi signifikan antar individu. Durasi penggunaan media sosial rata-rata berada pada angka 3,04 jam per hari, sementara persepsi terhadap pentingnya perawatan kulit memiliki rata-rata sebesar 3,17 dari skala maksimum 5. Nilai minimum dan maksimum dari setiap variabel menunjukkan rentang distribusi yang cukup lebar, khususnya pada variabel pengeluaran bulanan dan durasi media sosial, yang dapat menjadi indikator adanya outlier atau variasi gaya hidup antar responden. Informasi ini memberikan gambaran awal mengenai karakteristik sampel yang dapat digunakan dalam analisis lebih lanjut untuk memahami hubungan antar variabel.

c. Korelasi Antar Fitur

Pada Gambar 4, ditampilkan heatmap korelasi antar fitur numerik dalam dataset. Visualisasi ini berfungsi sebagai alat eksplorasi awal yang penting dalam proses analisis data, karena dapat mengungkap sejauh mana variabel-variabel saling berkaitan satu sama lain, termasuk hubungannya dengan variabel target.

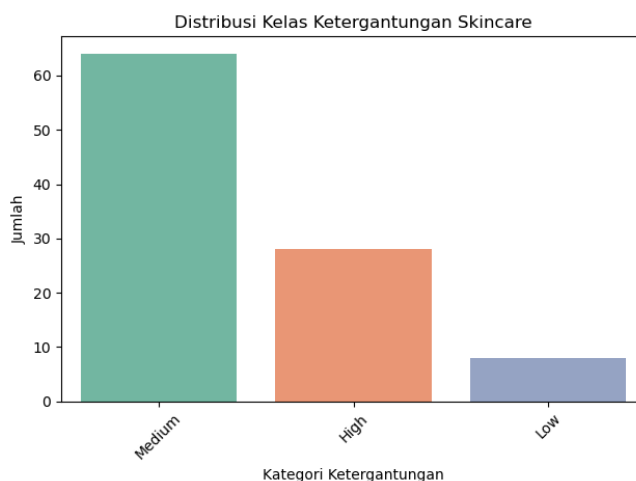


Gambar 4. Heatmap Korelasi Antar Fitur Numerik

Gambar 4 menunjukkan bahwa korelasi antar fitur numerik umumnya rendah, menandakan hubungan linear yang lemah. Korelasi negatif paling menonjol terdapat antara *stress_level* dan *sleep_hours* (-0.25), yang mengindikasikan bahwa stres cenderung mengurangi durasi tidur. Sementara itu, *skincare_importance* memiliki korelasi positif lemah dengan *age* (0.17) dan *stress_level* (0.15). Tidak ditemukan indikasi multikolinieritas tinggi, sehingga seluruh fitur numerik layak digunakan dalam pemodelan prediktif.

d. Visualisasi Distribusi Data

Visualisasi distribusi kelas ketergantungan skincare (*dependency_label*) terlihat pada Gambar 5 menunjukkan bahwa mayoritas responden berada pada kategori *Medium*, diikuti oleh *High*, sementara kategori *Low* memiliki jumlah paling sedikit. Ketidakseimbangan ini perlu diperhatikan karena dapat memengaruhi akurasi model klasifikasi, sehingga pada tahap selanjutnya akan dilakukan penanganan menggunakan teknik SMOTE.



Gambar 5. Distribusi Kelas Sebelum SMOTE

Gambar 5 memperlihatkan kategori ketergantungan skincare '*Medium*' merupakan yang paling dominan, dengan jumlah responden tertinggi dibandingkan kategori lainnya. Kategori '*High*' berada di posisi kedua, sedangkan kategori '*Low*' memiliki jumlah yang paling sedikit. Ketimpangan distribusi ini menunjukkan bahwa data bersifat tidak seimbang (*imbalanced*), yang dapat memengaruhi kinerja model klasifikasi dan perlu penanganan khusus pada tahap prapemrosesan, seperti penggunaan teknik SMOTE untuk menyeimbangkan kelas. Hasil EDA menunjukkan adanya ketidakseimbangan kelas yang signifikan, di mana kelas *Medium* mendominasi, sehingga berpotensi menyebabkan bias pada model saat proses pelatihan. Temuan ini menjadi dasar untuk menerapkan teknik oversampling seperti SMOTE guna memperbaiki distribusi kelas sebelum membangun model klasifikasi.

3.2 Pre-Processing

Tahap ini dilakukan untuk menyiapkan data sebelum pemodelan, meliputi encoding fitur kategorikal, dan normalisasi agar data siap digunakan secara optimal.

a. Encoding Variabel Kategorik

Selanjutnya adalah menerapkan proses *Encoding* untuk mengubah data kategorikal menjadi format numerik agar dapat diproses oleh algoritma machine learning bisa dilihat pada Gambar 6.

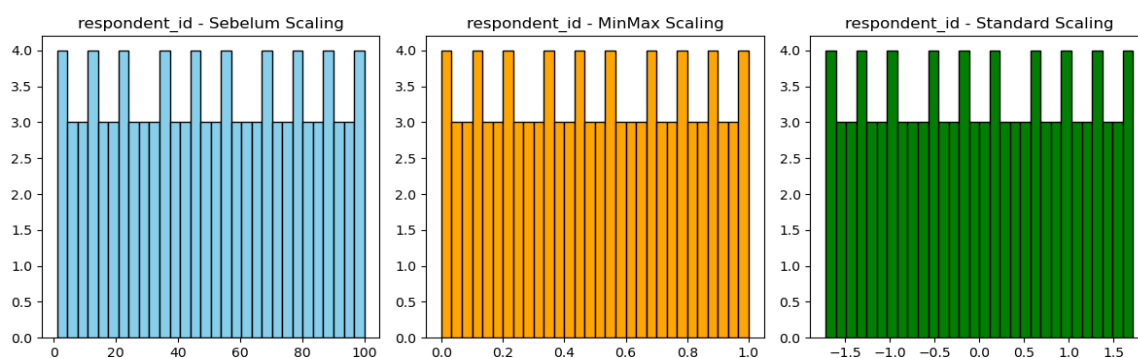
[INFO] One-Hot Encoding selesai. Dimensi fitur sekarang: (100, 17)

Gambar 6. One-Hot Encoding

Gambar 6 menggambarkan hasil dari proses *One-Hot Encoding* yang diterapkan terhadap fitur kategorikal seperti *gender*, *skin_type*, dan *routine_frequency*. Setiap kategori unik dari fitur-fitur tersebut dikonversi menjadi kolom biner, sehingga model dapat memahami informasi kategorikal dalam bentuk numerik. Hasil encoding ini meningkatkan jumlah fitur dalam dataset dan merupakan tahap penting sebelum pemodelan dilakukan.

b. Normalisasi Data

Pada Gambar 7, ditampilkan perbandingan distribusi nilai salah satu fitur sebelum dan sesudah dilakukan Min-Max Scaling dan Standard Scaling. Visualisasi ini menunjukkan bagaimana transformasi mempengaruhi sebaran data, namun tetap mempertahankan bentuk distribusinya.

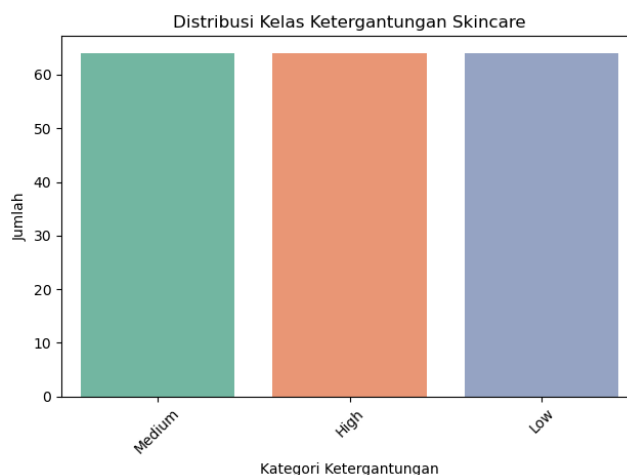


Gambar 7. Normalisasi Data

Gambar 7 menunjukkan sebelum dilakukan *scaling* menunjukkan pola yang merata, karena fitur ini merupakan ID unik yang berurutan dari 1 hingga 100 dan tidak mengandung informasi prediktif, sehingga umumnya diabaikan dalam pemodelan. Setelah diterapkan Min-Max Scaling, nilai *respondent_id* berada dalam rentang [0, 1] dengan distribusi tetap seragam, sesuai karakteristik metode tersebut. Sementara itu, penerapan *Standard Scaling* (*Z-score*) mengubah skala data menjadi terpusat di sekitar nol dengan penyebaran standar deviasi satu, menghasilkan nilai negatif dan positif tanpa mengubah bentuk distribusi yang tetap uniform.

c. Penerapan SMOTE

Gambar 8 di bawah ini merupakan hasil proses SMOTE (*Synthetic Minority Over-sampling Technique*).



