

SISTEM PREDIKSI WAKTU KEBERANGKATAN PESAWAT BERBASIS MACHINE LEARNING

David Jehuda Putnarubun¹, Citra Fathia Palembang^{2*}, Jefri Esna Thomas Radjabaycolle³, Emanuella M. C. Wattimena⁴

^{1,2,3,4}Program Studi Ilmu Komputer, Fakultas Sains dan Teknologi, Universitas Pattimura Jl. Ir. M. Putuhena, Ambon, 97233, Indonesia

Keywords:

Prediksi Keberangkatan, Random Forest, Support Vector Machine, Bandara Pattimura, Streamlit.

Correspondent Email:

citra.palembang@lecturer.unpatti.ac.id

Abstrak. Keterlambatan keberangkatan (departure delay) merupakan tantangan operasional utama di Bandara Pattimura Ambon yang berdampak signifikan pada efisiensi maskapai dan kepuasan penumpang. Penelitian ini bertujuan membangun sistem prediksi waktu keberangkatan pesawat menggunakan algoritma Machine Learning, yaitu Random Forest dan Support Vector Machine (SVM), dengan memanfaatkan data operasional penerbangan dan data cuaca. Dataset terdiri dari 4.901 data keberangkatan tahun 2024 yang diperoleh dari AirNav dan BMKG. Penelitian ini menerapkan tahapan pra-pemrosesan yang ketat, termasuk penghapusan variabel bocoran (data leakage) seperti waktu aktual keberangkatan (ATD), serta penanganan ketidakseimbangan kelas menggunakan teknik SMOTE. Hasil evaluasi menunjukkan bahwa algoritma Random Forest lebih superior dibandingkan SVM. Random Forest mencapai akurasi sebesar 94,09% dan mampu mendeteksi kejadian keterlambatan (Recall) sebesar 34,21%. Kinerja ini lebih baik dibandingkan SVM yang memiliki akurasi 93,17% namun hanya mencatatkan Recall keterlambatan sebesar 10,53%. Berdasarkan hasil tersebut, model Random Forest diimplementasikan ke dalam aplikasi berbasis web menggunakan Streamlit, yang memungkinkan petugas operasional memprediksi status penerbangan secara real-time berdasarkan jadwal dan kondisi cuaca.



Copyright © [JITET](http://www.jitet.org) (Jurnal Informatika dan Teknik Elektro Terapan). This article is an open access article distributed under terms and conditions of the Creative Commons Attribution (CC BY NC)

Abstract. *Departure delay is a major operational challenge at Pattimura Ambon Airport, significantly impacting airline efficiency and passenger satisfaction. This study aims to build an aircraft departure time prediction system using Machine Learning algorithms, specifically Random Forest and Support Vector Machine (SVM), by utilizing flight operational data and weather data. The dataset consists of 4,901 departure records from 2024 obtained from AirNav and BMKG. This study applies rigorous pre-processing stages, including the removal of leakage variables such as Actual Time of Departure (ATD), and handling class imbalance using the SMOTE technique. Evaluation results show that the Random Forest algorithm is superior to SVM. Random Forest achieved an accuracy of 94.09% and was able to detect delay events (Recall) of 34.21%. This performance outperforms SVM, which had an accuracy of 93.17% but only recorded a delay Recall of 10.53%. Based on these results, the Random Forest model was implemented into a web-based application using Streamlit, allowing operational officers to predict flight status in real-time based on schedules and weather conditions.*

1. PENDAHULUAN

Perjalanan udara merupakan bentuk suatu perjalanan transportasi yang menggunakan

pesawat terbang untuk berpindah dari satu tempat ke tempat yang lain. Kemudahan, kecepatan, dan efisiensi menjadi keunggulan

bagi perjalanan udara dibandingkan dengan transportasi lain namun salah satu tantangan yang biasanya dihadapi dalam perjalanan udara yaitu penundaan penerbangan [1]. Penundaan penerbangan meningkat karena pesatnya pertumbuhan permintaan sistem transportasi udara dan pengaruh globalisasi abad ke-21 [2]. Penundaan penerbangan mengakibatkan biaya yang besar, mengganggu rencana perjalanan sehingga terjadi penurunan kepuasan penumpang. Berdasarkan standar Federal Aviation Administration (FAA), keterlambatan penerbangan dihitung sebagai keterlambatan apabila keterlambatan penerbangan melebihi ambang batas 15 menit [3]. Prediksi penundaan penerbangan merupakan upaya untuk mengantisipasi penundaan menggunakan data historis. Prediksi penundaan penerbangan sangat penting untuk meningkatkan kepuasan penumpang dan efisiensi operasional bandara [4].

Berbagai metode digunakan dalam beberapa tahun terakhir untuk memecahkan masalah prediksi penundaan penerbangan menggunakan teknik pembelajaran mesin (*machine learning*) [4]. Salah satu metode pembelajaran mesin yaitu *Random Forest*.

Random Forest adalah algoritma pembelajaran mesin yang termasuk didalam teknik yang digunakan untuk melakukan klasifikasi dan regresi dalam pembelajaran mesin [5]. Keluaran *Random Forest* adalah kelas yang dipilih oleh sebagian besar pohon dalam kasus klasifikasi [6]. Klasifikasi *Random Forest* terdiri dari beberapa pohon keputusan yang beroperasi secara independen satu sama lain, pemungutan suara yang dikompilasi dari kelas-kelas yang ditetapkan ke input oleh setiap pohon keputusan dalam pengklasifikasi memutuskan kelas input [7]. Untuk mengukur efektivitas dan akurasi dari *Random Forest*, penelitian ini juga menggunakan metode *Support Vector Machine* (SVM) sebagai algoritma pembanding [8]. SVM memiliki tujuan untuk menemukan fungsi pemisah (*classifier hyperplane*) terbaik guna memisahkan dua buah kelas pada *input space*.

Selain pemilihan algoritma yang tepat, tantangan utama dalam prediksi penundaan penerbangan adalah karakteristik data yang cenderung tidak seimbang (*imbalanced data*). Dalam data operasional penerbangan, jumlah penerbangan yang tepat waktu (kelas

mayoritas) jauh lebih mendominasi dibandingkan penerbangan yang mengalami keterlambatan (kelas minoritas). Ketidakseimbangan ini sering kali menyebabkan model menjadi bias dan cenderung memprediksi kelas mayoritas, sehingga gagal mendeteksi kejadian keterlambatan yang sebenarnya menjadi fokus utama. Untuk mengatasi permasalahan tersebut, penelitian ini menerapkan teknik *Synthetic Minority Over-sampling Technique* (SMOTE). SMOTE bekerja dengan cara mensintesis sampel baru dari kelas minoritas untuk menyeimbangkan distribusi data latih, sehingga model dapat mempelajari pola keterlambatan dengan lebih sensitif dan akurat.

Meskipun metode *Random Forest* dan SVM memiliki kemampuan prediksi yang baik secara teoritis, algoritma ini perlu diimplementasikan ke dalam sebuah sistem terintegrasi agar dapat memberikan nilai manfaat praktis bagi operasional bandara. Model yang hanya berjalan dalam lingkungan pengembangan (*code-based*) sulit digunakan oleh petugas operasional yang membutuhkan pengambilan keputusan cepat. Oleh karena itu, hasil pelatihan model akan diintegrasikan ke dalam antarmuka berbasis web menggunakan kerangka kerja Streamlit. Pengembangan *dashboard* ini bertujuan untuk menyajikan hasil prediksi secara real-time dan visual, sehingga dapat digunakan sebagai alat bantu pendukung keputusan (*decision support system*) yang efektif dalam mengantisipasi potensi keterlambatan penerbangan [9].

Bandara Pattimura merupakan bandara utama yang melayani wilayah Kota Ambon dan Provinsi Maluku. Sebagai gerbang utama transportasi udara di kawasan Indonesia Timur, Bandara Pattimura memiliki peran strategis dalam menghubungkan Maluku dengan berbagai kota besar di Indonesia [10]. Kondisi geografis Maluku yang berupa kepulauan dengan cuaca tropis yang tidak menentu, terutama musim penghujan dan angin kencang, seringkali menjadi faktor penyebab keterlambatan penerbangan di bandara ini. Data historis menunjukkan pola keterlambatan yang bervariasi hingga keterlambatan dengan durasi tertentu, sehingga pengembangan sistem prediksi keterlambatan penerbangan menggunakan metode *Random Forest* dan SVM di Bandara Pattimura menjadi sangat

relevan untuk meningkatkan efisiensi operasional dan kualitas layanan penerbangan di wilayah Maluku.

