

Klasifikasi Cuaca Jawa Barat menggunakan *Ensemble Learning* pada Data Meteorologi

Weather Classification in West Java using Ensemble Learning on Meteorological Data

¹Cynthia Nur Azzahra, ²Yulison Herry Chrisnanto*, ³Gunawan Abdillah
^{1,2,3}Jurusan Informatika, Fakultas Sains dan Informatika, Universitas Jenderal Achmad Yani
^{1,2,3}Jl. Terusan Jenderal Sudirman, Kota Cimahi, Jawa Barat 40525, Indonesia
*e-mail: yhc@if.unjani.ac.id

(received: 23 May 2025, revised: 23 June 2025, accepted: 24 June 2025)

Abstrak

Klasifikasi cuaca di wilayah Jawa Barat menghadapi tantangan, terutama terkait ketidakseimbangan kelas dalam data dan kompleksitas variabel. Penelitian ini bertujuan untuk meningkatkan akurasi klasifikasi cuaca dengan mengusulkan pendekatan *stacking classifier*, yang menggabungkan *Support Vector Machine* (SVM) dan *Random Forest* sebagai *base learners*, serta *Logistic Regression* sebagai *meta-classifier*. Untuk mengatasi ketidakseimbangan kelas, diterapkan *teknik Synthetic Minority Oversampling Technique* (SMOTE), sementara optimasi model dilakukan melalui *GridSearchCV*. Data cuaca diperoleh dari BMKG periode bulan Desember 2024 diproses melalui tahap transformasi, normalisasi, serta penanganan *outlier*. Dataset kemudian dibagi ke dalam rasio pelatihan dan pengujian 70:30, 80:20, 90:10. Model *stacking classifier* tanpa SMOTE mencatat akurasi tertinggi sebesar 86.73%, namun menunjukkan *overfitting* dengan gap akurasi *training-validation* sebesar 13.27%. Penerapan SMOTE meningkatkan *recall* kelas minoritas hingga 76.3% serta mengurangi *overfitting* dengan gap menjadi <1%. Kinerja model paling stabil dicapai pada rasio pembagian data 80:20, di mana model dengan SMOTE dan optimasi *hyperparameter* mencapai akurasi 85.97%, *f1-score* 68.99% dan hasil t-test signifikan ($p < 0.001$). Hasil ini menunjukkan bahwa *stacking classifier*, SMOTE, dan optimasi *hyperparameter* mampu mengatasi bias kelas dan meningkatkan generalisasi model secara efektif, serta unggul dibandingkan model klasifikasi tunggal dalam menangani data cuaca yang tidak seimbang.

Kata kunci: klasifikasi cuaca, ensemble learning, stacking classifier, SMOTE, optimasi hyperparameter

Abstract

Weather classification in West Java presents several challenges, particularly related to class imbalance in the dataset and the complexity of meteorological variables. This study aims to improve classification accuracy by proposing a stacking classifier approach that combines Support Vector Machine (SVM) and Random Forest as base learners, with Logistic Regression serving as the meta-classifier. To address the class imbalance, the Synthetic Minority Oversampling Technique (SMOTE) was applied, while model optimization was conducted using GridSearchCV. Weather data from the Indonesian Meteorological, Climatological, and Geophysical Agency (BMKG) for December 2024 was used and processed through transformation, normalization, and outlier handling. The dataset was then split into training and testing sets with ratios of 70:30, 80:20, and 90:10. The stacking classifier without SMOTE achieved the highest accuracy of 86.73%, but suffered from overfitting, indicated by a 13.27% gap between training and validation accuracy. The application of SMOTE improved the recall for minority classes to 76.3% and reduced overfitting, with the accuracy gap narrowing to less than 1%. The most stable performance was achieved with an 80:20 train-test split, where the SMOTE-applied and hyperparameter-optimized model reached an accuracy of 85.97%, an F1-score of 68.99%, and a statistically significant t-test result ($p < 0.001$). These findings demonstrate that the combination of stacking classifiers, SMOTE, and hyperparameter tuning

effectively mitigates class bias and enhances model generalization, outperforming single-model classifiers in handling imbalanced weather data.

Keywords: *weather classification, ensemble learning, stacking classifier, SMOTE, hyperparameter optimization.*

1 Pendahuluan

Perubahan iklim global telah meningkatkan frekuensi dan intensitas kejadian cuaca ekstrem di Indonesia, seperti banjir bandang, hujan es, dan angin puting beliung [1]. Data Badan Nasional Penanggulangan Bencana (BNPB) tahun 2023 menunjukkan Provinsi Jawa Barat sebagai salah satu wilayah paling terdampak dengan 844 kejadian bencana *hidrometeorologi*. Tingginya angka ini dipengaruhi oleh kondisi geografis yang didominasi pegunungan dan perbukitan, jaringan sungai yang padat, serta kepadatan penduduk yang tinggi [2]. Oleh karena itu, klasifikasi cuaca yang lebih akurat diperlukan untuk mendukung prediksi yang lebih efektif dan meminimalkan risiko bencana alam di wilayah tersebut.

Di tengah tantangan tersebut, pemilihan metode klasifikasi yang tepat menjadi faktor penting dalam mengolah data cuaca yang lebih akurat dan stabil. *Support Vector Machine* (SVM) terbukti efektif dalam kasus *multiclass classification*, dengan akurasi mencapai 96.64% untuk mengklasifikasikan kondisi cuaca *drizzle, rain, sun, snow, dan fog* [3]. Keunggulan SVM juga terlihat dalam klasifikasi cuaca di Takari, Nusa Tenggara Timur memiliki akurasi lebih baik dibandingkan *Naïve Bayes* pada kasus prediksi risiko tanah longsor [4]. Kedua penelitian ini memperkuat posisi SVM sebagai metode yang efektif dalam klasifikasi cuaca berbasis data meteorologi. Namun, penggunaan SVM dalam klasifikasi cuaca masih kurang optimal, khususnya dalam menangani data yang tidak seimbang serta kurangnya optimasi *hyperparameter* yang baik.

Beberapa solusi telah diusulkan untuk mengatasi keterbatasan tersebut. Penggunaan *GridSearchCV* untuk optimasi *hyperparameter* mampu meningkatkan akurasi model hingga 21% pada klasifikasi kualitas udara [5]. Sementara, metode *Synthetic Minority Over-Sampling* (SMOTE) efektif dalam menangani ketidakseimbangan data [6]. Selain SVM, metode *Random Forest* juga telah terbukti efektif dalam klasifikasi karena mampu mengurangi *overfitting*, menangani data dengan *outlier*, mengenali hubungan non-linear antar fitur, dan memberikan tingkat akurasi yang stabil [7], [8]. Penelitian oleh [6] menunjukkan bahwa *Random Forest* mampu menghasilkan hasil klasifikasi lebih baik daripada model tunggal lainnya. Namun, *Random Forest* juga memiliki keterbatasan, seperti kecenderungan bias pada kelas mayoritas jika data tidak seimbang.

Metode *ensemble learning* untuk prakiraan cuaca dengan menggabungkan metode *Naïve Bayes, Random Forest, dan MLP* dengan pendekatan *soft voting classifier* menunjukkan potensi peningkatan performa, dengan akurasi mencapai 99.03% menggunakan variabel meteorologi seperti tekanan udara, kelembaban, suhu, kecepatan angin, dan titik embun [9]. Selain itu, kombinasi SVM dan *Random Forest* dengan metode *ensemble learning* juga terbukti meningkatkan akurasi hingga 97.5%, lebih baik dibandingkan dengan akurasi SVM tunggal dengan akurasi 91.81% untuk prediksi penyakit jantung [10]. Namun, penerapan *ensemble learning* untuk klasifikasi cuaca di Jawa Barat masih terbatas.

Berdasarkan analisis tersebut, penelitian ini mengusulkan metode *ensemble learning* dengan teknik *stacking classifier* yang mengkombinasikan metode SVM dan *Random Forest* sebagai *base model*, serta *Logistic Regression* sebagai *meta-classifier*, diperkuat oleh SMOTE untuk penanganan data tidak seimbang dan *GridSearchCV* untuk optimasi *hyperparameter*. Implementasi dilakukan pada dataset cuaca Jawa Barat dengan fokus pada tujuh kategori cuaca (hujan ringan, hujan sedang, hujan lebat, hujan petir, berawan tebal, berawan, dan cerah berawan) menggunakan parameter meteorologi utama (suhu, kelembaban, arah dan kecepatan angin), dengan mengesampingkan curah hujan dan penyinaran matahari karena keterbatasan konsistensi data. Pendekatan ini tidak hanya menawarkan solusi klasifikasi yang lebih akurat untuk karakteristik iklim spesifik Jawa Barat, tetapi juga memberikan kerangka kerja yang dapat diadaptasi untuk wilayah tropis lainnya dengan tantangan serupa.