Gambar 8. Distribusi Kelas Setelah SMOTE

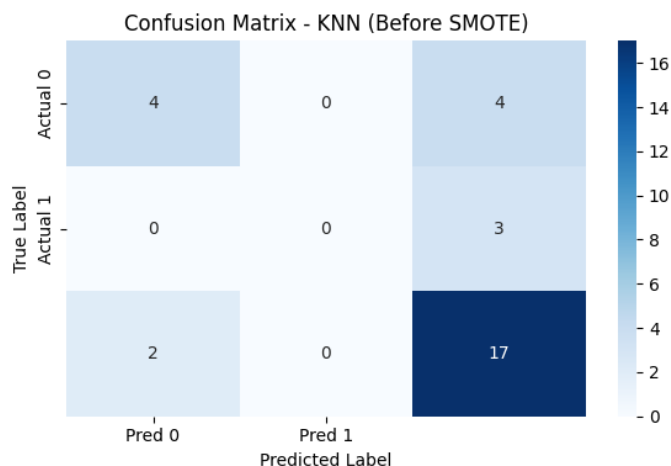
Gambar 8 menunjukkan hasil distribusi kelas *dependency_label* setelah dilakukan teknik oversampling menggunakan *SMOTE*. Terlihat bahwa jumlah data dari masing-masing kelas (*Low*, *Medium*, *High*) kini seimbang,

yang sebelumnya sangat timpang. Penyeimbangan ini penting agar model klasifikasi tidak bias terhadap kelas mayoritas dan dapat belajar secara adil dari ketiga kategori yang ada.

3.3 K-Nearest Neighbor

a. Evaluasi KNN Sebelum SMOTE

Pada Gambar 9, menunjukkan *confusion matrix* dari model *K-Nearest Neighbors* (KNN) sebelum dilakukan penanganan data tidak seimbang menggunakan SMOTE (*Synthetic Minority Over-sampling Technique*).



Gambar 9. Confusion Matrix KNN Sebelum SMOTE

Pada Gambar 9 menunjukkan confusion matrix dari model *K-Nearest Neighbor* (KNN) sebelum diterapkannya teknik *Synthetic Minority Over-sampling Technique* (SMOTE). Berdasarkan matriks tersebut, diketahui bahwa model memiliki tingkat ketidakseimbangan klasifikasi yang cukup signifikan. Dari total prediksi, terdapat 4 data kelas aktual 0 yang diprediksi dengan benar (*True Negative*), dan 17 data kelas aktual 1 yang diprediksi benar (*True Positive*). Namun demikian, terdapat pula kesalahan klasifikasi sebanyak 4 data kelas 0 yang diprediksi sebagai kelas 1 (*False Positive*), serta 2 data kelas 1 yang salah diklasifikasikan sebagai kelas 0 (*False Negative*). Selain itu, terdapat 3 data pada kelas tertentu yang tidak terklasifikasi dengan baik, yang terlihat dari adanya baris tanpa prediksi pada label tertentu. Hal ini menunjukkan bahwa model KNN masih belum mampu menangkap pola pada data minoritas secara optimal. Kinerja yang kurang seimbang ini menjadi indikasi bahwa diperlukan pendekatan penanganan ketidakseimbangan data seperti SMOTE untuk meningkatkan sensitivitas model terhadap kelas minoritas dan memperbaiki distribusi prediksi agar lebih proporsional.

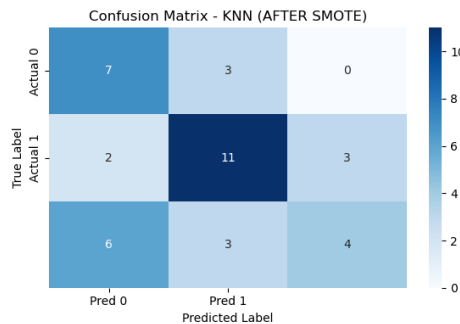
	precision	recall	f1-score	support
0	0.666667	0.500000	0.571429	8.0
1	0.000000	0.000000	0.000000	3.0
2	0.708333	0.894737	0.790698	19.0
accuracy	0.700000	0.700000	0.700000	0.7
macro avg	0.458333	0.464912	0.454042	30.0
weighted avg	0.626389	0.700000	0.653156	30.0

Gambar 10. Evaluasi KNN Sebelum SMOTE

Berdasarkan *confusion matrix* dan metrik evaluasi pada Gambar 9 dan Gambar 10 sebelum penerapan SMOTE, terlihat bahwa model KNN menunjukkan kecenderungan yang kuat dalam memprediksi kelas mayoritas (kelas 2) dengan akurasi keseluruhan sebesar 70%, namun gagal mengenali kelas minoritas, khususnya kelas 1 yang memiliki nilai *precision*, *recall*, dan *f1-score* sebesar 0.00. Confusion matrix mengonfirmasi bahwa seluruh data kelas 1 salah diklasifikasikan sebagai kelas 2, mencerminkan ketidakmampuan model dalam mengenali pola pada kelas dengan representasi rendah. Meskipun performa pada kelas 2 tergolong baik (*recall* 89%, *f1-score* 79%), buruknya performa pada kelas 0 dan 1 menyebabkan rata-rata makro *precision*, *recall*, dan *f1-score* menurun drastis hingga sekitar 45%. Ketimpangan ini menunjukkan bahwa tanpa penyeimbangan data, model menjadi bias terhadap kelas dominan dan tidak mampu melakukan klasifikasi yang adil terhadap seluruh kelas, sehingga diperlukan strategi seperti *oversampling* atau SMOTE untuk mengatasi masalah ketidakseimbangan dan meningkatkan kemampuan generalisasi model.

b. Evaluasi KNN Sesudah SMOTE

Pada Gambar 11, menunjukkan confusion matrix dari model *K-Nearest Neighbors* (KNN) sesudah dilakukan penanganan data tidak seimbang menggunakan SMOTE (*Synthetic Minority Over-sampling Technique*).



Gambar 11. Confusion Matrix KNN Sesudah SMOTE

Pada Gambar 11 Setelah proses *oversampling* kelas minoritas, model mulai menunjukkan peningkatan performa dalam mengklasifikasikan masing-masing kelas, khususnya kelas 1 yang sebelumnya sama sekali tidak terdeteksi. Dari total 16 data aktual kelas 1, sebanyak 11 diklasifikasikan dengan benar, sementara sisanya salah diklasifikasikan sebagai kelas 0 (2 data) dan kelas 2 (3 data). Untuk kelas 0, terdapat 7 prediksi yang tepat, meskipun masih terdapat 3 kasus salah klasifikasi ke kelas 1. Kelas 2 menunjukkan performa yang masih perlu ditingkatkan, dengan hanya 4 dari 13 data yang diklasifikasikan dengan benar, dan sisanya tersebar ke kelas 0 dan 1. Secara umum, penerapan SMOTE berhasil membantu model dalam mengenali pola dari kelas minoritas, meskipun masih terjadi kesalahan klasifikasi lintas kelas. Hal ini mencerminkan bahwa SMOTE dapat memperbaiki ketimpangan data, namun optimalisasi parameter dan pemilihan fitur masih diperlukan untuk meningkatkan akurasi klasifikasi secara keseluruhan.

	precision	recall	f1-score	support
High	0.466667	0.700000	0.560000	10.000000
Low	0.647059	0.687500	0.666667	16.000000
Medium	0.571429	0.307692	0.400000	13.000000
accuracy	0.564103	0.564103	0.564103	0.564103
macro avg	0.561718	0.565064	0.542222	39.000000
weighted avg	0.575594	0.564103	0.550427	39.000000

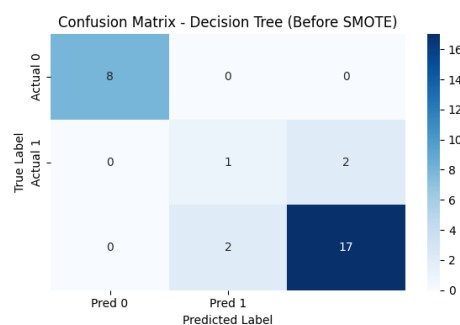
Gambar 12. Evaluasi KNN Sesudah SMOTE

Berdasarkan *confusion matrix* dan metrik evaluasi pada Gambar 11 dan Gambar 12 setelah penerapan SMOTE, model KNN menunjukkan peningkatan dalam hal kemampuan mengenali ketiga kelas. Meskipun akurasinya secara keseluruhan menurun menjadi sekitar 56,4%, namun nilai recall pada kelas minoritas seperti 'High' dan 'Low' meningkat hingga mendekati 70%, serta kelas 'Medium' yang sebelumnya diabaikan, kini mulai dikenali meskipun masih rendah (*recall* 30%). Hal ini konsisten dengan confusion matrix sesudah SMOTE, di mana prediksi sudah tersebar ke seluruh kelas, dan tidak lagi terpusat hanya pada satu kelas. Meskipun distribusi prediksi masih belum sempurna dan *f1-score* tertinggi hanya sekitar 66% pada kelas Low, namun metrik *macro average* dan *weighted average* menunjukkan peningkatan keseimbangan prediksi antar kelas. Ini mengindikasikan bahwa SMOTE berhasil mengurangi bias terhadap kelas mayoritas dan membantu model menjadi lebih adil terhadap kelas-kelas minoritas.