2. TINJAUAN PUSTAKA

2.1. *Machine Learning*

Machine Learning adalah proses dimana mesin belajar melalui proses yang berulang ulang. Eksplorasi data, visualisasi, dan eksperimen diperlukan dalam proses pembelajaran mesin karena langkah-langkah tersebut harus dieksplorasi, dimodifikasi, dan diaudit secara independen [11]. Arsitektur *Machine Learning* dimulai dari memasukan berupa data mentah seperti gambar, teks, audio, atau numerik. Kemudian masuk ke tahapan *feature extraction* yang merupakan tahapan mengubah data mentah menjadi representasi matematis. Pada tahapan klasifikasi, model *Machine Learning* memetakan fitur kategori menggunakan algoritma seperti yang digunakan dalam penelitian ini yaitu *Random Forest* [12].

2.2. *Random Forest*

Random Forest merupakan model *Machine Learning* yang digunakan dalam klasifikasi dan prediksi. *Random Forest* merupakan teknik pembelajaran mesin yang menggabungkan sejumlah pohon keputusan untuk mengurangi korelasi di antara data fitur. *Random Forest* menggunakan pemilihan fitur dan sampel secara acak untuk mengurangi hubungan antara *decision tree*. Pertama, jumlah data yang setara dipilih secara acak dari sampel pelatihan dalam data pelatihan asli. Setelah itu, pohon keputusan dibangun dengan sebagian fitur yang dipilih secara acak.

Kinerja *Random Forest* dipengaruhi oleh beberapa parameter penting, antara lain jumlah pohon (*n_estimators*), jumlah fitur yang dipilih pada setiap pemisahan (*max_features*), kedalaman pohon (*max_depth*), serta jumlah minimum sampel untuk membagi *node* (*min_samples_split*). Pemilihan dan pengaturan parameter yang tepat sangat berpengaruh terhadap keseimbangan antara kompleksitas model dan performa prediksi, sehingga *Random Forest* mampu mengurangi risiko *overfitting* sekaligus meningkatkan akurasi model [13].

2.3. *Support Vector Machine*

Support Vector Machine (SVM) merupakan salah satu metode *Machine Learning* yang digunakan untuk menyelesaikan permasalahan klasifikasi maupun regresi. Algoritma SVM merupakan suatu pendekatan diskriminatif yang dirancang untuk mencari pemisah terbaik yang dikenal dengan istilah *hyperplane*, guna mengidentifikasi perbedaan antara berbagai kelas [8].

SVM bekerja dengan menentukan sebuah *hyperplane* optimal yang memaksimalkan jarak (margin) terhadap titik data terdekat dari masing-masing kelas, yang disebut sebagai *support vector*. *Hyperplane* optimal berada tepat di tengah dua garis batas margin, sehingga menghasilkan pemisahan kelas yang paling stabil. Semakin besar margin yang terbentuk, semakin baik kemampuan model dalam melakukan generalisasi terhadap data baru.

Secara matematis, *hyperplane* pemisah dalam algoritma SVM didefinisikan melalui Persamaan (1) berikut:

$$w \cdot x + b = 0 \quad (1)$$

2.4. *Normalisasi Data*

Dalam *Machine Learning*, perbedaan skala antar fitur dapat memengaruhi kinerja algoritma berbasis jarak seperti SVM secara signifikan. Tanpa normalisasi, fitur dengan rentang nilai yang lebih besar akan mendominasi perhitungan jarak, yang mengakibatkan model menjadi bias dan tidak optimal.

Salah satu teknik yang paling umum digunakan adalah *Standard Scaler* atau dikenal juga sebagai *Standardization*. Teknik ini bekerja dengan mengubah distribusi data pada setiap fitur secara terpisah sehingga memiliki nilai rata-rata (*mean*) sebesar 0 dan standar deviasi sebesar 1. Transformasi ini dilakukan dengan cara mengurangi nilai asli dengan rata-rata fitur tersebut, lalu membaginya dengan standar deviasinya.

Proses transformasi data menggunakan *Standard Scaler* dilakukan berdasarkan rumus yang ditunjukkan pada Persamaan (2):

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

2.5. *Synthetic Minority Over-Sampling (SMOTE)*

SMOTE adalah teknik *oversampling* yang diperkenalkan untuk menyeimbangkan distribusi kelas dengan cara menciptakan data

sintetis baru, bukan sekadar menduplikasi data yang sudah ada seperti pada metode *random oversampling* konvensional. Pendekatan duplikasi data sering kali menyebabkan *overfitting*, sedangkan SMOTE menghindari hal tersebut dengan membentuk sampel baru yang memiliki karakteristik serupa dengan data asli namun tidak identik [14].

Mekanisme kerja SMOTE didasarkan pada algoritma *k-Nearest Neighbors* (kNN). Prosesnya dimulai dengan mengambil sampel dari kelas minoritas, kemudian algoritma akan mengidentifikasi k tetangga terdekat dari sampel tersebut dalam ruang fitur. Selanjutnya, sebuah garis imajiner ditarik antara sampel asli dan salah satu tetangga terdekatnya, dan titik data sintetis baru dibangkitkan di sepanjang garis tersebut melalui interpolasi linear [15].

Secara matematis, proses pembentukan sampel sintetis (p_{ij}) pada algoritma SMOTE dirumuskan sebagai berikut:

$$p_{ij} = x_i + \text{rand}(0,1) \times (x_{ij} - x_i) \quad (3)$$

2.6. Evaluasi Model

Evaluasi model adalah langkah krusial dalam siklus *Machine Learning* untuk menentukan sejauh mana model dapat melakukan prediksi. dengan baik terhadap data yang belum pernah dilihat sebelumnya [16], [17]. Pada penelitian ini, evaluasi dilakukan menggunakan beberapa metrik utama yaitu:

1. *Confusion matrix*: salah satu instrumen yang digunakan untuk mengevaluasi hasil klasifikasi model.
2. *Accuracy*: untuk mengukur seberapa sering model membuat prediksi yang benar secara keseluruhan. Nilai *Accuracy* dihitung untuk mengukur ketepatan prediksi model secara keseluruhan, sebagaimana dirumuskan dalam persamaan (4):

$$\text{Accuracy} = \frac{\text{Jumlah Prediksi Benar}}{\text{Jumlah Total Prediksi}} \quad (4)$$

3. *Precision*: mengukur seberapa akurat prediksi positif yang dibuat oleh model. Berikut ini adalah persamaan dari *Precision*:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad (5)$$

4. *Recall*: mengukur seberapa baik model dapat menemukan semua kasus positif

yang ada. Berikut ini adalah persamaan dari *Recall*:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad (6)$$

5. F1-score: rata-rata harmonik dari Precision dan Recall. Untuk menyeimbangkan nilai Precision dan Recall, digunakan metrik F1-Score yang perhitungannya mengacu pada Persamaan (7):

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (7)$$

2.7. Streamlit

Streamlit merupakan *framework* Python yang bersifat *open source* yang digunakan untuk membangun aplikasi web terutama pada bidang *Data Science* dan *Machine Learning* [18]. Keunggulan utama Streamlit adalah kemampuannya dalam memfasilitasi pembuatan antarmuka aplikasi secara cepat tanpa mengharuskan pengembang menulis kode *frontend* yang rumit seperti HTML, CSS, atau JavaScript. Hal ini menjadikan Streamlit sangat efisien digunakan untuk membuat *dashboard* sistem prediksi yang interaktif dan visualisasi data yang mudah dipahami oleh pengguna akhir [19].

3. METODE PENELITIAN

Jenis penelitian ini adalah penelitian kuantitatif, dengan pendekatan eksperimen komputasional. Pendekatan ini digunakan karena penelitian ini berfokus pada pengolahan data numerik pada penerbangan.