2 Tinjauan Literatur

Penelitian terkait penerapan metode *machine learning* dalam klasifikasi cuaca telah dilakukan secara luas, namun masih terdapat tantangan yang belum sepenuhnya teratasi. Salah satunya penelitian [3] menggunakan SVM dan berhasil mencapai tingkat akurasi 96.64% dalam membedakan berbagai kategori cuaca. Penelitian tersebut menunjukkan bahwa SVM dengan *kernel linear* tidak hanya unggul dalam akurasi, tetapi juga efisien secara statistik. Namun, penelitian tersebut terbatas pada penggunaan *kernel linear* dan *polynomial* yang cenderung sesuai untuk data linear. Tidak ada pembahasan mengenai penerapan SVM untuk data non-linear, mengingat pola cuaca sering kali kompleks. Oleh karena itu, penelitian ini membuka peluang untuk mengeksplorasi penggunaan *kernel Radial Basis Function* (RBF) untuk meningkatkan akurasi klasifikasi cuaca.

Penelitian [7] membandingkan algoritma Regresi Logistik dan *Random Forest* untuk klasifikasi cuaca di wilayah Jawa Tengah. Hasil penelitian menunjukkan bahwa *Random Forest* memperoleh nilai AUC tertinggi dengan rata-rata 0.7419, mengungguli Regresi Logistik dengan AUC 0.6923. Berdasarkan nilai AUC tersebut, *Random Forest* memberikan performa klasifikasi yang lebih baik. Namun, penelitian ini tidak menjelaskan apakah distribusi kelas yang digunakan seimbang atau tidak, sehingga belum diketahui sejauh mana ketidakseimbangan data mempengaruhi performa model. Penerapan teknik *balancing data*, seperti *oversampling* atau *undersampling*, berpotensi meningkatkan kinerja model lebih lanjut.

Perkembangan metode *ensemble learning* menawarkan pendekatan untuk meningkatkan akurasi dan stabilitas model klasifikasi. Hasil penelitian [11] membandingkan performa antara model klasifikasi tunggal dan model *ensemble*. Model tunggal seperti KNN, SVM, dan *Naïve Bayes* mencapai akurasi masing-masing 86%, 87%, dan 90.75%. Setelah diterapkan metode *ensemble* seperti *Bagging*, *Gradient Boosting*, dan *AdaBoost*, akurasi meningkat menjadi 94.43%, 94.96%, dan 95.54%. Meskipun menunjukkan peningkatan performa, penelitian tersebut belum melakukan optimasi *hyperparameter* pada setiap model. Sehingga masih terdapat ruang untuk peningkatan performa model melalui optimasi *hyperparameter*.

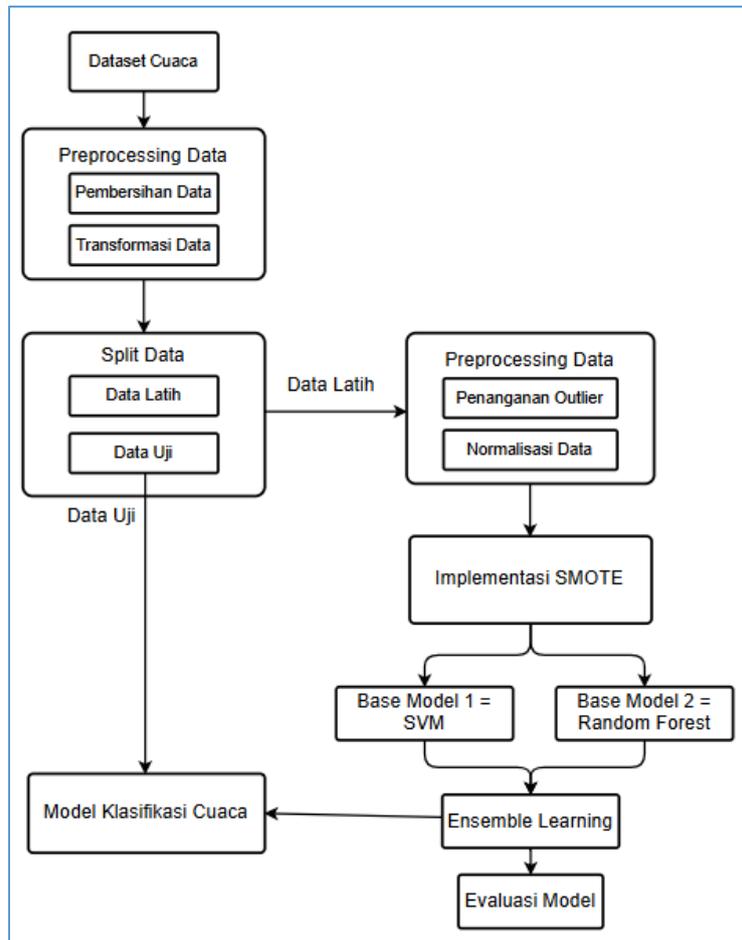
Penelitian lain oleh [9] mengembangkan model prakiraan cuaca berbasis *ensemble learning* dengan menggunakan dataset yang terdiri dari 33.746 baris, mencakup variabel seperti suhu, titik embun, kelembapan, kecepatan angin, tekanan, dan curah hujan. Beberapa metode *machine learning* tunggal diuji, termasuk *K-Nearest Neighbor* (KNN), *Logistic Regression*, *Random Forest*, *Naïve Bayes*, dan *Multi-Layer Perceptron* (MLP). *Naïve Bayes* menghasilkan akurasi sebesar 99,00%, dan kombinasi melalui pendekatan *soft voting* antara *Random Forest*, MLP, dan *Naïve Bayes* hanya sedikit meningkatkan akurasi menjadi 99.03%. Meskipun peningkatannya relatif kecil, hasil ini tetap menunjukkan bahwa *ensemble learning* berpotensi meningkatkan stabilitas prediksi kelas dengan menggabungkan keunggulan dari beberapa model.

Dalam bidang kesehatan, penelitian [10] menggunakan *ensemble learning* dengan menggabungkan SVM dan *Random Forest* untuk prediksi penyakit jantung. Model mencapai akurasi 98% dan *f1-score* 0.99 dengan optimasi *hyperparameter*, yang memungkinkan setiap model bekerja optimal. Meskipun menghasilkan performa yang cukup baik, penelitian ini terbatas pada domain kesehatan, ukuran dataset yang relatif kecil, serta belum diterapkan pada konteks klasifikasi cuaca yang umumnya memiliki karakteristik data non-linear dan jumlah fitur yang kompleks.

Dari berbagai studi tersebut, dapat disimpulkan bahwa metode *ensemble learning*, khususnya teknik *stacking*, memiliki potensi besar untuk meningkatkan kinerja klasifikasi, termasuk dalam kasus klasifikasi cuaca. Namun, masih terdapat ruang untuk eksplorasi lebih lanjut, terutama terkait optimasi model dan penanganan ketidakseimbangan data.

3 Metode Penelitian

Penelitian ini bertujuan untuk mengklasifikasi kondisi cuaca di wilayah Jawa Barat dengan menerapkan metode *ensemble learning* menggunakan *stacking classifier*. Proses penelitian mencakup sejumlah tahapan, mulai dari pemuatan dataset, *preprocessing*, pembagian data latih dan data uji, pelatihan model, hingga evaluasi performa sebagaimana diilustrasikan pada Gambar 1.



Gambar 1. Tahapan penelitian

3.1 Dataset

Dataset yang digunakan dalam penelitian ini diperoleh dari portal data terbuka milik BMKG melalui situs resminya (<https://data.bmkg.go.id>). Penelitian difokuskan pada wilayah Jawa Barat selama bulan Desember 2024, dengan data sebanyak 6.520 baris. Dataset ini mencakup berbagai parameter meteorologi, sebagaimana ditunjukkan pada Tabel 1.