3.4 Decision Tree

a. Evaluasi Decision Tree Sebelum SMOTE

Pada Gambar 13, ditampilkan *confusion matrix* untuk model *Decision Tree*, sebelum dilakukan penanganan data tidak seimbang menggunakan SMOTE (*Synthetic Minority Over-sampling Technique*).



Gambar 13. Confusion Matrix Decision Tree Sebelum SMOTE

Model *Decision Tree* pada Gambar 13 menampilkan *confusion matrix* dari model *Decision Tree* sebelum diterapkannya teknik *Synthetic Minority Over-sampling Technique* (SMOTE). Berdasarkan matriks tersebut, model menunjukkan performa yang sangat baik dalam mengklasifikasikan kelas 0 dan kelas 2, dengan masing-masing 8 dan 17 data berhasil diprediksi dengan benar. Namun, kelemahan terlihat pada performa terhadap kelas 1, di mana hanya 1 dari 3 data yang diklasifikasikan secara akurat, sementara 2 sisanya salah diklasifikasikan sebagai kelas 2. Ketidakeimbangan prediksi ini mengindikasikan bahwa model cenderung bias terhadap kelas mayoritas, yaitu kelas 2, dan belum mampu mengenali pola dari kelas minoritas dengan baik. Hal ini menegaskan pentingnya penerapan metode penyeimbangan data seperti SMOTE untuk meningkatkan sensitivitas model terhadap kelas-kelas dengan jumlah data yang lebih sedikit, sehingga akurasi dan generalisasi model dapat ditingkatkan secara menyeluruh.

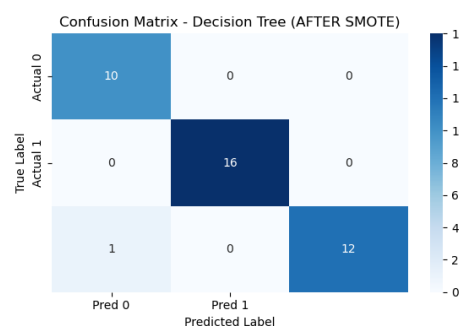
	precision	recall	f1-score	support
0	1.000000	1.000000	1.000000	8.0
1	0.000000	0.000000	0.000000	3.0
2	0.863636	1.000000	0.926829	19.0
accuracy	0.900000	0.900000	0.900000	0.9
macro avg	0.621212	0.666667	0.642276	30.0
weighted avg	0.813636	0.900000	0.853659	30.0

Gambar 14. Evaluasi Decision Tree Sebelum SMOTE

Berdasarkan *confusion matrix* dan *evaluation metrics* sebelum SMOTE pada Gambar 13 dan Gambar 14, terlihat bahwa model *Decision Tree* memiliki performa yang baik secara keseluruhan dengan akurasi sebesar 90%, namun performanya belum seimbang dalam mengklasifikasikan seluruh kelas. Kelas 0 dan kelas 2 berhasil diklasifikasikan dengan sangat baik, masing-masing dengan *f1-score* sebesar 1.000 dan 0.9268, sedangkan kelas 1 sama sekali tidak terdeteksi oleh model dengan nilai precision, recall, dan *f1-score* sebesar 0.000. Ketimpangan ini disebabkan oleh distribusi data yang tidak merata, di mana kelas 1 hanya memiliki support sebanyak 3 data, jauh lebih sedikit dibandingkan kelas 2 yang memiliki 19 data. Akibatnya, model cenderung bias terhadap kelas mayoritas dan mengabaikan kelas minoritas, sebagaimana tercermin dari nilai *macro average f1-score* yang hanya mencapai 0.6423. Kondisi ini menunjukkan perlunya penerapan teknik penyeimbangan data seperti SMOTE agar model dapat meningkatkan sensitivitas terhadap kelas minoritas dan menghasilkan performa klasifikasi yang lebih adil dan akurat.

b. Evaluasi Decision Tree Sesudah SMOTE

Pada Gambar 15, ditampilkan *confusion matrix* untuk model *Decision Tree*, sesudah dilakukan penanganan data tidak seimbang menggunakan SMOTE (*Synthetic Minority Over-sampling Technique*).



Gambar 15. Confusion Matrix Decision Tree Sesudah SMOTE

Setelah diterapkannya SMOTE, sebagaimana terlihat pada Gambar 15, menunjukkan *confusion matrix* dari hasil prediksi algoritma *Decision Tree* setelah dilakukan penyeimbangan data menggunakan teknik SMOTE. Visualisasi ini memperlihatkan performa klasifikasi yang sangat baik, dengan akurasi tinggi pada ketiga kelas. Kelas 0 berhasil diklasifikasikan seluruhnya dengan benar sebanyak 10 data, tanpa adanya kesalahan prediksi ke kelas lain. Kelas 1 menunjukkan performa sempurna dengan 16 data yang diklasifikasikan tepat, dan tidak terdapat satupun yang salah klasifikasi. Sementara itu, pada kelas 2, sebanyak 12 dari 13 data berhasil diklasifikasikan dengan benar, dengan hanya satu data yang salah diklasifikasikan sebagai kelas 0. Hasil ini menunjukkan bahwa penerapan SMOTE pada model *Decision Tree* berhasil mengatasi ketidakseimbangan kelas secara efektif dan memungkinkan model untuk mengenali pola pada kelas secara adil. Minimnya kesalahan klasifikasi mencerminkan bahwa model memiliki generalisasi yang baik terhadap data latih yang telah diseimbangkan, sehingga layak dipertimbangkan sebagai salah satu model terbaik dalam studi ini.

	precision	recall	f1-score	support
High	0.909091	1.000000	0.952381	10.000000
Low	1.000000	1.000000	1.000000	16.000000
Medium	1.000000	0.923077	0.960000	13.000000
accuracy	0.974359	0.974359	0.974359	0.974359
macro avg	0.969697	0.974359	0.970794	39.000000
weighted avg	0.976690	0.974359	0.974457	39.000000

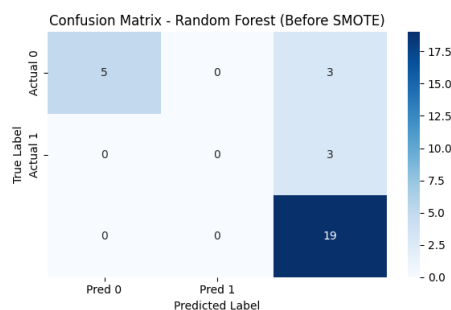
Gambar 16. Evaluasi Decision Tree Sesudah SMOTE

Berdasarkan *confusion matrix* dan evaluasi metrik setelah penerapan SMOTE pada Gambar 15 dan Gambar 16, terlihat bahwa kinerja model *Decision Tree* meningkat secara signifikan, terutama dalam mengklasifikasikan ketiga kelas secara lebih seimbang. *Confusion matrix* menunjukkan bahwa masing-masing kelas (*Low*, *Medium*, dan *High*) berhasil diprediksi dengan sangat baik dan tepat: kelas *Low* dan *High* memiliki prediksi sempurna, sedangkan kelas *Medium* hanya mengalami satu kesalahan klasifikasi. Hal ini tercermin dalam metrik evaluasi, di mana *precision*, *recall*, dan *f1-score* untuk setiap kelas berada di atas 0.90, dengan akurasi keseluruhan mencapai 97.44%, jauh lebih tinggi dibandingkan sebelum SMOTE (90.00%). Rata-rata makro dan tertimbang juga menunjukkan peningkatan yang signifikan, menandakan bahwa distribusi prediksi antar kelas menjadi lebih adil. Dengan demikian, SMOTE berhasil mengatasi ketidakseimbangan kelas dan meningkatkan performa model secara keseluruhan

3.5 Random Forest

a. Evaluasi Random Forest Sebelum SMOTE

Pada Gambar 17, ditampilkan *confusion matrix* untuk model *Random Forest*, sebelum dilakukan penanganan data tidak seimbang menggunakan SMOTE (*Synthetic Minority Over-sampling Technique*).