3.1. Evaluasi Model

Penelitian ini diawali dengan tahap pengumpulan data, di mana dataset utama diperoleh secara langsung dari lokasi penelitian, yaitu AirNav, yang berisi data historis penerbangan di Bandara Pattimura. Selain itu, data cuaca tahun 2024 dikumpulkan dari Badan Meteorologi, Klimatologi, dan Geofisika (BMKG) sebagai data pendukung untuk memperkaya analisis dan meningkatkan akurasi model prediksi [20].

3.2. Pra-Pemrosesan Data

Tahapan pra-pemrosesan data dilakukan melalui serangkaian langkah untuk memastikan data siap digunakan dalam pemodelan. Proses diawali dengan *filtering* data untuk menyeleksi data keberangkatan yang relevan, dilanjutkan dengan penggabungan antara data operasional

penerbangan dengan data cuaca. Selanjutnya, dilakukan penentuan atribut target (*labeling*) serta seleksi fitur untuk menghapus atribut yang berpotensi menyebabkan kebocoran informasi (*data leakage*).

Tahapan berikutnya meliputi penanganan *missing value* untuk menjaga kelengkapan data, serta proses *encoding* untuk mengubah atribut kategorial menjadi format numerik. Setelah data terstruktur, dilakukan pembagian data menjadi fitur dan target, yang kemudian dipisahkan menjadi data latih dan data uji. Khusus pada data latih, diterapkan proses normalisasi menggunakan teknik *Standard Scaler* untuk menyamakan skala antar fitur numerik, serta teknik SMOTE untuk menyeimbangkan distribusi kelas (*imbalanced data*) sebelum data tersebut digunakan untuk melatih model.

3.3. Perancangan dan Pelatihan Model

Pada tahapan ini, dilakukan pembangunan model klasifikasi menggunakan dua algoritma pembelajaran mesin yang berbeda, yaitu *Random Forest* dan SVM. Kedua model ini dirancang secara terpisah untuk membandingkan performa masing-masing dalam memprediksi keterlambatan penerbangan. Proses pelatihan model melibatkan optimasi parameter (*hyperparameter tuning*) menggunakan teknik GridSearchCV yang diterapkan khusus pada data latih. Teknik ini menggunakan skema *cross-validation* untuk mengevaluasi kinerja kombinasi parameter secara internal sebelum model final diuji pada data uji.

3.3.1. Perancangan Model Random Forest

Pencarian parameter optimal dilakukan menggunakan GridSearchCV dengan skema *5-fold cross-validation*. Parameter yang diuji meliputi jumlah pohon (*n_estimators*), kedalaman maksimum pohon (*max_depth*), jumlah minimum sampel untuk pemisahan *node* (*min_samples_split*), serta jumlah fitur maksimal yang dipertimbangkan (*max_features*). Kemudian model dilatih menggunakan data latih yang telah dipersiapkan dengan kombinasi parameter terbaik yang diperoleh dari tahap *tuning*. Setelah model terbentuk, dilakukan analisis tingkat kepentingan fitur (*feature importance*) untuk mengidentifikasi atribut operasional atau

cuaca yang memiliki kontribusi terbesar terhadap prediksi keterlambatan penerbangan.

3.3.2. Perancangan Model SVM

Optimasi parameter dilakukan menggunakan algoritma GridSearchCV untuk mengevaluasi kombinasi nilai parameter C (regularisasi), Gamma, serta jenis Kernel yang digunakan, yaitu *Radial Basis Function* (RBF) dan *Polynomial*. Model SVM dilatih menggunakan data yang telah dinormalisasi menggunakan *Standard Scaler* agar perhitungan jarak antar fitur menjadi optimal. Oleh karena itu, penggunaan parameter terbaik dari hasil optimasi bertujuan untuk memaksimalkan margin pemisah antara kelas *DELAY* dan Tidak *DELAY* dengan menyesuaikan batas keputusan (*decision boundary*) terhadap karakteristik skala data asli.

3.4. Evaluasi Model

Analisis hasil performa model dilakukan dengan menggunakan beberapa metrik evaluasi. Metrik evaluasi yang digunakan yaitu *Accuracy*, *Precision*, *Recall*, *F1-score*, dan *Confusion Matrix*.

3.5. Integrasi Aplikasi Streamlit

Membuat *dashboard* sistem prediksi menggunakan Streamlit. Streamlit memungkinkan pengguna untuk berinteraksi langsung dengan model dan mendapatkan prediksi keterlambatan penerbangan secara instan.

4. HASIL DAN PEMBAHASAN

4.1. Pengumpulan Data

Data penerbangan yang diperoleh dari Airnav sebanyak 9857 *record* dari data tersebut diperoleh atribut yang dapat dilihat pada tabel 1. Sedangkan data cuaca yang diperoleh dari BMKG untuk periode tahun 2024 memiliki atribut yang dapat dilihat pada tabel 2.

Tabel 1. Atribut Data Penerbangan

Atribut	Tipe Data	Deskripsi
Tanggal	Datetime	Tanggal
ATYPE	Kategorial	Jenis Pesawat
ADEP	Kategorial	Bandara Keberangkatan
ADES	Kategorial	Bandara Tujuan
EOBT	Datetime	Perkiraan Waktu di Luar Blok
PUSHBACK	Datetime	Proses mendorong mundur pesawat dari <i>gate</i> menggunakan

TAXI	Datetime	kendaraan khusus (<i>tractor</i>). Pergerakan pesawat di darat menuju landasan pacu atau menuju <i>gate</i> setelah mendarat.
ATD	Datetime	Waktu Keberangkatan Sebenarnya
ETA	Datetime	Perkiraan Waktu Kedatangan
ATA	Datetime	Waktu Kedatangan Sebenarnya
RIU	Numerik	Unit Instruksi Pelaporan
POB	Numerik	Jumlah Penumpang di Pesawat
MTOW	Numerik	Berat Lepas Landas Maksimum
AIRLINES	Kategorial	Maskapai Penerbangan

Tabel 2. Atribut Data Cuaca

Atribut	Tipe Data	Deskripsi
TM	Numerik	Temperatur minimum (°C)
TX	Numerik	Temperatur maksimum (°C)
TAVG	Numerik	Temperatur rata-rata (°C)
RH_AVG	Numerik	Kelembapan rata-rata (%)
RR	Numerik	Curah hujan (mm)
SS	Numerik	Lamanya penyinaran matahari (jam)
FF_X	Numerik	Kecepatan angin maksimum (m/s)
DDD_X	Numerik	Arah angin saat kecepatan maksimum (°)
FF_AVG	Numerik	Kecepatan angin rata-rata (m/s)
DDD_CA R	Numerik	Arah angin terbanyak (°)

Mengingat penelitian ini berfokus pada prediksi waktu keberangkatan, maka data yang diolah lebih lanjut hanya data keberangkatan (*departure*) yang berjumlah 4.901 *record*. Berdasarkan analisis awal pada data tersebut, ditemukan adanya ketidakseimbangan kelas (*class imbalance*) yang cukup besar. Jumlah penerbangan yang tepat waktu sangat mendominasi, sedangkan jumlah penerbangan yang mengalami keterlambatan jauh lebih sedikit. Proporsi perbandingan jumlah data antara kelas DELAY dan Tidak DELAY secara lebih rinci dapat dilihat pada Gambar 1.



Gambar 1. Perbandingan jumlah data pada target

Berdasarkan visualisasi pada Gambar 1, terlihat adanya ketimpangan distribusi kelas yang sangat signifikan dalam dataset. Kelas mayoritas (TIDAK DELAY) mendominasi dengan jumlah 3.769 data (96,1%), sedangkan kelas minoritas (DELAY) hanya berjumlah 151 data (3,9%). Ketidakseimbangan proporsi yang ekstrem ini mengindikasikan bahwa model berpotensi mengalami bias jika tidak ditangani dengan benar, sehingga menegaskan perlunya penerapan teknik *resampling* seperti SMOTE pada tahap pra-pemrosesan.