Tabel 1. Deskripsi atribut

Atribut Data	Tipe Data	Deskripsi
Lokasi_id	Integer	Kode daerah atau wilayah terjadinya perubahan cuaca (bukan Lokasi/stasiun pengamatannya)
Waktu	Object	Tanggal dan waktu terjadinya cuaca
Tn	Float	Suhu udara minimum, diukur dalam satuan °C
Tx	Float	Suhu udara maksimum, diukur dalam satuan °C
Tavg	Integer	Suhu udara rata-rata harian berdasarkan pembacaan berkala, diukur dalam satuan °C
RH_n	Float	Kelembaban udara minimum (%)
RH_x	Float	Kelembaban udara maksimum (%)
RH_avg	Integer	Rata-rata kelembaban udara harian (%)
ddd	Object	Arah datangnya angin berdasarkan delapan arah utama (N, NE, E, SE, S, SW, W, NW)
ff	Integer	Kecepatan rata-rata angin (m/s)
Kelas	Object	Label klasifikasi cuaca yang terdiri dari beberapa kategori: Hujan Ringan, Hujan Sedang, Hujan Lebat, Hujan Petir, Berawan Tebal, Berawan, dan Cerah berawan)

3.2 Preprocessing

Tahap *preprocessing* diperlukan karena data yang baru diperoleh seringkali tidak bisa langsung digunakan. Tahapan *preprocessing* yang dilakukan meliputi:

a. Pengecekan Data Hilang

Langkah awal dalam proses *preprocessing* adalah melakukan pengecekan terhadap nilai kosong (*missing value*) pada seluruh fitur dalam dataset. Jika ditemukan, beberapa metode imputasi yang dapat digunakan, diantaranya interpolasi linear, *mean*, *median*, *mode imputation*. Namun, penerapan teknik ini hanya dilakukan jika terdapat nilai kosong [12].

b. Transformasi Data

Setelah proses pengecekan data hilang, dilakukan transformasi terhadap beberapa fitur. Fitur arah angin yang bersifat kategorikal diubah menjadi bentuk numerik agar sesuai dengan algoritma *machine learning*.

c. Pembagian Dataset

Dataset dibagi kedalam dua subset utama, yaitu data latih dan data uji. Data latih digunakan dalam proses pembentukan dan pelatihan model klasifikasi, sedangkan data uji digunakan untuk mengevaluasi performa model terhadap data yang belum pernah dilihat sebelumnya. Pembagian dilakukan secara acak (*random split*) untuk menjaga representativitas setiap kelas dalam kedua subset.

d. Penanganan Outlier

Outlier merupakan data yang menyimpang jauh dari distribusi utama dan berpotensi mengganggu kinerja model [8]. Penelitian ini menggunakan metode *Interquartile Range (IQR)* untuk mendeteksi *outlier*, yang kemudian dihapus jika tidak dapat divalidasi secara substantif. Langkah ini dilakukan untuk meningkatkan kualitas data dan akurasi model klasifikasi.

e. Normalisasi Data

Proses normalisasi diperlukan karena algoritma *Support Vector Machine (SVM)* sensitif terhadap skala nilai fitur. Normalisasi dilakukan menggunakan metode *z-score* untuk memastikan bahwa data memiliki rata-rata nol dan standar deviasi satu, sebagaimana ditunjukkan dalam Persamaan (2).

$$x = \frac{x - \mu}{\sigma} \quad (2)$$

Keterangan:

x : Data yang akan dinormalisasi

μ : Rata rata

σ : Deviasi standar

f. Synthetic Minority Oversampling Technique (SMOTE)

SMOTE digunakan untuk menangani masalah ketidakseimbangan kelas pada dataset meteorologi. Ketidakseimbangan ini berpotensi menyebabkan model cenderung bias terhadap kelas mayoritas, sehingga menurunkan kemampuan generalisasinya terhadap kelas minoritas. Metode ini menghasilkan data sintetis dari kelas minoritas dengan cara melakukan interpolasi linier antara data data yang berdekatan dalam ruang fitur multivariat. Implementasi SMOTE dilakukan setelah proses pembagian data dan normalisasi fitur, tetapi sebelum tahap pelatihan model, untuk mencegah terjadinya *data leakage* yang dapat mengganggu validitas evaluasi [6].

3.3 Support Vector Machine (SVM)

SVM merupakan algoritma *supervised learning* yang bekerja dengan membentuk *hyperplane* pemisah optimal antar kelas dengan memaksimalkan margin [12]. Pada kasus klasifikasi cuaca dimana hubungan antar fitur meteorologi bersifat *non-linear*, kernel *Radial Basis Function (RBF)* dipilih karena kemampuannya menangani kompleksitas pada data meteorologi. SVM menawarkan pendekatan yang efektif dengan memanfaatkan formulasi matematis untuk mengenali pola dan mengklasifikasikan cuaca secara akurat [14]. Rumus kernel RBF ditunjukkan pada Persamaan (3).

$$K(x, x_i) = \exp(-\gamma \|x - x_i\|^2) \quad (3)$$

Keterangan:

x dan x_i : Vektor fitur input

x_2, y_2 : Parameter kernel yang menentukan radius pengaruh titik data

Agar model bekerja secara optimal, dilakukan proses optimasi *hyperparameter* menggunakan *GridSearchCV*, yaitu pencarian kombinasi parameter terbaik melalui *k-fold cross-validation* [10]. Parameter utama yang dioptimalkan meliputi:

- C (Parameter Regularisasi)

Berfungsi untuk mengontrol keseimbangan antara ukuran margin pemisah dan jumlah kesalahan klasifikasi pada data pelatihan. Nilai C yang kecil membuat margin lebih lebar tetapi toleransi terhadap misklasifikasi lebih tinggi. Sebaliknya, nilai C besar membuat margin lebih sempit dengan toleransi misklasifikasi rendah [10]. Fungsi objektif SVM melibatkan parameter C dapat dilihat persamaan berikut:

$$\text{minimize } \frac{1}{2} \|w\|^2 + c \sum_{i=1}^n \max(0, 1 - y_i(w * x_i - b)) \quad (4)$$

- γ (gamma)

Menentukan radius pengaruh setiap titik data. Nilai γ yang tinggi mengindikasikan kedekatan, dan nilai yang rendah mengindikasikan jarak. Parameter ini menentukan bentuk batas keputusan [10].

Optimasi kedua parameter ini penting dilakukan untuk memastikan bahwa model SVM dapat memberikan kinerja klasifikasi cuaca lebih optimal.

3.4 Random Forest

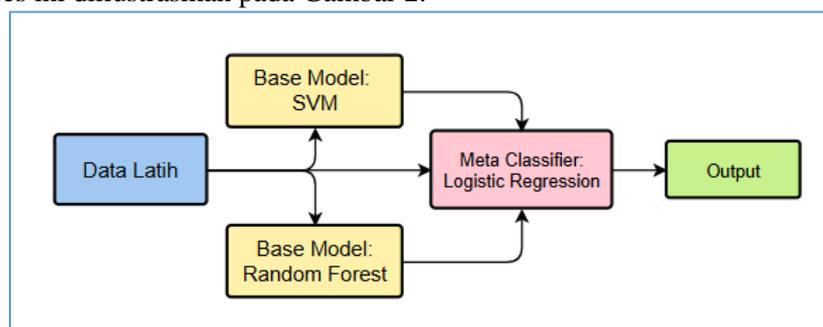
Random Forest merupakan salah satu metode *ensemble learning* yang bekerja dengan membentuk sejumlah pohon Keputusan (*decision trees*) dan menggabungkan hasil prediksi dari masing-masing pohon untuk meningkatkan akurasi serta stabilitas model [13]. Algoritma ini dipilih karena kemampuannya dalam menangani struktur data yang kompleks, ketahanannya terhadap *outliers* dan mengurangi risiko *overfitting* melalui penyesuaian *hyperparameter* [14]. Parameter yang digunakan dijelaskan pada Tabel 2.

Tabel 2. Parameter random forest

Parameter	Deskripsi	Rentang Nilai
n_estimators	Jumlah pohon dalam <i>random forest</i> .	[100, 150, 200]
max_depth	Kedalaman maksimum dari setiap pohon Keputusan.	[None, 5, 10, 15]
min_samples_split	Jumlah minimum sampel yang diperlukan untuk membagi node menjadi cabang baru.	[2, 5]
min_samples_leaf	Jumlah minimum sampel yang harus dimiliki sebuah <i>leaf node</i> setelah pemisahan.	[1, 2]

3.5 Ensemble Learning

Metode *ensemble learning* merupakan pendekatan yang menggabungkan model untuk meningkatkan akurasi dan kestabilan prediksi. Penelitian ini menggunakan pendekatan *stacking classifier*, yang bekerja dengan mengkombinasikan keunggulan beberapa model *machine learning* melalui arsitektur dua tingkat. Pada tingkat pertama (*base models*), digunakan dua algoritma, yaitu SVM dan *Random Forest* masing-masing model dilatih untuk menghasilkan prediksi awal. Selanjutnya pada tingkat kedua (*meta-classifier*), dengan model *Logistic Regression* yang menerima hasil prediksi dari *base models* sebagai input dan mengkombinasikannya untuk menghasilkan prediksi akhir [11]. Proses ini diilustrasikan pada Gambar 2.



Gambar 2. Diagram arsitektur stacking classifier

Untuk mengurangi risiko *overfitting* saat pelatihan *meta-classifier*, digunakan *cross-validation* yang memungkinkan pelatihan dan evaluasi dilakukan secara bergantian pada subset data yang berbeda [15]. Pendekatan ini dirancang untuk meningkatkan kemampuan model dalam melakukan generalisasi terhadap data baru yang belum pernah digunakan selama proses pelatihan.