Gambar 17. Confusion Matrix Random Forest Sebelum SMOTE

Model Sebelum diterapkannya SMOTE, seperti terlihat *confusion matrix* pada Gambar 17, model *Random Forest* menunjukkan performa yang kurang optimal terhadap data yang tidak seimbang. Terlihat bahwa model cenderung sangat bias terhadap kelas mayoritas, yakni kelas 2, dengan keberhasilan klasifikasi sebanyak 19 data yang tepat sasaran. Sebaliknya, performa terhadap kelas minoritas sangat rendah, di mana seluruh data kelas 1 gagal dikenali sama sekali dan diklasifikasikan salah sebagai kelas 2. Hal serupa terjadi pada kelas 0, yang hanya sebagian (5 dari 8 data) berhasil diklasifikasikan dengan benar, sementara sisanya juga diklasifikasikan sebagai kelas 2. Tidak adanya prediksi terhadap kelas 1 serta dominasi prediksi pada kelas 2 mengindikasikan bahwa model tidak mampu membedakan pola dari kelas yang lebih jarang muncul. Hal ini memperkuat urgensi penerapan teknik penyeimbangan data, seperti SMOTE, untuk mengurangi bias klasifikasi dan meningkatkan keadilan distribusi prediksi antar kelas dalam konteks klasifikasi multikelas yang tidak seimbang.

	precision	recall	f1-score	support
0	1.000000	0.625000	0.769231	8.0
1	0.000000	0.000000	0.000000	3.0
2	0.760000	1.000000	0.863636	19.0
accuracy	0.800000	0.800000	0.800000	0.8
macro avg	0.586667	0.541667	0.544289	30.0
weighted avg	0.748000	0.800000	0.752098	30.0

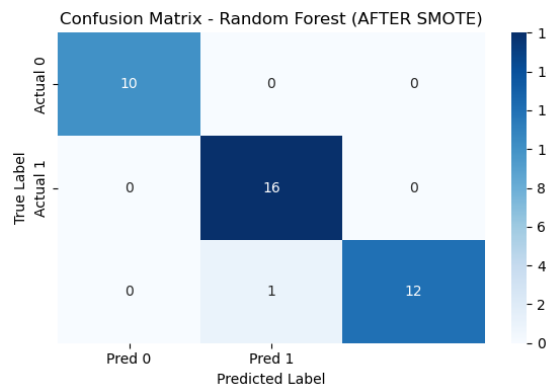
Gambar 18. Evaluasi Random Forest Sebelum SMOTE

Berdasarkan *confusion matrix* dan evaluasi metrik sebelum SMOTE pada Gambar 17 dan Gambar 18 untuk model *Random Forest*, sebelumnya terlihat bahwa algoritma ini cenderung bias terhadap kelas mayoritas, yaitu kelas 2. *Confusion matrix* menunjukkan bahwa seluruh data dari kelas 2 diprediksi dengan benar (19 dari 19), tetapi kelas

1 tidak berhasil diklasifikasikan sama sekali (seluruhnya diprediksi sebagai kelas 2), dan kelas 0 mengalami kesalahan sebanyak 3 dari 8 data. Hal ini tercermin dalam metrik evaluasi, di mana kelas 1 memiliki *precision*, *recall*, dan *f1-score* = 0.000, yang menandakan kegagalan total dalam mengklasifikasikan kelas tersebut. Meskipun akurasi keseluruhan terlihat cukup tinggi sebesar 80%, namun macro average (rata-rata performa per kelas) sangat rendah (*precision*: 0.5867, *recall*: 0.5417, *f1-score*: 0.5443), menandakan ketidakseimbangan performa antar kelas. Hasil ini menunjukkan bahwa model belum mampu menangani distribusi kelas yang tidak seimbang, sehingga perlu dilakukan penanganan seperti SMOTE untuk meningkatkan kinerja pada kelas minoritas.

b. Evaluasi Random Forest Sesudah SMOTE

Pada Gambar 19, ditampilkan *confusion matrix* untuk model *Random Forest*, sesudah dilakukan penanganan data tidak seimbang menggunakan SMOTE (*Synthetic Minority Over-sampling Technique*).



Gambar 19. Confusion Matrix Random Forest Sesudah SMOTE

Setelah diterapkan SMOTE, sesuai yang terlihat *confusion matrix* pada Gambar 19, menunjukkan peningkatan performa yang signifikan dibandingkan sebelum SMOTE, ditandai dengan keberhasilan model dalam mengenali seluruh kelas secara lebih proporsional. Kelas 0 dan kelas 1 diklasifikasikan secara sempurna tanpa kesalahan, masing-masing dengan 10 dan 16 prediksi yang tepat. Untuk kelas 2, sebanyak 12 dari 13 data berhasil diklasifikasikan dengan benar, sementara satu data mengalami salah klasifikasi ke kelas 1. Capaian ini mengindikasikan bahwa SMOTE berhasil mengurangi bias terhadap kelas mayoritas dan meningkatkan kemampuan generalisasi model terhadap data dari kelas minoritas. Dominasi klasifikasi yang sebelumnya condong hanya pada satu kelas kini telah tergantikan oleh distribusi prediksi yang lebih seimbang, menjadikan *Random Forest* sebagai model yang robust dalam menangani dataset dengan distribusi kelas yang tidak seimbang.

	precision	recall	f1-score	support
High	1.000000	1.000000	1.000000	10.000000
Low	0.941176	1.000000	0.969697	16.000000
Medium	1.000000	0.923077	0.960000	13.000000
accuracy	0.974359	0.974359	0.974359	0.974359
macro avg	0.980392	0.974359	0.976566	39.000000
weighted avg	0.975867	0.974359	0.974235	39.000000

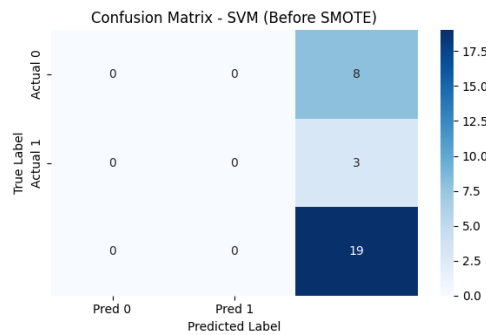
Gambar 20. Evaluasi Random Forest Sesudah SMOTE

Berdasarkan *confusion matrix* dan evaluasi metrik setelah SMOTE pada Gambar 19 dan Gambar 20 model *Random Forest*, terlihat bahwa performa klasifikasi meningkat secara signifikan dan menjadi jauh lebih seimbang antar kelas. *Confusion matrix* menunjukkan bahwa hampir seluruh data berhasil diprediksi dengan benar, dengan hanya satu kesalahan minor pada kelas *Medium*, sementara kelas *High* dan *Low* diprediksi sempurna. Hal ini didukung oleh metrik evaluasi, di mana *precision*, *recall*, dan *f1-score* untuk seluruh kelas berada di atas 0.92, bahkan mencapai 1.0 untuk kelas *High*. Akurasi keseluruhan juga sangat tinggi, yaitu 97.44%, dengan nilai *macro average f1-score* sebesar 0.9766 dan *weighted average* sebesar 0.9742. Hasil ini membuktikan bahwa penerapan SMOTE secara efektif memperbaiki ketidakseimbangan data dan meningkatkan kemampuan model dalam mengklasifikasikan semua kelas secara merata.

3.6 Support Vector Machine

a. Evaluasi SVM Sebelum SMOTE

Pada Gambar 21 ditampilkan *confusion matrix* untuk model *Support Vector Machine* (SVM), sebelum dilakukan penanganan data tidak seimbang menggunakan SMOTE (*Synthetic Minority Over-sampling Technique*).