4.2. Pra-Pemrosesan Data

Tahap Pra-pemrosesan data dilakukan untuk memastikan dataset yang digunakan dalam kondisi bersih dan siap digunakan dalam proses pemodelan. Tahapan yang dilakukan dalam Pra-pemrosesan data yaitu *filtering* data, penambahan atribut, penggabungan data cuaca dan data keberangkatan, pengecekan terhadap *missing value*, melakukan *encoding* pada atribut dan target, penentuan fitur dan target, dan pembagian data latih dan data uji.

4.2.1. Seleksi Fitur

Tahap pra-pemrosesan diawali dengan penyeleksian data secara menyeluruh. Berdasarkan hasil filter pada tahap pengumpulan data, ditetapkan sebanyak 4.901 *record* data keberangkatan (*departure*) sebagai dataset utama yang akan diproses. Hasil dari proses filter data ini menghasilkan sebanyak 4.901 *record* data penerbangan keberangkatan yang dapat dilihat pada gambar 2.

No.	Tanggal	ATYP	ADEP	ADES	EDPT	PUSHBACK	TAXI	DEP/ARR/LOCAL	ATD	...	TN	TX	TAWO	BL_AWO	RR	SS	FE_3	
0	11/12/2024	BT38	WAPP	WAAA	2:40:00	2:53	2:55	D	3:00:00		25.5	31.4	28.2	85.0	NAN	10.0	3	
1	4	11/12/2024	BT38	WAPP	WIII	7:35:00	7:30	7:23	D	7:28:00		25.5	31.4	28.2	85.0	NAN	10.0	3
2	8	11/12/2024	AT76	WAPP	WAPS	22:30:00	22:37	22:39	D	22:45:00		25.5	31.4	28.2	85.0	NAN	10.0	3
3	9	11/12/2024	AT76	WAPP	WAPN	22:35:00	22:28	22:32	D	22:41:00		25.5	31.4	28.2	85.0	NAN	10.0	3
4	10	11/12/2024	AS20	WAPP	WIII	23:05:00	23:05	23:10	D	23:16:00		25.5	31.4	28.2	85.0	NAN	10.0	3
4896	8948	12/31/2024	AT43	WAPP	WAPM	0:30:00	0:30	0:34	D	0:38:00		24.8	32.4	27.8	87.0	0.0	5.0	4
4897	8949	12/31/2024	C130	WAPP	WAPM	1:10:00	1:01	1:05	D	1:09:00		24.8	32.4	27.8	87.0	0.0	5.0	4
4898	8951	12/31/2024	AT43	WAPP	WAPM	4:00:00	3:56	3:58	D	4:02:00		24.8	32.4	27.8	87.0	0.0	5.0	4
4899	8952	12/31/2024	BT38	WAPP	WAAA	4:30:00	4:39	4:43	D	4:48:00		24.8	32.4	27.8	87.0	0.0	5.0	4
4900	8956	12/31/2024	AS20	WAPP	WIII	22:55:00	22:58	23:02	D	23:08:00		24.8	32.4	27.8	87.0	0.0	5.0	4

Gambar 2. Jumlah Record Data

Pada dataset ini, kemudian dilakukan seleksi fitur lanjutan untuk menjamin validitas model prediksi. Atribut-atribut yang teridentifikasi sebagai informasi masa depan atau indikator langsung dari kejadian keberangkatan seperti *Actual Time of Departure* (ATD), *Actual Time of Arrival* (ATA), serta durasi *Taxi* dan *Pushback* dieliminasi dari daftar fitur input. Penghapusan ini dilakukan untuk mencegah terjadinya kebocoran data (*data leakage*), sehingga model dipastikan hanya belajar dari atribut yang tersedia sebelum pesawat berangkat.

4.2.2. Seleksi Fitur

Pada tahap berikutnya, data cuaca tahun 2024 yang diperoleh dari BMKG, mulai dari bulan Januari hingga Desember, digabungkan menjadi satu dataset cuaca. Dataset cuaca tersebut kemudian diintegrasikan dengan dataset penerbangan berdasarkan kesesuaian waktu dan periode kejadian, sehingga menghasilkan satu dataset terpadu yang memuat informasi penerbangan dan kondisi cuaca secara bersamaan.

4.2.3. Pengecekan Missing Value

Setelah proses penggabungan dataset, dilakukan pengecekan terhadap *missing value*. Penanganan *missing value* pada atribut numerik dilakukan dengan menggunakan nilai rata-rata (*mean*), sedangkan pada atribut kategorikal dilakukan dengan mengganti nilai yang hilang menggunakan label “*Unknown*”. Langkah ini dilakukan untuk menghindari kehilangan data yang berpotensi memengaruhi proses pelatihan model.

4.2.4. Encoding Atribut

Tahap selanjutnya adalah melakukan proses *encoding* terhadap atribut kategorikal. *Encoding* dilakukan pada atribut ADEP (*Aerodrome of Departure*) dan ADES (*Aerodrome of Destination*) agar data dapat diproses oleh algoritma *Machine Learning*. Selain itu, *encoding* juga diterapkan pada atribut target, yaitu KETERANGAN_DELAY,

untuk mengubah label kategorikal menjadi bentuk numerik yang sesuai dengan kebutuhan model.

Setelah seluruh atribut siap digunakan, dilakukan penentuan fitur dan target. Pada penelitian ini, atribut KETERANGAN_DELAY ditetapkan sebagai variabel target, sedangkan atribut lainnya digunakan sebagai fitur untuk membangun model prediksi.

4.2.5. Pembagian Data

Tahap akhir pra-pemrosesan data adalah pembagian dataset menjadi data latih dan data uji. Dataset dibagi dengan proporsi 80% sebagai data latih dan 20% sebagai data uji. Pembagian ini bertujuan untuk melatih model menggunakan sebagian besar data serta menguji kemampuan generalisasi model terhadap data yang belum pernah dilihat sebelumnya.

4.2.6. Normalisasi Data

Setelah data dibagi menjadi data latih dan data uji, langkah selanjutnya adalah normalisasi fitur menggunakan teknik *Standard Scaler*. Langkah ini sangat penting, terutama untuk algoritma SVM yang sensitif terhadap variansi skala data. Normalisasi dilakukan dengan mengubah nilai fitur numerik agar memiliki rata-rata 0 dan standar deviasi 1.

4.2.7. Penanganan Ketidakseimbangan Data

Tahap akhir dalam pra-pemrosesan adalah penanganan masalah *imbalanced* data menggunakan metode SMOTE. Berdasarkan analisis awal, jumlah data kelas DELAY jauh lebih sedikit dibandingkan kelas Tidak DELAY. Oleh karena itu, SMOTE diterapkan khusus pada data latih untuk meningkatkan sampel sintesis pada kelas minoritas hingga jumlahnya seimbang dengan kelas mayoritas. Data uji tidak mengalami proses SMOTE dan tetap dibiarkan asli untuk memastikan evaluasi model mencerminkan kondisi dunia nyata. Perubahan jumlah data sebelum dan sesudah penerapan SMOTE disajikan secara rinci pada Tabel 3 berikut.

Tabel 3. Perbandingan jumlah sampel sebelum dan sesudah upsampling

Kategori kelas	Label	Jumlah Sampel (Sebelum SMOTE)	Jumlah Sampel (Sesudah SMOTE)
Tidak DELAY	0	3769	3769
DELAY	1	151	3769
Total Data Latih			7538

4.3. Pelatihan Model

Tahapan pelatihan model dilakukan menggunakan data latih yang telah dipersiapkan pada tahap pra-pemrosesan. Proses ini bertujuan untuk membangun model pembelajaran mesin yang mampu mempelajari pola hubungan antara fitur operasional penerbangan dan cuaca terhadap status keberangkatan pesawat. Pelatihan dilakukan pada dua algoritma berbeda, yaitu *Random Forest* dan SVM, dengan menerapkan teknik optimasi *hyperparameter* menggunakan GridSearchCV untuk mendapatkan performa terbaik dan menghindari overfitting.

4.3.1. Pelatihan Model *Random Forest*

Pelatihan model *Random Forest* dilakukan menggunakan data latih yang telah melalui proses penyeimbangan kelas (*resampling*) menggunakan teknik SMOTE. Langkah ini sangat krusial mengingat data asli memiliki ketimpangan distribusi yang ekstrem, sehingga penggunaan data hasil SMOTE bertujuan agar algoritma dapat mempelajari karakteristik kelas DELAY dengan bobot yang setara tanpa bias terhadap kelas mayoritas.