3.6 Evaluasi Model

Evaluasi model dilakukan menggunakan *confusion matrix* sebagai dasar evaluasi, yang menyajikan perbandingan label aktual dan hasil prediksi model klasifikasi. Terdiri dari *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Struktur dasar *confusion matrix* dapat dilihat pada Gambar 3. Penelitian ini mengadopsi empat metrik utama: (1) Akurasi mengukur prediksi benar secara keseluruhan, (2) presisi mengevaluasi ketepatan prediksi positif, (3) *recall* mengukur kemampuan mendeteksi kasus positif, dan (4) *f1-score* sebagai *harmonic mean* presisi dan *recall* [16].

		Actual Values	
		Positive	Negative
Predicted Values	Positive	TP	TF
	Negative	FP	FN

Gambar 3 Confusion matrix

Persamaan empat metrik utama yang digunakan pada Persamaan (5) hingga (8) [16].

$$accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (5)$$

$$precision = \frac{TP}{TP+FP} \quad (6)$$

$$recall = \frac{TP}{FN+TP} \quad (7)$$

$$f1 - score = 2 \times \frac{precision \times recall}{precision+recall} \quad (8)$$

Validasi model menggunakan *stratified k-fold cross-validation* untuk memastikan estimasi performa yang *robust*, dimana dataset dibagi secara proporsional dengan distribusi kelas terjaga. Setiap fold secara bergantian berperan sebagai data uji sementara fold lainnya sebagai data latih, kemudian hasil evaluasi dirata-ratakan. Pendekatan ini tidak hanya mencegah *overfitting* tetapi juga memungkinkan optimasi *hyperparameter* yang lebih stabil untuk data meteorologi yang bersifat temporal dan tidak seimbang [10].

Evaluasi perbedaan performa antar model dilakukan dengan pengujian *paired t-test* sebagai metode pengujian hipotesis. Teknik statistik ini digunakan untuk membandingkan nilai rata-rata

antara dua kelompok data [17]. Jika nilai *t-test* lebih kecil dari 0.05 maka H_0 ditolak, menggunakan perhitungan yang tercantum pada persamaan (9).

$$t = \frac{\bar{x} - \mu_0}{\frac{s}{\sqrt{n}}} \quad (9)$$

Keterangan:

t: Nilai statistik *t-test*

\bar{x} : Nilai rata-rata sampel

μ_0 : Rata-rata populasi yang diasumsikan

s: Standar deviasi sampel

n: Ukuran sampel

4 Hasil dan Pembahasan

Berdasarkan tahapan yang telah dijelaskan sebelumnya, dilakukan implementasi dan evaluasi terhadap model klasifikasi cuaca. Proses ini mencakup *preprocessing* data, pembangunan model, dan evaluasi kinerja model.

4.1 Preprocessing

Data yang digunakan akan melalui tahap *preprocessing* untuk memastikan data siap untuk digunakan dalam proses pelatihan model.

a. Pembersihan Data

Data yang diperoleh dari BMKG melalui *preprocessing* untuk memastikan data siap digunakan dalam pemodelan dan analisis. Berdasarkan hasil pengecekan, tidak ditemukan adanya nilai kosong pada seluruh fitur. Oleh karena itu, teknik imputasi *missing value* tidak diperlukan dalam penelitian ini.

b. Transformasi Data

Beberapa fitur pada dataset meteorologi, seperti *ddd* (arah angin) dan kelas (label cuaca), memiliki tipe data kategorikal. Karena algoritma *machine learning* yang akan digunakan hanya dapat memproses data numerik, dilakukan transformasi menggunakan *label encoding* untuk mengubah nilai kategorikal menjadi nilai numerik.

c. Pembagian Data

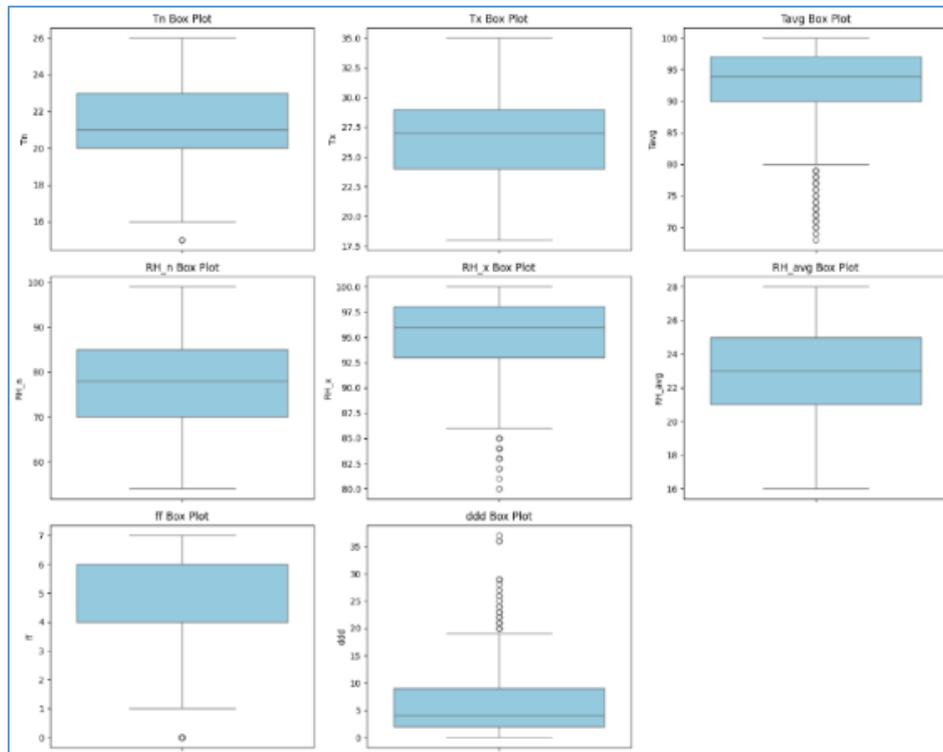
Setelah proses transformasi, dataset dibagi ke dalam dua subset, yaitu data latih dan data uji dengan beberapa rasio pembagian 90:10, 80:20, dan 70:30. Pembagian dilakukan secara acak menggunakan fungsi *train_test_split* dari *library scikit-learn*. Untuk menjaga proporsi distribusi kelas, terutama kelas minoritas, digunakan parameter *stratify* pada label kelas. Hal ini bertujuan agar model dapat dilatih dan diuji secara adil terhadap seluruh kelas. Rincian jumlah data pada masing-masing rasio pembagian disajikan ke dalam Tabel 3.

Tabel 3. Pembagian data penelitian

Data	Jumlah Data Latih	Jumlah Data Uji	Total
70:30	4564	1956	6520
80:20	5216	1304	6520
90:10	5868	652	6520

d. Penanganan Outlier

Dataset cuaca rentan terhadap adanya *outlier* akibat variasi ekstrem pada fitur seperti suhu, kelembaban, kecepatan angin, atau curah hujan. Dalam penelitian ini, data yang diperoleh dari BMKG juga menunjukkan keberadaan *outlier*, sebagaimana terlihat pada Gambar 4. Beberapa fitur menunjukkan nilai yang secara signifikan menyimpang dari distribusi rata-rata, mengindikasikan ketidakwajaran dalam pengukuran atau fenomena cuaca yang ekstrem.



Gambar 4. Box plot fitur

Penanganan *outlier* dilakukan setelah pembagian data, dan hanya diterapkan pada data latih saja. *Outlier* ditangani dengan menggunakan fungsi *mask* dari *library scikit-learn* untuk menyaring nilai-nilai yang berada di luar ambang batas yang ditentukan. Nilai-nilai tersebut kemudian dihapus dari data latih untuk menjaga kualitas model dalam proses pelatihan. Setelah proses ini, ukuran data uji akan tetap, namun ukuran data latih berkurang, sebagaimana ditunjukkan pada Tabel 4.

Tabel 4. Data setelah outlier

Data	Sebelum Outlier			Setelah Outlier		
	Data Latih	Data Uji	Total	Data Latih	Data Uji	Total
70:30	4564	1956	6520	4023	1956	5979
80:20	5216	1304	6520	4596	1304	5900
90:10	5868	652	6520	5173	652	5825

Langkah ini bertujuan untuk meningkatkan stabilitas model dan akurasi prediksi, dengan menghilangkan pengaruh nilai-nilai ekstrem yang berpotensi mengganggu proses pembelajaran.

e. Normalisasi Data

Proses normalisasi (*scaling*) diterapkan pada data latih untuk menyamakan skala nilai numerik antar fitur. Proses ini penting karena model SVM sensitif terhadap perbedaan skala antar fitur. Normalisasi dilakukan menggunakan metode *standardization* (*z-score normalization*) yang mengubah nilai setiap fitur kedalam distribusi standar (*mean* = 0, standar deviasi = 1). Contoh data sebelum dan sesudah normalisasi ditampilkan pada Tabel 5 dan Tabel 6, masing-masing menampilkan 5 entri data awal.