Gambar 21. Confusion Matrix SVM Sebelum SMOTE

Model *Support Vector Machine* Sebelum penerapan SMOTE, sebagaimana terlihat *confusion matrix* pada Gambar 21, model *Support Vector Machine* (SVM) menunjukkan performa yang sangat tidak seimbang. Seluruh data dari ketiga kelas diprediksi sebagai kelas 2, sehingga menghasilkan akurasi sempurna hanya untuk kelas mayoritas (19 dari 19), namun gagal total dalam mengenali kelas 0 (0 dari 8) dan kelas 1 (0 dari 3). Hal ini menegaskan bahwa model sangat bias terhadap kelas mayoritas dan tidak mampu mengenali pola dari kelas minoritas akibat distribusi data yang tidak seimbang. Akurasi yang tinggi menjadi metrik yang menyesatkan karena tidak menggambarkan kemampuan model dalam melakukan klasifikasi secara menyeluruh dan adil. Temuan ini menegaskan pentingnya penanganan distribusi data, seperti penerapan SMOTE atau penyesuaian parameter SVM, agar model mampu menggeneralisasi dan memberikan hasil klasifikasi yang lebih representatif di semua kelas.

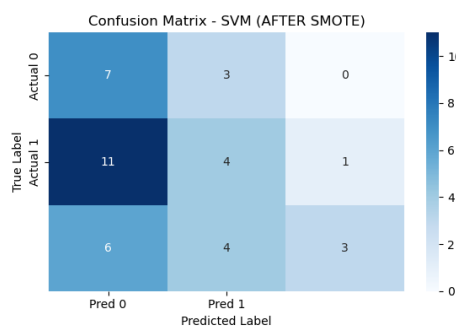
	precision	recall	f1-score	support
0	0.000000	0.000000	0.000000	8.000000
1	0.000000	0.000000	0.000000	3.000000
2	0.633333	1.000000	0.775510	19.000000
accuracy	0.633333	0.633333	0.633333	0.633333
macro avg	0.211111	0.333333	0.258503	30.000000
weighted avg	0.401111	0.633333	0.491156	30.000000

Gambar 22. Evaluasi SVM Sebelum SMOTE

Berdasarkan *confusion matrix* dan evaluasi metrik sebelum SMOTE pada Gambar 21 dan Gambar 22 untuk model SVM, terlihat bahwa model sangat bias terhadap kelas mayoritas (kelas 2) dan gagal total dalam mengenali kelas 0 dan 1. *Confusion matrix* menunjukkan bahwa seluruh data dari kelas 0 dan 1 diprediksi sebagai kelas 2, sementara hanya kelas 2 yang diprediksi dengan benar (19 dari 19 data). Hal ini berdampak buruk pada metrik evaluasi, di mana kelas 0 dan 1 memiliki *precision*, *recall*, dan *f1-score* = 0.000, menunjukkan tidak ada prediksi benar untuk kedua kelas tersebut. Meskipun akurasi model masih tampak sebesar 63.33%, nilai tersebut sangat menyesatkan karena hanya mencerminkan performa terhadap kelas dominan. Nilai *macro average f1-score* yang sangat rendah (0.2585) dan *precision weighted avg* hanya 0.4011 menegaskan bahwa model tidak mampu mengklasifikasikan data secara adil antar kelas. Hasil ini memperjelas pentingnya penanganan ketidakseimbangan data menggunakan teknik seperti SMOTE agar model dapat mengenali semua kelas dengan lebih baik.

b. Evaluasi SVM Sesudah SMOTE

Pada Gambar 23 ditampilkan *confusion matrix* untuk model *Support Vector Machine* (SVM), sesudah dilakukan penanganan data tidak seimbang menggunakan SMOTE (*Synthetic Minority Over-sampling Technique*).



Gambar 23. Confusion Matrix SVM Sesudah SMOTE

Setelah SMOTE diterapkan, seperti tampak *confusion matrix* pada Gambar 23, distribusi prediksi model *Support Vector Machine* (SVM) menjadi lebih merata terhadap ketiga kelas. Model mulai mampu mengenali sebagian data dari kelas 0 (7 dari 10 prediksi benar) dan kelas 2 (3 dari 13 prediksi benar). Namun, model masih mengalami

kesalahan yang cukup besar dalam mengklasifikasikan kelas 1 (hanya 4 dari 16 prediksi benar), dengan sebagian besar kasus justru diprediksi sebagai kelas 0. Hal ini menunjukkan bahwa meskipun SMOTE berhasil memperbaiki sebagian performa model, khususnya dalam mengurangi bias terhadap kelas mayoritas, performa keseluruhan model *Support Vector Machine* (SVM) masih belum optimal dan akurat dibanding model lain. Dibandingkan dengan *Random Forest* atau *Decision Tree*, kinerja SVM setelah SMOTE masih tertinggal dalam hal akurasi dan kestabilan prediksi.

	precision	recall	f1-score	support
High	0.291667	0.700000	0.411765	10.000000
Low	0.363636	0.250000	0.296296	16.000000
Medium	0.750000	0.230769	0.352941	13.000000
accuracy	0.358974	0.358974	0.358974	0.358974
macro avg	0.468434	0.393590	0.353667	39.000000
weighted avg	0.473970	0.358974	0.344785	39.000000

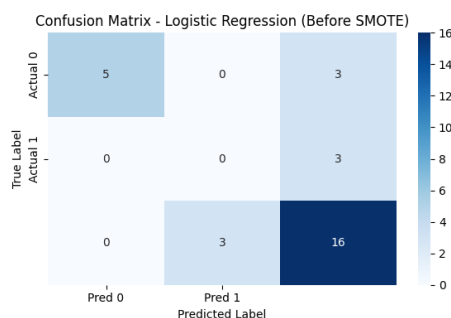
Gambar 24. Evaluasi SVM Sesudah SMOTE

Berdasarkan *confusion matrix* dan *evaluation metrics* hasil pada Gambar 23 dan Gambar 24 dari model SVM setelah dilakukan SMOTE, terlihat bahwa walaupun jumlah data antar kelas telah diseimbangkan, performa model secara keseluruhan masih rendah. *Confusion matrix* menunjukkan bahwa prediksi untuk masing-masing kelas masih banyak kesalahan, terutama pada kelas aktual 1 dan 2 (*Medium* dan *Low*), di mana sebagian besar data justru diklasifikasikan sebagai kelas 0 (*High*). Pada Evaluasi, di mana nilai akurasi hanya 35.89% dan nilai *recall* untuk kelas *Medium* dan *Low* sangat rendah (sekitar 23-25%), meskipun *precision* untuk kelas *Medium* tergolong tinggi (0.75). Artinya, model lebih sering salah dalam mengidentifikasi kelas, meskipun ketika memprediksi *Medium*, prediksinya cukup tepat. Kesimpulannya, penerapan SMOTE belum berhasil meningkatkan kinerja model secara signifikan, terutama dari segi generalisasi terhadap kelas minoritas.

3.7 Logistic Regression

a. Evaluasi Logistic Regression Sebelum SMOTE

Pada Gambar 25, ditampilkan *confusion matrix* untuk model *Logistic Regression* sebelum dilakukan penanganan data tidak seimbang menggunakan SMOTE (*Synthetic Minority Over-sampling Technique*).



Gambar 25. Confusion Matrix Logistic Regression Sebelum SMOTE

Confusion Matrix Logistic Regression pada Gambar 25 menunjukkan kecenderungan bias terhadap kelas mayoritas, yakni kelas 2, dengan 16 dari 19 data berhasil diklasifikasikan dengan benar. Namun, performa terhadap kelas minoritas masih jauh dari optimal. Model hanya mampu mengenali 5 dari 8 data kelas 0, sementara seluruh data kelas 1 (0 dari 3) salah diklasifikasikan sebagai kelas 2, tanpa satu pun prediksi yang tepat. Hasil ini mengindikasikan bahwa *Logistic Regression* kesulitan dalam mengenali pola dari kelas dengan representasi yang lebih kecil dalam data latih. Ketidakseimbangan ini menyebabkan model lebih ‘percaya diri’ terhadap kelas dominan, sehingga akurasi keseluruhan tampak cukup baik, namun menyesatkan karena tidak mencerminkan kualitas klasifikasi di semua kelas. Oleh karena itu, diperlukan teknik penyeimbangan data seperti SMOTE untuk membantu model mempelajari distribusi pola yang lebih merata, sehingga mampu menghasilkan prediksi yang lebih adil dan representatif terhadap seluruh kelas.