Proses optimasi *hyperparameter* dilakukan dengan menguji berbagai kombinasi parameter menggunakan skema *5-fold cross-validation*. Pendekatan ini memastikan bahwa model dievaluasi secara berulang pada bagian data yang berbeda untuk menjamin validitas hasil. Berdasarkan eksperimen menggunakan GridSearchCV, diperoleh konfigurasi parameter terbaik yang menghasilkan performa model paling optimal. Rincian parameter terbaik yang digunakan untuk membangun model akhir *Random Forest* dapat dilihat pada Tabel 4.

Tabel 4. Hasil Hyperparameter tuning menggunakan GridSearchCV untuk *Random Forest*

Parameter	Hasil
Jumlah Pohon	100
Kedalaman Maksimum Pohon	None
Jumlah Minimum Sampel untuk Split	2
Minimum sampel di leaf node	1
Jumlah fitur yang dipertimbangkan pada setiap <i>split</i>	sqrt
Bootstrap	True

Berdasarkan tabel di atas, *model Random Forest* terbaik dibangun menggunakan 100 pohon (*n_estimators*) dengan parameter kedalaman maksimum (*max_depth*) diatur sebagai *None*. Nilai *None* mengindikasikan bahwa pohon diperbolehkan tumbuh sepenuhnya tanpa pembatasan kedalaman eksplisit hingga setiap simpul (*node*) menjadi murni atau mencapai batas minimum sampel pemisahan. Penggunaan teknik bootstrap diaktifkan (*True*) untuk memastikan keragaman sampel pada setiap pohon, sedangkan jumlah fitur yang dipertimbangkan pada setiap pemisahan (*split*) diatur menggunakan metode akar kuadrat (*sqrt*). Kombinasi parameter ini dipilih karena memberikan kemampuan terbaik bagi model untuk menangkap pola detail pada data tanpa mengorbankan kemampuan generalisasi.

4.3.2. Penanganan Ketidakseimbangan Data

Pelatihan model SVM dilakukan menggunakan data latih yang telah melalui tahapan normalisasi dan penyeimbangan kelas SMOTE. Langkah normalisasi diterapkan karena algoritma SVM bekerja berdasarkan perhitungan jarak (*distance-based*), sehingga sangat sensitif terhadap perbedaan skala antar fitur. Sementara itu, penggunaan data hasil SMOTE bertujuan untuk mencegah bias mayoritas pada pembentukan *hyperplane* pemisah.

Proses pencarian parameter optimal (*hyperparameter tuning*) dilakukan menggunakan teknik GridSearchCV untuk mengevaluasi kombinasi nilai C, Gamma, dan jenis *kernel* (RBF dan *Polynomial*). Parameter C diuji untuk mengatur toleransi kesalahan klasifikasi, sedangkan Gamma digunakan untuk menentukan cakupan pengaruh data latih terhadap batas keputusan.

Evaluasi terhadap seluruh kombinasi *hyperparameter* dilakukan menggunakan skema *5-fold cross-validation*. Pendekatan ini memastikan bahwa model yang dihasilkan memiliki kemampuan generalisasi yang baik dan stabil. Berdasarkan hasil optimasi, diperoleh konfigurasi parameter terbaik yang disajikan pada Tabel 5.

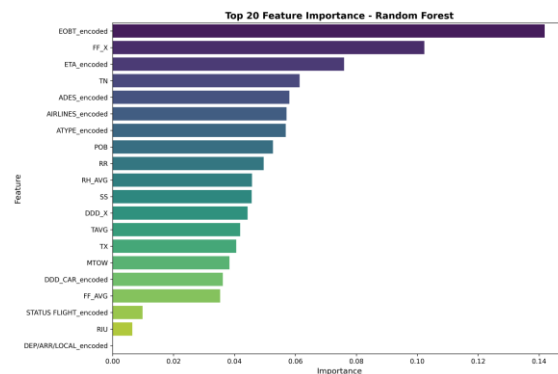
Tabel 5. Hasil *Hyperparameter tuning* menggunakan GridSearchCV untuk SVM

Parameter	Hasil
C	100
Gamma	0.1
Kernel	rbf
Degree	2

Berdasarkan tabel di atas, konfigurasi optimal model terbentuk dengan menggunakan *Kernel Radial Basis Function* (RBF), nilai parameter C sebesar 100, dan *Gamma* sebesar 0.1. Terpilihnya nilai C yang tinggi (100) mengindikasikan bahwa model memberikan penalti yang ketat terhadap kesalahan klasifikasi, yang bertujuan untuk membentuk batas keputusan (*decision boundary*) yang presisi dalam memisahkan kelas DELAY dan Tidak DELAY. Hal ini sangat relevan mengingat data latih telah melalui proses SMOTE, sehingga model dipaksa untuk belajar membedakan pola data asli maupun sintesis dengan akurat. Selain itu, nilai Gamma sebesar 0.1 menunjukkan jangkauan pengaruh *support vector* yang moderat, yang memungkinkan *kernel* RBF menangkap pola non-linear yang kompleks antar fitur yang telah dinormalisasi dengan efektif.

4.3.2. Analisis Feature Importance

Salah satu keunggulan utama algoritma *Random Forest* adalah kemampuannya untuk memberikan informasi mengenai tingkat kepentingan fitur (*feature importance*). Analisis ini bertujuan untuk mengidentifikasi atribut mana yang memiliki kontribusi terbesar dalam menentukan apakah sebuah penerbangan akan mengalami keterlambatan (DELAY) atau tidak. Nilai kepentingan ini dihitung berdasarkan seberapa besar setiap fitur mampu mengurangi ketidakmurnian (*impurity*) pada saat pembentukan pohon keputusan.



Gambar 3. Feature Importance

Hasil analisis menunjukkan bahwa atribut EOBT (*Estimated Off-Block Time*) merupakan fitur yang paling berpengaruh dengan nilai kepentingan (*importance score*) tertinggi mencapai di atas 0.14. Hal ini mengindikasikan bahwa waktu jadwal keberangkatan memiliki korelasi yang sangat kuat dengan potensi DELAY. Pola ini menunjukkan bahwa keterlambatan penerbangan di Bandara Pattimura cenderung terjadi pada jam-jam operasional tertentu, yang kemungkinan berkaitan dengan kepadatan lalu lintas udara pada jam sibuk (*peak hours*) atau rotasi pesawat harian.

Selain faktor jadwal, variabel cuaca terbukti memiliki kontribusi yang sangat signifikan. Atribut FF_X (Kecepatan Angin Maksimum) menempati peringkat kedua sebagai penentu utama prediksi. Tingginya pengaruh kecepatan angin ini menegaskan bahwa kondisi kestabilan udara di sekitar wilayah bandara merupakan faktor krusial yang dipertimbangkan dalam keputusan izin lepas landas (*take-off*) demi alasan keselamatan. Variabel cuaca lainnya, seperti TN (Temperatur Minimum) dan RR (Curah Hujan), juga masuk dalam jajaran fitur penting, yang semakin memperkuat peran kondisi lingkungan dalam sistem operasional penerbangan.

Di sisi operasional logistik, atribut ETA (*Estimated Time of Arrival*) menempati posisi ketiga. Hal ini menunjukkan bahwa jadwal kedatangan pesawat (*inbound flight*) sangat mempengaruhi ketepatan waktu keberangkatan berikutnya, yang sering kali berkaitan dengan efek domino keterlambatan. Atribut lain seperti Bandara Tujuan (ADES) dan Maskapai (AIRLINES) juga memberikan kontribusi yang relevan, menandakan bahwa rute dan karakteristik operasional maskapai tertentu

memiliki pola keterlambatan yang dapat dikenali oleh model.