Tabel 5. Data Sebelum Normalisasi

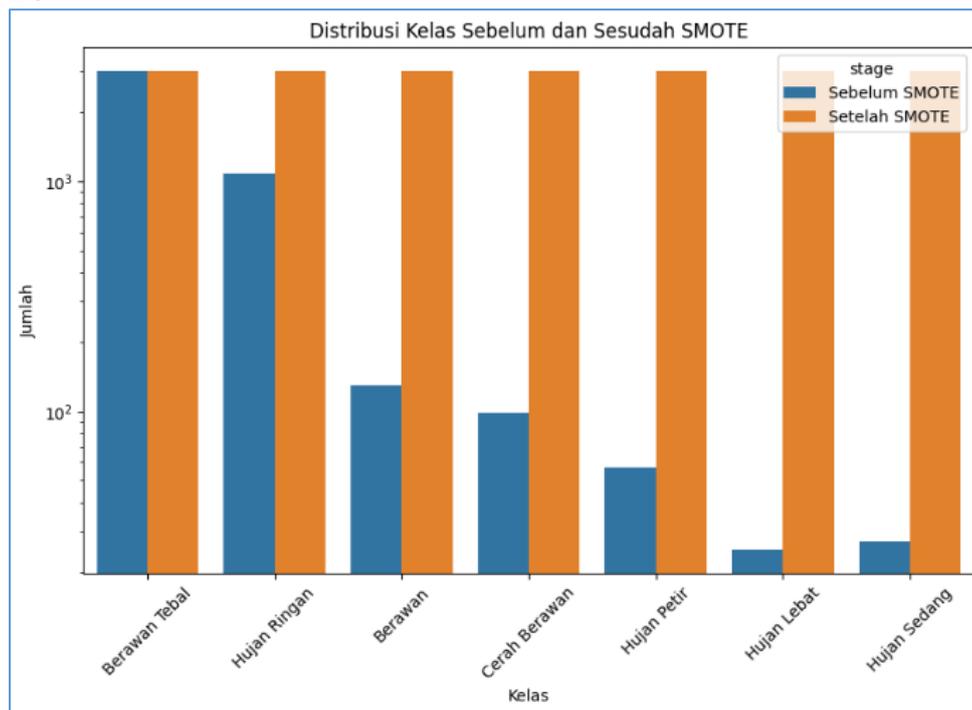
Tn	Tx	RH_n	RH_x	Tavg	RH_avg	ff	ddd	kelas
23	29	67	92	24	86	6	6	Hujan Ringan
21	31	70	94	26	92	6	4	Berawan Tebal
24	26	84	94	24	93	6	15	Hujan Ringan
21	30	74	84	23	89	3	5	Hujan Ringan
20	29	65	88	23	83	2	12	Berawan Tebal

Tabel 6. Data setelah normalisasi

Tn	Tx	RH_n	RH_x	Tavg	RH_avg	ff	ddd	kelas
1.8603	0.7014	-1.8367	-2.0015	1.3894	-1.8367	0.5303	2.6104	Hujan Ringan
0.8336	0.0522	-0.7329	-0.7405	0.9312	-0.7329	0.5303	0.9076	Berawan Tebal
1.8603	0.0522	-1.1744	-2.3168	1.3894	-1.1744	0.5303	0.2691	Hujan Ringan
-1.2197	-0.9215	-0.0706	0.5203	-1.3600	-0.0706	-0.0497	0.4819	Hujan Ringan
-0.7064	-1.5707	1.2539	1.1508	-1.3600	1.2539	-1.2100	0.2691	Berawan Tebal

f. SMOTE

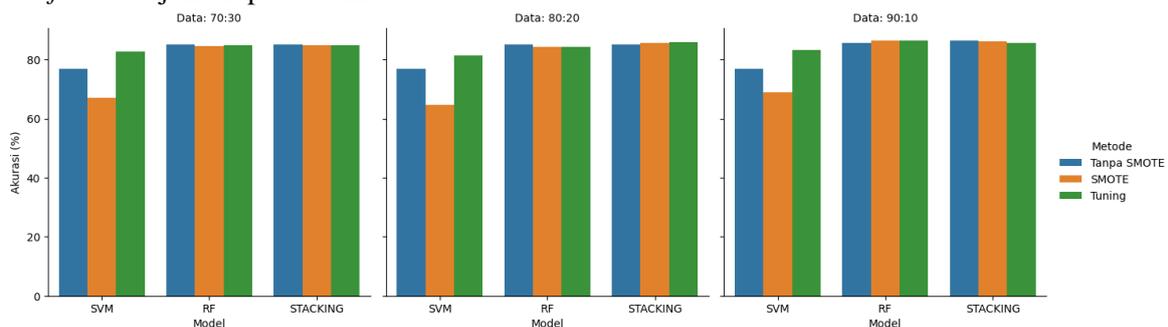
Ketidakseimbangan distribusi kelas dalam dataset diatasi menggunakan teknik metode SMOTE. Penerapan SMOTE dilakukan setelah proses normalisasi dan sebelum pelatihan model untuk menghindari *data leakage*. Perbandingan distribusi data sebelum dan sesudah SMOTE ditunjukkan pada Gambar 5.



Gambar 5. Distribusi kelas

4.2 Hasil Klasifikasi

Klasifikasi cuaca pada penelitian ini dilakukan menggunakan *stacking classifier* yang menggabungkan dua model *machine learning*, yaitu SVM dan *Random Forest*. Model dilatih dan diuji menggunakan berbagai skenario pembagian data serta teknik penanganan ketidakseimbangan kelas dengan SMOTE dan optimasi *hyperparameter*. Akurasi dari setiap *base-models* dan *stacking classifier* ditunjukkan pada Gambar 6.



Gambar 6. Grafik hasil akurasi model

Gambar 6 memperlihatkan akurasi dari masing-masing model. *Random Forest* secara konsisten menunjukkan akurasi tinggi untuk ketiga rasio perbandingan, baik dengan SMOTE maupun tanpa penerapan SMOTE. SVM menunjukkan performa yang lebih bervariasi dan cenderung lebih rendah, terutama tanpa SMOTE. Sementara itu, *stacking classifier* unggul tipis namun stabil di semua skenario, menunjukkan bahwa *ensemble learning* mampu memberikan hasil yang lebih *robust* dibanding model tunggal.

Hasil evaluasi kinerja model lebih rinci, termasuk presisi, *recall*, dan *f1-score* untuk seluruh skenario ditampilkan pada Tabel 7.

Tabel 7. Hasil evaluasi model stacking classifier

Data Size	Skenario Pengujian	Model	Metrik Evaluasi			
			Akurasi (%)	Presisi (%)	Recall (%)	F1-score (%)
70:30	Tanpa SMOTE	SVM	56.65	34.26	69.08	38.54
		<i>Random Forest</i>	85.07	82.80	49.12	56.06
		<i>Stacking Classifier</i>	85.48	86.47	49.85	57.07
	SMOTE	SVM	66.97	42.70	70.96	48.89
		<i>Random Forest</i>	84.46	77.23	59.89	65.53
		<i>Stacking Classifier</i>	84.76	76.96	58.24	64.43
	SMOTE & Optimasi Hyperparameter	SVM	82.98	69.33	59.71	62.93
		<i>Random Forest</i>	84.46	77.23	59.89	65.53
		<i>Stacking Classifier</i>	84.56	76.16	54.42	63.14
80:20	Tanpa SMOTE	SVM	77.99	36.87	26.59	29.32
		<i>Random Forest</i>	87.04	84.58	56.78	63.48
		<i>Stacking Classifier</i>	86.73	83.33	55.99	62.59
	SMOTE	SVM	68.79	43.63	73.49	50.46
		<i>Random Forest</i>	85.20	75.76	60.63	65.31
		<i>Stacking Classifier</i>	85.66	78.05	64.91	69.69
	SMOTE & Optimasi Hyperparameter	SVM	83.67	70.38	68.86	68.23
		<i>Random Forest</i>	85.58	76.48	60.86	65.69
		<i>Stacking Classifier</i>	85.97	78.76	63.21	68.99
90:10	Tanpa SMOTE	SVM	76.84	36.02	24.94	27.51
		<i>Random Forest</i>	85.58	84.87	57.62	65.10
		<i>Stacking Classifier</i>	86.50	92.15	58.28	65.88
	SMOTE	SVM	69.02	44.13	74.16	51.33
		<i>Random Forest</i>	86.35	77.97	70.02	73.45
		<i>Stacking Classifier</i>	85.89	78.55	67.34	71.95
	SMOTE & Optimasi Hyperparameter	SVM	83.13	67.47	74.42	70.41
		<i>Random Forest</i>	85.89	78.85	68.94	73.23
		<i>Stacking Classifier</i>	85.74	78.18	72.36	74.93