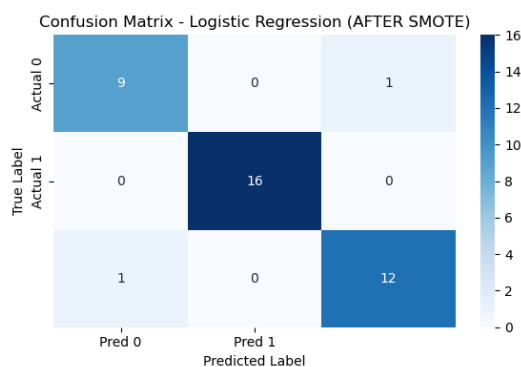
	precision	recall	f1-score	support
0	1.000000	0.625000	0.769231	8.0
1	0.000000	0.000000	0.000000	3.0
2	0.727273	0.842105	0.780488	19.0
accuracy	0.700000	0.700000	0.700000	0.7
macro avg	0.575758	0.489035	0.516573	30.0
weighted avg	0.727273	0.700000	0.699437	30.0

Gambar 26. Evaluasi Logistic Regression Sebelum SMOTE

Berdasarkan *confusion matrix* dan *evaluation metrics* pada Gambar 25 dan Gambar 26 dari model *Logistic Regression* sebelum diterapkannya SMOTE, terlihat bahwa model sangat bias terhadap kelas mayoritas (label 2) dan gagal memprediksi kelas minoritas (label 1). Hal ini tampak dari *confusion matrix* di mana tidak ada satupun instance dari kelas 1 yang berhasil diklasifikasikan dengan benar (*recall* = 0.0, *precision* = 0.0, *f1-score* = 0.0). Sebaliknya, kelas 2 memiliki performa yang jauh lebih baik (*f1-score* = 0.78 dan *recall* = 0.84) karena didominasi oleh jumlah data yang lebih banyak. Kelas 0 masih menunjukkan performa yang cukup baik dengan *f1-score* = 0.77. Secara keseluruhan, meskipun akurasi model mencapai 70%, nilai *macro average recall* hanya 0.49, yang menunjukkan ketidakseimbangan dalam prediksi antar kelas. Ini mengindikasikan bahwa sebelum SMOTE, model lebih cenderung mengabaikan kelas minoritas, sehingga kurang dapat diandalkan dalam kasus klasifikasi yang memerlukan sensitivitas tinggi terhadap semua kelas.

b. Evaluasi Logistic Regression Sesudah SMOTE

Pada Gambar 27, ditampilkan *confusion matrix* untuk model *Logistic Regression* sebelum dilakukan penanganan data tidak seimbang menggunakan SMOTE (*Synthetic Minority Over-sampling Technique*).



Gambar 27. Confusion Matrix Logistic Regression Sesudah SMOTE

Setelah diterapkannya SMOTE, sebagaimana terlihat pada Gambar 27, performa model *Logistic Regression* meningkat secara signifikan di semua kelas. Model berhasil mengklasifikasikan seluruh data kelas 1 dengan benar (16 dari 16), serta menunjukkan peningkatan akurasi pada kelas 0 (9 dari 10) dan kelas 2 (12 dari 13). Hanya terdapat sedikit kesalahan, yaitu satu data kelas 0 diprediksi sebagai kelas 2 dan satu data kelas 2 diprediksi sebagai kelas 0. Secara keseluruhan, penerapan SMOTE berhasil mengurangi bias terhadap kelas mayoritas dan membantu model mengenali pola dari semua kelas dengan lebih baik dan seimbang.

	precision	recall	f1-score	support
High	0.900000	0.900000	0.900000	10.000000
Low	1.000000	1.000000	1.000000	16.000000
Medium	0.923077	0.923077	0.923077	13.000000
accuracy	0.948718	0.948718	0.948718	0.948718
macro avg	0.941026	0.941026	0.941026	39.000000
weighted avg	0.948718	0.948718	0.948718	39.000000

Gambar 28. Evaluasi Logistic Regression Sesudah SMOTE

Berdasarkan *confusion matrix* dan *evaluation metrics* pada Gambar 27 dan Gambar 28 dari model *Logistic Regression* setelah diterapkannya SMOTE, terlihat bahwa performa model meningkat secara signifikan dan seimbang dalam mengklasifikasikan ketiga kelas (*High*, *Medium*, dan *Low*). *Confusion matrix* menunjukkan bahwa hampir seluruh prediksi berada pada diagonal utama, yang berarti prediksi sesuai dengan label sebenarnya, hanya terdapat dua kesalahan kecil. Hal ini selaras dengan nilai evaluasi yang sangat tinggi, yaitu *precision*, *recall*, dan *f1-score* untuk setiap kelas berada di atas 0.90, bahkan kelas *Low* mencapai nilai sempurna (1.0) pada ketiganya. Akurasi model pun meningkat drastis menjadi 94.87%, jauh lebih baik dibanding sebelum SMOTE. Rata-rata makro dan rata-rata berbobot dari semua metrik juga mendekati 0.95, menandakan bahwa model tidak hanya akurat secara keseluruhan, tetapi juga adil dalam performa antar kelas. Ini menunjukkan bahwa penerapan SMOTE berhasil mengatasi ketidakseimbangan data dan secara nyata meningkatkan kualitas prediksi model.

3.8 Evaluasi Perbandingan Model

a. Perbandingan Sebelum dan Sesudah SMOTE

Perbandingan kinerja lima model klasifikasi, yaitu *K-Nearest Neighbor* (KNN), *Decision Tree*, *Random Forest*, *Support Vector Machine* (SVM), dan *Logistic Regression* sebelum dan sesudah diterapkannya teknik *Synthetic Minority Over-sampling Technique* (SMOTE) dapat dilihat pada Tabel 3. Evaluasi dilakukan berdasarkan empat metrik utama: akurasi, *precision*, *recall*, dan *f1-score*. Notasi "→" digunakan untuk menunjukkan perubahan nilai dari sebelum ke sesudah penerapan SMOTE.

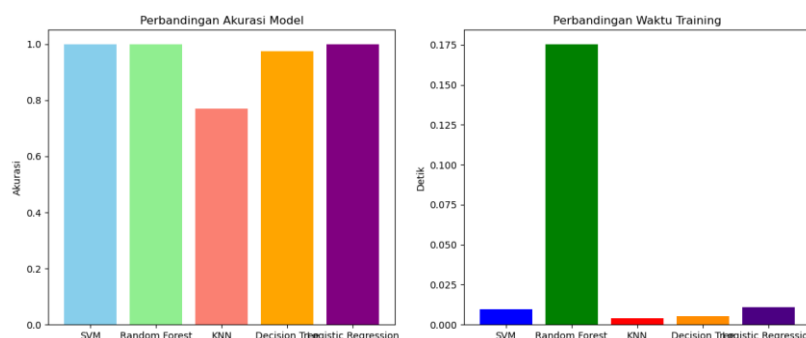
Tabel 3. Perbandingan Sebelum dan Sesudah SMOTE

Algoritma	Akurasi	Precision	Recall	F1-Score
K-Nearest Neighbor (KNN)	0.70 → 0.56	0.46 → 0.56	0.46 → 0.57	0.45 → 0.54
Decision Tree	0.90 → 0.97	0.62 → 0.97	0.67 → 0.97	0.64 → 0.97
Random Forest	0.80 → 1.00	0.59 → 1.00	0.54 → 1.00	0.54 → 1.00
Support Vector Machine (SVM)	0.63 → 0.36	0.21 → 0.47	0.33 → 0.39	0.26 → 0.35
Logistic Regression	0.70 → 0.95	0.58 → 0.94	0.49 → 0.94	0.52 → 0.94

Tabel 3 Terlihat bahwa SMOTE secara umum memberikan dampak positif terhadap peningkatan kinerja model, khususnya pada *Decision Tree*, *Random Forest*, dan *Logistic Regression* yang menunjukkan peningkatan signifikan di semua metrik evaluasi. *Random Forest* bahkan mencapai nilai sempurna setelah SMOTE diterapkan. Sebaliknya, model SVM mengalami penurunan performa, menandakan bahwa SMOTE tidak selalu memberikan perbaikan tergantung pada karakteristik model. Secara keseluruhan, penerapan SMOTE terbukti efektif dalam mengurangi bias terhadap kelas mayoritas dan meningkatkan keseimbangan prediksi antar kelas.

b. Evaluasi Model setelah SMOTE

Perbandingan akurasi model dan waktu pelatihan (*training time*) dari lima algoritma klasifikasi— *K-Nearest Neighbor (KNN)*, *Decision Tree*, *Random Forest*, *Support Vector Machine (SVM)* dan *Logistic Regression*—dilihat pada Gambar 29 yang menunjukkan performa masing-masing model dalam hal ketepatan prediksi serta efisiensi komputasi. Grafik tersebut menyajikan bagaimana setiap algoritma bekerja dalam mengklasifikasikan tingkat ketergantungan pengguna terhadap skincare berdasarkan dataset yang digunakan, baik dari sisi akurasi prediksi maupun waktu komputasi yang dibutuhkan selama proses pelatihan.