4.4. Evaluasi Model

Evaluasi model bertujuan untuk mengukur sejauh mana algoritma yang dibangun mampu melakukan prediksi dengan tepat terhadap data uji yang belum pernah dilihat sebelumnya. Pengukuran kinerja dilakukan menggunakan metrik evaluasi standar meliputi *Accuracy*, *Precision*, *Recall*, dan *F1-Score*. Selain itu, analisis *Confusion Matrix* digunakan untuk membedah distribusi kesalahan prediksi pada masing-masing kelas, khususnya dalam mendeteksi kejadian keterlambatan (DELAY) yang merupakan fokus utama penelitian ini.

4.4.1. Evaluasi Model Random Forest

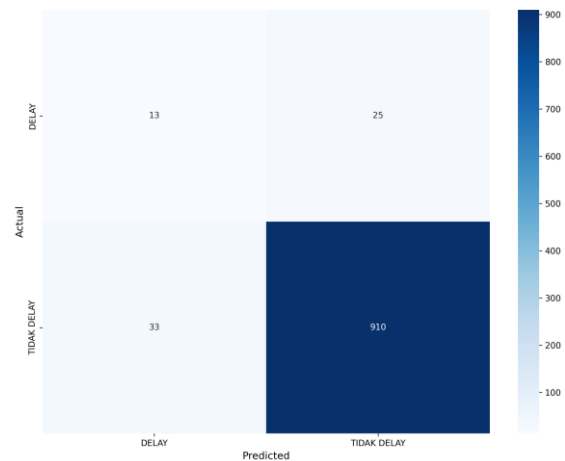
Berdasarkan hasil pengujian pada data uji, model *Random Forest* menunjukkan performa yang sangat stabil dan andal. Kinerja model dievaluasi secara menyeluruh menggunakan metrik *Accuracy*, *Precision*, *Recall*, dan *F1-Score*. Rincian hasil evaluasi model *Random Forest* dapat dilihat pada Tabel 6 berikut.

Tabel 6. Classification Report Random Forest

Kelas	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
DELAY	0.2826	0.3421	0.3095
TIDAK DELAY	0.9733	0.9650	0.9691
<i>Accuracy</i>			0.9409
<i>Macro Avg</i>	0.6279	0.6536	0.6393
<i>Weighted Avg</i>	0.9465	0.9409	0.9436

Model *Random Forest* mencapai akurasi tinggi sebesar 94,09%, namun kinerjanya sangat bias akibat ketidakseimbangan jumlah data. Model ini sangat andal dalam mengidentifikasi kelas dominan TIDAK DELAY (*precision* 97,33%, *recall* 96,50%, *F1-score* 96,91%), tetapi kurang sensitif dalam mendeteksi kelas DELAY (*precision* 28,26%, *recall* 34,21%, *F1-score* 30,95%). Ketimpangan performa antar kelas ini dipertegas oleh nilai *macro average* (*F1-score* 63,93%) yang jauh lebih rendah dibandingkan *weighted average* (*F1-score* 94,36%), sehingga membuktikan bahwa tingginya angka evaluasi secara keseluruhan semata-mata didorong oleh banyaknya data pada penerbangan yang tepat waktu.

Selanjutnya, distribusi detail prediksi benar dan salah dapat dilihat melalui visualisasi *Confusion Matrix* pada Gambar 4 berikut.



Gambar 4. Confusion Matrix Random Forest

Berdasarkan confusion matrix *Random Forest*, dari 38 data DELAY, model hanya mampu memprediksi dengan benar 13 data, sedangkan 25 data lainnya salah diklasifikasikan sebagai TIDAK DELAY. Hal ini menunjukkan bahwa kemampuan model dalam mendeteksi keterlambatan masih rendah, sesuai dengan nilai *recall* kelas DELAY sebesar 34,21%. Selain itu, terdapat 33 data TIDAK DELAY yang salah diprediksi sebagai DELAY, sehingga menghasilkan *precision* kelas DELAY sebesar 28,26%.

Sebaliknya, pada kelas TIDAK DELAY, model menunjukkan performa yang sangat baik dengan 910 dari 943 data berhasil diklasifikasikan dengan benar. Kondisi ini menghasilkan *precision* 97,33% dan *recall* 96,50%, yang menegaskan bahwa model sangat andal dalam memprediksi penerbangan tidak DELAY. Meskipun model mencapai akurasi tinggi sebesar 94,09%, confusion matrix memperlihatkan bahwa kinerja tersebut didominasi oleh kelas TIDAK DELAY, sehingga peningkatan sensitivitas terhadap kelas DELAY masih diperlukan.

4.4.2. Evaluasi Model SVM

Evaluasi terhadap model *SVM* dilakukan untuk mengukur kinerja algoritma dalam memprediksi status keberangkatan pesawat pada data uji. Evaluasi dilakukan menggunakan data uji yang telah dinormalisasi (*scaled*) sesuai dengan proporsi data latih, serta menggunakan parameter hasil optimasi ($C=100$ dan $\text{Gamma}=0.1$). Hasil evaluasi kinerja model berdasarkan laporan klasifikasi dapat dilihat pada Tabel 7 berikut.

Tabel 7. Classification Report SVM

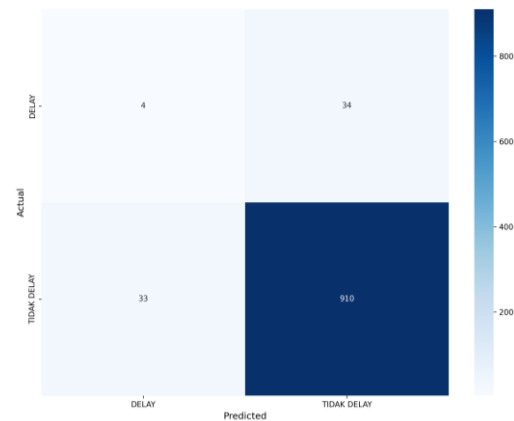
Kelas	Precision	Recall	F1-Score
DELAY	0.1081	0.1053	0.1067
TIDAK DELAY	0.9640	0.9650	0.9645
Accuracy			0.9317
Macro Avg	0.5360	0.5351	0.5356
Weighted Avg	0.9308	0.9317	0.9313

Model SVM menghasilkan akurasi sebesar 93,17%, yang menunjukkan bahwa model mampu mengklasifikasikan data penerbangan dengan tingkat ketepatan yang cukup tinggi. Namun demikian, performa model berbeda secara signifikan antara kelas DELAY dan TIDAK DELAY.

Pada kelas DELAY, model SVM memperoleh precision sebesar 10,81%, recall sebesar 10,53%, dan F1-score sebesar 10,67%. Nilai-nilai ini menunjukkan bahwa kemampuan model dalam mendeteksi penerbangan yang mengalami keterlambatan sangat rendah. Artinya, sebagian besar data DELAY tidak berhasil dikenali oleh model dan cenderung diprediksi sebagai TIDAK DELAY. Kondisi ini mengindikasikan bahwa SVM kurang sensitif terhadap kelas minoritas, yang kemungkinan besar dipengaruhi oleh ketidakseimbangan data.

Sebaliknya, pada kelas TIDAK DELAY, model menunjukkan performa yang sangat baik dengan precision 96,40%, recall 96,50%, dan F1-score 96,45%. Hal ini menandakan bahwa SVM sangat andal dalam mengidentifikasi penerbangan yang tidak mengalami keterlambatan. Nilai macro average F1-score sebesar 53,56% menunjukkan adanya perbedaan performa yang cukup besar antar kelas, sedangkan weighted average F1-score sebesar 93,13% menegaskan bahwa kinerja keseluruhan model sangat dipengaruhi oleh dominasi kelas TIDAK DELAY.

Visualisasi distribusi prediksi benar dan salah pada SVM dapat dilihat lebih rinci pada Confusion Matrix di Gambar 5 berikut.