Berdasarkan hasil yang diperoleh, model *Random Forest* menunjukkan akurasi tertinggi sebesar 87.04% pada skenario pembagian data 80:20 tanpa penerapan SMOTE. Namun, setelah SMOTE diterapkan, akurasi menurun menjadi 85.20%, tapi terdapat peningkatan pada kemampuan model dalam mengenali kelas minoritas, ditandai dengan naiknya nilai *recall* sebesar 3.85%. Model SVM dengan penerapan SMOTE dan optimasi *hyperparameter* terbukti secara signifikan meningkatkan performa, ditunjukkan oleh peningkatan akurasi dan *f1-score* dibandingkan saat model digunakan tanpa keduanya. Pada pembagian data 70:30, akurasi SVM meningkat dari 56.65% menjadi 82.98% dan untuk *f1-score* dari 38.54% menjadi 62.93% setelah SMOTE dan optimasi *hyperparameter* diterapkan. Meskipun demikian, performa SVM tetap lebih rendah dibandingkan *Random Forest* dan *stacking classifier*, terutama pada konsistensi nilai *recall* dan *f1-score*.

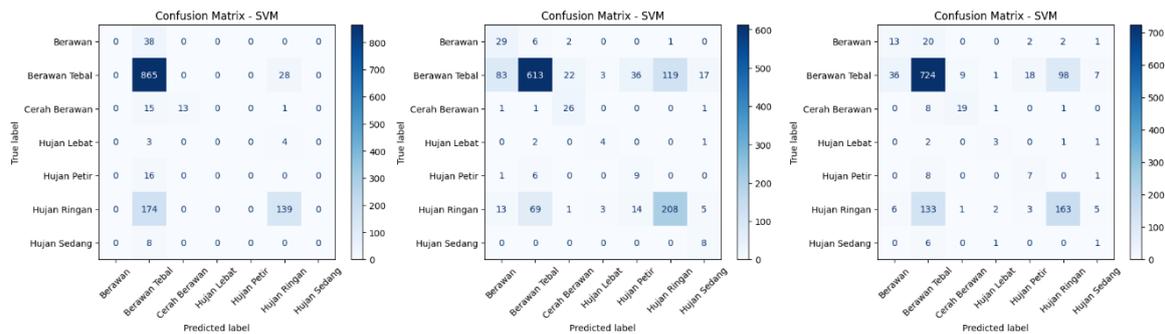
Model *stacking classifier* menunjukkan performa yang paling unggul, dengan *f1-score* bernilai 74.93% pada rasio 90:10 dengan penerapan SMOTE dan optimasi *hyperparameter*, mengungguli base

<http://sistemasi.ftik.unisi.ac.id>

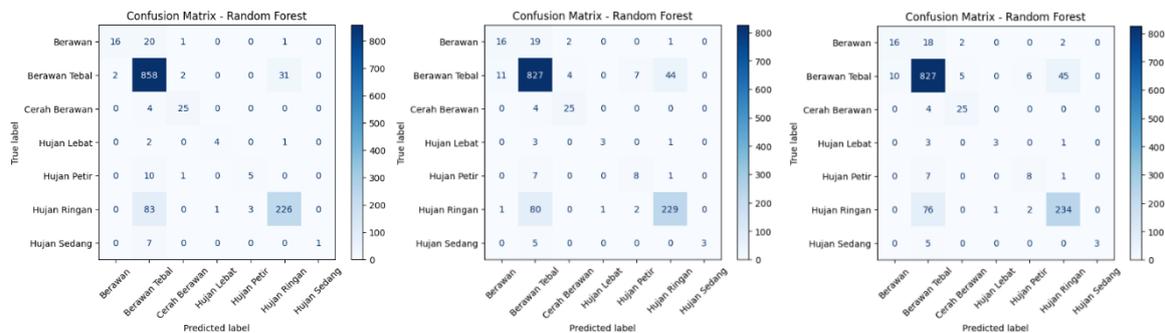
models SVM dan *Random Forest*. Hasil ini sejalan dengan temuan pada penelitian [9] yang menunjukkan bahwa kombinasi model tunggal dalam metode *ensemble learning* dapat mengurangi varians dan bias, sehingga meningkatkan performa secara keseluruhan.

Berdasarkan perbandingan performa model pada tiga skenario pembagian data, rasio 80:20 menunjukkan hasil yang paling seimbang. Pada rasio ini, *stacking classifier* yang dilatih dengan SMOTE dan optimasi *hyperparameter* menghasilkan kombinasi akurasi tinggi 85.97%, presisi 78.76%, recall 63.21%, dan F1-score 68.99%. Dengan rasio ini, jumlah data uji yang digunakan cukup besar, untuk memastikan bahwa penilaian performa model lebih mewakili kondisi nyata dan tidak memberikan hasil yang terlalu baik seperti pada pembagian data 90:10.

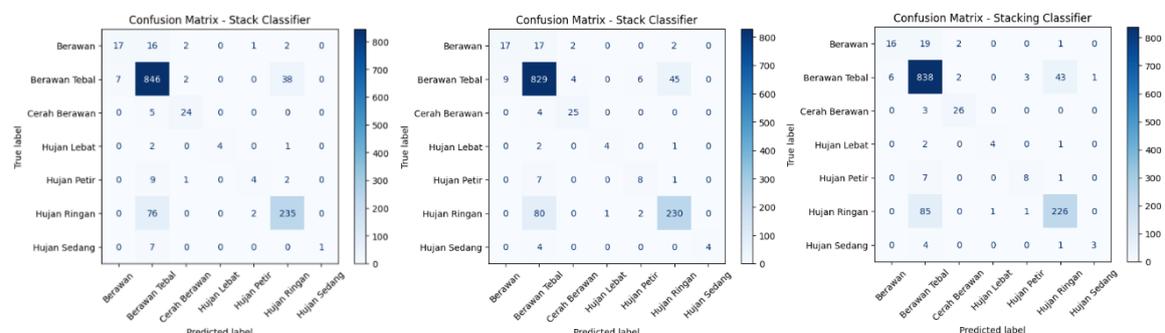
Untuk memahami performa model terhadap masing-masing kelas, digunakan *confusion matrix*. Analisis ini membantu mengevaluasi kemampuan klasifikasi, terutama pada data dengan distribusi tidak seimbang. Gambar 7-9 menunjukkan *confusion matrix* untuk masing-masing model pada rasio pembagian data 80:20, yang memperlihatkan jumlah prediksi benar dan salah untuk setiap kelas cuaca.



Gambar 7. Confusion Matrix SVM (a) Tanpa SMOTE (b) SMOTE (c) Hyperparameter



Gambar 8. Confusion Matrix Random Forest (a) Tanpa SMOTE (b) SMOTE (c) Hyperparameter



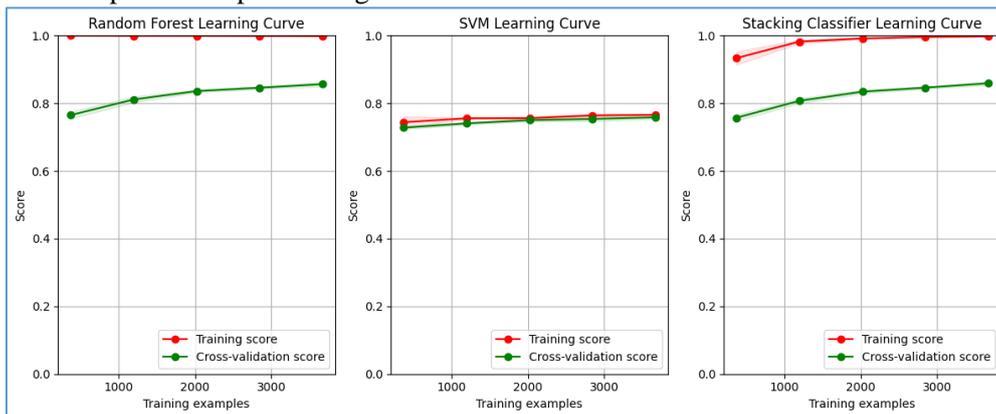
Gambar 9 Confusion Matrix Stacking Classifier (a) Tanpa SMOTE (b) SMOTE (c) Hyperparameter