Gambar 29. Perbandingan Akurasi dan Waktu Training

Berdasarkan visualisasi pada Gambar 29, model *Random Forest*, SVM, dan *Logistic Regression* menunjukkan akurasi tertinggi, mendekati atau mencapai 100%, menandakan bahwa model tersebut sangat baik dalam mengklasifikasikan data ketergantungan skincare. Sementara itu, model KNN memiliki akurasi terendah di antara kelima model. Namun, dalam hal waktu pelatihan, model *Random Forest* membutuhkan waktu yang paling lama, jauh di atas model lainnya, meskipun menghasilkan akurasi tinggi. Sebaliknya, KNN dan *Decision Tree* memiliki waktu pelatihan yang sangat singkat. Hal ini menunjukkan adanya *trade-off* antara akurasi dan efisiensi waktu, yang perlu dipertimbangkan dalam pemilihan model untuk implementasi dunia nyata. Ringkasan Hasil Perbandingan dapat dilihat pada Tabel 4.

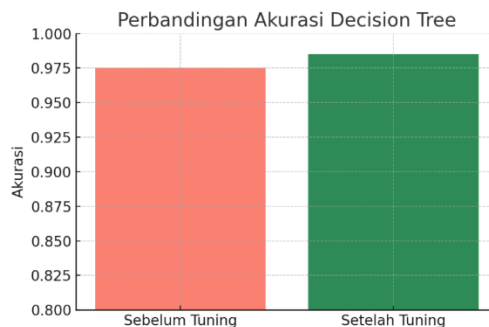
Tabel 4. Perbandingan Akurasi dan Waktu Training

NO	Algoritma	Akurasi	Waktu Pelatihan (detik)
1	K-Nearest Neighbor (KNN)	0.77	0.004
2	Decision Tree	0.97	0.006
3	Random Forest	1.00	0.175
4	Support Vector Machine (SVM)	1.00	0.009
5	Logistic Regression	1.00	0.010

Dari Tabel 4 dipilih karena mampu memberikan akurasi tinggi (97%), waktu pelatihan yang cepat, serta interpretasi hasil yang mudah dipahami, sehingga menjadi model yang paling seimbang antara kinerja dan efisiensi dalam analisis ketergantungan skincare.

3.9 Hyperparameter Tuning

Setelah memilih model terbaik, dilakukan proses *hyperparameter tuning* untuk mengoptimalkan kinerja model. Tahapan ini bertujuan menemukan kombinasi parameter yang paling tepat agar model dapat menghasilkan prediksi yang lebih akurat dan stabil, dapat dilihat pada Gambar 30.



Gambar 30. Perbandingan Sebelum dan Sesudah Tuning

Berdasarkan Gambar 30 grafik perbandingan akurasi, dapat disimpulkan bahwa proses *tuning* pada model *Decision Tree* memberikan peningkatan kinerja yang signifikan. Akurasi model sebelum *tuning* berada pada angka sekitar 97,5%, sedangkan setelah dilakukan *tuning* meningkat menjadi sekitar 98,5%. Hal ini menunjukkan bahwa optimasi parameter pada model mampu meningkatkan performa prediksi, sehingga model menjadi lebih akurat dalam mengklasifikasikan data. Peningkatan ini juga membuktikan pentingnya tahap *tuning* dalam proses pengembangan model machine learning untuk mencapai hasil yang lebih optimal. Lihat Tabel 5 untuk melihat angka perbandingan.

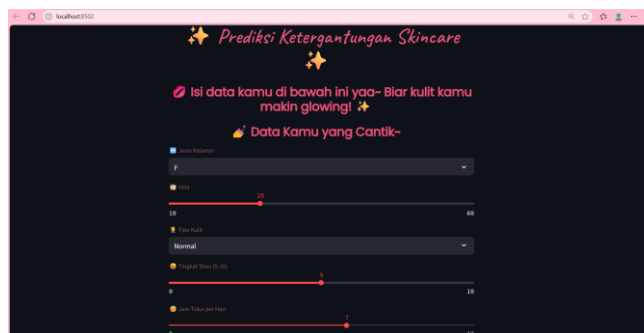
Tabel 5. Perbandingan Sebelum dan Sesudah Tuning

Model	Akurasi
Sebelum Tuning	0.975
Setelah Tuning	0.985

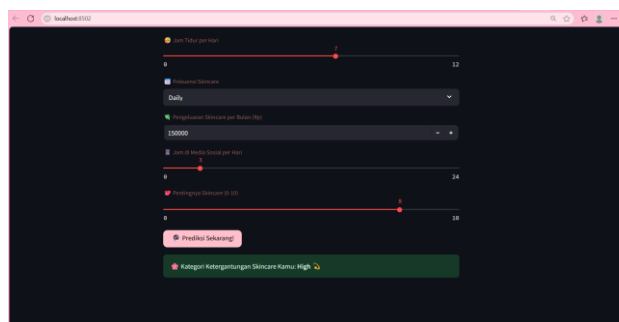
Setelah dilakukan *tuning*, akurasi model meningkat menjadi sekitar 98,5%. Kenaikan ini menunjukkan bahwa proses penyesuaian parameter mampu meningkatkan kemampuan model dalam memahami pola data yang kompleks. Dengan demikian, hasil ini memperkuat alasan bahwa *Decision Tree* yang telah dioptimalkan layak digunakan sebagai model utama dalam sistem prediksi ketergantungan skincare.

3.10 Implementasi Streamlit

Setelah memperoleh model terbaik, tahap implementasi dilakukan untuk memudahkan pengguna dalam memanfaatkan sistem prediksi ketergantungan skincare. Tampilan Hasil Interface Streamlit dapat dilihat pada Gambar 31 dan Gambar 32.



Gambar 31. Deployment Streamlit



Gambar 32. Deployment Streamlit

Aplikasi streamlit telah diuji coba pada Gambar 31 dan Gambar 32 dengan menggunakan data pengguna yang merepresentasikan kebiasaan perawatan kulit seperti usia, jenis kelamin, tipe kulit, tingkat stres, dan frekuensi

penggunaan skincare. Pengguna aplikasi diminta mengisi data secara interaktif melalui tampilan antarmuka yang menarik dan ramah pengguna. Berdasarkan input tersebut, aplikasi berhasil memberikan hasil prediksi kategori ketergantungan skincare dengan output akhir berupa klasifikasi, misalnya "High". Uji coba ini menunjukkan bahwa aplikasi mampu mengklasifikasikan tingkat ketergantungan skincare secara intuitif dan dapat digunakan sebagai alat bantu analisis kebiasaan kecantikan secara personal.

4. KESIMPULAN

Penelitian ini membangun model klasifikasi untuk memprediksi tingkat ketergantungan pengguna terhadap skincare berdasarkan fitur-fitur seperti usia, jenis kulit, tingkat stres, serta frekuensi penggunaan skincare. Proses dimulai dengan eksplorasi data, dilanjutkan dengan tahapan pra-pemrosesan mencakup *Encoding* Kategori Numerik, normalisasi, dan teknik *oversampling* menggunakan SMOTE untuk mengatasi ketidakseimbangan kelas. Lima algoritma pembelajaran mesin diterapkan, yaitu *Support Vector Machine (SVM)*, *Random Forest*, *K-Nearest Neighbor (KNN)*, *Decision Tree*, dan *Logistic Regression*. Hasil evaluasi menunjukkan bahwa *Decision Tree* menghasilkan akurasi tertinggi sebesar 97,5% sebelum tuning dan meningkat menjadi 98,5% setelah dilakukan *hyperparameter tuning*. Peningkatan ini menunjukkan bahwa penyesuaian parameter dapat secara signifikan meningkatkan kinerja model. Selain akurasi tinggi, *Decision Tree* juga unggul dalam hal interpretabilitas dan efisiensi pelatihan, menjadikannya pilihan yang sesuai untuk implementasi dalam sistem pendukung keputusan yang membutuhkan transparansi dan kecepatan analisis. Kelebihan ini memperkuat posisi *Decision Tree* sebagai model utama dalam penelitian. Meskipun demikian, masih terdapat keterbatasan pada ukuran dataset yang relatif kecil dan distribusi kelas yang tidak seimbang pada awal proses. Model yang telah dilatih kemudian diintegrasikan ke dalam aplikasi berbasis web menggunakan Streamlit, sehingga pengguna dapat secara langsung mengakses sistem melalui antarmuka interaktif. Dalam aplikasi ini, pengguna cukup mengisikan data pribadi terkait kebiasaan skincare seperti usia, jenis kelamin, tipe kulit, tingkat stres, dan frekuensi penggunaan pada formulir input yang tersedia. Pengembangan lebih lanjut dapat dilakukan dengan memperbesar cakupan data, menerapkan validasi silang, serta menguji algoritma lanjutan seperti *XGBoost* atau *LightGBM* guna memperoleh hasil yang lebih *robust* dan *generalisabel* pada data nyata.