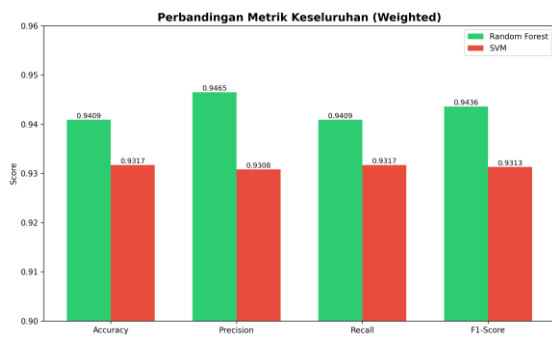
Gambar 5. Confusion Matrix SVM

Berdasarkan confusion matrix SVM, dari 38 data DELAY, model hanya mampu memprediksi dengan benar 4 data (10,53%), sedangkan 34 data (89,47%) lainnya salah diklasifikasikan sebagai TIDAK DELAY. Hal ini menunjukkan bahwa kemampuan SVM dalam mendeteksi keterlambatan penerbangan sangat rendah, selaras dengan nilai recall kelas DELAY sebesar 10,53%. Selain itu, terdapat 33 data TIDAK DELAY yang salah diprediksi sebagai DELAY, sehingga menghasilkan precision kelas DELAY sebesar 10,81%.

Sebaliknya, pada kelas TIDAK DELAY, model menunjukkan performa yang sangat baik dengan 910 dari 943 data (96,50%) berhasil diklasifikasikan dengan benar. Kondisi ini menghasilkan precision 96,40% dan recall 96,50%, yang menegaskan bahwa SVM sangat andal dalam mengidentifikasi penerbangan tidak delay. Meskipun model mencapai akurasi sebesar 93,17%, confusion matrix menunjukkan bahwa kinerja tersebut sangat didominasi oleh kelas TIDAK DELAY, sehingga SVM kurang efektif apabila digunakan untuk tujuan utama mendeteksi keterlambatan penerbangan.

4.4.2. Perbandingan Evaluasi

Setelah dilakukan evaluasi secara terpisah pada masing-masing algoritma, tahapan selanjutnya adalah melakukan komparasi performa secara menyeluruh (overall metrics) untuk menentukan model terbaik yang akan diimplementasikan pada sistem. Perbandingan ini meninjau lima metrik utama: Accuracy, Precision, Recall, dan F1-Score. Visualisasi perbandingan kinerja antara Random Forest dan SVM disajikan pada Gambar 6.



Gambar 6. Perbandingan kinerja Random Forest dan SVM

Berdasarkan grafik perbandingan dan rincian laporan klasifikasi, Random Forest terbukti lebih baik dibandingkan SVM di seluruh indikator pengujian. Dalam hal stabilitas keseluruhan, Random Forest memimpin dengan skor akurasi (Accuracy) sebesar 94,09% dan Weighted F1-Score sebesar 94,36%. Nilai ini mengungguli kinerja SVM yang berada di posisi akurasi 93,17% dan Weighted F1-Score 93,13%. Selisih ini menunjukkan bahwa Random Forest lebih andal dalam menangani variasi data secara umum dibandingkan SVM.

Keunggulan Random Forest semakin terlihat jelas pada kemampuan deteksi keterlambatan (Delay), yang merupakan aspek paling krusial dalam penelitian ini. Model ini mampu mencapai nilai Recall untuk kelas Delay sebesar 34,21%, yang berarti berhasil mengenali lebih dari sepertiga kejadian keterlambatan aktual. Sebaliknya, SVM hampir gagal sepenuhnya dalam mempelajari pola keterlambatan dengan nilai Recall yang sangat rendah, yaitu hanya 10,53%. Selain itu, dari segi ketepatan prediksi (Precision), Random Forest juga jauh lebih baik dengan nilai 28,26% dibandingkan SVM yang hanya 10,81%. Hal ini membuktikan bahwa peringatan yang dikeluarkan oleh Random Forest jauh lebih valid, sedangkan SVM cenderung memberikan banyak peringatan palsu (false alarm). Dengan dominasi mutlak pada aspek akurasi, sensitivitas, dan presisi tersebut, Random Forest ditetapkan sebagai model terpilih untuk diimplementasikan.

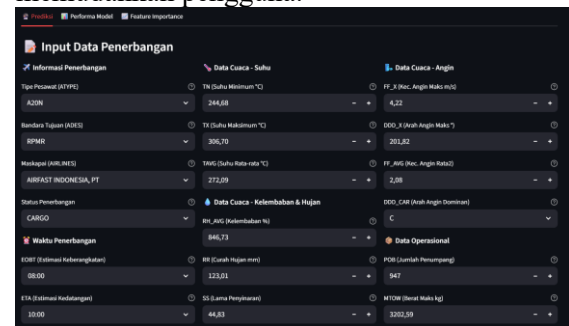
4.5. Integrasi Aplikasi

Tahap integrasi aplikasi merupakan langkah implementasi model prediksi terbaik yang diperoleh dari hasil evaluasi, yaitu Random Forest, ke dalam sebuah antarmuka

berbasis web. Pengembangan sistem ini memanfaatkan kerangka kerja (framework) Streamlit yang memungkinkan pembuatan aplikasi data sains yang interaktif dan real-time. Tujuan dari integrasi ini adalah untuk menyediakan alat bantu keputusan (decision support tool) yang praktis bagi petugas operasional bandara dalam mengantisipasi potensi keterlambatan.

4.5.1. Perbandingan Evaluasi

Antarmuka input dirancang untuk menangkap variabel-variabel krusial yang berpengaruh terhadap prediksi keterlambatan. Sebagaimana terlihat pada Gambar 7, formulir input dibagi menjadi dua kategori utama untuk memudahkan pengguna.



Gambar 7. Tampilan dashboard input data

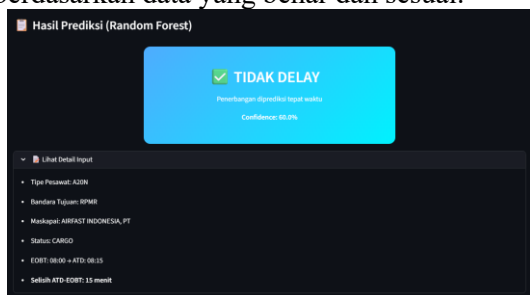
Antarmuka input data pada sistem dirancang dengan tata letak yang terstruktur untuk memudahkan petugas operasional dalam memasukkan parameter prediksi secara efisien. Halaman ini memuat formulir yang dikelompokkan ke dalam kategori logis guna memastikan kelengkapan data sebelum proses prediksi dijalankan. Pada panel sebelah kiri, pengguna diminta untuk mengisi informasi identitas penerbangan yang meliputi Tipe Pesawat (ATYPE), Bandara Tujuan (ADES), dan Maskapai (AIRLINES), serta parameter waktu perencanaan yang terdiri dari Estimasi Keberangkatan (EOBT) dan Estimasi Kedatangan (ETA). Bagian tengah hingga sisi kanan atas didedikasikan untuk input parameter lingkungan atau data prakiraan cuaca yang mencakup variabel Suhu (TN, TX, TAVG), Kelembapan, Curah Hujan, serta kondisi Angin (FF_X, DDD_X). Selanjutnya, pada bagian kanan bawah, tersedia kolom input untuk data operasional logistik, yaitu Jumlah Penumpang (POB) dan Berat Maksimum Lepas Landas (MTOW). Pengelompokan ini memastikan bahwa seluruh variabel prediktor yang dimasukkan merupakan data perencanaan yang

tersedia sebelum keberangkatan (pre-departure), sehingga sistem dapat menghasilkan prediksi status penerbangan yang valid dan objektif.

4.5.2. Tampilan Hasil Prediksi

Setelah pengguna memasukkan seluruh parameter yang diperlukan dan menekan tombol proses, sistem akan menampilkan halaman hasil prediksi yang dirancang untuk memberikan informasi secara cepat dan transparan. Sebagaimana terlihat pada Gambar 4.8, hasil klasifikasi utama dari model Random Forest ditampilkan dalam kotak visual yang menonjol di bagian tengah layar. Sistem memberikan status prediksi yang tegas, misalnya "TIDAK DELAY", yang dilengkapi dengan deskripsi pendukung "Penerbangan diprediksi tepat waktu" serta indikator warna dan ikon visual untuk memudahkan interpretasi petugas operasional secara sekilas.