Sebelum penerapan SMOTE, seluruh model cenderung bias terhadap kelas mayoritas “Berawan Tebal”, dengan SVM gagal mengenali kelas minoritas seperti “Berawan”, “Hujan Sedang”, dan “Hujan Lebat” dengan *recall* 0%. Setelah SMOTE diterapkan, *recall* kelas minoritas seperti

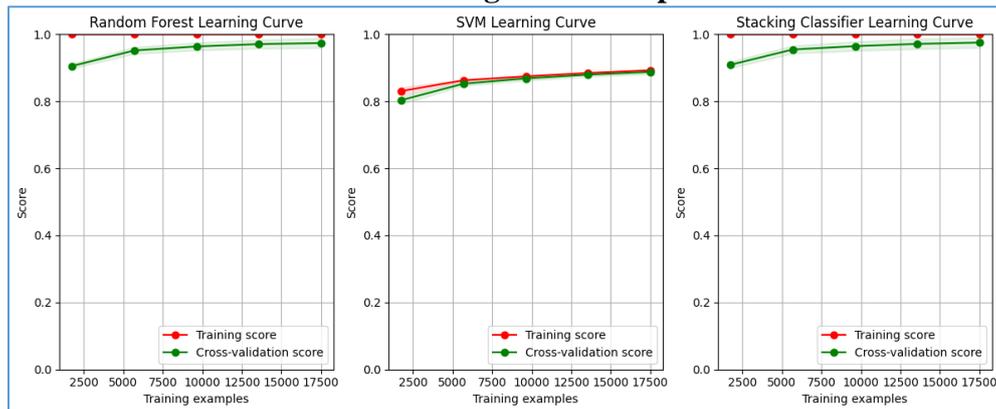
“Berawan” pada SVM meningkat signifikan dari 0% menjadi 76.3% (29 dari 38 data), meskipun diikuti penurunan presisi kelas “Berawan Tebal” menurun dari 98.2% menjadi 89.4%. *Random Forest* menunjukkan penurunan tak terduga pada *recall* kelas “Hujan Petir” dari 53.3% menjadi 40%. Sementara, *stacking classifier* menunjukkan hasil yang beragam, meningkat untuk beberapa kelas tetapi menurun untuk kelas lainnya.

Penerapan SMOTE tidak selalu meningkatkan akurasi model, sebagaimana dikemukakan dalam penelitian [18], karena keterbatasannya dalam menangani distribusi kelas minoritas yang kompleks dan kecenderungan menghasilkan duplikasi sampel yang tidak merepresentasikan batas antar kelas secara kuat. Setelah optimasi *hyperparameter*, ketiga model menunjukkan kemampuan yang lebih seimbang, dengan peningkatan presisi dan *recall* pada beberapa kelas minoritas.

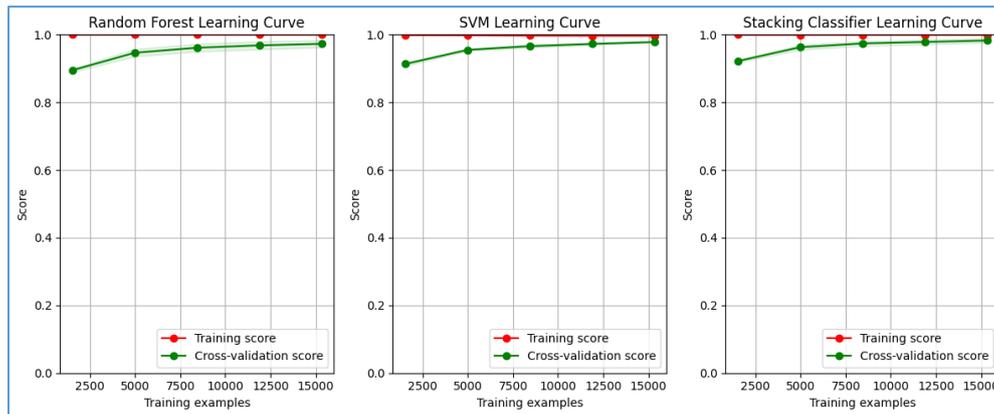
Namun demikian, penggunaan SMOTE dan optimasi *hyperparameter* berhasil mengurangi gejala *overfitting*. Hal ini dapat dilihat pada *learning curve* Gambar 10-12, ketiga model menunjukkan pengurangan *overfitting* termasuk *stacking classifier*. Hasil ini sesuai dengan visualisasi dari *confusion matrix* pada rasio perbandingan data 80:20.



Gambar 10. Learning Curve Tanpa SMOTE



Gambar 11 Learning Curve SMOTE



Gambar 12. Learning Curve Smote dan Optimasi Hyperparameter

Hasil *cross-validation* dan visualisasi *learning curve* memperkuat hal ini. Model tanpa penerapan SMOTE, terlihat adanya gejala *overfitting* pada model *Random Forest* dan *stacking classifier*, dengan akurasi pelatihan mendekati 100% namun akurasi validasi hanya berkisar 86.73%. Hal ini menciptakan gap yang signifikan sebesar 13.27%. Sementara SVM menunjukkan stabilitas model yang lebih baik, dengan nilai yang konsisten antara pelatihan dan validasi

Setelah penerapan SMOTE dan optimasi *hyperparameter*, gap performa pada ketiga model menurun menjadi kurang dari 1% serta kurva pelatihan dan validasi menunjukkan pola yang semakin *konvergen*. Meskipun pada awal kurva masih terlihat perbedaan, seiring bertambahnya jumlah data pelatihan, jarak antar kurva mengecil, menunjukkan bahwa model mulai belajar secara lebih general dan *robust* terhadap data baru. Temuan ini menunjukkan bahwa model memiliki kemampuan generalisasi yang baik pada data baru.

Untuk mengetahui lebih lanjut apakah peningkatan performa setelah penerapan SMOTE dan optimasi *hyperparameter* signifikan secara statistik, dilakukan *t-test* untuk masing masing model akan dibandingkan dengan tiga skema skenario secara bergantian. Hasilnya disajikan pada tabel berikut:

Tabel 8. Hasil uji statistik t-test

Model A	Model B	Rata-rata A	Rata-rata B	Selisih Rata-rata	t-statistika	p-value	Signifikan si ($\alpha=0.05$)
Stacking Classifier + SMOTE	Stacking Classifier (tanpa SMOTE)	0.975 8	0.863 9	+0.111 9	25.777 7	9.5901×10^{-10}	Signifikan
Stacking Classifier + SMOTE + optimasi <i>hyperparameter</i> <i>r</i>	Stacking Classifier (tanpa SMOTE)	0.983 9	0.863 9	+0.12	29.921 0	2.6722×10^{-10}	Signifikan
Stacking Classifier + SMOTE + optimasi <i>hyperparameter</i> <i>r</i>	Stacking Classifier + SMOTE	0.983 9	0.975 8	+0.008 1	8.0402	2.1265×10^{-5}	Signifikan
Stacking Classifier + SMOTE + optimasi <i>hyperparameter</i> <i>r</i>	SVM + SMOTE + optimasi <i>hyperparameter</i> <i>r</i>	0.983 9	0.978 7	+0.005 2	8.7447	1.0796×10^{-5}	Signifikan

r							
Stacking Classifier + SMOTE + optimasi hyperparameter	Random Forest + SMOTE + optimasi hyperparameter	0.983 9	0.975 6	+0.008 3	7.3500	4.3269 $\times 10^{-5}$	Signifikan
r	r						

Hasil *paired t-test* menunjukkan bahwa setiap model menghasilkan nilai $p\text{-value} < 0.05$. Hal ini mengindikasikan bahwa perbedaan performa antar model pada setiap skema perbandingan signifikan secara statistik. Penerapan SMOTE pada *stacking classifier* meningkatkan $f1\text{-score}$ dari 0.8639 menjadi 0.9758 (selisih +0.1119) dan setelah ditambahkan optimasi *hyperparameter*, performa meningkat lebih lanjut menjadi 0.9839 (selisih +0.12).

Pengujian dilakukan dengan *10-fold cross-validation* untuk memastikan reliabilitas evaluasi. Menggunakan metrik $f1\text{-score}$, karena dataset yang digunakan tidak seimbang, sehingga $f1\text{-score}$ dinilai lebih representatif dalam mengukur kinerja model secara keseluruhan terhadap semua kelas.

Secara umum, *stacking classifier* menunjukkan keunggulan performa yang konsisten. Dalam konfigurasi terbaiknya (dengan SMOTE dan optimasi *hyperparameter*), *stacking classifier* tetap lebih unggul dibanding SVM dan *Random Forest* yang juga telah dioptimasi, dengan selisih $f1\text{-score}$ masing-masing sebesar +0.0052 dan +0.0083.