REFERENCES

- [1] A. Arianto, J. Reveny, Nazliniwaty, and H. Bangun, *Formulasi dan Teknologi Kosmetik Perawatan Kulit*. Medan: USUpres (Penerbit Universitas Sumatera Utara), 2022. [Online]. Available: <https://usupres.usu.ac.id>
- [2] Badan Pengawas Obat dan Makanan Republik Indonesia, *Peraturan Badan Pengawas Obat dan Makanan Nomor 12 Tahun 2023 tentang Pengawasan Pembuatan dan Peredaran Kosmetik*. Jakarta, 2023. [Online]. Available: <https://peraturan.bpom.go.id>
- [3] P. Minerva, *Kosmetik Perawatan Kulit (Skin Care Cosmetics)*, First Edition. Padang: CV. Berkah Prima, 2022. [Online]. Available: <https://sinta.kemdikbud.go.id/authors/profile/6187973/?view=books>
- [4] Tim Redaksi UNAIR News, "Perilaku Konsumtif Perempuan untuk Produk Perawatan Kulit," Surabaya, Oct. 2024. [Online]. Available: <https://unair.ac.id/perilaku-konsumtif-perempuan-untuk-produk-perawatan-kulit/>
- [5] J. Lee, H. Yoon, S. Kim, C. Lee, J. Lee, and S. Yoo, "Deep learning-based skin care product recommendation: A focus on cosmetic ingredient analysis and facial skin conditions," *Journal of Cosmetic Dermatology (JoCD)*, vol. 23, no. 6, pp. 2066–2077, Jun. 2024, doi: 10.1111/jocd.16218.
- [6] A. Annur Rohman, G. Alfa Trisnapradika, and K. Kunci, "Perbandingan Algoritma NBC, SVM, Logistic Regression untuk Analisis Sentimen Terhadap Wacana KaburAjaDulu di Media Sosial X," *Building of Informatics, Technology and Science (BITS)*, vol. 7, no. 1, 2025, doi: 10.47065/bits.v7i1.7261.
- [7] I. I. Purnomo and A. Andie, "Perancangan Sistem Evaluasi Tingkat Kompetensi Alumni dengan Algoritma Support Vector Machine (SVM) Linear Method Berbasis Framework Streamlit," *Jurnal Informatika Universitas Pamulang*, vol. 8, no. 3, pp. 407–412, Sep. 2023, doi: 10.32493/informatika.v8i3.33766.
- [8] T. Ayhan and T. Uçar, "Determining customer limits by data mining methods in credit allocation process," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 2, pp. 1910–1915, Apr. 2022, doi: 10.11591/ijece.v12i2.pp1910-1915.
- [9] I. T. Ali, Y. Rahayu, and A. Setiawan, "Web Application Based on Machine Learning for Diabetes Detection using Microstrip Resonator and Streamlit," *Indonesian Journal of Electronics, Electromedical Engineering, and Medical Informatics (IJEEMI)*, vol. 6, no. 3, pp. 2656–8624, 2024, doi: 10.35882/ijeemi.v6i3.2.
- [10] R. S.P., S. M., and J. R., "Streamlit-based Web Application for Parkinson's Detection using Machine Learning," *Journal of Artificial Intelligence and Capsule Networks (JAICN)*, vol. 6, no. 4, pp. 466–478, Jan. 2025, doi: 10.36548/jaicn.2024.4.006.
- [11] I. T. Rahmawati and D. Alita, "Perbandingan Algoritma SVM, Random Forest, KNN untuk Analisis Sentimen Terhadap Overclaim Skincare pada Media Sosial X," *Building of Informatics, Technology and Science (BITS)*, vol. 6, no. 4, 2025, doi: 10.47065/bits.v6i4.6782.
- [12] P. Aulia Azhar, M. Arya Pratama, and R. Fitriani, "Prediksi Harga Mobil Audi Bekas Menggunakan Model Regresi Linear dengan Framework Streamlit," *Journal of Technology and Informatics (JoTI)*, vol. 6, no. 1, pp. 22–28, Oct. 2024, doi: 10.37802/joti.v6i1.763.
- [13] Y. Yadin and D. Ayu Megawaty, "Perbandingan Algoritma K-Nearest Neighbor dan Support Vector Machine Pada Pengenalan Pola Tanda Tangan Digital," *Building of Informatics, Technology and Science (BITS)*, vol. 6, no. 4, 2025, doi: 10.47065/bits.v6i4.6982.



- [14] G. Ayu Syafarina and I. Kalimantan Muhammad Arsyad Al Banjari Banjarmasin, “Implementasi Framework Streamlit Sebagai Prediksi Harga Jual Rumah Dengan Linear Regresi,” *METIK JURNAL*, vol. 7, no. 2, p. 2023, 2023, doi: 10.47002/metik.v7i2.680.
- [15] E. Arya Pranata, F. Budiman, and D. Kurniawan, “Analisis Sentimen Ulasan Mobile JKN pada Playstore dengan Perbandingan Akurasi Algoritma Naïve Bayes dan SVM,” *Building of Informatics, Technology and Science (BITS)*, vol. 7, no. 1, 2025, doi: 10.47065/bits.v7i1.7334.
- [16] H. Hilmi Zain, R. Maulana Awangga, and W. Isti Rahayu, “Perbandingan Model Svm, Knn Dan Naïve Bayes Untuk Analisis Sentiment Pada Data Twitter: Studi Kasus Calon Presiden 2024,” *JIMPS: Jurnal Ilmiah Mahasiswa Pendidikan Sejarah*, vol. 8, no. 3, pp. 2083–2093, 2024, doi: 10.24815/jimps.v8i3.25342.
- [17] D. Hartama *et al.*, “Perbandingan Algoritma Decision Tree, ID3, dan Random Forest dalam Klasifikasi Faktor-Faktor yang Mempengaruhi Karier Mahasiswa Ilmu Komputer,” *Jurnal Indonesia : Manajemen Informatika dan Komunikasi (JIMIK)*, vol. 6, no. 1, 2025, doi: <https://doi.org/10.35870/jimik.v6i1.1113>.
- [18] R. Sukma Andarujaya and R. R. Suryono, “Perbandingan Kinerja Algoritma Random Forest, KNN, dan SVM dalam Analisis Sentimen Cryptocurrency,” *Building of Informatics, Technology and Science (BITS)*, vol. 6, no. 4, 2025, doi: 10.47065/bits.v6i4.6572.
- [19] N. G. Ramadhan and A. Khoirunnisa, “Klasifikasi Data Malaria Menggunakan Metode Support Vector Machine,” *Jurnal Media Informatika Budidarma (MIB)*, vol. 5, no. 4, p. 1580, Oct. 2021, doi: 10.30865/mib.v5i4.3347.
- [20] L. Kencono and D. Darwis, “Perbandingan Algoritma NBC, SVM dan Random Forest untuk Analisis Sentimen Implementasi Starlink pada Media Sosial X,” *Building of Informatics, Technology and Science (BITS)*, vol. 6, no. 4, 2025, doi: 10.47065/bits.v6i4.6813.
- [21] H. D. Purnomo, T. Gonsalves, E. Mailoa, F. Y. Santoso, and M. R. Pribadi, “Metaheuristics Approach for Hyperparameter Tuning of Convolutional Neural Network,” *Jurnal Rekayasa Sistem dan Teknologi Informasi (RESTI)*, vol. 8, no. 3, pp. 340–345, Jun. 2024, doi: 10.29207/resti.v8i3.5730.
- [22] J. Rusman, B. Z. Haryati, and A. Michael, “Optimisasi Hiperparameter Tuning pada Metode Support Vector Machine untuk Klasifikasi Tingkat Kematangan Buah Kopi,” *Jurnal Komputer dan Informatika*, vol. 11, no. 2, pp. 195–202, Oct. 2023, doi: 10.35508/jicon.v11i2.12571.