Selain status klasifikasi biner, fitur krusial yang ditambahkan pada tampilan ini adalah nilai probabilitas atau Confidence Score (misalnya 73.5%). Nilai ini memberikan transparansi mengenai seberapa yakin model terhadap prediksi yang dikeluarkannya, sehingga pengguna dapat mengambil keputusan dengan lebih bijak, terutama pada kasus-kasus dengan probabilitas ambang batas. Di bagian bawah, tersedia panel "Lihat Detail Input" yang dapat diperluas. Panel ini menyajikan ringkasan parameter yang telah dimasukkan sebelumnya seperti Tipe Pesawat, Rute, Jadwal, dan kondisi cuaca mikro yang berfungsi sebagai mekanisme validasi ulang bagi pengguna untuk memastikan bahwa prediksi dihasilkan berdasarkan data yang benar dan sesuai.



Gambar 8. Hasil Prediksi

5. KESIMPULAN

- Penelitian ini berhasil membangun model prediksi keterlambatan keberangkatan pesawat di Bandara Pattimura dengan memanfaatkan data jadwal penerbangan

dan data cuaca tahun 2024. Variabel yang paling berpengaruh dalam prediksi adalah Estimasi Waktu Keberangkatan (EOBT) dan Kecepatan Angin Maksimum (FF_X), yang mengindikasikan bahwa pola keterlambatan sangat dipengaruhi oleh jam sibuk operasional dan kondisi cuaca ekstrem.

- Berdasarkan hasil evaluasi komparatif, algoritma Random Forest terbukti memiliki kinerja yang lebih unggul dibandingkan Support Vector Machine (SVM). Random Forest mencatatkan tingkat akurasi (Accuracy) sebesar 94,09%, melampaui kinerja SVM yang memperoleh akurasi 93,17%.
- Keunggulan utama Random Forest terletak pada efektivitasnya dalam mendeteksi kelas minoritas (Delay). Model ini mampu mencapai nilai Recall sebesar 34,21% dan Precision 28,26%. Kinerja ini jauh lebih baik dibandingkan SVM yang hanya menghasilkan Recall 10,53% dan Precision 10,81%. Hal ini membuktikan bahwa Random Forest lebih adaptif dalam menangani ketidakseimbangan data dibandingkan SVM.
- Implementasi model ke dalam aplikasi berbasis web menggunakan framework Streamlit telah berhasil dilakukan. Sistem ini menyediakan antarmuka input data perencanaan dan cuaca, serta menampilkan hasil prediksi status penerbangan beserta nilai kepercayaan (confidence score), yang dapat digunakan sebagai alat bantu pendukung keputusan (decision support tool) bagi petugas operasional bandara.
- Penelitian ini berhasil membangun model prediksi keterlambatan penerbangan dengan memanfaatkan data operasional dari AirNav dan data cuaca dari BMKG periode tahun 2024. Model dikembangkan menggunakan algoritma Random Forest dan Support Vector Machine (SVM) melalui tahapan pra-pemrosesan data yang komprehensif serta optimasi hyperparameter.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada pihak-pihak terkait yang telah memberi dukungan terhadap penelitian ini.

DAFTAR PUSTAKA

- [1] I. Hatipoğlu and Ö. Tosun, "Predictive Modeling of Flight Delays at an Airport Using Machine Learning Methods," *Appl. Sci.*, vol. 14, no. 13, 2024, doi: 10.3390/app14135472.
- [2] Y. Wu, G. Mei, and K. Shao, "Revealing influence of meteorological conditions and flight factors on delays Using XGBoost," *J. Comput. Math. Data Sci.*, vol. 3, no. March, p. 100030, 2022, doi: 10.1016/j.jcmds.2022.100030.
- [3] Federal Aviation Administration, "Types of Delay," U.S. Department of Transportation. Accessed: Oct. 20, 2024. [Online]. Available: https://aspm.faa.gov/aspmhelp/index/Types_of_Delay.html
- [4] M. Dai, "A hybrid machine learning-based model for predicting flight delay through aviation big data," *Sci. Rep.*, vol. 14, no. 1, pp. 1–16, 2024, doi: 10.1038/s41598-024-55217-z.
- [5] Prof. Swati Dhadake, Tuljai Kadam, Amanoddin Shaikh, Sumit Sabale, and Bhagyashri Shinde, "Flight Delay Prediction by Machine Learning," *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 3, no. 2, pp. 233–236, 2023, doi: 10.48175/ijarsct-8384.
- [6] S. M. Malakouti, A. R. Ghiasi, and A. A. Ghavifekr, "AERO2022-flying danger reduction for quadcopters by using machine learning to estimate current, voltage, and flight area," *e-Prime - Adv. Electr. Eng. Electron. Energy*, vol. 2, no. November, p. 100084, 2022, doi: 10.1016/j.prime.2022.100084.
- [7] A. Renaldi and W. Maharani, "Depression Detection of User in Media Social Twitter Using Random Forest," *J. Inf. Syst. Res.*, vol. 3, no. 4, pp. 410–416, 2022, doi: 10.47065/josh.v3i4.1837.
- [8] W. Apriliah *et al.*, "Prediksi Kemungkinan Diabetes pada Tahap Awal Menggunakan Algoritma Klasifikasi Random Forest," vol. 10, pp. 163–171, 2021.
- [9] Y. Akkem, B. S. Kumar, and A. Varanasi, "Streamlit Application for Advanced Ensemble Learning Methods in Crop Recommendation Systems – A Review," *INDIAN J. Sci. Technol.*, 2023.
- [10] M. S. Lewa, P. Ariawan, and P. Budiarnaya, "Evaluasi Perkerasan Landasan Pacu Pada Bandara Pattimura Dengan," vol. 3, no. 2, pp. 1–8, 2020.
- [11] V. Ganesan, "Machine Learning in Mobile Applications," *Int. J. Comput. Sci. Mob. Comput.*, vol. 11, no. 2, pp. 110–118, 2022, doi: 10.47760/ijcsmc.2022.v11i02.013.
- [12] A. Rajamanickam and C. Kamalakannan, "Land Use and Land Cover Prediction in Tamilnadu of India, Using Random Forest Machine Learning Technique," *Curr. World Environ.*, vol. 20, no. 1, pp. 206–220, 2025, doi: 10.12944/cwe.20.1.16.
- [13] H. A. Salman, A. Kalakech, and A. Steiti, "Random Forest Algorithm Overview," *Babylonian J. Mach. Learn.*, vol. 2024, pp. 69–79, 2024, doi: 10.58496/bjml/2024/007.
- [14] N. Ulfa and K. Surendro, "SMOTE-LOF for noise identification in imbalanced data classification," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 6, pp. 3413–3423, 2022, doi: 10.1016/j.jksuci.2021.01.014.
- [15] S. Wang, Y. Dai, J. Shen, and J. Xuan, "Research on expansion and classification of imbalanced data based on SMOTE algorithm," *Sci. Rep.*, no. 0123456789, pp. 1–11, 2021, doi: 10.1038/s41598-021-03430-5.
- [16] Y. Li and Y. Mu, "Research and performance analysis of random forest-based feature selection algorithm in sports effectiveness evaluation," pp. 1–15, 2024.
- [17] V. Sheth, U. Tripathi, and A. Sharma, "A Comparative Analysis of Machine Learning Algorithms for Classification Purpose Application Analysis of Machine Learning Algorithms for," *Procedia Comput. Sci.*, vol. 215, pp. 422–431, 2022, doi: 10.1016/j.procs.2022.12.044.
- [18] J. M. Nápoles-Duarte, A. Biswas, M. I. Parker, J. P. Palomares-Baez, M. A. Chávez-Rojo, and L. M. Rodríguez-Valdez, "Stmol: A component for building interactive molecular visualizations within streamlit web-applications," *Front. Mol. Biosci.*, vol. 9, no. September, pp. 1–10, 2022, doi: 10.3389/fmolb.2022.990846.
- [19] B. Gavrilović and J. Mitrović, "Comparative analysis of the traffic accidents in the territory of the city of Užice for 2021 and 2022 using open data and the Streamlit application," *Vojnoteh. Glas.*, vol. 71, no. 3, pp. 616–633, 2023, doi: 10.5937/vojtehg71-44007.
- [20] BMKG, "Data Online." [Online]. Available: <https://dataonline.bmkg.go.id/>