Model *stacking* dengan *Logistic Regression* sebagai *meta-classifier* terbukti optimal untuk klasifikasi cuaca pada data tidak seimbang, dengan capaian akurasi 85.97% dan $f1\text{-score}$ 68.99%. Penerapan SMOTE terbukti efektif dalam meningkatkan recall pada kelas minoritas, namun perlu dikombinasikan dengan optimasi *hyperparameter* untuk meminimalkan *trade-off* terhadap nilai presisi.

Temuan ini memberikan wawasan penting bagi pengembangan sistem prakiraan cuaca berbasis *machine learning*, khususnya di wilayah dengan distribusi kelas yang tidak merata.

5 Kesimpulan

Berdasarkan hasil dan pembahasan sebelumnya, dapat disimpulkan bahwa model *stacking classifier* yang menggabungkan SVM dan *Random Forest* dengan *Logistic Regression* menunjukkan performa yang efektif dalam klasifikasi cuaca di Jawa Barat. Penerapan SMOTE dengan optimasi *hyperparameter*, berhasil mengurangi bias terhadap kelas mayoritas dengan meningkatnya *recall* untuk kelas “Berawan” dari 0% menjadi 76.3%. Selain itu, gejala *overfitting* yang sebelumnya terlihat pada model tanpa SMOTE dengan gap akurasi antara data latih dan validasi sebesar 13.27% (lihat Gambar 10) berhasil ditekan menjadi kurang dari 1%, menandakan peningkatan kemampuan generalisasi model yang baik. Pencapaian ini diperkuat dengan peningkatan $f1\text{-score}$ sebesar 6.4% dan stabilitas antar-fold tertinggi dengan akurasi 85.97% dibanding model tunggal serta signifikansi statistik ($p < 0.01$) dalam *t-test* terhadap penerapan SMOTE. Temuan ini tidak hanya menjawab tantangan spesifik klasifikasi cuaca di Jawa Barat, tetapi juga membuka jalan bagi penerapan di wilayah dengan karakteristik serupa. Penelitian selanjutnya disarankan mengeksplorasi teknik *balancing* lain seperti ADASYN atau pendekatan *ensemble* berbasis *boosting* untuk hasil klasifikasi yang lebih optimal di wilayah lain dengan karakteristik serupa.

Ucapan Terima Kasih

Penulis mengucapkan terima kasih kepada Badan Meteorologi, Klimatologi, dan Geofisika (BMKG) atas penyediaan dan akses terhadap data cuaca yang digunakan dalam penelitian ini. Data tersebut berperan penting dalam proses pengembangan serta evaluasi model klasifikasi cuaca yang menjadi fokus utama penelitian ini.

Referensi

- [1] S. Ardhasena, Marjuki, A. F. Radjab, and H. T. Djatmiko, “At the Front Line of Climate Action,” *Kedeputan Bidang Klimatologi, BMKG*, 2024.

- [2] A. Rosyida, M. Aziz, Y. Firmansyah, T. Setiawan, K. P. Pangesti, and F. Kakanur I., *Data Bencana Indonesia 2023*, Vol. 3. Pusat Data Informasi dan Komunikasi Kebencanaan Badan Nasional Penanggulangan Bencana, 2024. [Online]. Available: <https://bpbd.kepriprov.go.id/files/buku-data-bencana-indonesia-tahun-2023.pdf>
- [3] E. Dritsas, M. Trigka, and P. Mylonas, "A Multi-class Classification Approach for Weather Forecasting with Machine Learning Techniques," in *2022 17th International Workshop on Semantic and Social Media Adaptation & Personalization (SMAP)*, Corfu, Greece: IEEE, Nov. 2022, pp. 1–5. doi: 10.1109/SMAP56125.2022.9942121.
- [4] S. I. Fallo, M. A. Aprihartha, and J. Prasetya, "Optimization of Early Warning System for Landslides based on Rainfall using Naive Bayes Classifier and Multiclass Support Vector Machine Algorithm in Takari Region," 2024.
- [5] A. Toha, P. Purwono, and W. Gata, "Model Prediksi Kualitas Udara dengan Support Vector Machines dengan Optimasi Hyperparameter GridSearch CV," *Buletin Ilmiah Sarjana Teknik Elektro*, Vol. 4, No. 1, pp. 12–21, May 2022, doi: 10.12928/biste.v4i1.6079.
- [6] I. Srivani, M. Sridhar, K. C. T. Swamy, and D. Venkata Ratnam, "Multi-Class Classification of Ionospheric Scintillations using SMOTE-Super Learner Ensemble Technique," *Advances in Space Research*, Vol. 73, No. 7, pp. 3845–3854, Apr. 2024, doi: 10.1016/j.asr.2023.09.039.
- [7] N. Larasati, "Perbandingan Regresi Logistik dan Random Forest pada Klasifikasi Cuaca Wilayah Jawa Tengah," *AKS*, Vol. 14, No. 2, pp. 172–181, Sep. 2023, doi: 10.26877/aks.v14i2.15985.
- [8] G. G. Ghiffary, N. T. Amanda, R. Ardhani, B. Sartono, and A. R. Firdawanti, "Analisis Kinerja Model Stacking berbasis Random Forest dan SVM dalam Klasifikasi Rumah Tangga berdasarkan Garis Kemiskinan Makanan di Provinsi Jawa Barat," *SCI TECH ED MATH*, Vol. 5, No. 3, pp. 2244–2265, Dec. 2024, doi: 10.46306/lb.v5i3.856.
- [9] S. Joses, D. Yulvida, and S. Rochimah, "Pendekatan Metode Ensemble Learning untuk Prakiraan Cuaca menggunakan Soft Voting Classifier," *J. Appl. Comput. Sci. Technol.*, Vol. 5, No. 1, pp. 72–80, Jun. 2024, doi: 10.52158/jacost.v5i1.741.
- [10] B. Selvanandhini and R. Karthikeyan, "Ensemble Heartguard: Integrating SVM and Random Forest for Robust Heart Disease Prediction," *eatp*, May 2024, doi: 10.53555/kuey.v30i5.5662.
- [11] P. Widiharso, S. Sendari, A. N. Handayani, and N. S. F. Putri, "Performa Metode Klasifikasi Tunggal dan Ensemble Model dalam Identifikasi Baku Mutu Air," *infotekmesin*, Vol. 13, No. 2, pp. 206–211, Jul. 2022, doi: 10.35970/infotekmesin.v13i2.1529.
- [12] S. Alam, M. S. Ayub, S. Arora, and M. A. Khan, "An Investigation of the Imputation Techniques for Missing Values in Ordinal Data Enhancing Clustering and Classification Analysis Validity," *Decision Analytics Journal*, Vol. 9, p. 100341, Dec. 2023, doi: 10.1016/j.dajour.2023.100341.
- [13] A. Setiawan, Y. Andalantama, M. Sidiq, and Kusriani, "Predictive Analysis of Monthly Flood Variables in the Palangkaraya Area using Multiple Regression Methods and MLR, NN, KNN, Random Forest, SVM Algorithms," in *2023 6th International Conference on Information and Communications Technology (ICOIACT)*, Yogyakarta, Indonesia: IEEE, Nov. 2023, pp. 224–229. doi: 10.1109/ICOIACT59844.2023.10455906.
- [14] F. D. Rahman, M. I. Z. Mulki, and A. Taryana, "Clustering dan Klasifikasi Data Cuaca Cilacap dengan menggunakan Metode K-Mean dan Random Forest," *J. SINTA: Sist. Inf. dan Teknol. Komputasi*, Vol. 1, No. 2, Apr. 2024, doi: 10.61124/sinta.v1i2.15.
- [15] H. Hou et al., "Load Forecasting Combining Phase Space Reconstruction and Stacking Ensemble Learning," *IEEE Trans. on Ind. Applicat.*, Vol. 59, No. 2, pp. 2296–2304, Mar. 2023, doi: 10.1109/TIA.2022.3225516.
- [16] I. D. Mienye and Y. Sun, "A Survey of Ensemble Learning: Concepts, Algorithms, Applications, and Prospects," *IEEE Access*, Vol. 10, pp. 99129–99149, 2022, doi: 10.1109/ACCESS.2022.3207287.

- [17] S. Zhao *et al.*, “Attach Importance of the Bootstrap *t*-test Against Student’s *t*-test in Clinical Epidemiology: A Demonstrative Comparison using COVID-19 as an Example,” *Epidemiol. Infect.*, Vol. 149, p. e107, 2021, doi: 10.1017/S0950268821001047.
- [18] A. Sakho, E. Malherbe, and E. Scornet, “Do We Need Rebalancing Strategies? A Theoretical and Empirical Study Around SMOTE and Its Variants,” May 22, 2025, *arXiv*: arXiv:2402.03819. doi: 10.48550/arXiv.2402.03819.